UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Design and Implementation of a Model-Driven Software Production Method: From Strategy to Code

February 12, 2023

Author: René Noel López

Advisors: PhD. Óscar Pastor López
PhD. José Ignacio Panach Navarrete

Agradecimientos

A Oscar, por recibirme en el grupo de investigación y guiarme en este camino. Por hacer las preguntas difíciles y por la tenacidad para ayudarme a lograr claridad conceptual en mis planteamientos. Por las conversaciones en el café y por conectarme con gente extraordinariamente brillante y generosa. A Ignacio, por su continuo apoyo, paciencia y pragmatismo para sacar adelante lo que fuera necesario sacar adelante. Gracias por estar siempre atento y pendiente a lo que pudiera necesitar.

A mi esposa, compañera y amiga Maggi. Gracias por acompañarme en esta aventura, por tu amor, generosidad y comprensión a toda prueba. Por empujarme hacia este desafío que sabías que estaba en mi imaginario desde que unimos nuestros caminos hace más de veinte años. Por tu dedicación amorosa a nuestra familia y por cuidar tan bien de nuestra Fiona.

A mi padre, quien cultivó mi curiosidad desde que tengo memoria, y con dosis iguales de amor y humor, ha sido mi ejemplo para superar cualquier obstáculo. A mi madre, que me hacía ver el estudio como un juego, siendo ella mi guía y compañera. A mi tía Lucy, quién me inspiró para ser profesor y a descubrir el mundo en sus ideas y paisajes.

A mis amores con quienes compartí estos años en Valencia. A Rosita, gracias por ser parte de nuestra familia y por amar y cuidar a Fiona. A Erik, Cristina, Elisa y Josefina, por ayudarme a aterrizar en Valencia en un hogar amoroso que se sintió de inmediato como familia. A Bea y Giovanni, su vuelta a Valencia fue una muy buena noticia y una linda oportunidad de reencontrarnos.

Agradezco especialmente a mi gente de Chile, que estuvo presente en estos cuatro años. A mis grandes amigos y colegas de la Universidad de Valparaíso, Carlos y Roberto, gracias por hacerme sentir en estos años que seguimos unidos en nuestra amistad y nuestra pasión por la educación. A mis amigos de la vida, que a pesar de la distancia, me han hecho sentir el calor familiar en nuestras numerosas y extensas videollamadas: Leo, Lolita y Víctor, Esteban y Carito, Fabián y Cris, Christian y Marioly, Margot. Volver es más fácil sabiendo que nos esperan lindos momentos juntos.

A Marcela, por esas mañanas en confinamiento que ayudaron a dibujar mi tesis y nuestras primeras publicaciones juntos. Claudia, gracias por el espacio de contención y el patio de juego para ideas. A Gianina, por compartir tu camino en estos años tan desafiantes. A Sergio, gracias por las conversaciones, las ideas, y los futuros proyectos. ¡Vamos por más!

A mis compas de PROS: Diana, Adrián, David, gracias por compartir conmigo un pedacito de sus vidas. Alberto, Mireia, Alejandro, por las risas en las partidas de rol. Jose, por el apoyo y estar siempre disponible para resolver lo que fuera necesario.

A la Agencia Nacional de Investigación y Desarrollo de Chile, por el financiamiento de mi proyecto de investigación. A la Universidad de Valparaíso, por darme la oportunidad de realizar el doctorado. Vuelvo con mucha ilusión de retribuir la confianza entregada con lo que he aprendido.

Abstract

New, disruptive technologies emerge daily, changing the world as we know it: how we learn, work, and socially interact. This ever-changing scenario pushes organisations to quickly adapt not only their products and services but also their structure and strategies to survive and thrive. As has been widely studied, aligning information technology to high-level goals is key for an organisation to adapt quickly to its environment. Model-driven development (MDD) methods have contributed to this by systematically including business goals in the software development process, providing traceability, quality and efficiency through model-to-model transformations. Yet, existing MDD methods have not included organisational strategy and structure in the development process. This thesis integrates organisational information into a baseline MDD method composed of the OO-Method, an object-oriented model-driven development method, and Communication Analysis, a communication-oriented business process modelling method. The baseline MDD method is extended by the main contributions of this thesis: LiteStrat, an organisational modelling method, and Stra2Bis, a method for designing strategically aligned business processes. LiteStrat supports modelling the external influences that drive new software development endeavours and the strategy and organisational structure to address such influence. Stra2Bis integrates LiteStrat and Communication Analysis through three model-to-model transformation guidelines, generating the scaffold of business processes aligned with the organisation's structure and strategy. Sound experimental validations were performed to assess the methods' improvements in completeness and accuracy and their effect on the method users' efficiency and satisfaction. Further work regards implementing the methods into industrial contexts and their continuous evolution.

Resumen

Cada día surgen nuevas tecnologías que cambian el mundo tal y como lo conocemos: cómo aprendemos, trabajamos y nos relacionamos. Este escenario lleva a las organizaciones a adaptar rápidamente no sólo sus productos y servicios, sino también su estructura y estrategias para sobrevivir y prosperar. Como se ha estudiado ampliamente, alinear la tecnología de la información con objetivos de alto nivel es clave para que una organización se adapte rápidamente a su entorno. Los métodos de desarrollo dirigidos por modelos (MDD) han contribuido a ello al incluir los objetivos de negocio en el proceso de desarrollo de software, proporcionando trazabilidad, calidad y eficiencia mediante transformaciones de modelo a modelo. Sin embargo, los métodos MDD existentes no han incluido la estrategia y la estructura de la organización en el proceso de desarrollo. Esta tesis integra la información organizacional en un método MDD existente compuesto por OO-Method, un método MDD orientado a objetos, y Análisis de Comunicaciones, un método de modelado de procesos de negocio orientado a la comunicación. A ellos, se integran las principales contribuciones de esta tesis: LiteStrat, un método de modelado organizacional, y Stra2Bis, un método para diseñar procesos de negocio alineados estratégicamente. LiteStrat permite modelar las influencias externas que demandan el desarrollo de nuevo software, y la estrategia y la estructura organizacional para abordar dicha influencia. Stra2Bis integra LiteStrat y Análisis de Comunicaciones a través de tres reglas de transformación, generando una estructura de procesos de negocio alineada con la organización. Hemos realizado validaciones experimentales de las mejoras de completitud y precisión de los modelos producidos por los métodos, y de la eficacia y satisfacción de sus usuarios. El trabajo futuro se centra en la aplicación de los métodos en la industria y en su evolución continua.

Resum

Cada dia sorgixen noves tecnologies que canvien el món tal com el coneixem: com aprenem, treballem i ens relacionem. Aquest escenari porta a les organitzacions a adaptar ràpidament no sols els seus productes i servicis, sinó també la seua estructura i estratègies per a sobreviure i prosperar. Com s'ha estudiat àmpliament, alinear la tecnologia de la informació amb objectius d'alt nivell és clau perquè una organització s'adapte ràpidament al seu entorn. Els mètodes de desenvolupament dirigits per models (MDD) hi han contribuït en incloure els objectius de negoci en el procés de desenvolupament de programari, proporcionant traçabilitat, qualitat i eficiència mitjançant transformacions de model a model. No obstant, els mètodes MDD existents no han inclòs l'estratègia i l'estructura de l'organització en el procés de desenvolupament. Aquesta tesi integra la informació organitzacional en un mètode MDD existent compost per OO-Method, un mètode MDD orientat a objectes, i Anàlisis de Comunicacions, un mètode de modelatge de processos de negoci orientat a la comunicació. A ells, s'integren les principals contribucions d'aquesta tesi: LiteStrat, un mètode de modelatge organitzacional, i Stra2Bis, un mètode per a dissenyar processos de negoci alineats estratègicament. LiteStrat permet modelar les influències externes que demanden el desenvolupament de nou programari, i l'estratègia i l'estructura organitzacional per a abordar aquesta influència. Stra2Bis integra LiteStrat i Anàlisi de Comunicacions a través de tres regles de transformació, generant una estructura de processos de negoci alineada amb l'organització. Hem realitzat validacions experimentals de les millores de completesa i precisió dels models produïts pels mètodes, i de l'eficàcia i satisfacció dels seus usuaris. El treball futur se centra en l'aplicació dels mètodes en la indústria i en la seua evolució contínua.

Contents

List of Figures

List of Tables

Part I

Problem Investigation

Chapter 1

Motivation

1.1 Including organisational information in model-driven development

Model-driven development (MDD) aims to systematically use models as the primary artefacts of the software engineering process. The approach is to design models, defined as “a coherent set of formal elements describing something” to transform them into real software systems (Mellor, A. N. Clark, and Futagami, 2003). MDD is an alternative to the technological dependence of traditional programming, i.e., designing, programming, and testing the system in specific programming languages and technologies. Programming at a conceptual level (Embley, Liddle, and Pastor, 2011) is expected to enable the reuse at the domain level, to produce a continuously increased quality of the models and this the generated systems, costs reduction, and major longevity of the designed solutions, since they are not technology dependant (Mellor, A. N. Clark, and Futagami, 2003).

One of the most successful contributions of model-driven methods is helping developers and analysts integrate and trace information of the context of the information system into the software development process. Standardised approaches such as Model-Driven Architecture (MDA) (The Object Management

Group, 2014) have proposed the separation of business and application logic from the underlying platform and technologies.

In MDA, Computation Independent Models (CIM) represent business information, while Platform Independent Models (PIM) represent high-level abstractions of the system, and Platform Specific Models (PSM) include implementation details that could support code generation. At the CIM level, models contain information relevant to understanding and specifying the requirements of the software development endeavour. Models at the CIM level can be traced to different artefacts at the PIM level and then to the PSM level, where model transformations can produce the working code of the system.

Even though MDA recommended modelling standard is the Unified Modelling Language (UML) (The Object Management Group, 2017), MDA provides a general framework where other modelling methods can be included, particularly at the CIM level (Kirikova, Finke, and Grundspenkis, 2010). Requirements engineering (RE) methods and techniques have been integrated into MDD methods to provide traceability and partial automation from business information to the model of the information system. In this way, RE methods for identifying the goals of the stakeholders of the Information System (IS) and designing business processes aligned with those goals provide a sound input for the IS design process. Following MDA principles (Brown, 2004) such as metamodel mappings and model-to-model transformations and method design methodologies (Henderson-Sellers, Ralyté, et al., 2014), RE methods have been integrated with each other and with MDD methods to promote traceability and automation in information system model generation (España, 2011; Ruiz, Costal, et al., 2015).

In the MDA context, RE methods are part of the CIM level since models at the CIM level aim to describe the environment of the system and the business model without showing the structure of the system (Belaunde et al., 2003). Having different models at the CIM level requires ensuring their consistency and alignment, which can be achieved through model-driven techniques such as model-to-model transformations to ensure information is preserved between models or through automatic consistency analysis, among others. Figure 1.1 illustrate the MDA and RE+MDD approaches, where goals and business process models are used for representing business information.

One of the key approaches in RE for representing business information is goal modelling. Most goal modelling frameworks define goals as a *"desired state of affairs"* of social actors (Yu, 2011b), which can be system users, business stakeholders, regulatory agencies, the organisation developing software itself, and

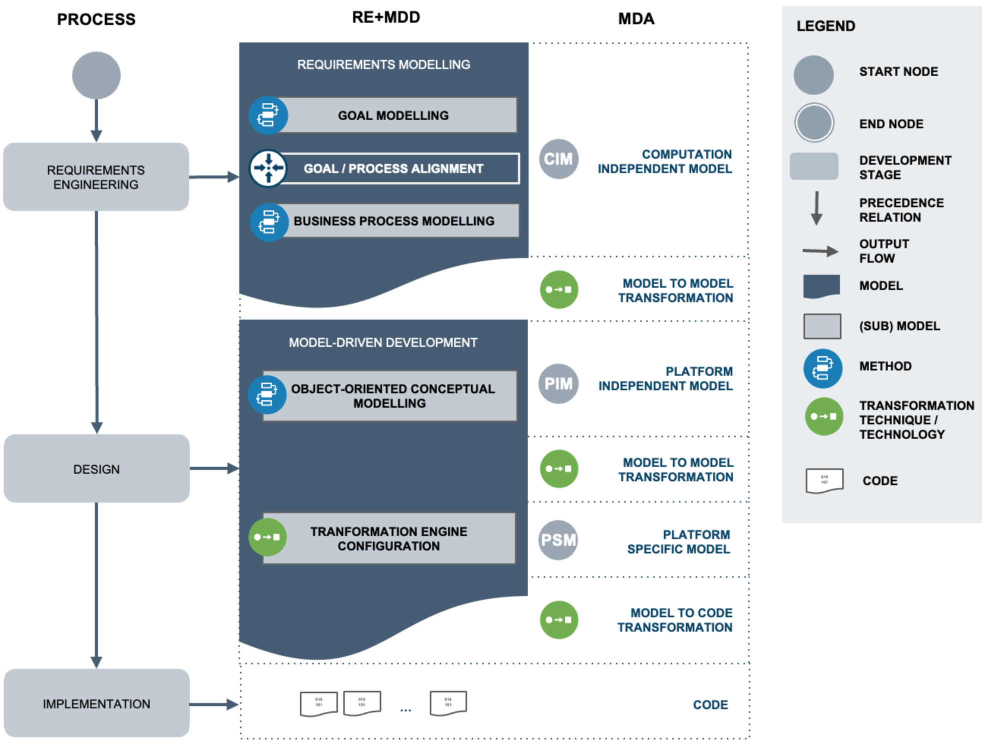


Figure 1.1: Model-Driven Architecture (MDA) and Requirements Engineering Model-Driven Development (RE+MDD) approaches

so on. In RE+MDD, goal models have been mainly used to represent system users and business stakeholders. On the other hand, high-level organisational information has been commonly used as input for long-term enterprise architecture efforts (The Object Management Group, 2015; The Open Group, 2022a) but not as a source of requirements. However, recent research on software organisations (Forsgren, Humbpotifle, and Kim, 2018) has shown the importance of including organisational information as a key input for designing the IS. By considering the organisational level information, software organisations aim to design the IS as a set of small software services aligned to the high-level business outcomes and the organisational structure (i.e., software development teams) responsible for achieving such business outcomes.

This thesis addresses the inclusion of organisational information into an MDD method by means of transforming organisational models of business strategy and organisational structure into business process models. Following the design science methodology (Wieringa, 2014), we designed and validated two artefacts to achieve this goal. The following sections describe the research context, goals, and methodology. In Section 1.2, we present the knowledge context of the research regarding model-driven initiatives to align goals and business processes and an overview of the open challenges. The motivation for expanding this knowledge area towards including organisational information is presented in Section 1.3, taking software organisations as the social context for the design science endeavour. The problem statement is detailed in Section 1.4, and the research and design methodology is presented in Section 1.5. Finally, Section 1.6 presents an outline of the thesis and maps the research goals with the chapters where they are addressed.

1.2 Model-driven alignment of goals and business processes

In the RE area, a particular topic of interest is aligning goals and business process models to design justified and purposeful processes according to goals. On the one hand, goal and agent-oriented modelling frameworks collect the strategic intentions of social actors, helping analysts to identify the overarching organisational needs that drive the development and evolution of the information system (Yu, 2011a). On the other hand, business process models represent the organisation's operation in terms of the flow of actions and interactions needed to achieve the goals (Rosing et al., 2015).

Hence, model-driven techniques have been applied to help analysts identify inconsistencies between goal and process models. For instance, goal and process alignment techniques can identify whether any business process does not cover a goal or ensure the inclusion of key goal model elements by transforming them into business process model elements. In a RE+MDD context, the completeness and accuracy of business process models are critical since they are a key input for most automatic and semi-automatic MDD techniques (Habba, Fredj, and Chaouni, 2019). Figure 1.2 illustrates the goal and process alignment approach.

A feature shared by most goal modelling frameworks is leaving what to represent as actors and goals to the modellers. Freedom in goal modelling has helped the application of goal modelling into different domains, from the detailed specification of the system users's goals to the specification of the organisation's high-level, strategic goals for innovation (Yu, 2009). Goal and process alignment initiatives also exploit this freedom, varying from claiming to support the

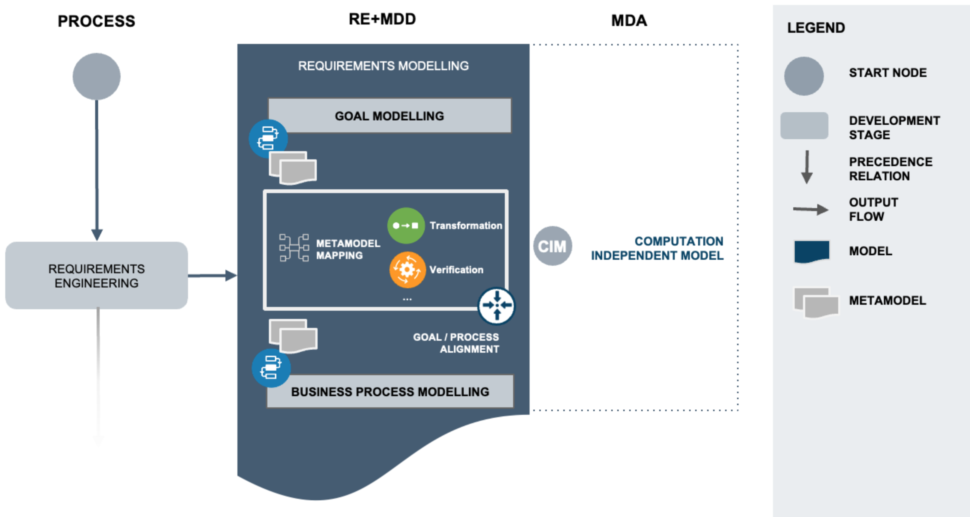


Figure 1.2: Goal and process alignment.

alignment of processes with high-level organisational goals (Vara, Sánchez, and Pastor, 2008; Sousa and Prado Leite, 2014) to mapping very system-specific goals and tasks to business process elements (Insfrán et al., 2017; Al-Kalbani et al., 2019).

Despite the benefits of freedom in goal modelling, a question arises when aiming to align goals and processes: *what goals must be modelled for supporting strategic alignment in an MDD method?*. Even though MDA's CIM has been widely exploited for specifying the information system requirements, other interpretations have included business model information (Kirikova, Finke, and Grundspenkis, 2010), the organisation's environment, and external forces (Huang and Fan, 2007) that could also set goals for developing the information system. Considering information systems as work systems (Steven Alter, 2013) and the framework depicted in Figure 1.3, different goals can be set by the system users, customers, and high-level organisational actors. The *participants* of the system can set their goals regarding what they expect when using the system. *Customers*, as the consumers of the organisation's products and services, can set goals for which using the system is an alternative or tool towards achieving a non-system-related goal. On the other hand, though more indirectly in most organisations, *strategy, environment, and infrastructure* can set high-level, strategic goals for developing the information system.

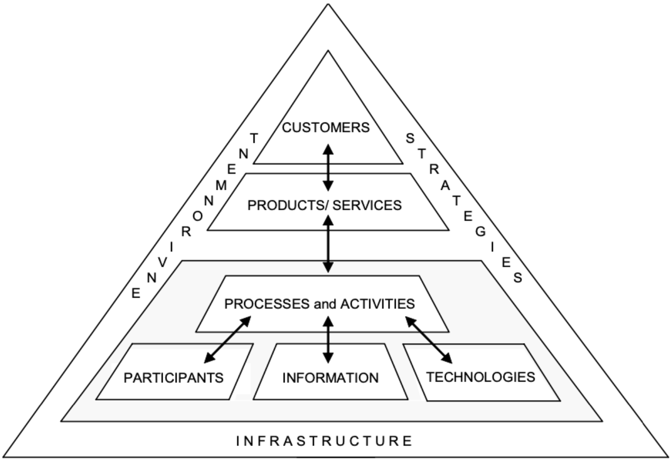


Figure 1.3: The work system framework (Steven Alter, 2013)

Integrating existing goal modelling and business process alignment into MDD methods could possibly support developing software according to participants' and customers' needs. However, representing strategic-level information could be challenging for existing alignment methods. Consider the example depicted in Figure 1.4 modelled using i*, one of the most used goal and agent-oriented modelling frameworks for goal and process alignment (Habba, Fredj, and Chaouni, 2019), and described below.

Example 1.1: Goal modelling example: Real Estate Agency.

A Real Estate company wants to seize the opportunity of increasing the market share by offering an on-line rental service to customers abroad. This is operationalised by offering an online rental service, which will be the responsibility of the Rentals Team. At the same time, legal compliance of the transaction is delegated to the Legal Department through the design of an abroad rental contract.

In the example, the same goal construct is used for defining high-level, strategic goals (Real Estate Co.'s "Market share increased"), and operational goals (Rental Team's "Online rental service offered"). Similarly, high-level actions such as "digitalise the rental service" coexist with the more specific "Book a property". Regarding the goal and process alignment, specific tasks such as

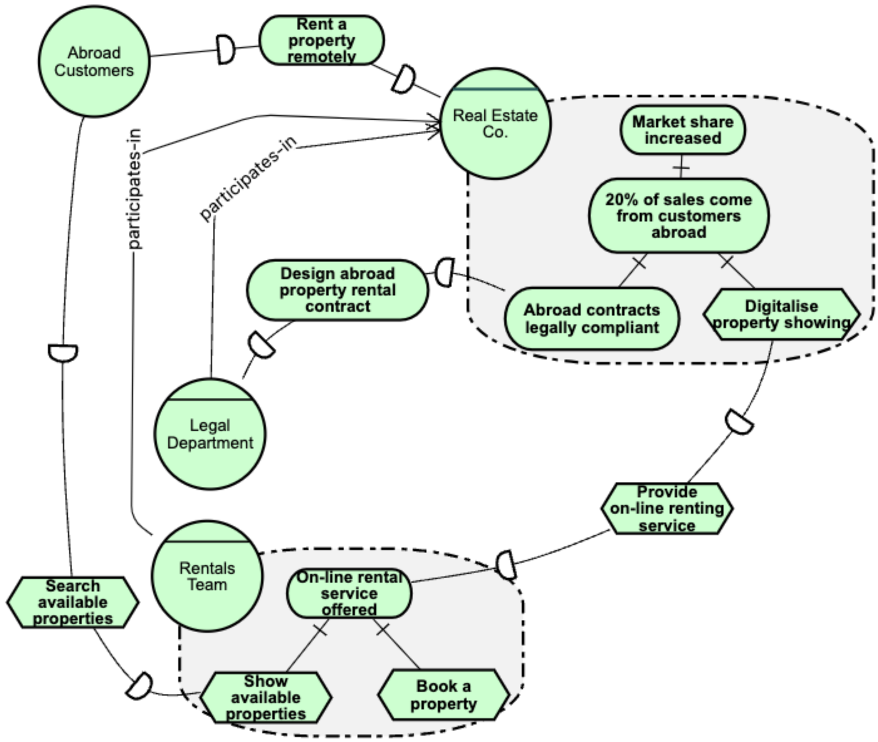


Figure 1.4: Goal model example.

the latter and "Show available properties" could also overlap with business process model tasks, leading to obvious alignments and model redundancy. On the other hand, since strategic goals could reach several parts of the organisation (e.g., the Legal Department and the Rentals Team), they should be aligned with different business processes, and some of them could not require the support of the IS (for instance, Legal Department's process). It could be possible to scope the alignment just to actors having operational goals (Rentals Team) by interpreting the *participates-in* relationships; however, the existing goal and process alignment frameworks do not include this type of link in their mappings.

The example above illustrated challenges for modelling and including some business strategy-related information: high-level ends and means and the organisational structure. While these topics could apparently not be relevant for the IS development process, recent research has provided evidence of the need to include them not only for the strategic alignment of processes but as a critical element in the design of the IS, as presented in the next section. The existing goal and process alignment techniques and the related methods are further studied in Chapter 2, where a detailed review of existing goal and business process alignment methods is presented. The challenges relevant for the present research are analysed in Chapter 4. Below, in Definition 1.1, we define the knowledge context for the research.

Definition 1.1: Knowledge Context

- Goal and agent-oriented modelling.
- Alignment of goal and business processes using model-driven methods and techniques.

1.3 Strategy in Software Organisations

Nowadays, companies whose core purpose is developing software, or *software organisations* (SO) (Kettunen and Laanti, 2017) have been able to scale agile software development (Aghina et al., 2017) across the whole organisation by aligning strategic elements with their software architecture (Forsgren, Humbpotifle, and Kim, 2018). Compared to traditional organisations, SOs have a different way of defining their strategic goals and how they structure their development teams around such goals, which has been found as a key enabler for agility and high-growth (Forsgren, Humbpotifle, and Kim, 2018; Salameh and Bass, 2022). Among other benefits, SOs' approach to business strategy also tackles the hindrances of MDD adoption related to the need for inter-team communication and coordination (Skelton and Pais, 2019). Nevertheless, more importantly for RE and MDD, the way SOs manage business strategy provides requirements for the design of the information system (Thoughtworks, 2016).

Software organisations (SO) are *"organisations or organisational units with new software production as the core purpose, or software-intensive customer organisations of those software producers"* (Kettunen and Laanti, 2017). In the last decade, there has been a rapid growth in the stock value of SOs such as Facebook and Twitter. Today, the consumption of software services provided by technology companies is massive. Sharing economy applications like Uber and Airbnb use software to connect customers and providers of various services, while social media platforms like Facebook, X (formerly Twitter), and Instagram use software to create online communities and networks. E-commerce giants like Amazon, Alibaba, and Shopify use software to enable online shopping and delivery; streaming services Spotify, Netflix, YouTube, and Twitch have disrupted the entertainment industry. The enabling technology for these services is also offered as software services: cloud computing providers like Microsoft, Google, and Amazon use software platforms to offer scalable and flexible computing resources. The advent of artificial intelligence technology, such as generative artificial intelligence and large language models, is enabling new software services whose capabilities and potential uses have yet to be explored, with disruptive potential to change the world as we know it.

Nowadays, the market cap (i.e., the total value of their outstanding shares) of tech giants like Facebook (meta) is close to a trillion dollars[1]. Other companies with more specific niches, such as Slack (a messaging app for business) and Spotify (a music streaming service), reach a market cap of close to 27 billion dollars [2][3]. Since growth is a key factor for valuing technology organisations[4], one of their main challenges for SOs is to be able to grow quickly without the efficiency of software development being an obstacle (Forsgren, Humbpotifle, and Kim, 2018).

Spotify is one of the benchmarks on how it is possible to develop software in a highly scalable way (Salameh and Bass, 2022). First released in 2012, the Spotify Model (Kniberg and Ivarsson, 2012) proposed a new way of carrying out business strategy and organisational structure. The Spotify Model is an autonomous, people-driven model that emphasises culture and networking. The Spotify model aims at an organisational structure composed of Squads, small cross-functional groups with a clear mission that own their business and software development processes and are responsible for a specific feature of the software product. The squads are free to choose their development methodologies and tools and define their progress and results measures. Squads also collaborate with other squads, tribes (groups of squads), chapters (groups of specialists) and guilds (communities of interest) to share knowledge, best practices and feedback. The products developed by the squads are communicated to others through services. One of the main activities for this approach is to monitor the dependencies between squads and tribes so that they remain as

[1] https://companiesmarketcap.com/meta-platforms/marketcap/
[2] https://companiesmarketcap.com/slack/marketcap/
[3] https://companiesmarketcap.com/spotify/marketcap/
[4] https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/valuing-high-tech-companies

independent as possible (Kniberg and Ivarsson, 2012). Figure 1.5 illustrates the structure of the Spotify model.

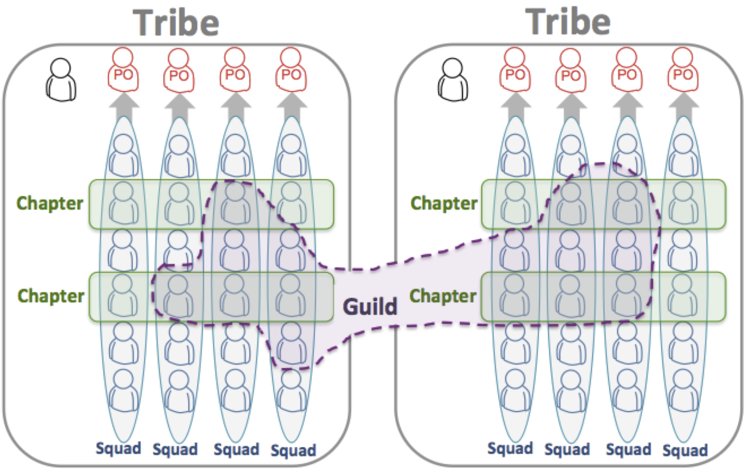


Figure 1.5: The Spotify model for organisational structure (Kniberg and Ivarsson, 2012). (PO: product owner).

The Spotify model has been widely adopted (and adapted) by technology consulting organisations and other technology companies (Brosseau et al., 2019). For example, Shopify, a commerce platform that allows starting, growing, and managing a digital business, explicitly references the Spotify model as the enabler of their growth (Shopify, 2022). Shopify market capitalisation is currently \$83.1 billion [5]. Software engineering practitioners have unravelled the nature behind the Spotify model in what is called Conway's Law, which states that organisations replicate their communication structure to everything they design (Conway, 1968), naming Spotify's approach as the "Inverse Conway' Maneouvre" (Thoughtworks, 2016; Forsgren, Humbpotifle, and Kim, 2018). The approach aims to continuously evolve the organisation's structure to match the desired system architecture without losing sight of the need to design the components of the structure around business strategy.

Next, an overview of SOs' approach to business strategy and organisational structure is presented, as well as their effect on system design.

[5] https://companiesmarketcap.com/shopify/marketcap/

1.3.1 Strategy and alignment in software organisations

SOs have designed and continuously evolved business models built on top of cloud infrastructure, focused on providing highly customised products. However, SOs must adapt their *strategy* to constantly change the offered products (Weinhardt et al., 2009) and the way they built those products. Strategy, in its dimension as a plan, can be defined as a *"plan to achieve the organisational goals"* (Mintzberg, 1987). Changes in organisational strategy require adjusting the way the organisation works to implement the strategy. SOs must conceal strategic decisions such as what product will be offered to which market, with the organisational structure, business processes, and supporting technology for implementing the strategy. This topic has been studied for decades under the concept of *strategic alignment* (Henderson and H. Venkatraman, 1999).

Strategic alignment deals with fitting the external domain of the organisation, i.e., the business goals and the means to achieve them, with the internal domain, which regards the organisation's structure, how the organisation designs (and redesigns) its business processes, and the information technology (IT) needed (Henderson and H. Venkatraman, 1999). The approach to managing the information technology needed for supporting the organisation, thus the *IT strategy*, has varied as companies' products and services increasingly rely on digital technologies. The IT strategy was initially considered a functional-level strategy (N. Venkatraman, 1994) that must be aligned but subordinated to business strategy. In the last decade, IT strategy became indistinguishable from business strategy, namely *digital business strategy* (Bharadwaj et al., 2013). For SOs, *digital business strategy is business strategy.*

SOs are born as digital organisations (Jarosiński, Sekliuckiene, Kozma, et al., 2023), so they continuously face the challenge of digital transformation (DT), which is finding new ways to create value through technology. As defined by (Vial, 2021), DT is *"a process in which digital technologies create disruptions that trigger strategic responses from organisations seeking to alter their pathways to create value while managing the structural changes and organisational barriers that affect the positive and negative results of this process"*. Hence, SOs must manage changes in their *business strategy* and *structure* to create value through technology.

SOs have specific practices to manage their business strategy and organisational structure, directly affecting their business processes and the design of the systems supporting them. This approach combines business strategy practices (called strategy agility or agile strategising by some authors (Scaled Agile, INC, 2021; Holbeche, 2018)), organisational design practices, and software design practices to exploit the strategic information to align the system architecture with the organisational structure. In the following subsections, we provide an overview of these practices.

1.3.2 Business strategy in software organisations

According to the classic management literature, the definition of *business strategy* is broad and could address the organisation's plan to achieve its business goals, a particular ploy to take advantage of a situation, the position of the organisation in the market, the definition of the organisation's perspective from the point of view of different stakeholders, or the pattern of action of the organisation to achieve a business goal (Mintzberg, 1987).

The business strategy approach of frameworks for agile organisations concerns the definition of *business goals and the courses of action to achieve them* (Holbeche, 2018; Highsmith, Luu, and Robinson, 2019; Scaled Agile, INC, 2021; Doerr, 2018). Most of these frameworks propose setting high-level, customer-focused goals and then breaking them down into more specific and measurable goals and actions to achieve them. For instance, Google's framework, *Objectives and Key Results* (OKR) (Doerr, 2018) define *objectives* as high-level aims, which are broken down into measurable *key results*. Similarly, the EDGE framework by the ThoughtWorks (Highsmith, Luu, and Robinson, 2019) proposes to define *goals* as high-level ends, the *measures* that will allow checking the achievement of the goal, and *targets*, which are the desired values for the measures.

A key element of agile organisations' strategy is that, since they have a customer-value focus, they have an *outside-in perspective* (Highsmith, Luu, and Robinson, 2019), following an adaptive strategy (Holbeche, 2018). In an adaptive strategy, the organisation identifies external elements that could affect the business goals and the strategic actions to react to these elements. External elements could include new market trends, technological advances, new customer needs, and competitors whose behaviour affects the organisation (Walter, 2021).

While business strategy is a long-term effort in traditional organisations, in software organisations, the capability of *continuously adapting the strategic direction* is currently considered a vital business success factor (Highsmith, Luu, and Robinson, 2019). Software organisations could change their business strategy and organisational structure as fast as quarterly (Aghina et al., 2017).

The approach to the business strategy of the above frameworks is consistent with the definition of *business strategy as a plan* (Mintzberg, 1987). However, the approaches recommended by the previously mentioned frameworks are not a heavy-weight process for strategic planning but a *lightweight approach*. An example of a lightweight approach in management is the GOST framework, (Horwath, 2014), which defines two levels regarding *what* the organisation wants to achieve (high-level goals and more specific objectives), and *how* to achieve such goals (high-level strategies and more specific tactics).

1.3.3 Organisational structure in software organisations

SOs' approach to organisational structure has been named "Inverse Conway's Maneouvre" by software development practitioners (Thoughtworks, 2016; Forsgren, Humbpotifle, and Kim, 2018). In a nutshell, ICM aims to evolve the organisational structure's design, the definition of organisation units and their dependencies, according to the desired technology architecture (Thoughtworks, 2016). In Figure 1.6, the schematic representation illustrates how the organisational structure, including organisational units (e.g., areas, departments, teams) and their communication interactions, is replicated at the process and technology architecture levels. At the process level, this replication involves mapping the organisational structure into the business processes of each participating organisation unit, as well as their business collaborations. Similarly, at the system architecture level, the organisational structure is reflected by the arrangement of software modules and their dependencies. As can be seen in the example, at the system architecture level, there are circular dependencies between the software modules, which is an antipattern in software design (Parnas, 1979; Martin, 2000) and has been shown to have a significant effect on the change proneness of system architecture (Oyetoyan et al., 2015).

Considering Figure 1.6, if the organisation seeks to improve the system architecture, e.g., reducing circular dependencies, the organisational structure must not have circular dependencies. Figure 1.7 depicts an alternative organisational structure for Figure 1.7, in which the circular dependency between OUA, OUB, and OUC has been eliminated, as well as the two-way communication relationship between OUD and OUE. As a result, both the business process and system architecture levels mirror these designs, eliminating the circular dependencies in the system architecture.

The approach of SOs to strategic alignment has been empirically studied. In (Forsgren, Humbpotifle, and Kim, 2018), the authors surveyed over 23,000 survey responses from 2,000 different software organisations from 5 to 10,000

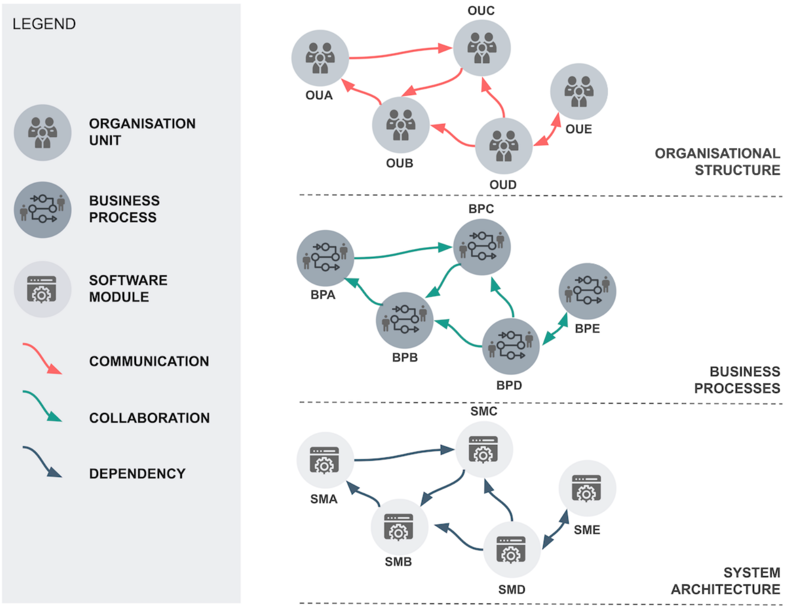


Figure 1.6: According to Conway's law, the organisational structure is replicated into everything the organisation designs (Conway, 1968).

employees to identify the best practices of high-performing technology organisations. The results show that, regardless of the type of the system, high-performance software delivery is possible as long as the systems and the teams that build and maintain them are *loosely coupled*.

For organisational structure, loose coupling deals with the interdependence of organisational actors; such dependency could pertain to different domains such as the organisation's hierarchical structure, workflow relationships, resource sharing and exchange, among others (Beekun and Glick, 2001). Forsgren et al. emphasise the need for loose coupling of development teams in different domains: surveyed practitioners declared they achieve higher performance when they do not have to communicate with other people outside the team to perform large-scale design changes for testing and deploying their systems and depending from other teams work or authorisation to complete their work (Forsgren, Humbpotifle, and Kim, 2018).

Restructuring the organisation might be challenging; however, organisations in a highly variable environment (such as SOs and organisations undergoing

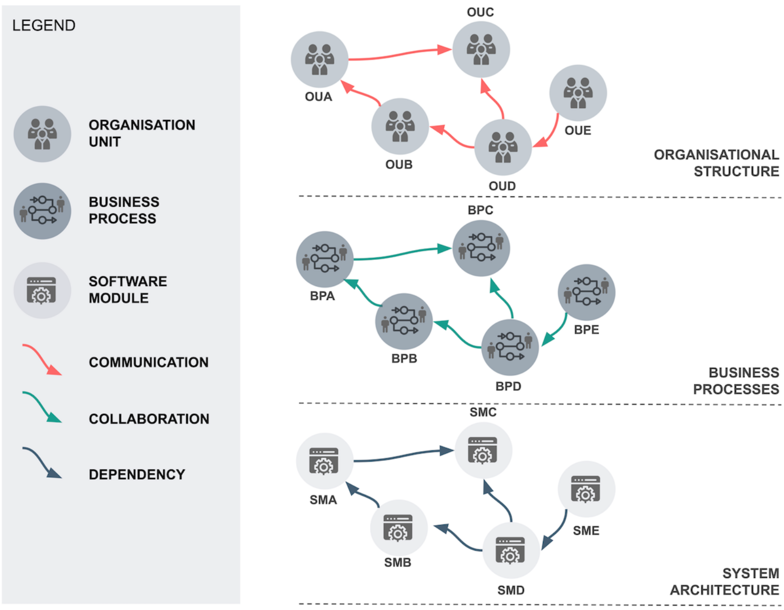


Figure 1.7: Inverse Conway's Manoeuvre seeks to design the organisational structure to mirror the desired system architecture.

DT) may benefit from *reconfiguring* the organisation (Girod and Whittington, 2017; Karim and Capron, 2016). While restructuring the organisation changes the fundamental principles of the organisational design (e.g., change from a division by function to a division by product or market), organisational reconfiguration aims for changing organisational units under the existing design principles of the organisation structure (Girod and Whittington, 2017). Hence, SOs constantly change their configuration by adding, recombining, splitting, and deleting teams. In the example in Figure 1.7, deleting the circular dependency between OUA, OUB, and OUC might be recombined by transferring part of the capabilities of OUC to OUB (Girod and Whittington, 2017). Another improvement opportunity is optimising the communication design between the organisational Units. In the example, the bi-directional relationship between units (as between OUE and OUD in Figure 1.7 might be solved by properly designing the interface between the units in such a way one organisation unit offers its value "as-a-service" to the other unit (as OUE to OUD in the example), this is, with minimum collaboration among the units (Skelton and Pais, 2019).

1.3.4 Software design in software organisations

To exploit the loose coupling of the organisational design requires a software design approach that allows mirroring the organisational structure into a loosely coupled software design. In information systems, loose coupling refers to the positive architectural feature of information services sharing only a small set of assumptions. Hence, the impact of its change is limited (Hohpe and Woolf, 2004; Kaye, 2003). This allows systems to be easily tested and deployed, even if the number of services and systems in the organisation grows.

The system design approach in terms of services is called Service-oriented architecture (SOA) (Raj and Bhukya, 2023). SOA has been applied to address the deployment and scaling problems with monolithic systems (Raj and Sadam, 2021), this is, systems that are independent of other systems and self-contained, but lacking flexibility (Mishra, Kunde, and Nambiar, 2018). However, monolithic systems can still affect the architecture coupling since they manage a business domain, requiring different organisation units (or teams) to communicate and coordinate. Figure 1.8 represents how a monolithic system couples different organisation units because of its business domain scope. In the figure, each organisation unit (OUA, OUB, OUC) is responsible for managing the lifecycle of information entities (DCOUA, DCOUB, DCOUC, respectively). Under a SOA architecture, the monolith could offer independent services (SDCOUA, SDCOUB, SDCOUC). However, since the entities are managed by the same monolithic system, even though the organisation units are cross-disciplinary and independent, they would need to coordinate and communicate their software development actions.

SOs have adopted the microservice architecture (MSA) approach to enable loosely coupled service design consistent with the organisational structure. Under MSA, each service is a small, loosely coupled, scalable, and reusable service that can be built and deployed separately (Thönes, 2015). To define the scope of the business domain of a microservice and avoid coupling among organisation units, design techniques such as domain-driven design (DDD) (E. Evans and E. J. Evans, 2004) have been employed. DDD aims to split the domain model in the same way as organisational units are, provided that organisational units have been designed to match the desired information system's architecture. Figure 1.9 represents a domain-driven microservice design for the monolithic architecture in Figure 1.8. The domain entities that formerly were connected (DCOUB, DCOUA, and DCOUC) are split into independent software modules (MSOUA, MSOUB, MSOUC) which offer similar services as in Figure 1.8 (SDCOUA, SDCOUB, SDCOUC), but that can be built, tested and deployed independently. The persistence of the domain entities is also split,

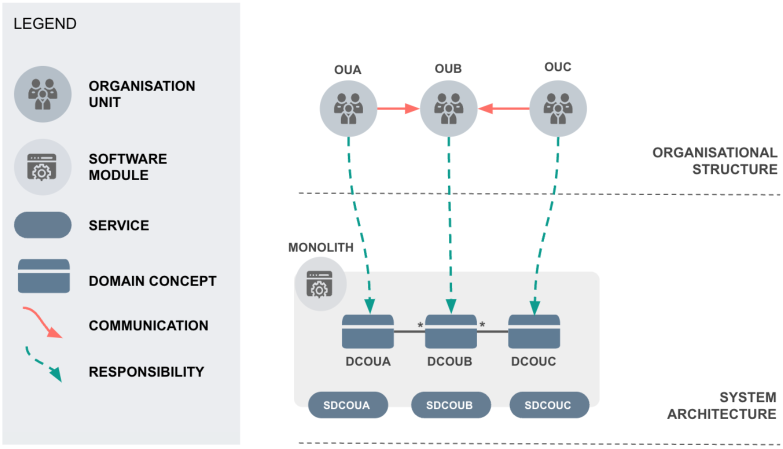


Figure 1.8: A monolithic application couples the organisational structure.

so each software module has its own persistence (known as the database per service pattern (Richardson, 2019), which is synchronised through lightweight mechanisms such as queues.

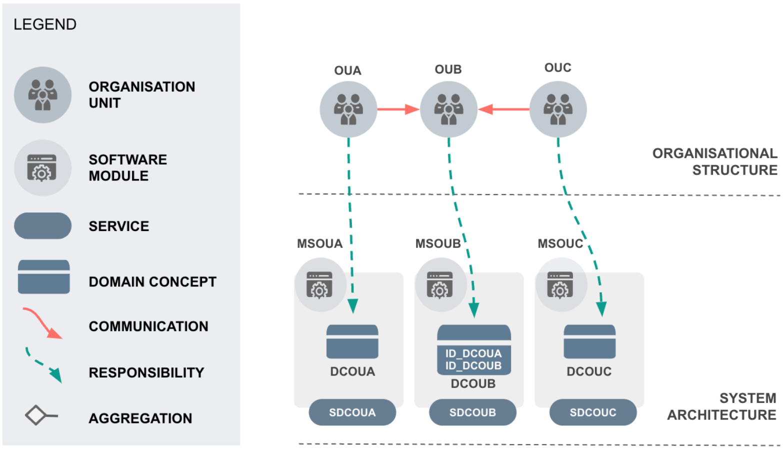


Figure 1.9: Domain-driven microservice design aims to split the system architecture into small, independent, and aligned software modules.

1.3.5 Requirements from the social context

SOs' approach to system design starts by designing an organisational structure that 1. allows deploying the business strategy of the organisation and 2. matches the desired architecture of the information system. The organisational structure, conceived as a set of cross-functional teams that own a part of the business process and domain, establishes the requirements for how the information system must be modularised. According to the concern-based taxonomy of requirements by Glinz (Glinz, 2007), this is a *constraint requirement*, as it "sets a restriction in terms of a prescribed solution element".

Even though the motivation for including organisational information in model-driven development methods comes from SOs, any type of organisation could benefit from this approach. However, it could be challenging for traditional, complex organisations to re-design their organisational structure to match the desired software design. Traditional organisations could also benefit from SO's approach by scoping organisational structure modelling to their software development teams and their interactions, which can be reconfigured (Girod and Whittington, 2017) to better assign software development responsibilities.

As a summary, we define the requirements for a model-driven development method so it can enable SOs' approach to software design.

Definition 1.2: Social Context Requirements

- Include organisational information on business strategy and organisational structure in the software development process.
- Use organisational information to design strategically aligned, scalable, and loosely-coupled information system.

1.3.6 Limitations

The above requirements deal with SOs' practices for business strategy and organisational structure design. However, as strategy is a broad concept, we scope it to business strategy as a plan. Examples of other management and strategy practices that SOs follow to scale the agility of their development teams which are out of the scope of our research are portfolio management, changing the organisations' mindset from projects to products, adopting an adaptive leadership, continuous learning culture, and conducting experiments to test value hypotheses (Highsmith, Luu, and Robinson, 2019; Scaled Agile, INC, 2021; Holbeche, 2018). Despite their importance for managing the software development process, these practices do not have documented effects on defining requirements for the software products.

On the other hand, other software design practices and patterns are key for enabling SOs' approach to a scalable, loosely coupled software design. Several design patterns aim to organise the subdomains into one or more deployable and executable components (Richardson, 2018), such as the saga pattern (Štefanko et al., 2019) to implement distributed transactions as a series of local transactions, and the API composition pattern, to resolve queries by joining data from separated databases (Richardson, 2018). Approaches such as *event-sourcing* (Alongi et al., 2022) help to manage the complexity of large, distributed systems by providing *observability*, this is, a system's attribute regarding how much and well the internal state of a system can be inferred through a monitoring infrastructure (Muller, 2018). These practices and patterns address the requirements set by the SOs approach for software design. Since this thesis focuses on collecting requirements, these practices are also out of the scope of the present research.

1.4 Problem Statement

As detailed in the next section, this research follows the Design Science methodology in the interpretation by Wieringa (Wieringa, 2014). In this context, the research problem is stated design problems that address the questions detailed below.

- *What must be designed by the researcher?*: This research deals with designing a model-driven software production method from requirements to code.
- *With what will the artefact interact?*: The model-driven method is expected to be implemented in organisations or organisational units with new software production as the core purpose or software-intensive customer organisations of those software producers (Kettunen and Laanti, 2017).
- *What desired properties must it have?*: The model-driven method must include organisational information, in particular, the organisation's strategy and structure and a way to convey this information as software requirements.

- *To whom should this interaction be useful?*: The model-driven method should be useful for software engineers and business analysts.
- *To achieve which of their goals?*: for designing strategically aligned, scalable and loosely-coupled information systems.

Following Wieringa's template (Wieringa, 2014), the design problem is stated below.

Definition 1.3: Problem Statement

> How to design a model-driven development method that includes organisational information for designing a strategically aligned, scalable, and loosely coupled information system?

To solve the problem defined above, the designed method should cover the full software development process in its entirety, from the early requirements stage to the generation of the working code of the information system. The present research builds upon existing requirements engineering and model-driven development methods, which have been methodologically integrated previously. We refer to this method as the *baseline model-driven software production method*, or *baseline method*, for short.

1.4.1 Overview the baseline model-driven software production method

As an initial standing point, this work follows the practice of *model-driven requirements engineering (RE+MDD)*, defined by España as "the practice of engineering requirements by emphasizing modelling an model transformations" (España, 2011). This practice allows the integration of model-driven requirements engineering and model-driven development methods. As depicted in Figure 1.10, the method aims to produce a requirements model that generates the information system's conceptual schema through semi-automatic model-to-model transformations. The conceptual schema is then compiled to generate the system's source code.

The baseline method integrates two requirement engineering methods and a model-driven development method. In the requirements model, i* (Yu, 2011b) is used for organisational modelling, which is integrated by a model-to-model transformation (Ruiz, Costal, et al., 2015) with Communication Analysis (España, González, and Pastor, 2009), a communication-oriented requirements

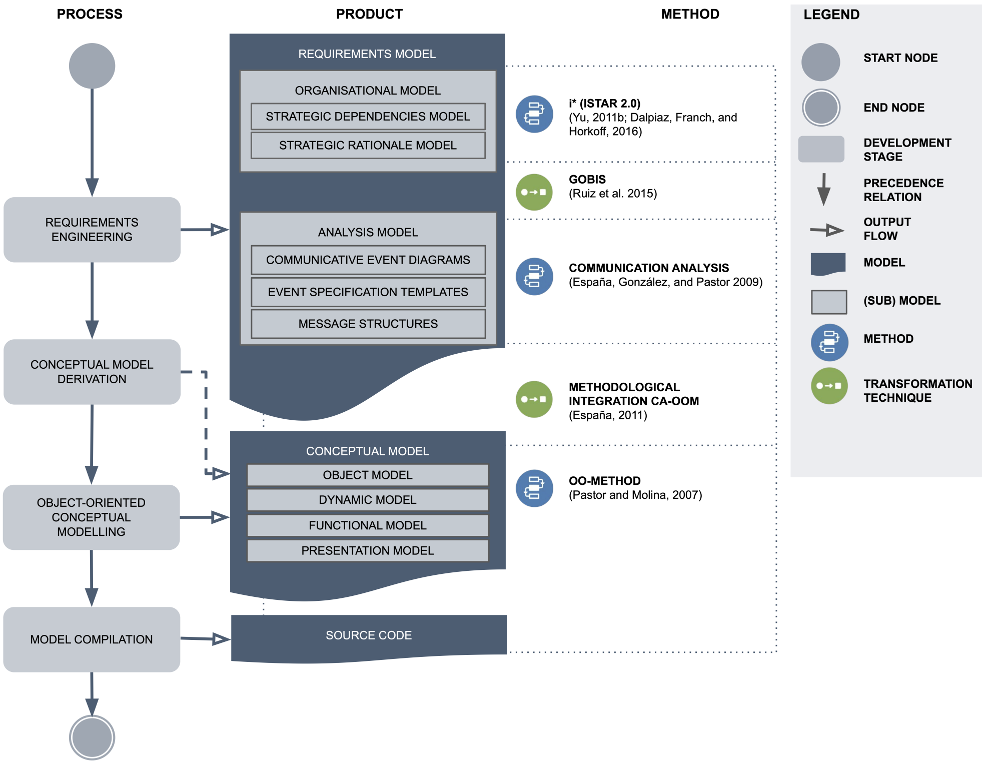


Figure 1.10: Activities and products of the baseline model-driven software production method.

engineering method. Through a semi-automatic model-to-model transformation technique, the conceptual model of the system is derived from the Communication Analysis models, specified through the OO-Method (Pastor and Molina, 2007). Through object-oriented conceptual modelling, the conceptual model of the system is completed and then compiled to generate the code of the information system. The methods and integration techniques are further detailed in Chapter 2.

This thesis focuses on the requirements engineering stage of the process, particularly in designing a method for organisational modelling and an alignment technique that enables the inclusion and integration of business strategy and organisational structure information. As addressed in Chapter 4, the baseline method presents a series of challenges for satisfying the requirements set by the social context. This motivates the need to design an artefact that 1. Takes into account the requirements of the software organisations, and 2. Presents the artefact as an alternative to the situation when the baseline method is applied to software organisations. The methodological support for achieving these goals is provided by Design Science (Wieringa, 2014) as research methodology, and Situational Method Engineering (Henderson-Sellers, Ralyté, et al., 2014) as design methodology.

1.5 Research and Design Methodology

1.5.1 Research Methodology

This research follows the design science research methodology, which supports designing and studying artefacts in context. In this research, the context is the baseline model-driven software production method introduced in Section 1.4.1, and the artefacts are new methods that will enable the development of strategically aligned information systems.

Though there are different interpretations of design science for information systems and software engineering (Johannesson and Perjons, 2014; Wieringa, 2014; Hevner and Chatterjee, 2010), this thesis follows the design science methodology proposed by Wieringa (Wieringa, 2014). Design science guides the researcher to achieve *research goals* by addressing *knowledge questions* about the problem to be solved and the effects of the solution, and solving *design problems* through the engineering of artefacts to be applied in a given context, this is, a *treatment*.

Design science proposes an engineering cycle of four tasks: problem investigation, treatment design, treatment validation, and treatment implementation, described below.

- Problem Investigation concerns studying the problematic phenomena in which a person or group of persons, named *stakeholders*, have desired states of affairs or *goals* that they are failing to achieve using their current treatments. In order to systematically elicit the problematic phenomena, the researcher has to characterise the constituent components and relationships of the current solution, thus, the *conceptual framework*, and elaborate a *theory* on the causes of the problem and how solving it would contribute to achieving the stakeholder goals. The theory must be validated through empirical research techniques and/or critical peer evaluation.

- Treatment Design regards *specifying the requirements* to design the new treatment, which must be consistent with the findings of the problem investigation task, and to validate whether the *requirements contribute to the goals* of the stakeholders. The *available treatments* that could possibly satisfy the requirements must be studied (usually jointly with the problem investigation task). In case no available treatments could satisfy the requirements, a *new treatment design* activity must ensure their satisfaction by applying a suitable design methodology.

- Treatment Validation regards the empirical assessment of the artefact's effects in the context. The assessment addresses whether *the treatment produces the desired effects* and whether *the effects satisfy the requirements.* Other studies, such as context sensitivity analyses and trade-offs between different artefacts, can be carried out.

- Treatment Implementation concerns applying the designed treatment in the original context of the problem. When the research context is a real-world problem (as in this thesis), the treatment implementation task refers to technology transfer, thus applying the research results to a context outside the academy.

This design science project is limited to the three first tasks of the engineering cycle (Wieringa, 2014), named the *design cycle*, leaving treatment implementation outside the research project. Design problems must be addressed by following specific *design methodologies*, which depend on the type of artefact needed to satisfy the requirements. In the following section, the design methodology of this research is summarised.

1.5.2 Design Methodology

The artefacts designed in this thesis follow the Situation Method Engineering (SME) methodology (Henderson-Sellers, Ralyté, et al., 2014), given its support for designing custom methods according to different situational factors. In SME, the researcher, named the method engineer, considers situational factors and method requirements to design a custom method from an existing method base and/or method parts. In this research, the situational factors regard whether the organisation adopting the MDD method has a traditional approach to business strategy or is more like software organisations since they have different strategic alignment approaches, as introduced in Section 1.3. The method requirements are elicited in the problem investigation task.

SME provides different approaches for constructing a method; this research follows the assembly-based approach in a similar way as the case study presented by Ralyté in (Ralyté, 2013). The assembly-based approach is depicted in Figure 1.11.

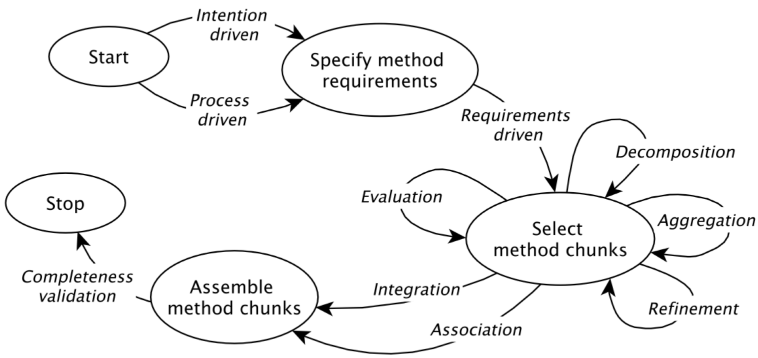


Figure 1.11: Process for assembly-based situational method engineering (from (Henderson-Sellers, Ralyté, et al., 2014)).

In order to specify the situational requirements, it is possible to adopt a *intention-driven* or a *process-driven* strategy. The intention-driven strategy supports adding new intentions and strategies for a new method, while the process-driven strategy is suitable for creating a new method. The requirements are specified in terms of a requirements map, using the same map notation (Rolland, 2007) as in Figure 1.11, which corresponds to a graph where the nodes represent the intentions the method must fulfil, and arcs represent the strategies to achieve those intentions.

The *requirements-driven* approach to *select method chunks* supports querying the existing method base, searching for method chunks that fulfil the requirements (Ralyté, 2013). A *method chunk* is an autonomous and coherent part of a method that specifies a product and how to produce it, i.e., a product and a process perspective. The query results can be refined through decomposition, aggregation, refinement and evaluation to select the more appropriate method chunks.

The *assemble of method chunks* can be performed by an *integration* strategy or by an *association strategy.* The association strategy is useful for positioning the selected method chunks into the new method and connecting them through guidelines (Ralyté, 2013), while the integration strategy is suitable when the method chunks overlap and have similar engineering goals.

With regard to the guidelines to connect method chunks, (Henderson-Sellers, Ralyté, et al., 2014) classifies them into Intention Selection Guidelines (ISG), Strategy Selection Guidelines (SSG) and Intention Achievement Guidelines (IAG). ISGs guide the method users to select to which intention to move from the current intention in case more than one option is available. SSG guides the method user to select which strategy to adopt to move from one intention to another; for instance, in Figure 1.11 to move from the intention *start* to the intention *specify method requirements*, an SSG should guide the method user on following an intention-driven or a process-driven strategy. Finally, IAG provides guidance on how to enact the current intention, in this case, how to Specify method requirements. It is worth noting that an IAG can be specified through a sub process, this is, in a new and embedded map.

For assembling the new model-driven method, we consider the existing model-driven method presented in Section 1.4.1 as an assembled method, composed by the method chunks i* (Yu, 2011a), Communication Analysis (CA) (España, González, and Pastor, 2009), and the OO-Method (OOM) (Pastor and Molina, 2007). The methods have been assembled by *association*; particularly, i* has been associated with CA through the GoBIS guidelines (Ruiz, Costal, et al., 2015), and CA has been associated with OOM through the systematic integration in (España, 2011).

From the point of view of SME, this thesis addresses two design problems: 1. Designing a new organisational modelling method to represent information relevant to the strategic alignment and 2. Designing a new model-driven development method that includes this information in the design of strategically aligned business processes. The way SME is applied to address these problems is detailed in the Research Method in Section 1.5.3.

1.5.3 Research Method

In the research method section, we describe the instantiation of the research and design methodologies. The research is structured around a main design research goal, which is *to improve the baseline model-driven software production method by including organisational information on business strategy and organisational structure for designing a strategically aligned, scalable, and loosely-coupled information system.*

Figure 1.12 presents the research definition under the design science framework. As described by the methodology, the social context provides relevance and requirements for the research project; in return, the research project produces an artefact that satisfies the context requirements. The research project is based on a knowledge context that provides theoretical and methodological tools, as well as existing solutions, which are taken as input for the design of the new artefact. In turn, the research projects contribute to new solutions. The investigation of the effects of the artefact is based on research methods from the knowledge context; in turn, the investigation returns empirical evidence on the effects of the artefact.

The social context for the research is software organisations, as presented in Section 1.3. Besides, the Universitat Politènica de València and the Valencian Research Institute of Artificial Intelligence-VRAIN provide funding and research facilities. Gendelf is a software organisation in the process of formalisation as a spin-off from the research of the PROS Research Centre in the bioinformatics area. Though the transference of the method to the real-world context is outside of this thesis, Gendelf is considered an organisation representative of the target of the method since it requires satisfying the goal of designing a strategically aligned, scalable, and loosely-coupled information system.

The knowledge context is the research areas of model-driven development, goal-based requirements engineering and strategic alignment. Also, the methods and transformation techniques from the baseline method are part of the knowledge context: i*, communication analysis, and the OO-Method, as well as the model transformation techniques. Since enterprise architecture modelling frameworks have addressed most business and organisational structure concepts, they are also considered part of the knowledge context.

The research goal is addressed through a single design cycle, according to Wieringa's design science interpretation (Wieringa, 2014). A set of knowledge

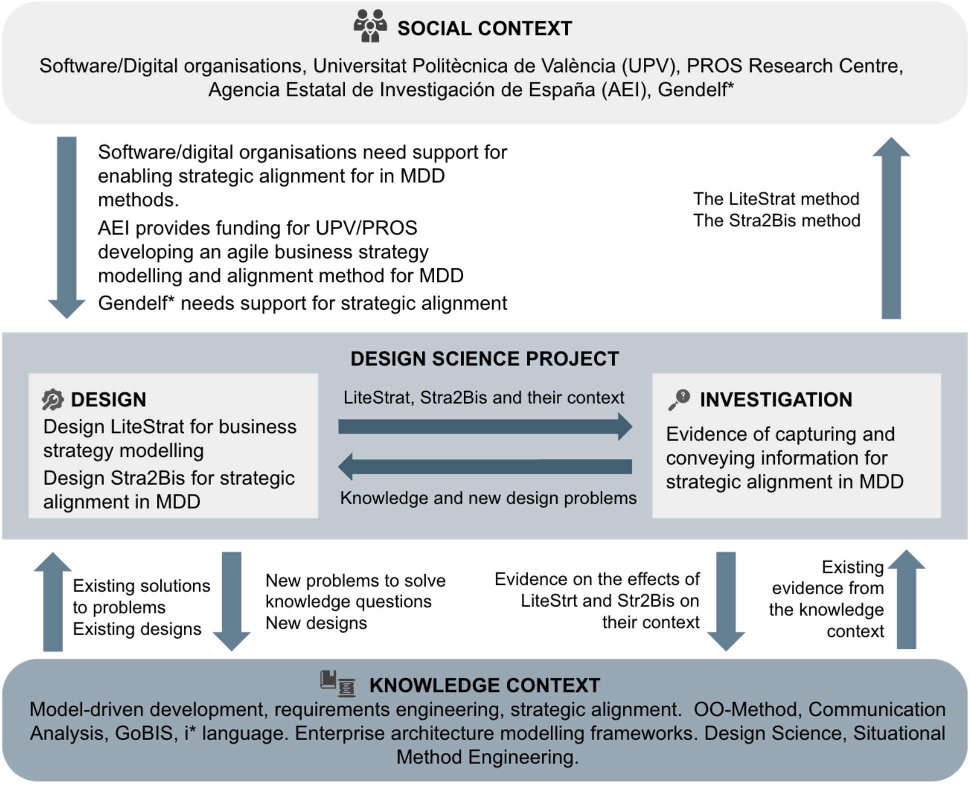


Figure 1.12: Research definition under the Design Science Methodology.

questions and design problems are addressed through the design cycle, which are enunciated in Figure 1.13 and detailed in the subsections below.

Problem Investigation

The problem investigation task focuses on identifying the existing model-driven approaches for designing strategically aligned information systems **(KQ1)** and the challenges of the baseline model-driven software production method for including organisational information on business strategy and organisational structure **(KQ2)**.

The research question KQ1 is addressed by an empirical cycle through the review of related works on the knowledge context defined in Section 1.2. This

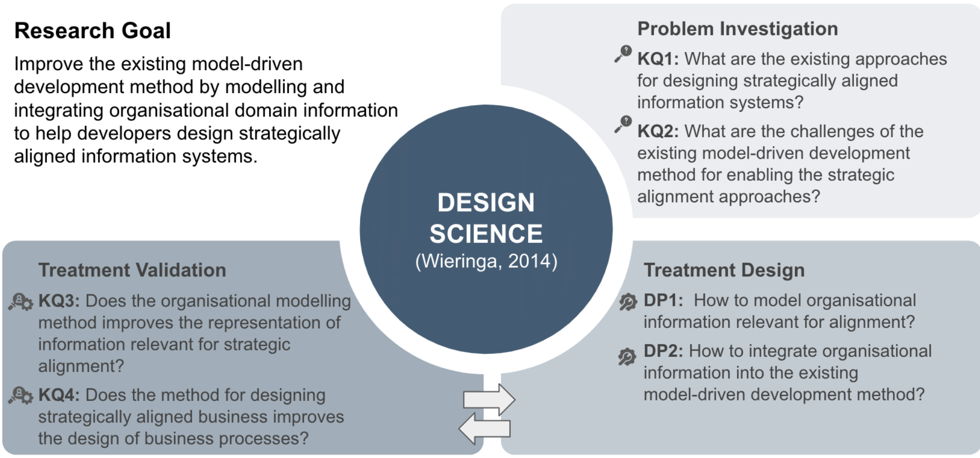


Figure 1.13: Research method structure.

task's outcome is identifying the modelling methods used in strategic alignment and the existing model-driven strategic alignment techniques. We also scope the research in the context of information systems theory and provide a domain conceptualisation through a theoretical framework.

The research question KQ2 is answered through an empirical cycle using the single-case mechanism experiment method and the critical peer validation of the findings (Wieringa, 2014). Single-case mechanisms experiments help study the inner architecture of an artefact. The experiment consists of a problem designed to test whether the architecture of the baseline model-driven software production method supports including organisational information on business strategy and organisational structure for strategic alignment. The issues identified are explained in terms of the architecture of the artefact. Possible solutions are theorised from the modelling methods and strategic alignment techniques identified in the state of the art. Critical peers with expert knowledge of the baseline model-driven software production method validate the findings. To support the analysis, a conceptualisation of the domain is also proposed.

The outcomes of the problem investigation task are the specification of the method requirements according to the SME design methodology described in 1.11.

Treatment Design

The treatment design addresses two design problems regarding the model-driven development of strategically aligned information systems: how to model organisational information relevant for strategic alignment **(DP1)**, and how to integrate organisational information into the baseline model-driven software production method **(DP2)**. The solution for the two problems produces two artefacts that jointly address the main research goal.

For the design problem DP1, the artefact is *LiteStrat* (Noel, Panach, and Pastor, 2021b), an organisational modelling method. The situational method engineering approach is exploited for designing the method out of the concepts of existing modelling frameworks used for strategic alignment, in particular, its existing conceptualisations from ArchiMate (The Open Group, 2022e), Business Motivation Model (The Object Management Group, 2015) and i* (Yu, 2011b). The outcome of this task is LiteStrat (Noel, Panach, and Pastor, 2021b), a lightweight organisational modelling method. From the point of view of SME, the method is constructed following an *assembly-based approach by integration strategy*. Under this approach, the method parts are concepts which belong to different modelling frameworks that share similar engineering goals. LiteStrat is the result of assembling such concepts through a set of guidelines and has a product dimension (the method's metamodel) and a process dimension (the modelling procedure).

For the design problem DP2, the artefact is a method for designing strategically aligned business processes. The method uses model-to-model transformations to convey organisational models' information to business process models, so these models are strategically aligned and could serve as input for designing a strategically aligned information system. In this case, SME is applied following an *assembly-based approach by association strategy*, since a new method is created by positioning the existing method chunks and providing guidelines for their association. In particular, we provide three model-to-model transformation guidelines for integrating LiteStrat (Noel, Panach, and Pastor, 2021b) Communication Analysis (España, González, and Pastor, 2009), the business process modelling method of the baseline model-driven software production method. The outcome of this task is the *Stra2Bis* method (Noel, Panach, Ruiz, et al., 2022), a modelling method that integrates LiteStrat and Communication Analysis through a set of transformation guidelines which materialise strategic alignment. From the point of view of SME, it is an assembled method from existing method chunks (LiteStrat and Communication Analysis), assembled through a set of guidelines.

Treatment Validation

The treatment validation focuses on whether the designed artefacts improve the existing solution and is decomposed into two knowledge questions. The first knowledge question is whether LiteStrat, the proposed organisational modelling method, improves the representation of relevant information for strategic alignment **(KQ3)**. The second knowledge question concerns whether Stra2Bis, the proposed method for designing strategically aligned business processes, improves the design of business processes **(KQ4)**.

The knowledge question KQ3 is addressed in an empirical cycle through a family of experiments aiming to test whether LiteStrat improves the modelling method used for the strategic alignment in the baseline model-driven software production method, i* (Yu, 2011a). The methods are compared in terms of which better represent the organisational domain information relevant for the model-driven development of strategically aligned information systems. The outcome of this task is the empirical evidence of the improvements produced by LiteStrat.

The knowledge question KQ4 is also answered in an empirical cycle through an experiment, which compares Stra2Bis with an unguided (ad-hoc) approach for modelling business processes given the strategy information of the organisation. The outcome of the task is the empirical evidence of the improvements produced by Stra2Bis.

1.6 Outline of the thesis

1.6.1 Overview of the Proposal

This research describes the design and validation of an alternative method for integrating organisational modelling into the baseline method depicted in Figure 1.10. The proposal, depicted in Figure 1.14, is composed of two artefacts: *LiteStrat*, a lightweight organisational modelling method focused on business strategy and organisational structure, and *Stra2Bis*, which integrates LiteStrat with Communication Analysis, in a similar way as GoBis integrates i* in the baseline method.

The proposal offers LiteStrat (Noel, Panach, and Pastor, 2021b) as an alternative to i* for organisational modelling, as it is specifically designed for modelling business strategy and organisational structure, satisfying the social context requirements. LiteStrat proposes to model a *strategic scenario*, which describes the drivers behind the software development endeavour. The strategic scenario addresses definitions that affect the design of business processes and information systems: the strategic ends, the actions to achieve them, and the organisational structure needed to implement the strategy. The scenario does not consider other long-term strategic concerns, such as capacity and resource development.

On the other hand, Stra2Bis (Noel, Panach, Ruiz, et al., 2022) proposes the integration of LiteStrat with Communication Analysis. The two methods are integrated through metamodel mappings and through three model-to-model transformation guidelines. The first guideline aims to transform organisational units in the LiteStrat Model into individual CA business process models. The second guideline transforms organisational unit links into inter-process communications between business process models. The third guideline transforms LiteStrat's objectives into CA communicative events, which report information on the performance of the objectives. This way, the strategy's performance is measured, and the organisational structure is mirrored in the analysis models, aiming to have separated analysis models for every organisation unit, resulting in the derivation of separated conceptual models. Following this approach, the model-driven software production method is enabled to follow the approach for system design of software organisations, previously presented in Section 1.3.

1.6.2 Outline of the thesis

Table 1.1 summarises the research methods and outcomes for the research questions and design problems and details the chapters of this document where the research is exposed. The table also details the main articles that have served to disseminate and validate the research among the scientific community.

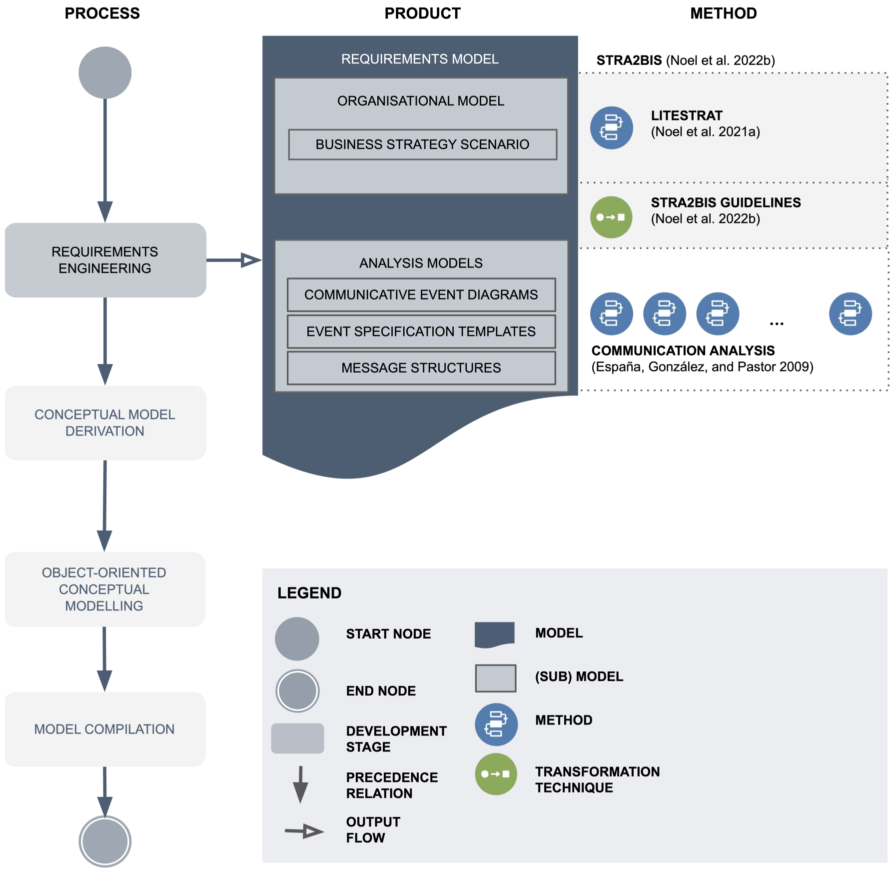


Figure 1.14: Overview the proposal.

Table 1.1: Thesis summary

Design Cycle Task	**KQ / DP**	**Outcome**	**Chapter**	**Article**
Problem Research	KQ1	State of the art	Chapter 2	-
	KQ2	Theoretical framework	Chapter 3	-
	KQ2	Treatment design requirements	Chapter 4	(Noel, Panach, and Pastor, 2022)
Treatment Design	DP1	Organisational modelling method	Chapter 5	(Noel, Panach, and Pastor, 2021b; Pastor, Noel, et al., 2021; Noel, Ruiz, et al., 2021)
	DP2	Business process alignment method	Chapter 6	(Noel, Panach, Ruiz, et al., 2022)
Treatment Validation	KQ3	Validated organisational modelling method	Chapter 7	(Noel, Panach, and Pastor, 2023)
	KQ4	Validated business process alignment method	Chapter 8	(Noel, Ruiz, et al., 2023)
-	-	Conclusions and future work	Chapter 9	-

Chapter 2

Related work

2.1 Motivation

The main motivation of this research is to advance the state of the model-driven development research field by incorporating information from the organisational level that, according to the practices of software organisations, poses requirements for the design of information systems. Although organisational information has been considered in different methods for aligning information technology and business strategy (Henderson and H. Venkatraman, 1999; The Open Group, 2022a; Zachman, 1987), the main target of the revision of related works are requirements engineering (RE) initiatives. This is because the proposed method (introduced in Section 1.6.1) aims to be enacted for a particular software development initiative in which the role of organisational models is to collect requirements affecting the design of the software.

Requirements engineering is the "disciplined application of proven principles, methods, tools, and notations to describe a proposed system's intended behaviour and its associated constraints" (Hsia, A. M. Davis, and Kung, 1993). A corner stone in RE is eliciting *early requirements* in terms of the intentions driving the need for designing or maintaining the information systems which support the organisation's operation. Goal and agent oriented modelling initiatives have served for this purpose, providing frameworks, modelling languages, and methods for eliciting representing the intentions of social actors which are relevant for the information system design. In goal modelling, *Goals*, defined as "is a condition or state of affairs in the world that the actor would like to achieve" (Yu, 2011b), where an actors are "active, autonomous entities that aim at achieving their goals by exercising their know-how, in collaboration with other actors" (Dalpiaz, Franch, and Horkoff, 2016).

Provided the support for representing social technical aspects of information system development, goal models have served for representing social actors intentions with different levels of specification. From representing the goals of actors regardless if they are achieve through technology or no, to business goals related to the information system to be designed, to system users' goals in terms of what they expect from a software system. Other applications are modelling software development processes, business processes, for extending enterprise architecture modelling, and strategic change (Yu, 2009). The flexibility and freedom of goal models have posed them as an input for designing business processes which are aligned with business goals. Consistently with the freedom of goal modelling languages, the goal/process alignment initiatives have covered from very software-specific goals to high-level organisational goals.

Goal and agent oriented modelling initiatives have the potential to represent the organisational structure (as social actors) and business strategy (though goals and associated actions), and the goal/process alignment initiatives have the potential to convey such information to the business process level, which would enable aligning business process and organisational information in MDD methods. In this chapter, we review goal and agent-oriented and goal/process alignment initiatives, to assess the need for a new method.

On the other hand, business strategy and organisational structure information has been conceptualised by enterprise architecture (EA) modelling frameworks. EA frameworks such as Archimate (The Open Group, 2022a), TOGAF (The Open Group, 2018), and ARIS (Santos Jr, Almeida, and G. Guizzardi, 2010) aim to map the whole organisation architecture, providing support for representing business and information technology (IT) elements. Although the scope and purpose of EA varies among different schools of thought (Lapalme, 2011), all relate to how business strategy can be enabled and supported coherently by information technology. While EA definitions must be considered in new software development endeavours, they are not a RE method. However, RE engineering methods have been integrated with EA frameworks to help including EA elements into the software development process.

This chapter reviews the works related to existing model-driven approaches for designing strategically aligned information systems as a means to address **(KQ1)**. The chapter continues as follows. In Section 2.2 we describe goal and agent oriented modelling initiatives, and in Section 2.3 we a analyse works on goal/process alignment. Section 2.4.1 presents an overview of other integration techniques. Finally, Section 2.7 summarises the findings of this section.

2.2 Overview of goal modelling methods

2.2.1 The i modelling framework*

The i* modelling framework provides a goal and agent-oriented modelling language, first introduced in the PhD thesis of Erik Yu in 1995 (Yu, 1997), *"Modelling strategic relationships for process reenginering"*. Among its applications in requirements engineering, it has also been applied for organisational and process modelling, among others (Yu, 2009). Focused on the early requirements stage of the software development lifecycle, i* proposes a socio-technical approach for representing the goals of the information system's stakeholders. The main construct of i* is social dependency, thus the representation of social actors depending on each other to achieve their goals. I* supports two levels of modelling: *strategic dependency model* and the *strategic rationale model.*

In the *strategic dependencies view*, the social actors involved in the development of the information system and their dependencies are represented through *goal*, *task*, *resource*, and *softgoal* links. The actors can be specified as *agents*, which represent real-world organisations or people, or as *roles*, which are abstractions of behaviours performed by a person. In goal dependencies, the source actor trusts the responsibility and know-how of achieving the goal to the target actor. For tasks, the dependency is more constrained to execute an action; similarly, resource dependencies regard achieving a specific resource. On the other hand, soft goals have no clear-cut satisfaction criteria, and checking their achievement is not straightforward since multiple aspects should be assessed. The most recent version of the language, iStar 2.0 (Dalpiaz, Franch, and Horkoff, 2016), departs from the difference between goals and soft goals, replacing soft goals for the *quality* construct, which is used for qualifying goals.

In the strategic rationale model, the details about the actors' inner goals, tasks, resources and qualities are modelled. This allows representing a configuration of *intentional elements* that allow actors to achieve their own goals and to satisfy the goals of their depending actors. The inner elements can be connected by *refinement*, *contribution*, *qualification* and *needed-by* links. The refinement links can be of two types: *OR* or *AND* refinements. In OR refinements, a parent intentional element is connected with two or more children elements, and satisfying at least one child element will satisfy the parent element. Similarly, AND refinements connect a parent to children elements, but in this case, all the children elements must be satisfied to satisfy the parent element. Contribution links specify whether a child element contributes to achieving the parent element and can be typed as *help*, *hurt*, *make*, or *break*. Qualification links connect a quality with the goal it is qualifying. Needed-by links connect a task with a resource needed for the task.

In both the strategic dependencies and strategic models, it is possible to draw the *participates-in* relationship between actors, agents, and roles. This relationship can mean different things; for instance, if the participant-in relationship's source is an agent representing a person and the target is a role, it must be interpreted as *agent plays the role*. On the other hand, this relationship allows representing hierarchical structures in organisations; however, as presented in Chapter 2, it is little to no exploited by current initiatives. Another relationship between actors is the *is-a* link, used for generalisation, which can be applied to specialise roles into other roles or general actors into other general actors.

Figure 2.1 presents examples for i* strategic dependencies (SD) and strategic rationale (SR) models, taken from (Dalpiaz, Franch, and Horkoff, 2016). As can be seen, the SD model shows a goal dependency from the role *Student* to the *Travel Agency* actor for representing the delegating of the goal *trip bundle booked*. The task dependency *buy flight tickets* between the same is also modelled but through a task dependency since it implies a specific action. Other relationships that can be seen in the SD model are participates-in and is-a. On the other hand, the SR model details the student's inner intentional elements. The goal *travel organised* is refined through AND relationships, meaning that it is needed to get the *authorisation obtained* and the *trip booked*. An example of OR refinement is shown for achieving the *request prepared* goal, which can be satisfied by *filling in the paper form* of *filling in the online form*. An example of the needed-by relationships is shown for the task *pay for tickets*, which requires the resource *credit card*. The qualification relationship can be observed in the *quick booking* soft goal, which qualifies the *trip parts booked* goal.

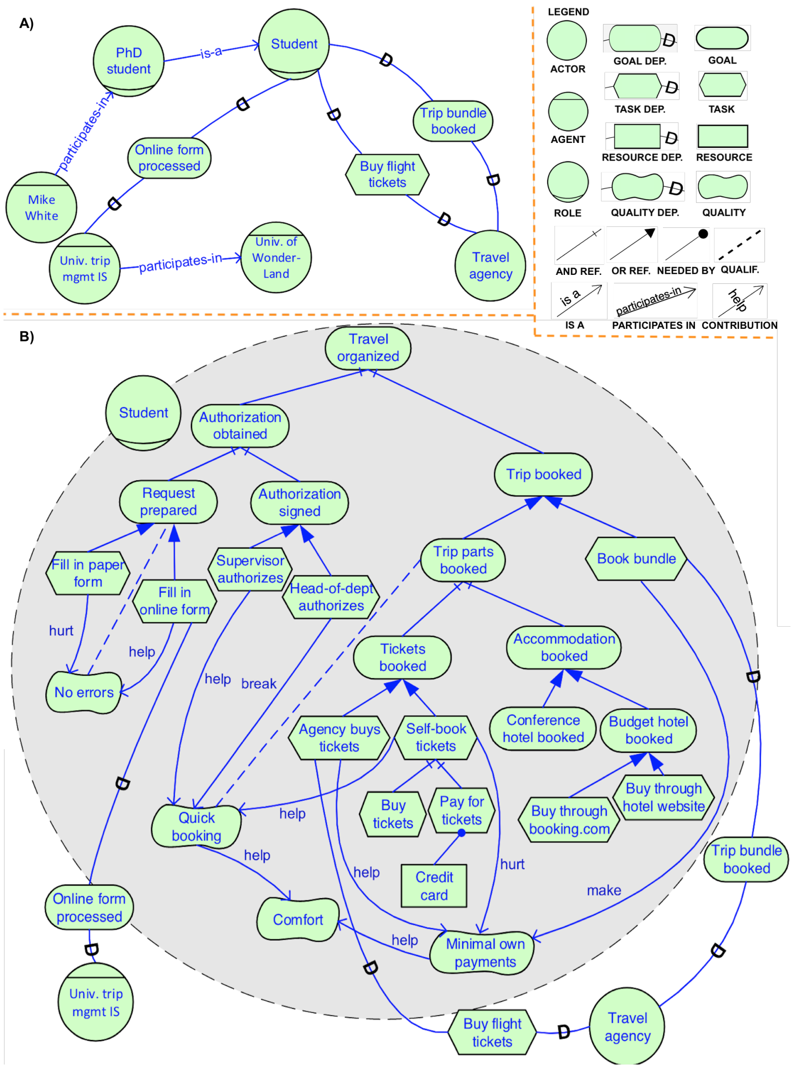


Figure 2.1: Examples of iStar 2.0 models from (Dalpiaz, Franch, and Horkoff, 2016). A) Strategic Dependency Model; B) Strategic Rational Model.

2.2.2 The Goal-Oriented Requirements Language

The Goal-Oriented Requirements Language (GRL) is a “ language for supporting goal-oriented modelling and reasoning about requirements, especially non-functional requirements and quality attributes" (ITU-T, 2018). GRL is part of the recommendation Z.151 of the International Communication Union (ITU), which specifies the User Requirements Notation (URN). GRL is based on i* and the NFR framework (Chung et al., 2012). The main constructs are similar to i*: actors, intentional elements and links between such elements; however, GRL also considers *indicators*. GRL's intentional elements are also similar to the initial version of i*: goals, tasks, resources, and soft goals, and add the *beliefs* construct. Beliefs are used to represent design rationale and support the reflection and justification of the decision-making process.

GRL supports the specification of qualitative or quantitative attributes, which can be automatically evaluated for analysing whether the modelled dependencies and intentional elements satisfy the actors' goals. Some of these attributes are the *importance*, which can be applied to actors, links, and intentional elements, and *indicators* to express quantitative and qualitative real-world values about the satisfaction of the intentional element. Figure 2.2 shows an example of an indicator an actor containing the task *Make connection over internet* with an importance of 100, the indicator *Failure rate for voice connection over internet* with an importance of 40, and a contribution of 100 from the indicator to the task.

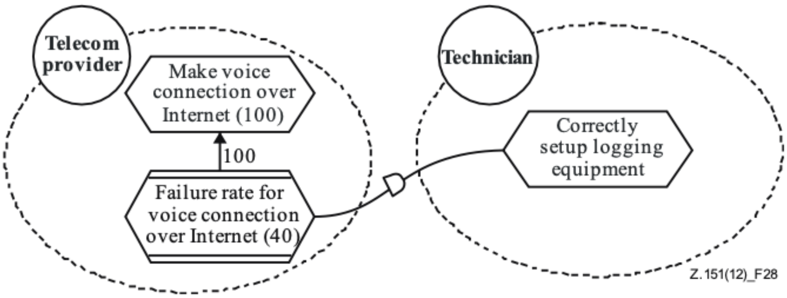


Figure 2.2: Examples of a GRL indicator and importance values from (ITU-T, 2018).

2.2.3 TROPOS

TROPOS (Bresciani et al., 2004) is a software-oriented methodology based on goal and agent-oriented modelling. TROPOS covers the entire software development lifecycle, including early and late requirements, architectural design, detailed design and Implementation. In the early requirements stage, TROPOS addresses similar concerns as i* and GRL. TROPOS includes a modelling language based on i*, although some constructs are changed and new ones are included.

TROPOS includes a modelling language based on i* for the early and late requirements stages, although some important modifications are introduced. For modelling social actors, TROPOS provides just the *actor* construct, which serves to represent i*'s actors, roles and agents. Regarding the intentional elements, TROPOS preserves i* goals and soft goals; however, the task construct is replaced by the *plan* construct. A plan is defined in TROPOS as "a way of doing something". Another construct not present in i* (but in GRL) is *beliefs*, representing an actor's knowledge of the world.

TROPOS provides other diagrams for supporting architectural design to requirements to implementation stages. Notably, in the *architectural design* stage, the *actors diagram* is introduced. The actor diagram allows specifying the delegation of goals (identified in the early and late requirements stages) to sub-actors, and thus defining the organisational architecture, as depicted in the example in Figure 2.3.A. The actor diagram is extended by the *capabilities diagram* (an example is shown in Figure 2.3.B), which depicts the dependencies among the sub-actors of an actor and the dependencies among sub-actors and other actors.

KAOS

KAOS was one of the first goal-modelling initiatives (Dardenne, Van Lamsweerde, and Fickas, 1993) that has evolved to consider system actors, responsibilities, and domain information (Nwokeji, T. Clark, and Barn, 2013). In KAOS, *goals* represent what the system is designed to achieve or the intentions of an actor in a system; among them, *conflicts* or trade-off situations can be represented. Goals can be *requirements* if they have clear satisfaction criteria, or *expectations* if not. KAOS also introduces the *obstacle* concept as an undesirable condition that hinders the satisfaction of a goal, while a *domain property* is a condition that must be held for satisfying a goal. Goals can be refined into more specific goals until they are assigned to *agents* (e.g., humans, machines, etc.), this is, active objects that can operate the system.

In Figure 2.4 an example of a KAOS model from (Matulevičius and Heymans, 2007b) is presented, showing the refinement of goals into subgoals, the responsibility assignment to a software agent (the scheduler), and the operations the agent must perform to satisfy the goals.

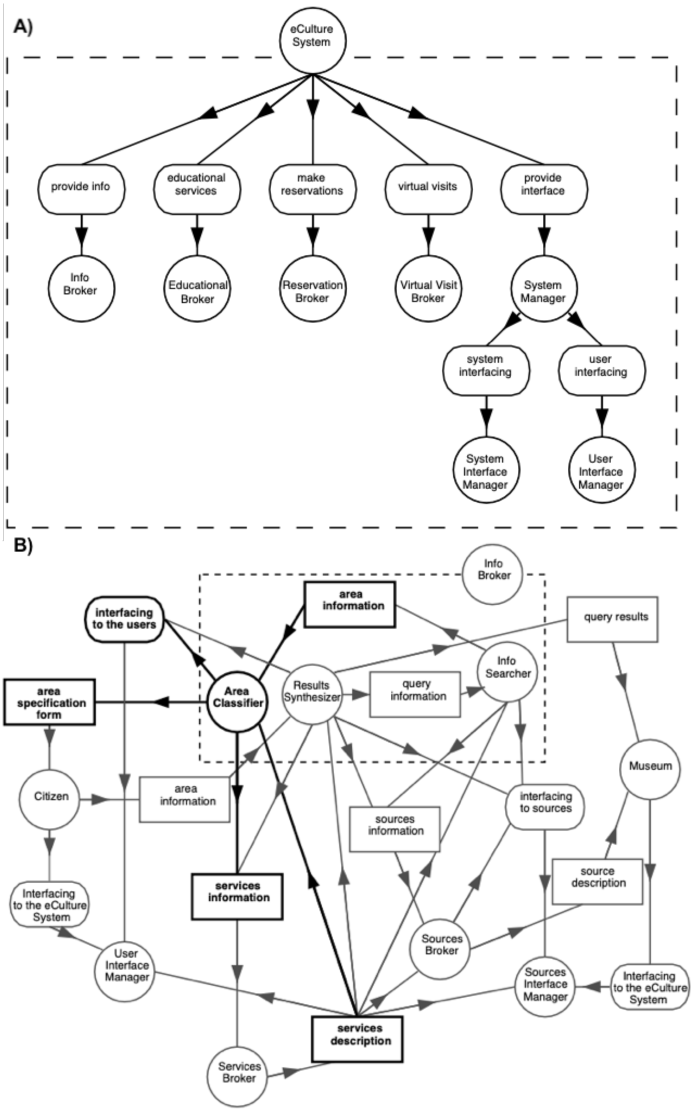


Figure 2.3: Examples of TROPOS actor diagram (A) and capabilities diagram (B) from (Bresciani et al., 2004).

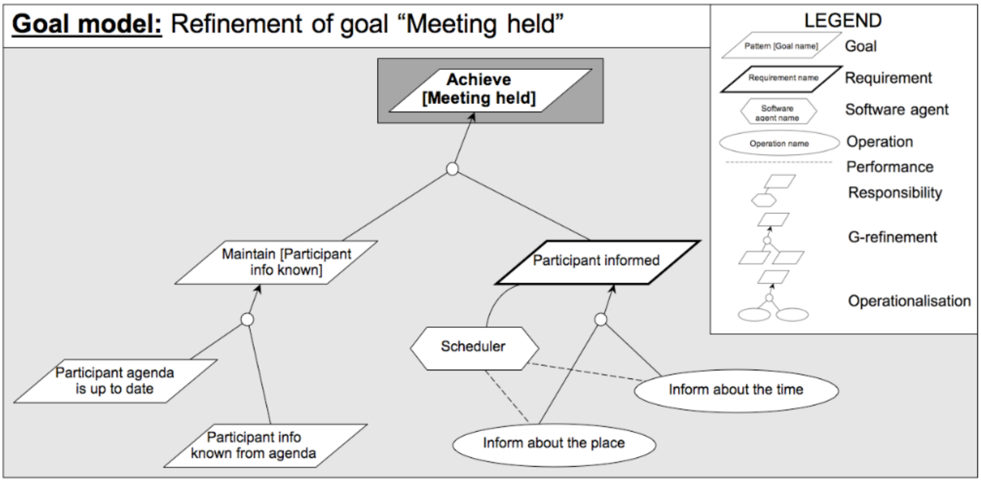


Figure 2.4: Example of a KAOS goal model from (Matulevičius and Heymans, 2007b).

MAPS

A map is a "directed labelled graph consisting of nodes representing intentions and edges to represent strategies" (Henderson-Sellers, Ralyté, et al., 2014). Though maps are widely exploited for modelling processes, they have been applied to capture strategic information at the organisational level (Vara, Sánchez, and Pastor, 2008). The core concept of a map is the *section*, which is composed of a *source intention*, a *target intention*, and a *strategy* that suggest how to achieve the goal of performing the desired task expressed in the target intention, from the source intention. A section is specified through three types of *guidelines*, which provide guidance on what target intention to select given a source intention, what strategy to select if there is more than one available to go to the selected target intention, and how to enact the selected strategy to achieve the selected target intention. A section can be refined as an entire map at a more specific level of granularity.

In Figure 2.5, an example of a map is shown from (Vara, Sánchez, and Pastor, 2008), depicting a booking process.

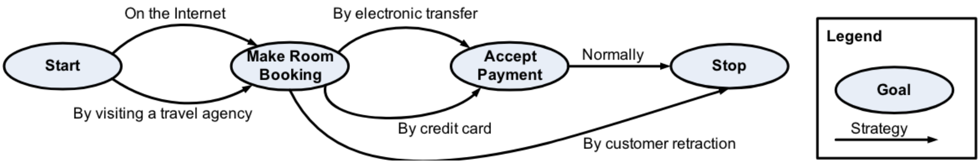


Figure 2.5: Example of a map diagram for a booking process from (Vara, Sánchez, and Pastor, 2008).

2.3 Goal and business process alignment

In Table 2.1 we summarise the reviewed works according to some of their differentiating characteristics. First, the table presents the organisational and business process modelling languages considered. Then, a proposed classification for the alignment approach, inspired in the alignment patterns by Habba, Fredj, and Chaouni (Habba, Fredj, and Benabdellah Chaouni, 2017). The reviewed initiatives can be classified as a*verification* if they provide automatic checking of consistency between models, as a *transformation*, for proposals for automatically transforming organisational model elements into business process model elements and vice-versa, or as an *analysis*, for those articles that offer a conceptual framework for reasoning about alignment. The table also describes the integration mechanism of the initiatives, which can be mapping rules, i.e., guidelines to match elements from one model to another, or a procedure for analysing the consistency of the models, or others. We also identify the most important mapped goal model concepts. Finally, we classify the goals provided by the authors to illustrate their proposals.

We base our classification of goals in Alter's work system theory (Steven Alter, 2013) as depicted in Figure 2.6. We name *system goals* to those who represent goals of the system itself, processes, and participants (as system users), *business goals* to those focused on the customer and in the products and services offered by the organisation, and *organisational goals* to those addressing strategy, environmental or infrastructure issues of the organisation. We excerpt a caption of the highest level goals in the examples provided by the authors to illustrate the classification. Next, we summarise the related works according to their alignment approach.

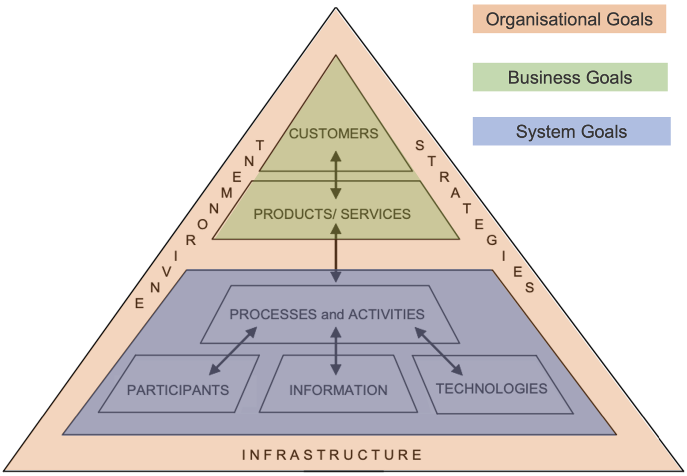


Figure 2.6: Goal classification proposal based on work system theory.

Table 2.1: Summary of initiatives for integrating goal and business process models.

Article	Source ML	Target ML	Approach	Integration Mechanism	Mapped Org. Concepts	Goal Class	Example
(Insfrán et al., 2017)	value@GRL	BPMN	Transformation	Mapping rules	Actors, intentional elements, dependencies/decomposition relationships.	System	"Plan route" (Green route [system])
(Ruiz, Costal, et al., 2015)	i*	CA	Transformation	Mapping rules	Actors and dependencies	System	"Process order" (Supplier)
(Al-Kalbani et al., 2019; Kraiem et al., 2014)	MAP	BPMN	Transformation	Mapping rules	Intentions, strategies	System	"Handle loan request"
(Li and Ye, 2021; Li, Zhou, et al., 2015)	GRL	BPMN (UCM)	Transformation	Mapping rules	Intentional elements and dependencies	Business	"Provide accommodation service" (Travel Agency)
(Horita, Hirayama, et al., 2019; Horita, Honda, et al., 2014)	KAOS	BPMN	Transformation	Mapping rules	Goals, subgoals	-	no example models provided
(Vara, Sánchez, and Pastor, 2013)	BPMN	Goal tree model*	Transformation	Mapping rules	Goal, subgoals	System	"Order processed"
(Sun et al., 2010)	RPGS (Goal)	RPGS (Process)	Transformation	Mapping rules	Goals	System	"Arrange travel plan"
(R. Guizzardi and Reis, 2015)	Tropos	BPMN	Validation	Mapping rules	Goals	Business	"Organize succesful conference track" (PC Chair)
(Amyot et al., 2022; Akhigbe et al., 2016)	GRL	UCM	Validation	Mapping rules	Actors, intentional elements, tasks	Business	"Improve health" (Worker)
(Gröner et al., 2014)	i*	BPMN	Validation	Mapping rules	All elements but "participates in" links	System	"Process order" (Back Store)
(Cortes-Cornax et al., 2015)	KAOS	BPMN	Validation	Procedure	Goals	Business	"Assess the relevance and the cost of the mission"
(Skobtsov and Kalenkova, 2019)	MAP	BPMN	Analysis	Procedure (automated)	Goal	System	"Create Invoice"
(Sousa and Prado Leite, 2014)	i*	BPMN (KPI)	Analysis	Means-ends links, intermediate model	Actor (1 representing the business), roles, intentional elements (tasks, goals)	Business	"Customers be met quickly" (General attendant)
(Vara, Sánchez, and Pastor, 2008)	MAP	BPMN	Analysis	Procedure (Manual analysis)	Intentions, strategies	Business	"Facilitate work item development"

2.4 Model-to-model transformation techniques

In (Cano-Genoves, Insfran, and Abrahão, 2022; Insfrán et al., 2017), thr authors present a proposal based on GRL, aiming to help prioritise the business process activities that must supported by technology according to the value propositions of the system's stakeholders. The initiative proposes adding business value information to GLR models, namely value@GRL, so that it can be traced to BPMN models. The approach is to propagate the importance values for intentional elements and links in GRL models to BPMN models, so each BPMN activity has a value based on the goal model. To do this, mapping rules for transforming GRL elements into BPMN models are proposed. Actors are transformed into process pool lanes, intentional elements (tasks) into process activities, and goal links into process flow links. It is worth noting that, from a RE perspective and according to the concern-based requirements taxonomy (Glinz, 2007), the proposal does not contribute with system requirements but with *project requirements*; however, under some taxonomies, project requirements are considered among non-functional requirements (Kotonya and Sommerville, 1998).

Kraiem et al. (Kraiem et al., 2014) present a mapping from a MAP model to a BPMN model as a means to convey higher-level intentions and strategies to the business process level. The approach is to start modelling the high-level intentions and strategies using MAP models and then refine the MAP model sections of the model into more detailed MAP models. MAP models that cannot be further specified can be transformed into BPMN models following mapping rules. Higher-level MAP sections are transformed into BPMN sub-processes; for refined MAP models, strategies linked to intentions are transformed into BPMN tasks. BPMN gateways are inferred from the different configurations of strategies and intentions. From a RE perspective, the proposal provides *functional requirements* as it helps identify the functions and behaviour of the system (Glinz, 2007).

The proposal by Ruiz et al. (Ruiz, Costal, et al., 2015) presents the integration of i* models with Communication Analysis (CA) (España, González, and Pastor, 2009) models to integrate business goals and business process models. The approach is based on the idea that dependencies among actors at the organisational level require communication among such actors at the business process level. I* model elements are transformed into CA model elements through a set of mapping rules. An i* model is transformed into a CA process, and each dependency among actors in i* is transformed into a communicative event between the same actors in the CA model. This way, business process models are guaranteed to consider the communication between actors for specifying and informing the achievement of goals. From a RE perspective, the approach helps elicit *functional requirements* since it supports completing the specification of business processes.

In (Li, Zhou, et al., 2015), the authors present a formal approach for transforming business goal models in GRL to business process models in BPMN. The method defines the metamodels for the modelling languages and a set of mapping rules for transforming goal models' intentional elements and dependencies into business process model elements. The method aims to convey business goals and dependencies among actors into business process activities and flow elements passing through a business scenario, modelled in the Use Case Maps notation (Buhr and Casselman, 1995). The article presents examples regarding the business goals of the organisation that provides the system, the dependencies with external organisations, and the goals of such organisations. From a RE perspective, the method aims to collect *functional requirements* from the organisational level: the goals and dependencies are mapped into BPMN subprocesses and collaborations, which are meant to be supported by the information system.

The work by Horita et al. (Horita, Honda, et al., 2014) proposes an approach for transforming goals and sub-goals from KAOS models into BPMN modelling elements. The proposal offers a set of transformation rules to be applied incrementally from top to lower-level goals. The mapping rules consider KAOS patterns as source elements, which must be interpreted by the analyst to be transformed into the correct business process model elements. Like other approaches, from a RE perspective, the proposal supports the elicitation of *functional requirements* since it helps identify business process elements to be supported by the information system.

In (Sun et al., 2010), Sun et al. describe a transformation approach to generate business process models from goal models represented using a proposed notation named O-RGPS. O-RPGS proposes a goal modelling language that decomposes functional or non-functional *goals* into more specific *operational goals* and represents whether a decomposed goal is mandatory, optional, or alternative, as well as dependencies and exclusions between goals. O-RPGS' business process modelling notation represents *composite* and *atomic* processes and precedence relationships among them; split and join gateways and start and end events complement the language's elements to specify the process flow. The method proposes mapping rules to transform goals into composite sub processes, operational goals into process tasks, and the goal links into precedence and gateway elements to represent the business process flow. The method supports the identification of functional*functional requirements* since the transformation ensures the inclusion of business process elements in the information system to be developed.

De la Vara et al. (Vara, Sánchez, and Pastor, 2013) propose a different approach of integration based on generating goal models from business process models using BPMN. For goal modelling, a goal-tree notation is proposed that provides the *goal* concept and AND and OR links to decompose them into more operational goals. The authors propose a set of guidelines for deriving goal elements from business process elements. BPMN's diagrams, sub-processes, loops, branching structures and data objects are mapped into goals, while process tasks and events are mapped into tasks at the goal modelling level. The article also proposes heuristics for refining the generated goals by analysing whether branching structures have alternative execution flows or not to generate OR or AND refinements, respectively. The article discusses when goal and business process models should be combined, as in the proposal. The recommendation is to use goal and business process models jointly when goals are expected to change consistently with new strategies from the organisational level. From a RE perspective, this proposal does not help identify new system requirements but fosters traceability between business goals and processes.

2.4.1 Other integration techniques

As seen in Table 2.1, another approach for integrating information from the organisational level into the business process level is *validation*. Under this approach, we classify proposals on which goal and business process models are modelled separately, thus, without using model-to-model transformations. The techniques propose checking goal and business process model consistency based on mapping rules or analysis procedures.

In (R. Guizzardi and Reis, 2015), the authors propose modelling goals using TROPOS and business processes using BPMN. The initiative seeks to help analysts identify which business process activities help achieve the organisation's goals and reasoning about the impact of such activities on the organisation's top goals. The article proposes an analysis procedure named goal and activity alignment, where the analyst selects the leaf goals of a goal model and assign them to an activity or sub-process in the BPMN model. As a result, the analyst could verify whether the organisational goals are being satisfied by current business processes; moreover, using TROPOS goals' importance attributes, the impact of business process activities for organisational goals can be estimated. From a RE perspective, the verification of business processes completeness supports eliciting *functional requirements*, with the impact analysis does not provide system requirements but helps identify *project requirements*, according to Glinz' requirements taxonomy (Glinz, 2007).

The work presented in (Gröner et al., 2014) seeks to validate whether business process activities realise business goals. The authors propose an automated approach based on the formal definition of the modelling notations and the mapping rules among them. The mapping rules help identify inconsistencies between the models, particularly for the purpose of business process orchestration and choreography. The proposal focuses on the goal model's tasks and their consistency with business processes, while other goal elements, such as goals and soft goals, are outside the scope of the article. From a RE point of view, even though the proposal does not recommend a specific way for resolving the inconsistencies, it can be exploited to redesign business processes so they fix to goal requirements, so it could help identify *functional requirements*.

Cortex-Cornax et al. (Cortes-Cornax et al., 2015) present another validation technique for checking the consistency of KAOS goal models and BPMN business process models. The article describes an analysis procedure to align goals and business processes by identifying *intentional fragments*, which are elements of the BPMN model which can be traced to one or many goals. The analysis procedure yield to the classification of intentional fragments as *justified* (a set of business process elements can be traced to a goal) or as *potential* (a set of business process elements that is not related to a business goal). As in the previous proposal, checking the consistency of business process models helps analysts to identify *functional requirements*, by identifying process redesign needs for matching business goals.

The user Requirements Notation (URN (ITU-T, 2018) is a standard by the International Telecommunications Union for requirements specification for requirements specification. URN proposes model goals using GRL and business processes using Use Case Maps (Buhr and Casselman, 1995). Additionally, the language supports user-defined, typed links that can be used to connect any pair of modelling elements, as well as adding tags and profiles for adding metadata to the modelling elements. Although the specification recommends two types of analysis (GRL model evaluation and UCM path transversal), it does not recommend a systematic approach for analysing the consistency of goal and business process models. However, in their revision of twenty years of application of the approach, Ammyot et al. (Amyot et al., 2022) visit URN applications for aligning goals and processes, including most of the previously reviewed works using GRL. A worth noting proposal is the one by Akhigbe et al. A(Akhigbe et al., 2016), which uses both GRL and UCM notations, connecting their elements through the user-typed links. The initiative proposes a set of consistency rules to check the alignment of goals and business processes. From an RE perspective, the initiative helps completing business process model elements, supporting the elicitation and validation of *functional requirements*.

Another approach for integration is providing a conceptual framework to analyse goal and business process models for their alignment without implying any kind of mappings or automation. We refer to this approach as *analysis* proposals. For instance, de la Vara, Sanchez, and Pastor (Vara, Sánchez, and Pastor, 2008) present a method for analysing the purpose of business processes based on organisational goals. The method considers an initial business process model representing the current organisation status. Then, goal modelling is performed using MAPs, considering organisational actors' intentions towards using the system as goals and the high-level activities supported by the system as strategies. This step might reveal that some goals have not been achieved yet, yielding the design of the existing business processes. The next step is to operationalise the MAP model by analysing what organisational goals are related to which business process model elements and identifying whether existing business process elements must be removed, maintained, or added. From an RE perspective, the approach helps elicit *functional requirements* since it allows identifying business process elements to be supported by the information system.

Another proposal for helping analysts to align organisational goals and business process models is presented by Sousa and Prado Leite (Sousa and Prado Leite, 2014). The GPI (goals, process and indicators) approach introduces an intermediate model between goal and business process models to specify *key performance indicators* as a means for bridging the gap between high-level organisational goals and more specific goals of organisational actors or, more precisely, organisational roles. In GPI, an i* goal model represents the high-level organisational goals and processes (using the i*'s *task* construct). The processes are refined into an intermediate model that links processes with the participating roles; more specific goals and sub-processes are modelled for each role. The intermediate model serves as input for business process modelling in BPMN, where each role is represented as a pool lane, and a process flow is modelled to specify the role's subprocess. An interesting outcome of the approach is that it supports validating whether business process models have the activities and information assets to assess the organisational goals that it is supposed to satisfy. From a RE perspective, it contributes to eliciting *functional requirements* for completing the business process models.

2.5 Discussion of goal and process alignment initiatives

Next, we discuss the reviewed initiatives in the context of our research , this is, how they could possibly support the integration of business strategy and organisational structure information into an MDD method to align business processes.

2.5.1 Including business strategy and organisational structure information

Regarding the integration of organisational structure, none of the articles proposes an explicit characterisation of the organisation under analysis and its inner organisation units. Some articles recommend and exemplify modelling the organisation under analysis as an actor (Amyot et al., 2022; Gröner et al., 2014), no specific constructs, tags or metadata profiles are proposed. Even though the *participates-in* link is present in most of the agent-oriented notations, none of the reviewed initiatives exploit it for preserving hierarchical relationships. On the other hand, most of the initiatives map the dependencies between actors at the organisational level (that could represent organisation units) into business process level links. However, most proposals exemplify the dependencies between actors at a very operational or system-specific level (dependencies to perform a task or to achieve a system goal), and not in terms of what are the actors (in this case, organisational units) high-level goals or *organisational commitments* (see Section 3.3.2).

Regarding the representation of business strategy (see the conceptual framework for organisational ends and means in Section 3.3.2), most initiatives present stakeholder goals at the operational or system level in their examples, which is appropriate for the scope of the proposed methods since they do not claim to provide strategic or high-level alignment (Ruiz, Costal, et al., 2014; Sun et al., 2010; Insfrán et al., 2017; Vara, Sánchez, and Pastor, 2013; Mario Cortes-Cornax et al., 2012). On the other hand, some alignment initiatives, which claim connecting high level goals with the operational levels, consider an iterative refinement of goals to get from strategic to operational level goals (G. Guizzardi and Wagner, 2004; Vara, Sánchez, and Pastor, 2013); however, they do not provide a clear cut criteria for the refinement, so strategic and operational goals are not conceptually distinguishable. Some approaches clearly separate organisational ends and means from process specific goals and actions by introducing intermediate models to bridge the gap (Sousa and Prado Leite, 2014; Li, Zhou, et al., 2015). We think this conceptual differentiation is helpful both for the method's users and for the automatic integration of business strategy into a MDD method.

2.5.2 Supporting system requirements elicitation

As presented in Table 2.1, most approaches integrate goal and process models aiming for help eliciting functional requirements by improving the quality of business process models. For quality we refer to business process model's completeness and alignment, this is, having all the needed process elements to achieve the goals, and not having unjustified process elements according to the goal models. Transformation approaches ensure that critical goal level elements are present in business process models, while validation and analysis approaches provide guidelines for checking consistency, helping identifying process re-engineering needs. In almost all the cases, the business process is presented as a single model and diagram, and no guidelines for structuring multiple processes or for analysing their dependencies is mentioned.

A notable exception is the work by Gröner et al. (Gröner et al., 2014), which addresses business process orchestration and choreography consistency with organisational dependencies. Under the Glinz taxonomy for non-functional requirements (Glinz, 2007), this approach helps eliciting *specific quality requirements* (o "ilities"); according to ISO/IEC 25010 standard, the specific quality regards requirements *interoperability*, defined as "The ability of two or more systems or components to exchange information and to use the information exchanged" (ISO/IEC, 2010).

As presented in Section 1.3.5, aligning organisational information and the information system requires preserving organisational structure information from organisational models to information system models, going through the business process models. As reviewed in the related works, this type of requirement (a*constraint requirements*, according to Glinz (Glinz, 2007)), is not supported by the existing goal and process alignment initiatives, setting a challenge for existing MDD methods.

2.5.3 *On the state of the art of goal and business process alignment approaches*

As reviewed, most of the related works were produced in the middle of the last decade; however, they are still relevant today. The reviewed proposals set the conceptual foundations for goal and process alignment, which has been exploited with automated, state-of-the-art techniques. For instance, an automated analysis method based on process mining is presented in (Skobtsov and Kalenkova, 2019), where heuristic approaches are applied to compare business process models in BPMN efficiently. The proposal aims to compare a BPMN model generated by process mining against a reference business process model in BPMN, previously designed in alignment with a MAP goal model using the method by Kraiem, Kaffela, and Khanjari (Kraiem et al., 2014), with a business process model mined from the information system logs. Similarly, the alignment approach by Horita et al. (Horita, Honda, et al., 2014) has been applied to analyse the alignment of business process models extracted from logs of organisational daily operations through process mining techniques (Horita, Hirayama, et al., 2019). Using a pattern-based method, the authors propose to repair goal models by dealing with the repair of business process models.

While the works reviewed provide a varied set of approaches for integrating goal model information, they do not support the alignment with organisational strategy and structure. We think there are three main reasons for this, based our claim on the results presented in Table 2.1 and on the papers review.

Firstly, most approaches exclusively focus on eliciting *functional requirements*, since they seek to have complete and consistent business process models in terms of activities, tasks, and other elements aligned to goal models. This leaves outside the scope deriving the *constraint requirement* needed for mapping the structure of the organisational actors (unit, roles) to the business processes and information systems structure.

Secondly, although some of the reviewed works explicitly mention their purpose of aligning processes with organisational goals, all the approaches presented methods, examples, or case studies addressing system or business goals, and not organisational goals. This is not a limitation for most proposals since they work with goals that can be straightforwardly addressed by a business process, thus helping elicit functional requirements. However, organisational goals are high-level, strategic goals that can affect several business processes. Mapping organisational goals business processes could require an explicit intermediate level of goals, such as introducing intermediate models as in (Sousa and Prado Leite, 2014; Li, Zhou, et al., 2015), or by having different constructs to distinguish high-level and more operational goals, such as *goals* and *objectives* in BMM (The Object Management Group, 2015).

Finally, except for GoBIS (Ruiz, Costal, et al., 2015), the alignment approach considered in our baseline MDD method, none of the related works provide a semantic rationale for connecting the goal and business process modelling constructs underlying the proposed mapping rules and analysis processes. This problem is shared by most model integration techniques (Mustafa and Labiche, 2017). In GoBis, a pivot ontology is used as an intermediate conceptualisation to connect goals and business processes, providing a sound conceptual integration.

2.5.4 Enterprise architecture and goal models

Besides goal modelling, other initiatives have combined frameworks addressing business strategy concerns. Business plans (modelled in Business Motivation Model (The Object Management Group, 2015)) (BMM) have been used jointly with i* to add intentionality to the process of enterprise architecture construction (Yu, Strohmaier, and Deng, 2006). The approach proposes an enterprise architecture construction process that introduces i* modelling steps. First, the current EA architecture is jointly modelled in with i* and BMM's business plans, provided the strategy-specific BMM's constructs for representing the organisational ends and means, such as *strategy*, *tactic*, *goal*, and *objective*, among others. The process then exploits i* capabilities to analyse business problems and rot causes, and then develop different EA configurations to satisfy strategic goals.

In (Engelsman, Quartel, et al., 2011) a proposal to provide intentionality modelling to EA is presented. The aim of the proposal is representing the underlying motivation of the EA construction, representing the stakeholder concerns and the goals that related to these concerns. 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 (The Open Group, 2022a), 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 (Wautelet, 2019), the author presents MoDrIGo, a model-driven framework for information technology (IT) governance. MoDrIGo enables the representation of business and IT objectives within a strategy and facilitates the assessment of how business IT services, whether existing or under development, align with both business and IT goals. Business strategy and the IT strategy are modelled using NFR trees. A model based on i* represents the portfolio of IT services needed to achieve the strategic goals, and i* models are designed to analyse how the management level goals can be fulfilled through other actors' contributions.

The above initiatives exploit goal models expressiveness of social actors' intentionality to help bridging the gap between business strategy and IT architecture. However, organisational structure is not considered in the above approaches, and the the integration with the operational level (i.e., business processes) is not addressed.

2.6 Works related to the baseline method

As introduced in Section 1.4.1, the baseline method integrates three methods for addressing the organisational, business process, and system modelling levels: i*, Communication Analysis, and the OO-Method. Even though the baseline method is part of the context for the present research, there is sound rationale for them to compose model-driven software production method from requirements to code, which are listed below.

- Organisational level business strategy representation: as previously reviewed i* and the family of agent and goal oriented modelling frameworks are the most used for representing business goals in RE initiatives. Moreover, i* is one of the most used frameworks for business goals modelling for strategic alignment (Yu, Strohmaier, and Deng, 2006; Johannesson, 2007; Louaqad and El Mohajir, 2014). Despite its many applications for modelling organisational and business goals, it is worth noting that strategic concepts such as strategy, influence, motivation, and tactics are not present in i* but have been mostly covered by enterprise architecture frameworks such as Archimate (The Open Group, 2013) and BMM (The Object Management Group, 2015).
- Information system level code generation: the modelling framework for the information system level must support code generation. As reviewed Sebastian et al. (Sebastián, Gallud, and Tesoriero, 2020), there are more than 50 MDA-based initiatives with code generation, being UML the most used language. However, many research initiatives lack of industrial adoption evidence. We choose the OO-Method (Pastor and Molina, 2007)

(OOM), since its tool support[1] has been applied for more than a decade in several software projects; also, OOM uses diagrams that are similar to UML's class and estate machine diagrams.

- Semantic consistency: As researched by Mustafa and Labiche (Mustafa and Labiche, 2017), one of the most challenging aspects to connect different modelling languages is to have meaningful connections among them. Even though BPMN is one of the most used language for business process modelling, we opted for the Communication Analysis method (España, González, and Pastor, 2009) for business process modeling, since it has been methodologically integrated with OOM (España, 2011) and with i* (Ruiz, Costal, et al., 2015).

The stack of methods is depicted in Figure 2.7. Below, we describe the methods and the integration techniques.

At the **organisational modelling level**, the baseline method considers modelling organisational goals and strategic elements using i* (Yu, 2011b), which we previously described in Section 2.2.1.

For the **transformation of goal models to business process models**, the baseline method considers the GoBIS technique (Ruiz, Costal, et al., 2015). GoBIS, previously described in Section 2.4, uses FRISCO (Falkenberg et al., 1998) as a pivot ontology (Giachetti, Valverde, and Marín, 2012) to ensure ontological consistency i* and Communication Analysis models. Its main construct is that the satisfaction of a dependency between actors involves a communicative interaction between these actors. GoBIS presents nine guidelines to cover the different types of dependencies of i* and map as much information as possible about the process flow. The GoBIS approach provides semi-automated assistance for the analyst in the model transformation process.

For **business process modelling**, we use the Communication Analysis (España, González, and Pastor, 2009) method (CA). CA is a communication-centred business process modelling method. Its main construct is the communicative interaction among actors. A communicative interaction is a fine-grained unit of valuable information about the problem space in the business process context (A. Gonzalez, Espana, and Pastor, 2009). The communicative interaction involves a primary actor that triggers the communication, a communicative event, that details the communication requirements, one or many receiver actors, that are the target of the communication, and ingoing and outgoing interactions. The communicative events can be specified in terms

[1] Integranova Software Solutions - http://www.integranova.com/es/

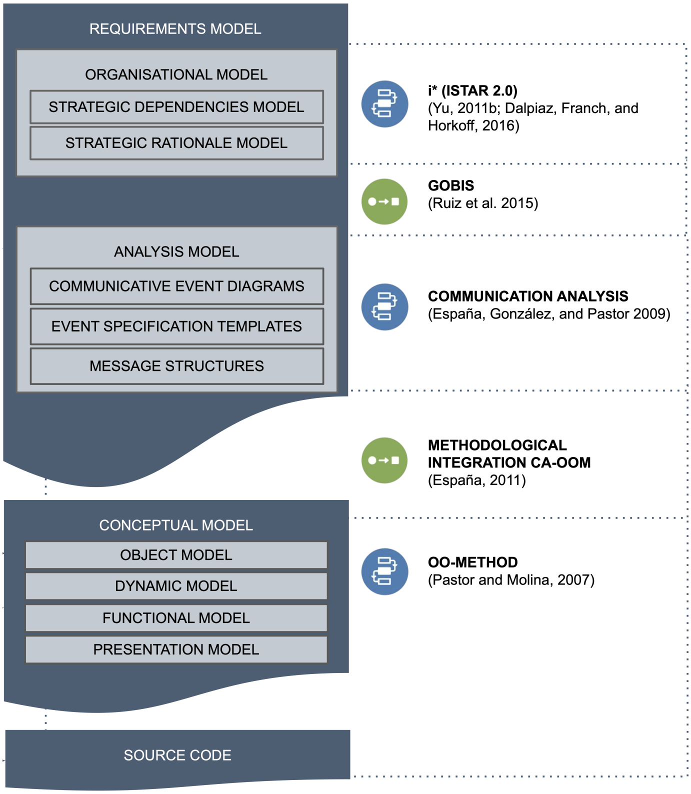


Figure 2.7: Baseline method.

of the contact, content, and reaction system requirements for supporting the communicative event. The ingoing and outgoing interactions can be specified in terms of the structure of the messages, allowing to specify data fields, types, and structure.

For the **transformation of business process models into information system models**, we use the technique presented in (España, 2011), which allows the derivation of OOM structure, behaviour, and functional models. The main idea is that the structure of the messages interchanged among the actors can be mapped into classes, attributes, and their relationships. Moreover, the process flow and the actors' interactions can be mapped into methods and, partially, into functionality. The technique also ensures semantic consistency among CA and OOM using FRISCO (Falkenberg et al., 1998) as pivotal ontology (Giachetti, Valverde, and Marín, 2012).

Finally, for the **information system modelling level**, the baseline method uses OO-Method (OOM) (Pastor and Molina, 2007). OOM is an example of an MDD method: it is a software production method that is based on a formal language for the object-oriented specification of information systems called OASIS (Pastor, Salavert, and Cerdá, 1995). OOM is composed of four modelling views: the object model, the dynamic model, the functional model, and the representation model. The object model allows specifying the system structure using object orientation, while the dynamic model represents the system's behaviour. The functional model allows to specify business logic, and the representation model permits defining abstract user interface components for using the system services. The specific platform requirements are modelled as attributes of the information system. The tool support for the OO-Method is INTEGRANOVA Model Execution System (Integranova Software Solutions, 2023), an industrial tool that fully supports OOM and generates codes in several technological platforms both for the back end and the front end of the information system.

2.7 Summary

Chapter 2: Related works

- This chapter reviewed works on existing model-driven approaches for designing strategically aligned information systems, addressing **(KQ1)**. We reviewed requirement engineering proposals which might help integrate organisational information into business process models. Since goal and agent-oriented modelling frameworks are useful for representing social actors and their intentions, we reviewed five goal modelling frameworks to understand their approach and differences.

- The overall approach of the works is the alignment between goals and processes. We identified different alignment techniques (transformation, verification, analysis), mostly supported by mapping rules. The aim of most of the initiatives was to improve the completeness and justification of business process modelling, thus helping elicit functional requirements of the supporting information system.

- Even though most of the related works were proposed in the middle of the last decade, their conceptual foundations are relevant today for applying state-of-the-art techniques for goal and process alignment, such as process mining and automated analysis through heuristics. However, new conceptual contributions addressing the need for integrating organisational structure or mapping high-level organisational requirements have yet to be proposed.

- Other reviewed works aimed to integrate goal and enterprise architecture methods, mostly to provide intentionality analysis to the enterprise architecture construction process. However, the scope of these initiatives is broader than requirements engineering since it seeks to align the whole organisation's strategy and IT architecture.

- As a conclusion, none of the goal and process alignment methods support including organisational information for goal and process alignment. Although goal modelling frameworks (such as i*, included in the baseline method) could possibly be applied to model and map organisational strategy goals, there is an open challenge in preserving organisational structure for business process alignment.

Chapter 3

Theoretical framework

3.1 Motivation

Since MDD puts models at the centre of the software development process, modelling methods and languages must provide a sound basis for helping its users to produce models with syntactic, semantic, and pragmatic quality (Lindland, Sindre, and Solvberg, 1994). Regarding the semantic quality of models, one of the critical elements is the ontological commitment needed to characterise a complex domain precisely. A complex domain such as business strategy and organisational structure can introduce conceptual ambiguities, potentially harming the quality of the modelling language and the overall quality of the method. The modelling language of a modelling method can be seen as a representation of a domain conceptualisation (G. Guizzardi, 2013).

Conceptual frameworks help to clarify the ambiguities inherent to the domain to be addressed (i.e., *what is a strategy?*) by providing precise and well-founded definitions. The conceptualisation must be based on well-known conceptual frameworks to be unambiguously interpreted. In particular, ontologies such as FRISCO (Falkenberg et al., 1998) have been used to conceptualise elements of information systems and software development methods. Foundational ontologies such as UFO (G. Guizzardi, Botti Benevides, et al., 2022) are used to disambiguate the semantics of modelling languages, particularly in domains such as business modelling (G. Guizzardi and Wagner, 2004) or goal-oriented requirements engineering (Bernabé et al., 2019).

This chapter proposes the base definitions relevant for characterising the organisational information needed to satisfy the social context requirements for aligning business processes and software. First, we scope the definition of organisational information based on the work system theory (Steven Alter, 2013), stating the difference between organisational, business, and system requirements. Based both on work systems theory and the social context requirements, we propose three complementary conceptual frameworks for characterising 1) the strategic scenario that drives the need for the organisation to adapt its processes and systems to create a new value offer, 2) the elements defining what the organisation needs to do to provide the new value offer -the business strategy plan-, and 3) the organisational structure needed to perform such actions. The aim and scope of these definitions is to describe organisational information to be included in a model-driven software development method. While previous work on enterprise ontologies exists (Uschold et al., 1998), we differ from enterprise architecture conceptualisations since our aim is not to describe the organisation as a whole in a static way but to consider scenarios in which the organisation makes strategic decisions that generate requirements for the (re)design of business processes and information systems.

To define the conceptual frameworks, we follow an approach inspired by Bernabe et al. (Bernabé et al., 2019), who characterised the Goal-Oriented Requirements Engineering domain by proposing concepts which extended foundational concepts from the Unified Foundational Ontology (UFO). Similarly to the work by Bernabe et al., we use UFO-C as a reference framework, which addresses social agents and their intentions.

The conceptualisations presented in this chapter aim to support the analysis of the challenges of the baseline model-driven software production method for including organisational information on business strategy and organisational structure, as a means to address **(KQ2).** This chapter first presents in Section 3.2 the work system framework as a theoretical foundation for organisational modelling. Then, in Section 3.3, we present the proposed conceptual framework in four parts. The first one, presented in Section 3.3.1, describes a subset of fundamental UFO-C concepts upon which the proposed conceptual frameworks are built. The second one, in section Section 3.3.2, describes the conceptual framework for strategic scenarios, i.e. the specific configuration of the organisation, the actors in its environment, and the constituent units of the organisation, which will generate the requirements for business process redesign and the information systems that support them. The third one, Section 3.3.2, presents the conceptual framework for characterising the organisation's ends and means, which are defined in the context of the strategic scenario, and the fourth one, Section 3.3.2, proposes a conceptual framework for characterising the organisational structure, particularly its hierarchical relationships and the communication needs between the organisation's constituent units. Finally, Section 3.4 summarises the chapter.

3.2 Work system theory and organisational modelling

3.2.1 Overview of the Work Systems Theory

Work Systems Theory (WST) (SL Alter, 1995) emerges as a response to the view of systems as just technical artefacts disregarding their importance as a key business element. WST integrates business and organisational elements as a central part of the system and not just as its context of use. WST aims to help analyse systems by focusing on generating business results, leading to better requirements for the system (Steven Alter, 2013).

A work system (WS) is "*a system in which human participants and/or machines perform work (processes and activities) using information, technology, and other resources to produce specific products/services for specific internal and/or external customers*" (Steven Alter, 2013). Information systems (IS) are a special case of WS where all the activities and processes are focused on processing and managing information (Steven Alter, 2008). WS theory provides a static characterisation of the system, the WS framework, and a dynamic view of how the system changes over time, the WS life cycle. Figure 3.1 depicts the WS framework. All the elements in the framework are relevant for designing systems, although they are focused on business concerns.

Processes and activities occur to produce the WS services and products; otherwise, the WS would do nothing. *Participants* are humans that work in the WS, whether they use the IS or not. *Information* in a WS considers information entities used, created, or processed by the WS, regardless of whether it is computerised. *Technology* includes tools used by the WS participants and automatic processes.

Products/services regards information, physical or actions produced by the WS for its customers. *Customers* are entities having purposes beyond their participation in the WS and are the recipients of the WS products/services. Customers could also be participants in the WS. There are external customers (outside the organisation) and internal customers (employed by the organisation).

The *environment* of the WS regards organisational, cultural, technical, regulatory, and other aspects that affect the WS's effectiveness and efficiency. In particular, the organisational environment deals with stakeholders, policies, procedures, organisational history, and politics, among others. *Infrastructure* considers human, information and technical resources outside the WS but are used by it. *Strategies* include three different levels: enterprise, department, and work system strategy that ideally should be in alignment.

Although all the above elements should be considered to reason about the WS, they differ in their belonging to the WS. Processes and activities, participants, and information are completely in the work system. Customers and products/services are partially in the WS (customers can also be participants in the WS, and products/services are produced in the WS). On the other hand, environment, infrastructure, and strategies are outside the WS but have direct effects on it.

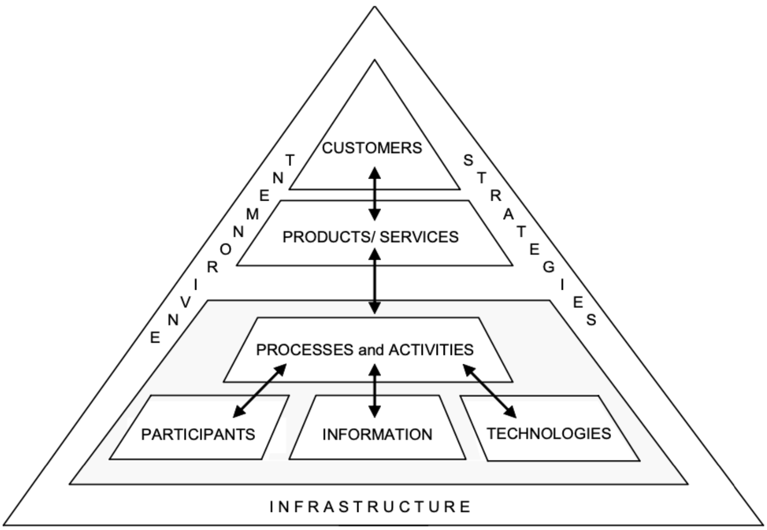


Figure 3.1: The work system framework, from (Steven Alter, 2013).

Regarding the dynamic perspective of WS, Figure 3.2 presents the WS life cycle. The WSLC is an iterative process of planned and unplanned or emergent changes. The *initiation* phase defines the vision and goals that drive the desired organisational change. In the *development* phase, the resources needed for the desired change in the organisation are created or acquired, including software development. *Implementation* deals with implementing the system in the organisation and does not refer to the computational implementation of the technical solution. Finally, the WSLC considers the *operation and maintenance* of the WS.

The WSLC considers the emergence of hindrances in each phase: unanticipated goals in the initiation phase, unanticipated opportunities in the development phase, and unanticipated adaptations in the implementation phase and in the operation and maintenance phase. The WSLC also describes return conditions for the phases, such as the recognition of infeasibility in vision, goals or resources in the development and implementation phases or the recognition of non-adoption or excessive workarounds in the operation and maintenance phase.

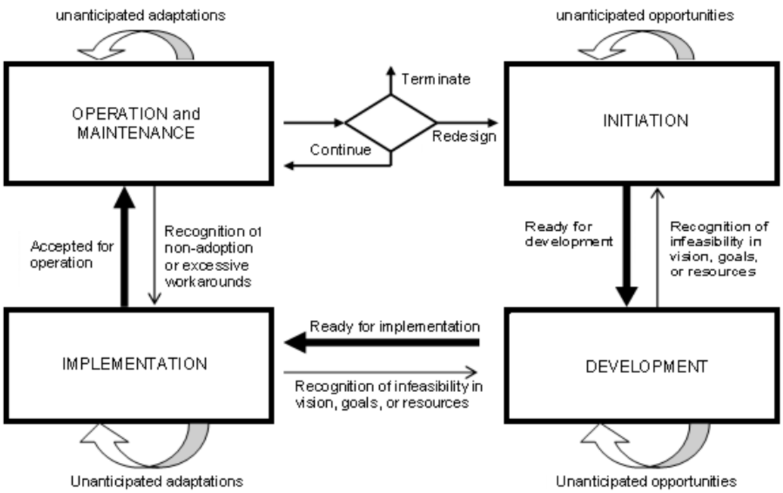


Figure 3.2: The work system life cycle, from (Steven Alter, 2013).

We characterise organisational modelling under the work system theory to achieve the goal of modelling organisational information to be included in the baseline MDD method. In the next subsections, we propose the static and dynamic perspectives for organisational modelling under the work system framework and the work system life cycle, respectively.

3.2.2 Organisational modelling in the work system framework

Considering how the nine WSF elements are grouped as inside, partially inside, or outside the work system, in Definition 3.1, we define organisational modelling in terms of the elements outside the WS but affecting it directly. The definitions below are provided for the IS as a particular case of WS.

Definition 3.1: Organisational modelling

Organisational modelling regards the representation of the *environment*, *infrastructure*, and *strategy* of the IS that have a direct effect on its design.

Under this definition, we aim to scope the environment, infrastructure, and strategy elements relevant for designing the IS under the approach of software organisations, presented in Chapter 1.

Modelling of the work system environment

Regarding the modelling of the environmental elements that affect the design of the IS, we consider the social context that motivates this research, i.e., the approach of software organisations for business strategy. According to the social context, a highly dynamic relationship exists between the environmental elements outside the organisation (e.g., stakeholders, policies and regulations, among others) and the internal elements (organisational vision, culture, etc.) that seek to adapt the organisation to its environment quickly. This approach is based on Vial's definition of digital transformation (Vial, 2021) presented in Chapter 1. In Definition 3.2, we propose the definition of a *business scenario* as a means to address the dynamics of the organisation reacting to an external influence to adapt its value creation processes, and so the IS supporting them.

Definition 3.2: Strategic scenario modelling

Strategic scenario modelling regards the representation of the environmental elements that drive the organisational change, the ends and means defined by the organisation to adapt to such influences, the infrastructure to deploy such definition, and the organisation's reaction to the environment by offering a new value proposition.

The above definition is encompassed by specific definitions for modelling the WS strategies and infrastructure elements, provided below.

Modelling of the work system strategies

We scope the representation of strategy elements of the WS to the enterprise level and to the department (or organisation unit level), leaving the strategies particular to the work system outside of the organisational modelling scope. In Definition 3.3, we propose a definition of organisational ends and means, based on Mintzberg's definition of *strategy as a plan*, this is, the definition of the organisational ends and the means to achieve them (Mintzberg, 1987).

Definition 3.3: Ends and means modelling

Ends and means modelling regards the representation of the ends of the organisation and its constituent units, as well as the means to achieve them in terms of high-level sets of actions.

Modelling of the work system infrastructure

From the definition of strategy scenario, we scope the definition of infrastructure to the elements relevant to the social context: the development teams and their dependencies. We scope the definition of infrastructure to human infrastructure, which we define in Definition 3.4 as organisational structure.

Definition 3.4: Organisational structure modelling

Organisational structure modelling regards the representation of the organisation's constituent units, as well as their hierarchy relationships and influences.

We emphasise that it is necessary to model not only the hierarchy among organisational units but also the influences among them since each organisational unit could be considered as a work system, with the other organisational units as the stakeholders in its environment.

3.2.3 Organisational modelling in the work system life cycle

Regarding the WSLC, our research in organisational modelling is focused on the *initiation* phase. In the initiation phase of the WSLC, the organisation's executives can think about whether it is needed to invest in a mere technical improvement or actually improve the work system and whether the improved version of the work system is aligned with the improvement of the business performance (Steven Alter, 2013). In the context of requirements engineering methods, we place the organisational modelling activity in the early requirements phase, similar to the goal and agent-oriented modelling methods reviewed in Section 2.2.

The expected outcomes of our organisational modelling proposal aim to mitigate the risk of the infeasibility of the vision, goals, and resources that drive the development of the WS, as depicted in Figure 3.2. Our proposal aims to 1) provide a modelling method for analysing and representing the organisation's high-level goals and resources, particularly human resources, in terms of the organisational structure and 2) provide a model-to-model transformation technique to preserve relevant organisational information to the development stage of the WSLC.

3.3 A conceptual framework for organisational modelling

In this section, we provide precise definitions for the concepts involved in organisational modelling following the definitions provided in Section 3.2, using the Unified Foundational Ontology (UFO) as a reference framework. We first describe the foundational concepts on which we base our proposal, and then the concepts for the strategic scenario, ends and means, and organisational structure modelling.

3.3.1 Foundational concepts

The conceptual framework for this thesis builds upon the Unified Foundational Ontology (UFO) (G. Guizzardi, Botti Benevides, et al., 2022). UFO has been developed following theories from philosophical logic, philosophy of language, linguistics, formal ontology and cognitive psychology (G. Guizzardi, 2005).

UFO comprises three sub-ontologies: UFO-A, which defines an ontology for endurants (objects); UFO-B is an ontology for perdurants (events); and UFO-C, an ontology for social entities. UFO-C is built upon the foundations provided by UFO-A and UFO-C (G. Guizzardi, Botti Benevides, et al., 2022). In particular, we base our proposal mainly on the categories of UFO-C, since it provides an ontological foundation for describing social agents such as organisations and their structure, as well as agents' intentionality and actions, which supports the definition of business strategy. Figure 3.3 shows a UML class diagram describing the subset of the UFO-C categories relevant to our proposal. The categories in grey are from UFO-C, while the categories in colour belong to UFO-A and UFO-B. Below, we define each of the classes. Please note that some classes may refer to other UFO classes, which have been underlined and coloured in light grey. Some referred classes are defined in this document, while others have been omitted for brevity and are marked in italic font.

- Agent: *endurant* that is either a physical agent or an institutional agent(G. Guizzardi and Wagner, 2004)
- Physical Agent: physical object that creates *action events* affecting other physical objects, that perceives events, possibly created by other physical agents, and to which we can ascribe a mental state. Examples: a dog; a human; a robot(G. Guizzardi and Wagner, 2004).
- Person: A human physical agent. (UFES, 2015b).
- Social Agent: Social Agents (e.g., an organisation, a society) are created by *communicative acts*. Social Agents are composed by a number of other agents, which can themselves be physical agents, or other social agents(UFES, 2015a).
- Organisation : A social agent involving people and other agents and facilities with an arrangement of responsibilities, authorities and relationships (UFES, 2015b).
- Normative Description: A normative description defines one or more rules or norms recognised by at least one social agent and that can define *nominal universals* such as *social moment universals* (e.g., social commitment types), *social objects* (the crown of the king of Spain), and social roles such as president, prime minister, PhD candidate or pedestrian. Examples of normative descriptions include a contry's constitution, a PhD program's regulations, and a set of directives on performing some actions within an organisation (UFES, 2015a).
- Role (UFO-A): A Role represents a phased-sortal role, i.e. anti-rigid and relationally dependent universal. For instance, the role *student* is played by an instance of the kind Person (UFES, 2015a).

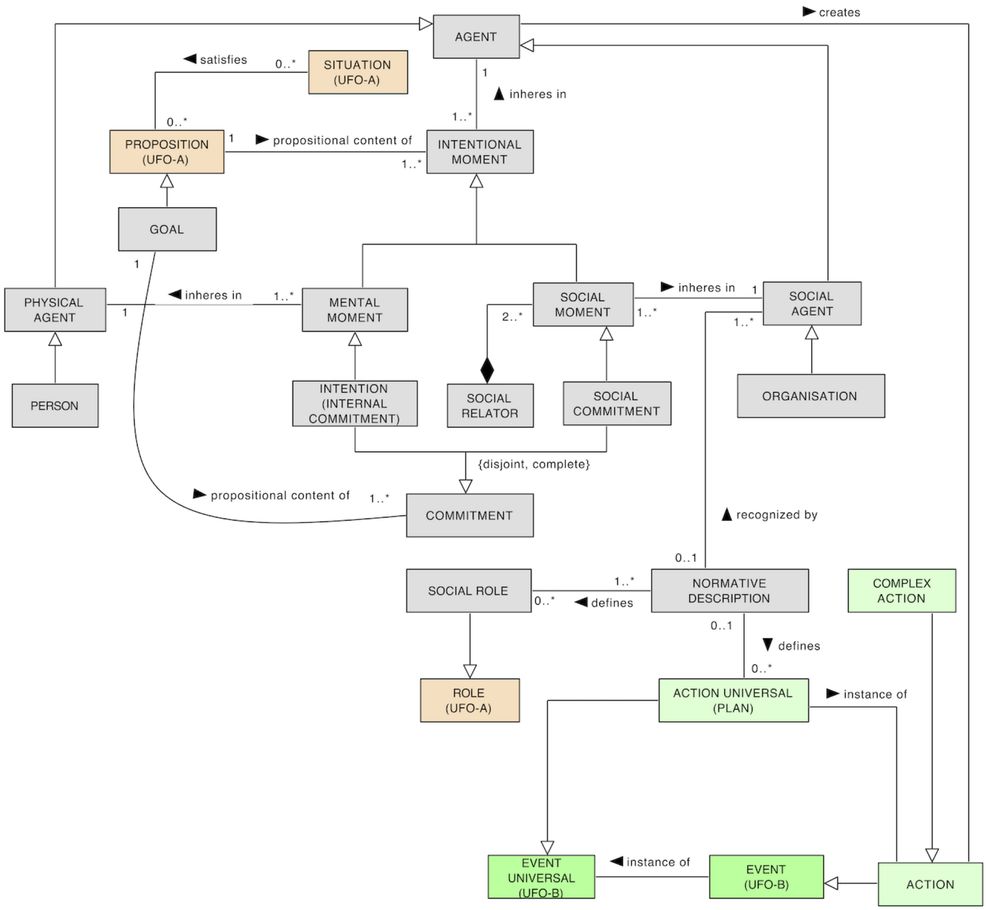


Figure 3.3: Foundational concepts from UFO-A, UFO-B and UFO-C.

- Social Role: Social Roles are special types of roles (i.e., anti-rigid and relationally dependent universals) which are characterised by a number of social moment universals, typically, commitments and claims (UFES, 2015a).

- Intentional Moment: Intention Moment (Intentionality) should be understood in a much broader context than the notion of "intending something", but as the capacity of some properties of certain individuals to refer to possible situations of reality. Every intentional moment has a type (e.g., Belief, Desire, Intention) and a propositional content (G. Guizzardi, Botti Benevides, et al., 2022).

- Mental Moment: *Intrinsic moment* that is existentially dependent on a particular agent, being an inseparable part of its mental state. Examples: a thought, a perception, a belief, a desire, an individual goal. Constraint: For all mm:mental moment; e:*endurant* — if mm inheres in e then e is physical agent (G. Guizzardi and Wagner, 2004).

- Social Moment: Social Moments are types of intentional moments that are created by the exchange of *communicative acts* and the consequences of these exchanges (e.g., goal adoption, delegation). For instance, suppose that John rents a car at a car rental service. When signing a business agreement, John performs a communicative act (a promise). This act creates a *social commitment* towards that organisation: a commitment to return the car in a certain state, etc. (the propositional content). Moreover, it also creates a social claim of that organisation towards John with respect to that particular propositional content. Commitments/Claims always form a pair that refers to a unique propositional content (UFES, 2015a).

- Commitment: Abstract concept to encapsulate different desired states of affairs for which an agent commits to pursue.

- Intention: Intentions are desired state of affairs for which the agent commits to pursuing (an intention is an internal commitment) (e.g., the intention of going to a beach resort for the next summer break) (UFES, 2015a). (Following the definition of

- mental moment, an intention regards a

- physical agent.

- Social Commitment: Desired state of affairs for which the social agent commits to pursuing.

- Social Relator: A social relator is an example of a relator composed of two or more pairs of associated commitments/claims (Social Moments). Finally, a commitment (internal or social) is fulfilled by an agent A if this agent performs an action x such that the post-state of that action is a *situation* that satisfies that *commitment* (UFES, 2015a).

- Relator (UFO-A): Mereological sum of two or more externally dependent modes (Fonseca et al., 2019).
- Proposition (UFO-A): Proposition is an abstract representation of a class of situations referred by an intentional moment. (G. Guizzardi, Botti Benevides, et al., 2022)
- Goal: A Goal is the propositional content of a commitment. In other words, since a goal is a proposition, we have that a particular situation (state of affairs) can be the truthmaker of that proposition (UFES, 2015a).
- Situation (UFO-A): Situations are special types of endurants. These are complex entities that are constituted by possibly many endurants (including other situations). Situations are taken here to be synonymous to what is named state of affairs in the literature, i.e., a portion of reality that can be comprehended as a whole (UFES, 2015a).
- Event (UFO-B): *perdurant* that is related to exactly two states (its pre-state and its post-state). An event is related to the states before and after it has happened.(G. Guizzardi and Wagner, 2004)
- Event Universal: Represents all the existing types of events.
- Action: Actions are intentional events, i.e., events with the specific purpose of satisfying (the propositional content of) some intention of an agent (In this sense, an action can be said to be caused by the intention) (UFES, 2015a).
- Complex Action: Complex actions are actions that are composed of other complex actions and/or *atomic actions*.
- Action Universal (Plan): Represents all the existing types of actions.

3.3.2 Conceptual framework

A conceptual framework for strategic scenarios

As described in the overview of the proposal in Section 1.6.1, our proposal for including business strategy information relevant to requirements elicitation is modelling a *strategic scenario.* A strategic scenario represents how an *organisation* reacts to address a stimulus from its environment (for example, a new competitor or a new regulation by a governmental agency) by adapting its inner elements to produce a stimulus to its environment (for example, a new product feature or an improved quality of service). In this way, the strategic scenario is a set of relationships between the organisation and *agents* in its environment, as well as the relationships between the inner elements of the organisation. These relationships are grounded on the intentionality of the agents and the organisation (and its inner elements). Since the organisation must adapt to the environment, the strategic scenario sets changes in the organisation's intentions and inner elements. We refer to the set of intentions of the organisation and its units as *organisational commitments* and to its propositional content as the *value offer* of the organisation. On the other hand, the new value offer causes the organisation to define actions to achieve them, or a *business strategy plan.*

To support the definitions of the strategic scenario, we present a conceptual framework built upon UFO-A, UFO-B, and UFO-C foundational concepts. Figure 3.4 depicts the framework, which concepts are detailed below.

- Strategic Scenario: A strategic scenario is a situation consistent of a configuration of influence relationships between intentional moments of agents outside the organisation and the organisational commitments of the organisation and its organisational units. The strategic scenario satisfies a value proposition as a result of deploying a business strategy (plan) to realise the organisational commitments of the organisation.

- Influence: Is a relator between the intentional moments of two agents, that represents that the intentional moment I of the agent A produces some effect in the intentional moment of the agent B.

- External Influence: Is an influence produced by an intentional moment of an agent of any type (a person, an organisation, a market) that has an effect over the organisational commitments of an organisation. For instance, a regulatory agency sets a policy that could affect the organisational commitments of an organisation. As a result, the organisation may change some of its organisational commitments to adapt its value proposition to the new regulatory policies.

- Internal Influence: Is an influence produced by the organisational unit purpose of an organisational unit that has an effect over the organisational unit purpose of another organisational unit. For instance, when a team commits to delivering a new feature, other teams may be affected since they have to adapt their

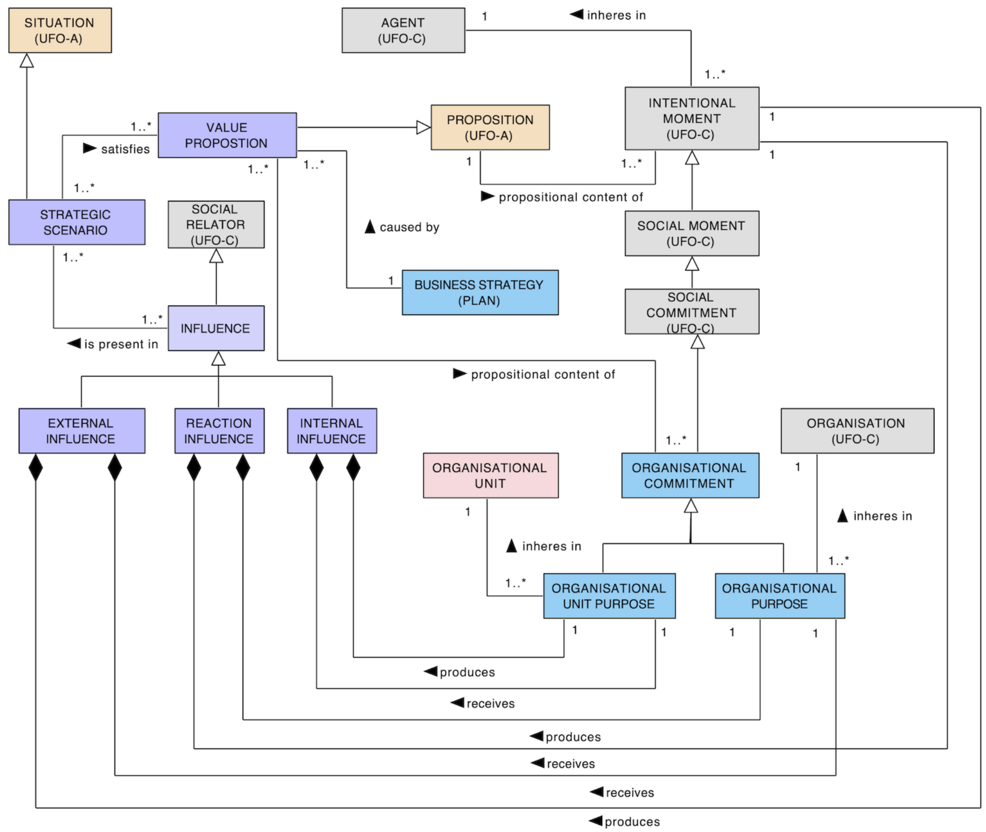


Figure 3.4: Conceptual model for the strategic scenario.

- Reaction Influence: Is an influence produced by an orgnisational commitment of an organisation that has an effect on an agent outside the organisation. For instance, an organisation commits to offer better product quality that affects the customer's intention to buy the product.

- Value Proposition: Value proposition is a proposition representing the propositional content of the set of organisational commitments of an organisation and its organisational units. The value proposition causes the definition of the actions the organisation must take to achieve them; this is, the business strategy.

A conceptual framework for organisational ends and means

In this section, we present a conceptual framework to represent organisational ends and means, i.e., *what the organisation wants* and its actions to achieve them. Our approach ontologically differs from goal-oriented requirements engines such as GORO 2.0 (Bernabé et al., 2019) since we do not focus on goal-based requirements from stakeholders but on the organisation's goals.

Organisational goals cannot be described in terms of the organisation's or its units' *intentions* since intentions are *mental moments* of *physical agents.* We address this by defining the concept of *organisational commitment*, mirroring the *intention* concept for the organisation and its inner elements. We propose two types of organisational commitments regarding the commitment of the whole organisation, namely *organisational purpose*, the commitment of the organisational units or *organisational unit purpose.* For these organisational commitments, we propose two different types of goals to express the propositional content of the organisational commitments: *organisational goals*, addressing high-level goals for the whole organisation, and *organisational objectives*, for well-defined and measurable goals.

On the other hand, the organisation's actions to achieve its goals are included in a *business strategy plan.* It comprises a set of *organisational actions* of two types: strategic action, which operationalises the *organisational goals*, and *tactical actions*, which are more specific and directly contribute to achieving *organisational objectives.* In Figure 3.5, we depict the conceptual framework built upon UFO-C foundational concepts. The definition of each concept is detailed below.

- Organisational Commitment: Represents a desired state of affairs for which an organisation as a whole and its constitutive organisation units commits at pursuing. Since organisations are not physical agents with intentions but social agents, the desired state of affairs comes from the normative description of the organisation. For instance, a for-profit organisation's commitment is profiting (to keep existing); a governmental organisation's purpose could be to provide a service to the community, and non-governmental organisations could have as purpose promoting a social cause.

- Organisational Purpose: Represents a type of organisational commitment characterised by a desired state of affairs of the whole organisation that defines its reason to be and to which is committed. Organisational purpose and mission are often used interchangeably in business and manage-

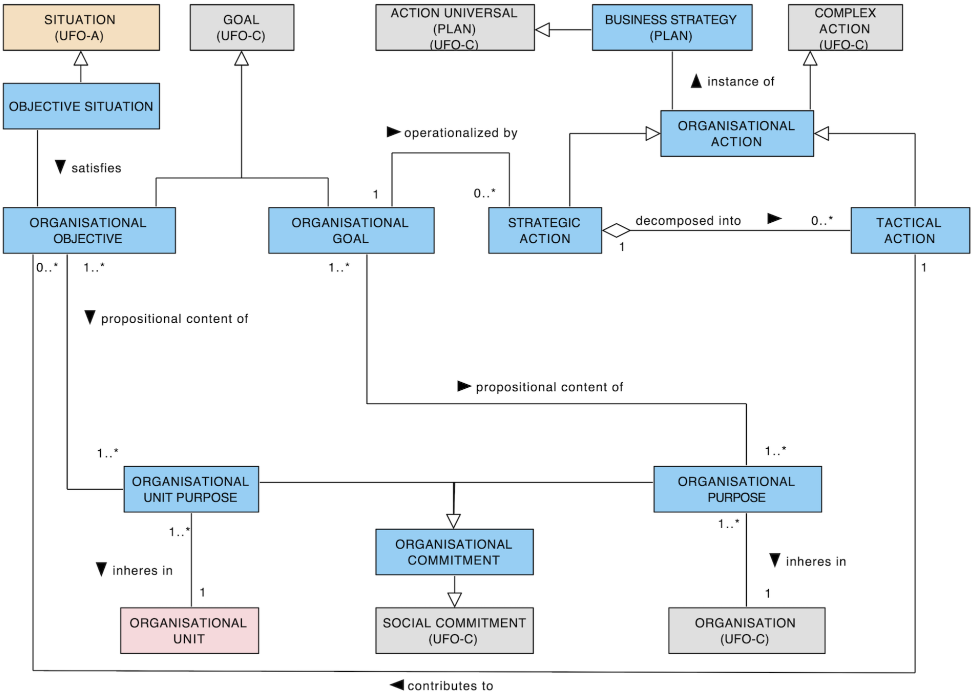


Figure 3.5: Conceptual model for organisational ends and means.

ment literature (Khalifa, 2012). While the organisational purpose as the "reason of being" of the organisation could be interpreted similarly to the organisational commitment (i.e., profiting), different authors consider it a statement that reflects the organisation exists beyond accomplishing its social commitment. For instance, in (Harvard Business Review, 2015), organisational purpose is defined as "the aspirational reason for being which inspires and provides a call to action for an organisation and its partners and stakeholders and provides benefit to local and global society.".

- Organisational Goal: An organisational goal defines the propositional content of the organisational purpose. An organisational goal states a particular situation which will contribute to the organisational purpose. Organisational goals might commit participants of the whole organisation, cross-cutting the organisational units.

- Organisational Unit Purpose: Represent a desired state of affairs of an organisational unit, which commitments are limited to the participants of the organisational unit. Similarly to the organisational purpose, it defined the reason to be of an organisational unit.

- Organisational Objective: Similarly to organisational goals, organisational objectives state a particular situation that will contribute to satisfy the organisational unit purpose. However, the situation is scoped to the commitments of the participants of the organisational unit, and must be objectively assessed. Unlike goals, the situation that satisfies an organisational objective is a well-defined and measurable situation, or objective situation.

- Objective Situation: Type of situation that can be objectively measured.

- Business Strategy: Is an special type of plan which represents the set of organisational actions that an organisation could perform to offer its value proposition in the context of a strategic scenario.

- Organisational Action: Is an special type of action which is performed by an organisation in the context of abusiness strategy plan. Organisational actions are complex actions, which means they are composed of other complex or *atomic actions*.

- Strategic Action: Is an organisational action that affects the whole organisation cross-cutting its organisational units. Strategic actions are high-level, complex actions that take place in the context of a business strategy plan. Strategies are designed to achieve organisational goals.

- Tactical Action: Is an organisational action with a limited scope. Tactical actions are complex actions; however, they are more operative, concrete and specific than organisational strategies. Organisational tactics aim to achieve practical and concrete organisational objectives.

A conceptual framework for organisational structure

Including organisational information for alignment requires identifying the concepts to support modelling the organisation's structure in terms of its constitutional units and the relationships among them. We propose a conceptual framework inspired by the Enterprise Ontology included on SEON: Software Engineering Ontology Network (UFES, 2015b).

In the first level, we conceptualise the *organisation* following UFO-C definition. Consistently with Conway's law (Conway, 1968) and the inverse Conway's manoeuvre (Thoughtworks, 2016), we focus on identifying working groups that will perform design activities in the organisation (*committees*, according to Conway (Conway, 1968). We name these working groups *organisational units*. To support the definition of the organisational structure, we consider two relationship types among organisational units. For hierarchical relationships, we define the *organisational unit delegation* relationship, a particular type of *delegation relationship*, in the sense that the organisation or an organisational unit delegates the responsibility of implementing part of the business strategy to another organisation unit.

On the other hand, to represent the relationships among organisation units that require communication, we draw the *influence* relationship from the conceptual framework for strategic scenario modelling presented in Section 3.3.2. This means that the influence of an organisational unit over another (for instance, a new software service delivered by a development team that must be integrated into another team's software product) sets the need for collaboration and coordination among teams, not only for designing the way to handle the influence (designing the integrating of the service and the product), but to use the designs produced collaboratively (in this case, operate and maintain the software product and the service). These consequences must be addressed at an operational level, whether in business process or information systems modelling, so we do not provide further typification at the organisational level.

Finally, to support the participation of persons as *organisational unit members*, we provide the concept of *organisational role*, which is performed by an *organisational unit member* (a person). Organisational roles receive their *allocation unit* through *organisational unit assignment*. Figure 3.6 presents the conceptual model for organisational structure. The concepts are defined below.

- Organisational Unit: Organisational units are organisations that exist inside an organisation. Similarly to organisations, organisational units involve people and other agents and facilities with an arrangement of responsibilities, originated by a delegation from the organisation to the organisational unit.

- Organisational Unit Delegation: is a *delegation* from the organisation to an organisational unit, involving people and other agents and facilities with an arrangement of responsibilities. An organisational unit delegation defines a hierarchical relationship between the organisation to the subordinate organisational unit.

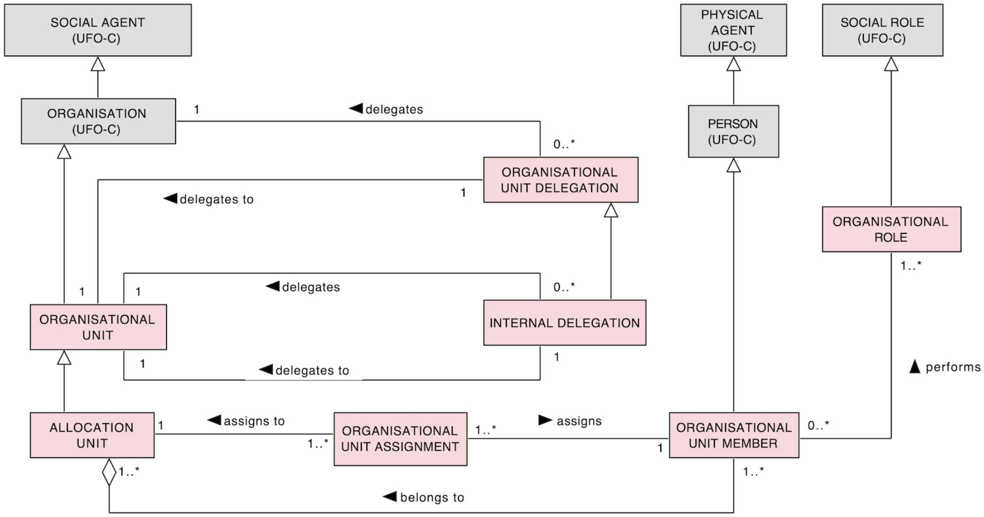


Figure 3.6: Conceptual model for organisational structure.

- Internal Unit Delegation: is a organisational unit delegation from an organisational unit to another organisational unit. It also defines a hierarchical relationship between the delegating organisation unit to the delegated organisational unit.

- Organisational Role: A social role, recognised by the organisation, assigned to agents when they are hired, included in a team, allocated or participating in activities. Examples: system analyst, designer, programmer (UFES, 2015b).

- Allocation Unit: Instance of organisational unit that have assigned organisational unit members.

- Organisational Unit Assignment: is a relator that connects an organisational unit member with their allocation unit.

- Organisational Unit Member: A Person that has been hired to work as part of an organisation and has been allocated to an organisational unit.

3.4 Summary

Chapter 3: Theoretical framework

- In this chapter, we presented a conceptual basis to support the study of the challenges of the baseline model-driven software production method for including organisational information on business strategy and organisational structure **(KQ2)**.
- To unambiguously define the organisational information to be included in a model-driven development method, we presented our approach to organisational modelling under the work system theory and a conceptual framework built upon the universal foundational ontology UFO.
- Following the definitions for the work system's organisational elements, we defined our approach for modelling the work system's environment (strategic scenario), infrastructure (organisational structure), and strategic (organisational ends and means).
- We propose a theoretical framework for modelling a strategic scenario from which the requirements for designing or redesigning business processes and information systems are derived. The framework presents the strategic scenario as a set of relationships between agents outside the organisation that influence the organisation's ends and means, the relationships between the inner elements of the organisation to handle such influence, and the relationships from the organisation to its environment attempting to influence it with a new value proposition.
- We propose a theoretical framework for defining organisational ends and means, including strategic and tactic actions as part of a business strategy plan and organisational goals and objectives as the propositional context for the organisation's purpose and social commitment.
- Finally, we also propose a theoretical framework for representing the organisational structure in terms of organisational units. We propose using delegation relationships to represent hierarchy and influence relationships among them to represent communication needs derived from the strategic scenario.

Chapter 4

Baseline Method Analysis

4.1 Motivation

The baseline MDD method this research aims to improve supports the modelling of social agents and their goals. As detailed in Section 2.6, the baseline method considers i* (Yu, 2011b) for modelling social agents and their goals, and GoBIS (Ruiz, Costal, et al., 2015) for conveying organisational information to the business process modelling level. Using i* and GoBIS presents numerous strengths: i* is one of the most widely used languages for agent and goal modelling, and can be used for different domains, including the strategic organisational level. On the other hand, GoBis provides semantic consistency with Communicative Analysis, the business process modelling method of the baseline MDD method, which differentiates it from other goal and business process alignment initiatives.

However, as reviewed in Chapter 2 and discussed in Section 2.5, the freedom that i* provides and the specific focus of GoBIS on mapping dependencies between actors presents challenges and gaps for modelling and integrating business strategy and organisational structure information into the baseline MDD method. Since modelling business strategy jointly with organisational structure is an enabling elements of the SO's approach to alignment, these challenges and gaps must be identified to be addressed and to specify the requirements for improving the baseline MDD method.

In this chapter, we analyse the challenges that using i* and GoBis pose for modelling business strategy and organisational structure and for preserving this information down to the business process modelling level. We define a *challenge* as an issue that hinders the design goals of this research, previously stated in Chapter 1 goals, i.e., the *traceability* and the *automatic transformation* of business strategy and organisational structure information into the business process level. We define traceability and practical automation goals as follows:

- *Traceability* is the capability to trace modelling elements through different stages of the modelling process (Estrada et al., 2006).
- *Practical Automation* is a significant and non-redundant automated transformation of modelling elements through different stages of the modelling process. Non-redundance means that there must be differences in the rationale and detail of the source and target modelling elements to avoid adding overwhelming details to more abstract levels just for the sake of having a completely automated transformation. A transformation is significant if it helps to provide a method quality feature, taking into account the features defined in (Estrada et al., 2006): refinement, modularity, repeatability, complexity management, expressiveness, reusability, scalability, and domain applicability.

Throughout this chapter, we present a series of *i* organisational models*. We refer to *organisational models* as we focus on *organisational goals and actions* coming from strategy, infrastructure, and environment, and not to *business goals* set by customers or *system goals* set by the system's participants, following the proposed classification based on work systems Section 2.3. On the other hand, the organisational information in the models is scoped to the concepts defined in the theoretical framework in Chapter 3. We consider business strategy and organisational structure information in the context of the strategic scenario that drives the need for redesigning business processes and supporting information systems.

Our analysis method is based on the abductive inference from a set of single-case mechanism experiments (Wieringa, 2014). A mechanism experiment consists of exposing an artefact to stimuli, observing its response, and explaining the response based on the internal mechanisms of the artefact, which yields answers that probably could explain the observed phenomena. In our case, the artefact is the baseline MDD method; the stimuli is an i* organisational model; we observe if the method helps represent and preserve the information from the organisational level to the business process modelling level, keeping traceability and practical automation.

A limitation of the single-case mechanism experiment method is that, as stated in (Wieringa, 2014), the explanations provided by the abductive inference method are not certain but probable. To explore whether other researchers share these explanations, we comment on the results with three experts in requirements engineering and model-driven engineering who also know the methods under analysis.

The analysis presented in this chapter addresses the knowledge question about the challenges of the baseline model-driven software production method for including organisational information on business strategy and organisational structure **(KQ2)**. In Section 4.2, we present the single-case mechanism experiment method. The analysis is detailed in Section 4.3, while in Section 4.4, the expert review of the results is discussed. Finally, we summarise the findings in Section 4.5.

4.2 Analysis method

In this analysis, we observe how the modelling methods and transformation techniques of the baseline MDD method described in Section 2.6 respond to the attempt to preserve the strategic information from the organisational to the business process level.

The working hypothesis is that it is possible to provide traceability and practical automation of business strategy and organisational structure information from the organisational modelling level to the business process modelling level. In Figure 4.1, we depict the mechanism experiments or cases we designed to test the working hypothesis. Each mechanism experiment is a modelling or model transformation situation that demands using the baseline MDD method to test the working hypothesis.

For each experiment, we present the following topics:

- The *mechanism experiment*, describing the modelling or model transformation situation.
- The *problem*, describing how the experimental situation hinders traceability and practical automation.

Figure 4.1: Summary of the analysis scope and cases.

- The *explanation*, proposing a cause of the problem based on the methods' inner concepts, relationships, or mechanisms.
- The *implications*, describing the effects of the problem in the development process, are commented.
- The *rationale*, classifying the problem in terms of quality attributes extracted from the literature review (information loss or transformation coverage) and its impact on traceability and practical automation on the baseline MDD method. We also comment on how the issue could be addressed, referencing existing methods and techniques.
- The *challenge*, as a statement of an improvement goal for the baseline MDD method.

4.3 Analysis of challenges

In the following subsections, we present the challenge analysis for including strategic information in the baseline MDD method. First, we analyse two cases that expose issues regarding the representation of strategic information at the organisational modelling level with i*. Then, we analyse three cases of transformations from the organisational level to the business process level, showing traceability or automation issues when attempting to preserve strategic information. It is worth noting that these challenges are not intrinsically an issue of the methods and techniques but arise from the necessity of using them for satisfying the social context requirements using the baseline method described in Section 2.6.

We introduce a real estate agency detailed in Example 4.1 as a working example.

Example 4.1: Real Estate Agency

The real estate agency provides house and apartment renting services. Currently, potential tenants ask the agency for houses or apartments (namely properties) that fulfil specific requirements. The company assigns an agent who offers a set of properties that might cover the requirements. The tenant makes a reservation, pays a booking fee, and submits his or her financial profile. The agency reviews the tenant's financial status and then confirms the reservation (or not). Currently, the agency is facing competitors that offer shorter times from the property requirements specification to the reservation confirmation. In order to react to this threat, the agency is re-engineering the renting process to go entirely online. The agency expects to maintain and even increase its market share.

4.3.1 Challenges in organisational modelling

At this modelling level, we look for challenges regarding how to design an organisational model using i* (presented in Section 2.2.1) modelled at the organisational level.

Case 1: Modelling Procedure for the Organisational Level

This case shows that more than the mere presence of concepts regarding organisational structure and strategic intentionality is needed to produce organisational-level models.

Mechanism experiment: An analyst is asked to design an organisational model as the first activity for re-engineering the renting process. The i* model in Figure 4.2.(A) describes the goal of the organisation (online renting service offered) that is refined by two tasks (receive booking and show available properties). These tasks depend on the Tenant, so two social dependencies are designed between the Agent and the Tenant. Then, the analyst designs the model in Figure 4.2.(B) as the first step for designing business process models to implement the business strategy.

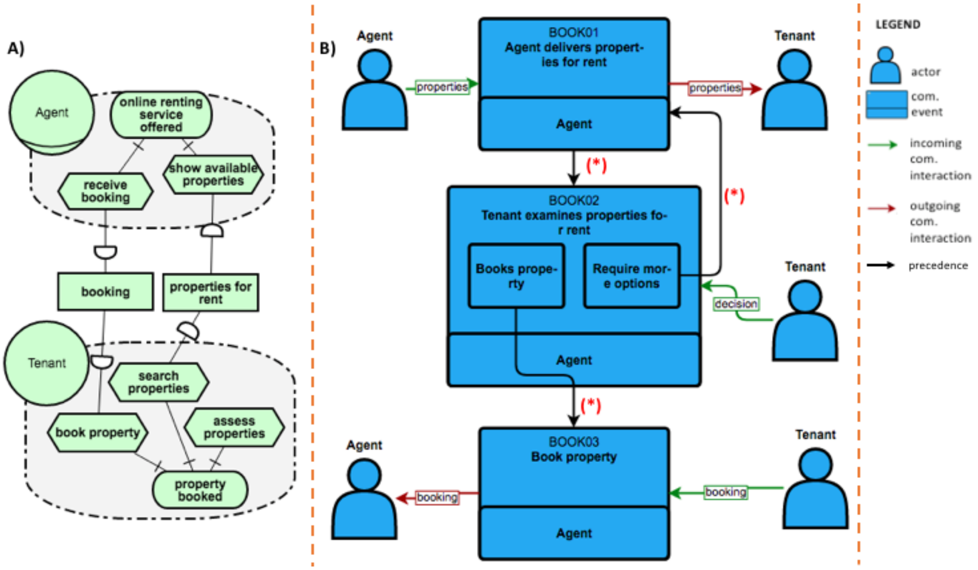


Figure 4.2: i* and Communication analysis diagrams showing the same level of detail for describing business process elements.

Problem: The i* model in Figure 4.2.(A) is semantically correct and expresses the organisational goal of offering an online service as the motivation for more detailed tasks. However, in the context of MDD, we identify a modelling issue when using the approach shown in Figure Figure 4.3(B), which introduces redundancy: both Figure 4.2.(A) and Figure 4.2.(B) models have the same level of detail. Overlapping business process specification introduces redundancy, hinders complexity management of the model, and introduces process modelling detail that could be overwhelming at the organisational level.

Explanation: i* does not prescribe a modelling procedure, so it can be freely applied by the analyst. However, for its integration into an MDD method, modelling guidelines for using i* at the organisational modelling level would be needed to prevent mixing business intention with business process specification. Figure 4.3 illustrates the difference between i* models representing (a) strategic ends and means and (b) a model with fine-grained tasks.

Implications: In practice, this would lead to the use of the same model to reflect on business strategies and goals and for defining operational details about who delivers a document to whom. To avoid this issue, the business

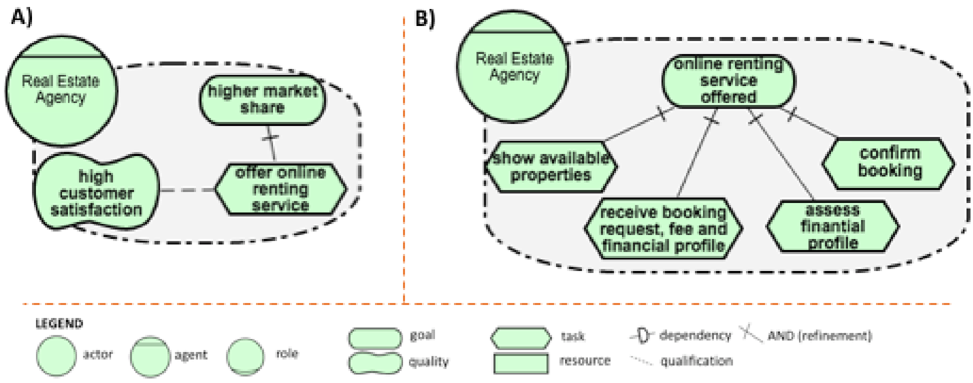


Figure 4.3: Two i* models for representing different levels of detail for organisational ends and means.

process details must be delegated to the CA model, and traceability must be provided from the motivation to the specification of business processes.

Rationale: the modelling issue can be classified as an *information loss issue.* Given the intention to model organisational strategy, not having a modelling procedure to avoid unnecessary detail at the organisational level could harm the *traceability* of strategic information in the MDD method. It also harms the *practical automation* of the method since not having a modelling procedure does not ensure that an analyst could get to model the concepts that can be transformed into elements of the business process modelling level.

Challenge: *Challenge 1 - Provide a modelling procedure to avoid overlapping between the organisational and business process modelling levels.*

Case 2: Modelling constructs for the organisational level

This case aims to identify if more specific concepts are needed to represent organisational goals and strategic elements.

Mechanism experiment: An analyst must represent the business strategy defined by the directors of the company. In addition to the goal of increasing market share and the strategic action of offering an online renting service, the agency's executives define that no more than 12 hours must pass between the moment when a tenant contacts the agency to manage a property and the online publication of the property. Also, executives state that tenants must be able to request the agency's publication services at any time from anywhere. The analyst represents this information in the model depicted in Figure 4.4.

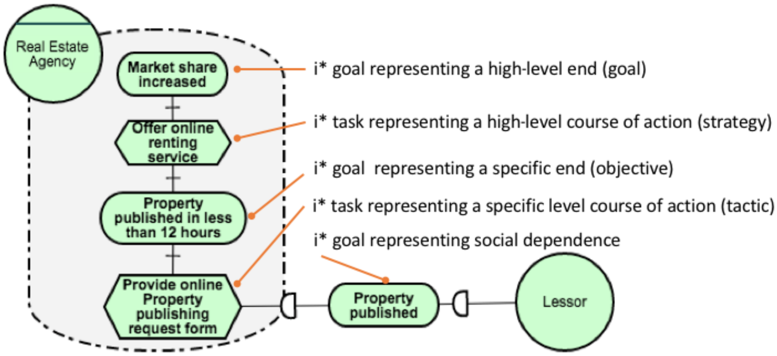


Figure 4.4: I* model showing how the same i* primitives can be used for modelling different organisational concepts.

Problem: The analyst applied the same i* modelling concepts (goals and tasks) to represent, in the same diagram, different business concepts. On the one hand, regarding organisational ends, "market share increased" is a general desire of the state of affairs. At the same time, "property published in less than 12 hours" is a more specific, measurable, desired state of affairs. On the other hand, concerning organisational means, offering an online renting service is a high-level strategic action that can impact several business processes. At the same time, "provide online property publishing request form" is a precise action that affects a specific set of activities of the organisation.

Explanation: There is a construct deficit (Rosemann, Green, and Indulska, 2004); this is, the i* constructs are not enough to represent relevant concepts of organisational modelling.

Implications: Given strategic information and the i* constructs (goals, tasks, resources and qualities), the decision of *what to model* could lead the analysts to omit high-level information, to represent as goals the more precise definitions, or to omit the more detailed information, to favour a more high-level model. Since each analyst decides what concepts are important to model, two models could not be compared nor evaluated in terms of completeness and consistency, as already identified by Estrada et al. (Estrada et al., 2006).

Rationale: Other organisational modelling initiatives such as the Business Motivation Model (The Object Management Group, 2015) and ArchiMate (The Open Group, 2013) define several concepts for the *ends* of an organisation (goals, vision, objectives) and for the strategical *means* (strategies, tactics, courses of action, business policies, etc.). The definition of these concepts and their relationships could improve the semantics of organisational models. This issue is related to *information loss* and hinders the *traceability* of strategic information.

Challenge: *Challenge 2 - Define organisational level constructs that are valuable for representing strategic information.*

4.3.2 Challenges in the transformation of the organisational model into the business process model

This section shows issues when transforming organisational models in i* to business process models in Communication Analysis (CA). We took as reference the GoBIS technique (Ruiz, Costal, et al., 2015) for transforming social dependencies between two actors in i* into communicative events between the same actors. In order to identify challenges for the MDD goals, we present three cases. Case 3 exposes the current voids in transforming the organisational structure and business strategy information into elements of the business process models. Cases 4 and 5 show the effects of not preserving the organisational information in the structure and logic of business processes.

Case 3: Preserving organisational structure and strategy information

This case shows that information at the organisational level that is not currently mapped could be important for designing strategically aligned business processes.

Mechanism experiment: The agency's executives have decided to create a new business area, the Sales Department, responsible for the reservations. The agents will belong to the Sales Department and be responsible for confirming the reservations in less than 12 hours. The Agent must receive a booking from another actor (Lessor) to achieve this goal, creating a social dependency. The analyst models these strategic definitions as shown in Figure 4.5.

Problem: The GoBIS (Ruiz, Costal, et al., 2015) technique allows mapping social dependencies among actors into communicative interactions; however, strategic concepts that are not directly related to interactions among actors are

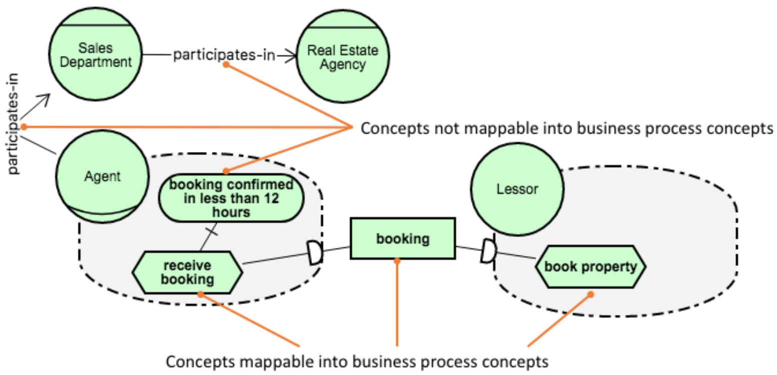


Figure 4.5: Organisational model in i* showing concepts that are mappable and non-mappable to business process models using GoBIS.

not mapped into any business process concepts in CA. As detailed in Figure 4.5, the Agent's goal will not be mapped into the business process model, and the analyst must manually keep track of the constraint "booking confirmation in less than 12 hours" when designing the business process. On the other hand, since the organisational structure is only modelled at the organisational level, the information on these concepts will be lost.

Explanation: When using i* for representing the actor's goals for an information system, the transformation technique helps connect the actor's goals with the business processes that support them. However, when using i* for organisational modelling, extending the transformation to cover other relevant information besides the actor's goals is needed.

Implications: The analyst may be ignoring relevant organisational definitions, hindering the strategic alignment of business processes. For example, the goal "booking confirmed in less than 12 hours" sets requirements for at least three communicative events or tasks in the business process model. First, it is necessary to register the date and hour of the reservation and then register the date and hour of the confirmation. Finally, it is necessary to report this information to the Sales Department. These requirements also have effects on the design of the information system.

Rationale: This is a *transformation coverage* issue since there are elements in the source metamodel that are not being mapped to the target metamodel. Concerning organisational structure, the target of a "participates-in" relationship in i* could be mapped as an organisational unit in CA since this concept already exists in the CA metamodel (España, 2011). In other notations, such as Business Process Model Notation (BPMN) (Rosing et al., 2015), the i* relationship could be mapped as the label of a pool group, where each pool represents an actor that belongs to the unit. Concerning goals and strategies, the CA metamodel allows specifying the goal of communicative events as an attribute of the Communication Event Template. It would be possible to map the i* intentional elements into a textual format to guide the analyst when designing the business process. Similarly, i* goals could be mapped to BPMN's process documentation since BPMN metamodel supports this attribute. This issue hinders the *traceability* of strategic information and its *practical automation*.

Challenge: *Challenge 3 - Transform organisational structure and goals into business process concepts.*

Case 4: Effects on the business logic of the business process model

Mechanism experiment: The Agency defines as part of the strategy that the online renting service and all the associated services must provide maximum customer satisfaction. As part of the online renting service, the analyst must model the process to attend to reimbursement claims by the Lessor. In Figure 4.6.(A), the customer satisfaction is not considered, and the Lessor's claim is assessed first and then compensated, while in Figure 4.6(B), the claim is immediately compensated and then assessed.

Problem: Unless the assessment of the claim is extremely fast, the model in Figure 4.6.(A) will be misaligned with the organisational goal of customer satisfaction.

Explanation: There is no concept in i* to represent a strategic behavioural statement that could favour the traceability and practical automation of business process flow decisions from organisational-level definitions.

Implications: There is a risk of designing business processes with logic not aligned with the organisational goals.

Rationale: In other organisational modelling frameworks, such as the Business Motivation Model (BMM), there is the concept of *directives* that can be *business policies* or *business rules*, both of which can be traceable to business process elements in BPMN (Rosing et al., 2015). Including a behavioural concept at the organisational level that could be mapped to the business process

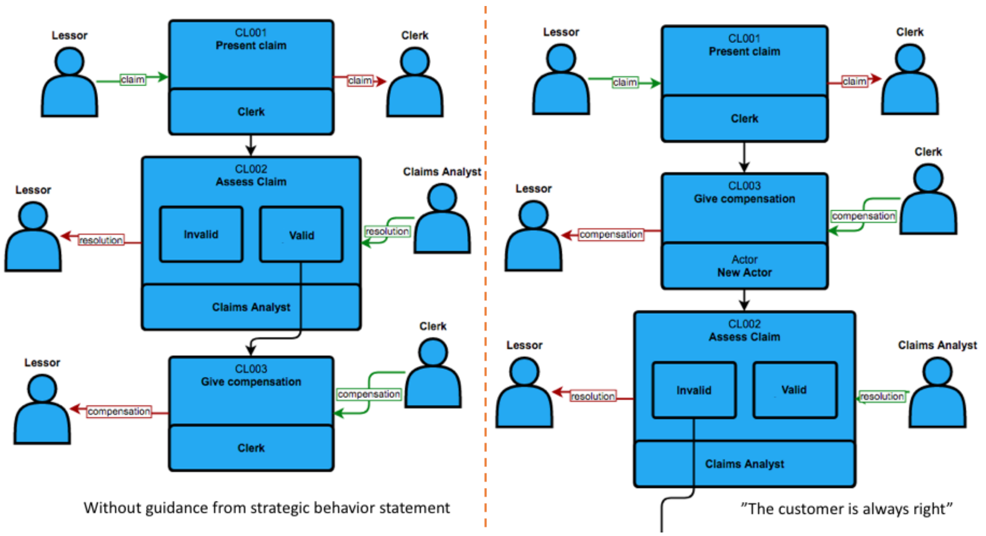


Figure 4.6: Two different business process designs with and without taking into account the customer satisfaction goal.

flow could help design strategically aligned business processes. This is an *information loss* problem and an improvement opportunity for *traceability* and *practical automation*. There is also an opportunity to improve practical automation by encapsulating business process patterns (Fellmann et al., 2020) or interaction patterns (Ruiz, Espana, et al., 2013) in these strategic behavioural statements. Including behavioural concepts at the strategic level would allow taking advantage of the existing pattern repositories, analysis techniques, and methods (Ramos-Merino et al., 2019).

Challenge: *Challenge 4: Define a strategic behavioural concept to guide the design of business processes.*

Case 5: Effects on the structure of the business process model

This case aims to show that losing information about organisational structure and goals generates coupling process elements that should belong to independent business processes.

Mechanism experiment: When defining the organisational goal of offering an online renting service, the agency also requires training for the Agents to use the online renting platform and an advertising campaign for the new service. For the first process, the agency depends on the Human Resources Department, specifically on the Chief of Human Resources; for the second, the agency depends on the Marketing Department, specifically on the Marketing Executive. The analyst models these definitions as shown in Figure 4.7 (the departments have been omitted for simplicity). Considering the transformation of social dependencies into parts of the business process, the agency's dependency on the Chief of Human Resources and the Marketing Executive will lead to two separate business process elements. Whether these process elements belong to the same business process model or not must be decided by the analyst.

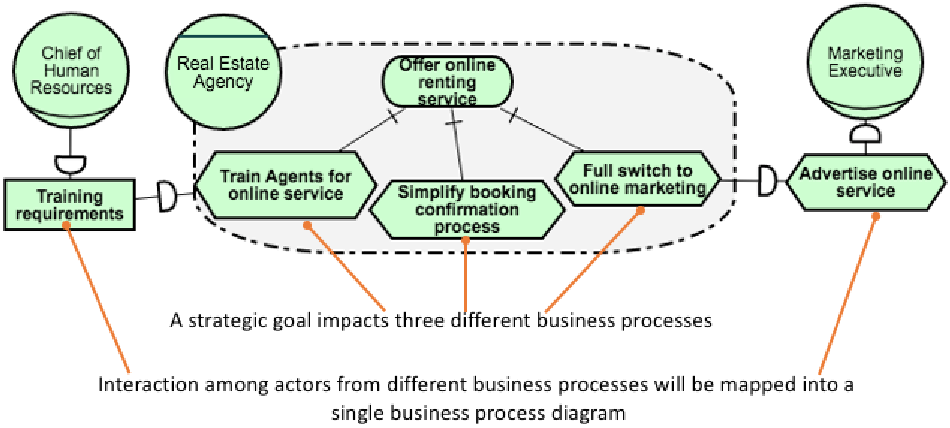


Figure 4.7: The refinement of a single goal can impact several business processes.

Problem: Three problems can arise when automatically transforming social dependencies of an organisational model into business process elements: (1) merging elements from different business processes, (2) mapping unnecessary process elements, and (3) spreading elements in different processes that are meant to be in the same re-engineered business process.

Concerning the first point, the dependencies with the Chief of Human Resources and the Marketing Executive displayed in Figure 4.7 could motivate the re-engineering of the training and advertisement processes (in addition to the re-engineering of the renting process). In this case, three different processes need to be modelled, and the elements generated by the transformation must be distributed according to the organisational structure of the dependencies.

For the second point, it could be the case that the Agency needs the training and advertisement processes to be performed in the same way as usual, thus not requiring changes to the existing business processes. In this case, it would be wrong to automatically transform these dependencies into elements for designing the new business process model. For the third point, consider that all the dependencies in Figure 4.7 have the same source goal. As discussed in the social context of this research in Section 1.3, organisational alignment tends to group business processes around business outcomes. Knowing that a set of dependencies belongs to the same organisational goal would help group the automatically generated process elements in the same business process model.

Explanation: The current transformation technique does not consider information about the context of a social dependency (e.g., the organisational structure or source organisational goal) that could help to group the generated process elements into different business process models.

Implications: In practice, this means making the analyst responsible for manually organising the automatically generated portions of business process elements into different diagrams without providing guidance to identify separate business processes and sub-processes. To better assist the analyst in the transformation, the current transformation technique must be extended to identify the source grouping concept at the organisational level and map it as a grouping concept for business process elements. Also, the technique can consider identifying if an organisational goal or strategy affects the current design of business processes.

Rationale: This is an *information loss* issue, as the mapping between elements could be misplaced and hinder *traceability* and *practical automation.* In other organisational modelling approaches (BMM (The Object Management Group, 2015; The Open Group, 2018)), it is possible to connect concepts that describe strategic courses of action with the affected business processes. In i*, there is no way to connect goals with the affected business processes or group the actors and concepts of the same business process. The transformation technique could be extended to identify organisational-level actions that affect business processes. Since a CA model can have several diagrams (España, 2011) to support modularity (Estrada et al., 2006), the transformation could be extended to define the target view in the business process model.

Challenge: *Challenge 5 - Organise the transformed business process elements using the strategic organisational context.*

4.4 Expert analysis

Mechanism experiments allow identifying *possible explanations* for the modelling issues found, which we named *challenges*. However, there are two threats to the validity of these challenges. First, the modelling issues might not be relevant in achieving traceability and practical automation in MDD. Second, the modelling issues could be already solved by other methods or techniques different from or complementary to those examined in the analysis.

To mitigate these threats, we surveyed three experts on the modelling methods presented in Section 2.6. These experts have worked with i*, Communication Analysis (CA), and the OO-Method (OOM) for at least five years. They are researchers from Utrecht University (Netherlands), Zurich University of Applied Sciences (Switzerland), and Universidad de Castilla la Mancha (Spain), respectively. They have no relationship with the contributions of this paper.

We described each challenge with the same examples similar to those presented in the previous section and we asked the questions detailed below.

1. From 1 to 10, what is the importance of the modelling issue? Comment.
2. From 1 to 10, what is the frequency of the modelling issue? Comment.
3. Do you know of any initiatives that mitigate the modelling issue? If not, could you suggest any ideas? (yes/no)
4. To tackle this issue, do you prefer a method that is a supported systematic approach or a free, manual approach? (S: systematic, NS: not systematic, D: depends)

Table 4.1 presents the data collected from the survey. In Figure 4.8, we present the importance and frequency addition of each challenge. We comment on the main findings below.

Challenge 1: Provide a modelling procedure to avoid overlapping organisational and business process modelling levels. While experts agree that guidance for avoiding overlapping is necessary, there is no consensus about its importance. Expert 2 gave the maximum rating to importance. The other two experts (who gave a rating of medium importance) agreed that separated models make it easy to maintain the whole model and help to avoid inconsistencies among models. Expert 3 stated that *"it is important to keep a link between goals and business processes"*. However, Expert 1 noted that

Table 4.1: Data from expert evaluation of challenges.

Challenge	Expert	Q1	Q2	Q3	Q4
Challenge 1	**Expert 1**	5	2	no	S
	Expert 2	10	8	no	S
	Expert 3	4	2	yes	S
Challenge 2	**Expert 1**	8	4	no	S
	Expert 2	10	10	no	S
	Expert 3	7	9	yes	S
Challenge 3	**Expert 1**	7	7	no	NS
	Expert 2	10	10	no	NS
	Expert 3	4	7	yes	S
Challenge 4	**Expert 1**	8	4	no	NS
	Expert 2	0	0	no	NS
	Expert 3	1	1	yes	NS
Challenge 5	**Expert 1**	6	5	no	S
	Expert 2	10	10	no	S
	Expert 3	6	3	no	S

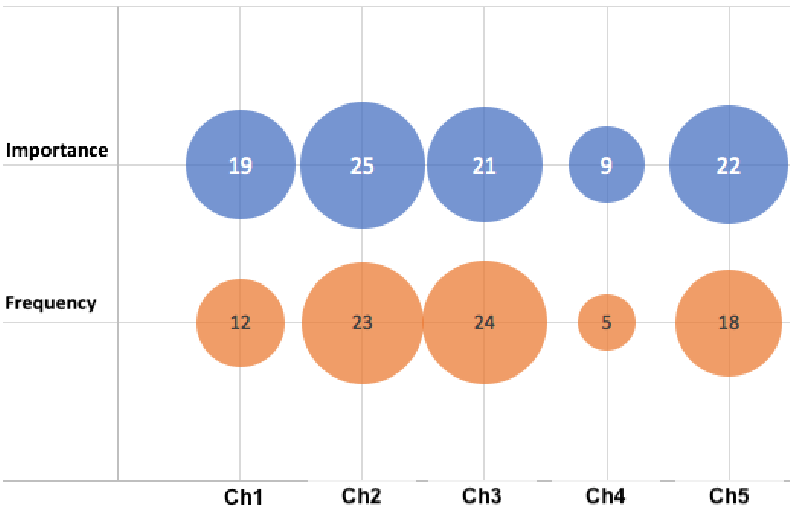


Figure 4.8: Summary of the analysis scope and cases.

"avoiding overlap, the transformation of process models from goal models is difficult".

Concerning the frequency of the problem, while Expert 2 gives a high rating to mixing goals and detailed tasks when modelling in i*, Experts 1 and 3 find it unusual. This contradiction could be due to the small size of academic projects. Finally, about existing initiatives to solve the problem, Expert 3 provided a reference to a proposal of a combination of different modelling methods (Vara, Sánchez, and Pastor, 2013), but at the same modelling levels: Maps (Rolland, 2007) and BPMN (Rosing et al., 2015), where the difference among levels is more clearly stated.

Challenge 2: Define organisational-level constructs that are valuable for representing strategic information. This challenge was the most supported by the experts of all of the other challenges. They all agreed on the need for systematic support to define organisational-level concepts and rated it as very important and frequent. Expert 3 commented that *"Different people can make very different models of the same phenomenon under study due to the lack of specific and homogeneous guidelines to follow."*. Concerning current initiatives that could bridge this challenge, Expert 3 provided references to the work of Professors Joao Araujo and Jaelson Castro, from Universidade Nova de Lisboa and Universidade Federal de Pernambuco, respectively.

We reviewed the work by professors Araujo and Castro on a systematic review of i* extensions (Gonçalves et al., 2018). A catalogue of i* extensions was developed based on the study's results[1]. A total of 24 concepts in the *organisation//business process* category were found, some of which are important for modelling strategic definitions (Marosin, Van Zee, and Ghanavati, 2016). However, language constructs which are appropriate for describing business strategies, such as strategy, tactic, and objective, are not present among these extensions. The work by Kitsios et al. (Kitsios and Kamariotou, 2019) also reviews strategy concepts in goal and enterprise architecture languages, including i* in the review. The findings show that some strategy concepts not included in i* are included in ArchiMate (The Open Group, 2013) or in Business Motivation Model (The Object Management Group, 2015) frameworks. However, these frameworks are much more extensive and complex than i*, lacking i*'s social approach to model actors' inner goals and strategies. On the other hand, no modelling procedures were found to ensure the consistent representation of organisational goals and strategies regarding requirements engineering.

Challenge 3: Transform organisational structure and goals into business process concepts. There was no consensus about supporting this challenge. Experts 1 and 2 agreed on the importance of the problem but dis-

[1] http://istarextensions.cin.ufpe.br/catalogue/

regarded automatic transformation in favour of the analyst being principally responsible. Moreover, Expert 2 indicates that *"Traceability could be created using partial automatic transformations along with additional information provided by the analyst."*. Experts agreed on the problem's high frequency and that, to their knowledge, there is no solution for this issue. The experts did not provide references to existing initiatives addressing this challenge.

Challenge 4: Define a strategic behavioural concept to guide the design of business processes. This was the least supported topic by the experts. The experts agree that there is no need to add behavioural elements to i*, and Experts 2 and 3 gave the lowest importance and frequency to the modelling issue. Regarding the existing initiatives for tackling this challenge, even though experts 1 and 2 commented on the existence of i* extensions for including process flow definitions, no specific works or authors were mentioned.

Challenge 5: Organise the generated business process elements using the strategic organisational context. All three experts supported this challenge; however, Expert 2 stated that the solution could be combined with a manual approach similar to Challenge 2. Expert 1 stated *"What I would propose is that the traceability is generated automatically, but I think that the modularity of business processes must be determined by the analyst"*. While the experts did not provide references about existing works to overcome this challenge, Expert 1 commented *"I think that a good way to solve this challenge could be joint modelling of i* and processes. "*.

From the above findings, we present three improvement goals for the baseline MDD method for software development to address the challenges most supported by the experts.

- **Improvement goal 1 - Use a specific modelling language and procedure for business strategy and organisational structure:** At the *organisational modelling level*, the modelling language must precisely focus on strategy elements and must have a modelling procedure that prevents from having several levels of refinement for strategic actions to keep a higher level of granularity than business process elements. This improvement goal addresses Challenge 1 and Challenge 2.
- **Improvement goal 2 - Use organisational models information to design modular business processes.** For the *transformation from organisational models to business process models*, there is a need for an integration method (e.g., a transformation, verification, or analysis method) to use the information about the organisational structure and the high-

level organisational goals to design modularised business processes. This recommendation addresses Challenge 3 and Challenge 5.

The above improvement goals drive the design activities of this research. The improvement goal 1 yield to the design of LiteStrat, presented in Chapter 5, and the improvement goal 2, to designing Stra2Bis, detailed in Chapter 6.

4.5 Summary

Chapter 4: Baseline method analysis

- This chapter presented the analysis the challenges of the baseline model-driven software production method for including organisational information on business strategy and organisational structure, addressing **(KQ2)**.
- We analysed the baseline MDD method through a set of mechanism experiments to test whether the current method supports modelling and preserving organisational information.
- We identified five challenges for the baseline MDD methods' goal modelling method (i*) and transformation technique (GoBIS) to model and preserve organisational information.
- Three experts in the baseline MDD method reviewed the analysis results and supported four out of the five challenges. Two improvement goals to address the supported challenge were distilled after the expert review.
- The first improvement goal is using a specific modelling language and procedure for business strategy and organisational structure in order to avoid overlapping between the organisational and business process modelling levels (Challenge 1) with constructs which are meaningful for the organisational domain (Challenge 2).
- The second improvement goal is using organisational model information to design modular business processes in order to exploit the organisational structure hierarchies and dependencies (Challenge 3) and avoid coupling independent business processes into a single model (Challenge 5).

Part II

Treatment Design

Chapter 5

Design of an organisational modelling method

5.1 Motivation

The baseline MDD method considers i* to model social actors' goals and GoBis to map this information into the business process modelling level. However, in Chapter 4, we identified challenges for the existing methods to represent business strategy and organisational structure and to preserve the information relevant to the alignment approach of Software Organisations (SO). In this chapter, we address the first improvement opportunity that such challenges set to the baseline MDD method: Use a specific modelling language and procedure for business strategy and organisational structure. We follow the design science (DS) methodology as the research method and the situational method engineering (SME) approach as the design method for creating a novel organisational modelling method, as described in Section 1.5.

Following the design science methodology (Wieringa, 2014), it is necessary to revisit the context of the artefact to be designed. The social context presents the domain knowledge relevant to enabling the strategic alignment approach of software organisations. The knowledge context discusses the model-driven development context for the new method. From the design methodology perspective, the social and knowledge contexts set the basis for applying Situational Method Engineering (SME) (Henderson-Sellers, Ralyté, et al., 2014) to assemble the organisational modelling method. The requirements for the method come from the need to enable an engineering practice, the SO's strategic alignment approach, presented in the knowledge context. The method base from which the new method is assembled comprises the methods presented in the knowledge context.

This chapter presents the design of the LiteStrat method, which addresses the design problem of how to model organisational information relevant for strategic alignment **(DP1)**. In Section 5.2, the social context from which the method requirements are elicited is presented. Section 5.3 presents the design of how the domain conceptualisation defined in Section 3.3 will be represented in the modelling method. Section 5.4 describes the requirements for the design of LiteStrat. In Section 5.5, we detail how the selection and assembly of method chunks from the method base, while Section 5.6 presents an application example. The implementation of a tool for supporting the method is described in Section 5.7, and finally, Section 5.8 summarises the chapter.

5.2 Social context for the organisational modelling method

5.2.1 Strategic alignment in software organisations

The social context for the new method is the *adoption of the existing model-driven development method by software organisations.* As reviewed in Chapter 2, software organisations have a strategic alignment approach where instead of analysing the goals of different actors across different departments or areas in the organisation, they focus on designing an organisational structure aligned with the strategic goals, forming independent and self-sufficient teams (Highsmith, Luu, and Robinson, 2019; Forsgren, Humbpotifle, and Kim, 2018; Scaled Agile, INC, 2021).

5.2.2 Effects of Software Organisation's alignment approach in software development

The approach of software organisations to strategy and organisational structure affects software development in the organisation of multidisciplinary, autonomous teams[1] (Highsmith, Luu, and Robinson, 2019) and, more important, in the design of software components and services. Each team takes full responsibility for a part of the business domain. Through modelling techniques such as Domain-Driven Design (E. Evans and E. J. Evans, 2004), the business domain model is split into sub-domains, and the interfaces between them are defined. The teams offer their part of the business domain *as-a-service* to other teams (Skelton and Pais, 2019). In this way, modularising the organisation units and the software components around business outcomes aligns business strategy with architecture.

In the above context, a model-driven development method that enables strategic alignment must include the organisational domain knowledge regarding *business strategy* and the *organisational structure* defined to implement the strategy. Below, we discuss these domain elements, while their relationship with the strategic alignment of information systems is discussed in Chapter 6.

5.2.3 Business strategy in Software Organisations

Strategy, as defined by classic management literature, encompasses various aspects, such as the organisation's plan to achieve business goals, specific ploys to exploit situations, market positioning, stakeholder perspectives, and patterns of action (Mintzberg, 1987). Existing frameworks for agile organisations, including those by Holbeche (Holbeche, 2018), Highsmith, Luu and Robinson (Highsmith, Luu, and Robinson, 2019), and the Scaled Agile Framework (Scaled Agile, INC, 2021), emphasise defining business goals and corresponding courses of action. These frameworks advocate setting high-level, customer-focused goals and then breaking them down into measurable objectives and key results, aligning with an outside-in perspective and adaptive strategy. For instance, Google's OKR model (Doerr, 2018) and the EDGE framework by Thoughtworks (Highsmith, Luu, and Robinson, 2019) propose similar approaches, focusing on measurable key results and targets for achieving high-level goals.

A distinctive feature of software organisations is their ability to continuously adapt their strategic direction, considering it a crucial success factor (High-

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

smith, Luu, and Robinson, 2019). Unlike traditional organisations, software organisations can swiftly modify their business strategy and organisational structure, sometimes on a quarterly basis (Aghina et al., 2017). Despite the dynamic nature of agile strategy, the frameworks align with the classic definition of *business strategy as a plan* by Mintzberg (Mintzberg, 1987). However, they stand out for promoting a lightweight strategic planning approach, contrasting with traditional heavyweight processes. An illustration of this lightweight approach is the GOST framework (Horwath, 2014), which outlines high-level goals and strategies and more detailed objectives and tactics as two distinct levels within the organisational pursuit of its ends.

5.2.4 Organisational structure in software organisations

On the other hand, *organisation structure* deals with the configuration of individuals and groups for allocating tasks, responsibilities, and authority within the organisation (Lunenburg, 2012). Different frameworks for software organisations propose a decentralised, matrix structure, where the *basic unit of organisation is a team* (which can have different names in different frameworks such as squad (Atlassian, 2021) or agile release train (Scaled Agile, INC, 2021), which is a *cross-functional organisation unit working towards a specific business goal.* The matrix structure is given by the fact that team members are also part of groups according to their technical specialities to manage the technical know-how. For instance, in the Spotify model (Atlassian, 2021), a team (named *squad*) is responsible for the playlist feature of the product, while the UI/UX specialist of the squad also belongs to the UI/UX *chapter*, which groups UI/UX specialists across the squads.

Agile frameworks emphasise the importance of defining *organisational structure since it affects the architecture of the organisation's software products.* Forsgren et al. (Forsgren, Humbpotifle, and Kim, 2018) found that independent, cross-disciplinary organisation units or teams yield loosely coupled systems, which improve software development performance and scalability. Most of the agile software development frameworks have adopted this approach (Scaled Agile, INC, 2021; Larman and Vodde, 2016), which is based on the principle that organisations replicate their communication structure to everything they design, following Conway's Law (Conway, 1968; Fowler, 2022). *Inverse Conway Manoeuvre* (Forsgren, Humbpotifle, and Kim, 2018; Fowler, 2022) is an approach for *evolving the organisational structure, so business architecture matches the desired system architecture.* A loosely-coupled organisation structure design sets requirements for designing loosely-coupled business processes and for the information systems that support them, which translates into greater maintainability, scalability, and a more efficient software development delivery (Forsgren, Humbpotifle, and Kim, 2018).

5.2.5 Business strategy and organisational structure relationship

While the formal relationship between business strategy and organisational structure is still a matter of study in management (Lunenburg, 2012), both classic and agile approaches point to organisational structure following business strategy; this is, the organisation is structured according to strategic definitions. In traditional organisations, strategic planning and organisation restructuring are long-time efforts and barely important to set requirements for a specific software development endeavour. However, in software organisations, business strategy is defined for the scope of a few months (Highsmith, Luu, and Robinson, 2019) and drives organisational *reconfiguration*. Organisational reconfiguration deals with adding, redeploying, recombining and divesting resources and organisational units (Karim and Capron, 2016) to implement the strategy. For instance, a software organisation could set a goal of reaching more customers of a particular segment through a new feature in the software product. The organisation should decide whether an existing team addresses the feature or form a new one.

5.2.6 Design goals from the social context

We summarised and categorised the social context elements into four design goals, presented in Table 5.1. These goals are specified in the method's requirements in Section 5.4.

5.3 Representation of the domain conceptualisation

The domain conceptualisations proposed in Section 3.3 provide the concepts to unambiguously describe the information about business strategy and organisational structure in the context of the strategic scenario that drives a software engineering endeavour. In order to design a modelling method which satisfies the method goals previously described, the modelling language of the method needs to commit to the domain conceptualisation. To achieve this, we follow two steps. First, we scope the modelling language to the domain conceptualisations relevant to achieving the method requirements. Then, we group and simplify the domain concepts to the smallest possible number of elements of

Table 5.1: Design goals for the LiteStrat method.

	Design Goal	Description
DG1	Business strategy as a plan	Support the organisational ends and the means to achieve them
DG2	Outside-In approach	Support modelling the external elements affecting the organisation as the trigger of the business strategy
DG3	Organisational structure configuration	Support modelling the organisational units and roles according to the strategy and the desired system architecture
DG4	Lightweight approach	Support modelling the above elements with lightweight techniques and a minimal set of constructs

the modelling language, provided that the concepts can still be unambiguously identified in the model.

5.3.1 Scoping the modelling language

We scope the modelling language to the domain conceptualisations presented in Section 3.3: strategic scenario, organisational ends and means, and organisational structure.

- **Strategic Scenario:** A model produced by the method represents a strategic scenario, as a configuration of intentional moments of agents. Particularly, between the external influence of agents on the organisation, the internal influences among the organisational units, and the reaction influence from the organisation towards agents outside the organisation. The model does not explicitly represent the value proposition, as it is considered the emergent meaning of the strategic scenario. Also, the organisational commitments (organisational purpose and organisational units' purpose) are stable definitions (i.e., changing them will transform the organisation into a different organisation), so we argue that it is not needed to model them for a particular software development endeavour.
- **Ends and Means:** A model produced by the method describes the business plan to address the strategic scenario, in terms of goals and the actions to achieve them. Particularly, describes the organisational goals, organisational objectives, the strategic actions, and the tactical actions.

With regard to the links between the concepts, a model produced by the method considers that an organisational goal is *operationalised by* a strategic action, a strategic action is *decomposed into* tactical actions, and a tactical action *contributes to* an organisational objective.

- **Organisational Structure:** A model produced by the method describes the organisation and its organisation units and the organisational roles allocated to them which are relevant for the strategic scenario. Regarding the links between the concepts, an organisational unit delegation supports the hierarchical relationship between the organisation and an organisation unit. On the other hand, connecting an organisational with an organisational unit requires several concepts of the domain: an organisational unit can be the allocation unit for the organisational unit assignment of an organisational unit member performing an organisational role.

5.3.2 Grouping and simplifying

For the strategic scenario, we propose grouping External influence, Internal Influence, and Reaction Influence into a unique modelling language element, *influence*. We group these concepts since the combination of the connected concepts and the direction of the relationship unambiguously describe external influence (from an actor to the organisation), internal influence (from an organisational unit to another organisational unit), and creation influence (from the organisation to an external agent.

For the ends and means concepts, we propose grouping the *operationalised by*, *decomposed into*, and *contributes to* links into a single type of link, namely a *refinement*. We chose this name based on existing goal modelling frameworks that connect goals and tasks (or actions) refinement links. It is possible to unambiguously determine the link type based on the items that the refinement connects. An organisational role refined into a strategic action represents an *operationalised by* link, a strategic action refined into a tactic action represents a *decomposed into*, and a tactic action refined into an organisational objective represents a *contributes to* link.

For the organisational structure, we merged the organisation and organisation unit into a single concept, the organisation unit. Besides reducing the number of concepts, we argue that this would allow the analysis of an organisational unit as if it were an independent organisation in a strategic scenario with its own action plans and internal structure. The organisation could still be identified in the model as the organisation unit that does not belong to another organisation unit. Regarding the relationships, we grouped the organisational unit delegations and the allocation of a role to an organisational unit as a single language construct, the *assignment*. It is possible to unambiguously distinguish the delegation from the unit allocation by following the edges of the links: organisational units assigned to other organisational units represent *delegations*, while organisational roles assigned to organisational units represent *allocations*.

The resulting domain concepts to be represented by the modelling method are *agent*, *organisation unit*, *organisational goal*, *organisational objective*, *strategic action*, and*tactic action*. The relationships are *influence*, *refinement* and *assignment*.

5.3.3 Knowledge context for the organisational modelling method: Method base

The knowledge context for designing the required organisational modelling method regards conceptual modelling frameworks that include the organisational domain in their conceptualisations. From the knowledge context discussed in the previous section, we focus on representing the organisational domain knowledge regarding *business strategy as a plan*, this is, the organisational goals and the means to achieve them, and the *organisational structure*.

As reviewed in Chapter 2, enterprise architecture frameworks such as Zachman's framework (Zachman, 1987), TOGAF (The Open Group, 2018), ARIS (Santos Jr, Almeida, and G. Guizzardi, 2010), ArchiMAte (The Open Group, 2022a), and the Business Motivation Model (BMM) (The Object Management Group, 2015) have conceptualised most of the organisational domain information which is relevant for aligning information systems with business strategy. For applying the Situational Method Engineering (SME) (Henderson-Sellers, Ralyté, et al., 2014) approach to design LiteStrat, we consider ArchiMate as the first source method, in particular the *strategy* and *motivation*, and *business structure* elements. Since version 2.1, ArchiMate has included motivation and strategy elements inspired by the Business Motivation Model (BMM).

However, as will be reviewed in the next subsections, BMM provides different abstraction elements for the organisational ends and means, which matches the need to break down the ends as means, as required by the social context presented in the previous section. Based on this, we also take into account BMM as a source modelling method. The third source modelling method is i*, in particular iStar 2.0 (Dalpiaz, Franch, and Horkoff, 2016), since it is the current method for addressing organisational elements in the existing model-driven development method. Below, we analyse the three modelling methods from the perspective of who they aim to represent business strategy (as a plan) and the organisational structure.

ArchiMate

The most recent versions of ArchiMate (The Open Group, 2022a) have extended the representation of the business layer by adding business strategy elements, having separated elements to describe the organisational ends and means. On the one hand, ArchiMate's Motivation Elements, depicted in Figure 5.1, support describing the organisational ends. The *goal* concept describes high-level statements of intent, direction, or desired end state for an organisation., while *outcomes* represent a result, effect, or consequence of a certain state of affairs which are tangible, possibly quantitative, and time-related.

On the other hand, the means to achieve the strategy are included in ArchiMate's Strategy Layer (The Open Group, 2022d) addresses the concept of *course of action* as approach or plan for configuring some *capabilities* and *resources* of the enterprise, undertaken to achieve a goal. The *value stream* concept represents a sequence of activities that create an overall result. The metamodel of ArchiMate's strategy layer is presented in Figure 5.2.

The relationship between the strategy layer and the motivation layer is depicted in Figure 5.3: strategy's *courses of action* realises motivation's *outcomes*, which at the same time realise the organisation stakeholders' goals.

On the other hand, regarding organisation structure, ArchiMate does not provide a layer or a set of elements under this concept. Still, in ArchiMate's business layer, the inner structural elements are specified (The Open Group, 2022b), as shown in Figure 5.4. As can be seen, *business role* and *business actors* are represented, as well as *business collaborations* where two or more structure elements work together. Please note that business collaborations specify time-limited collaborations, while more stable associations such as departments, areas or development teams are meant to be modelled as business actors (The Open Group, 2022b).

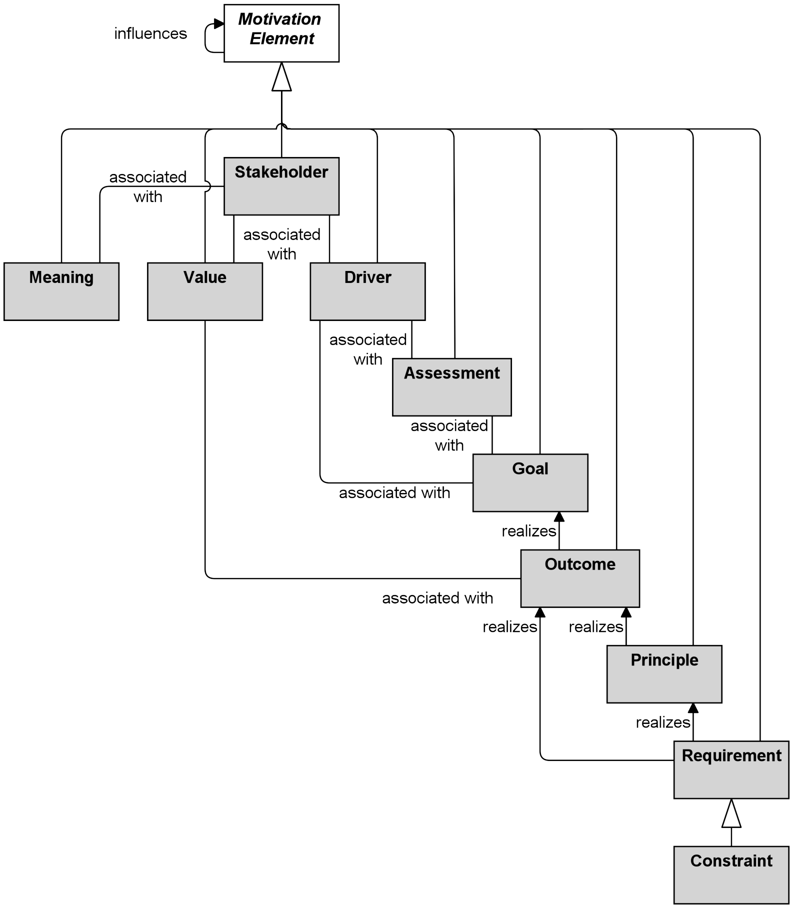


Figure 5.1: Metamodel for ArchiMate's motivation elements from (The Open Group, 2022c)

The Business Motivation Model

Unlike ArchiMate, the Business Motivation Model (BMM) modelling concepts are not split into layers. Figure Figure 5.5 presents BMM metamodel. Regarding the business strategy and planning concepts, two main concepts specifically address the organisational ends and means.

The *ends* concept is specialised into *vision*, concerning the long-term ends of the organisation, and into the more specific concept *desired results*, which in turn is specialised into the *goal* concept and the *objective concept*. While

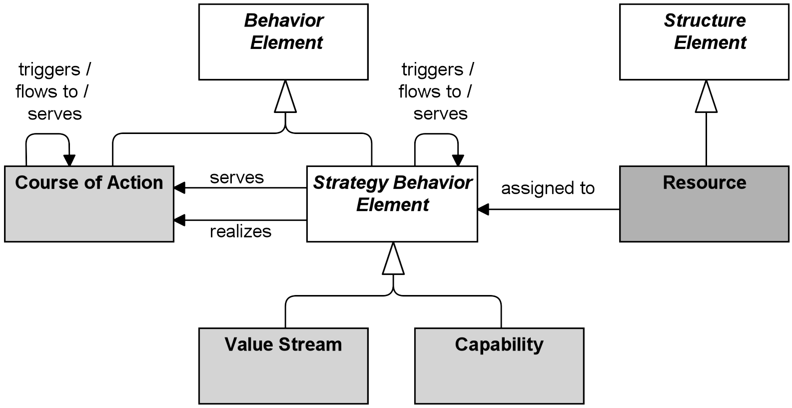


Figure 5.2: Metamodel for ArchiMate's strategy layer (from (The Open Group, 2022d))

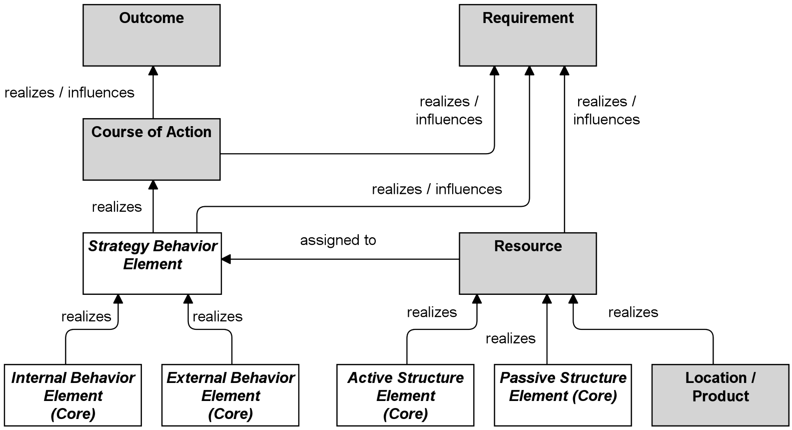


Figure 5.3: Metamodel for the relationship between ArchiMate's strategy layer and motivation elements (The Open Group, 2022d)

the goal concept matches the definitions in ArchiMate's motivation elements, objectives should always be time-targeted and measurable (The Object Management Group, 2015).

The *means* concept is specialised into the *mission*, *directive*, and *course of action* concepts. *Mission* represents the long-term definition of what the organisation does, while *directives* support specifying business rules and business policies. On the other hand, *Courses of Action* supports the achievement of the desired results and is specialised into the *strategy* and *tactic* concepts.

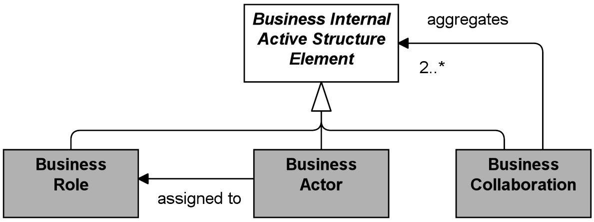


Figure 5.4: Metamodel for internal structural business elements in ArchiMate's (The Open Group, 2022b)

Strategy is a high-level definition of a course of action, representing what the organisation considers the right approach to achieve its *goals*. In turn, the *tactic* concept details strategies and channel efforts towards *objectives*. BMM does not conceptualise concepts related to material or informational elements to implement the courses of action, though it references *assets* and *resources* as *placeholders*; this is, belong to other conceptualisations but are related to BMM.

Regarding organisational structure, the Business Motivation Model specification version 3.1 also identifies the *organisation unit* concept as external to BMM, since it was meant to be addressed by the Object Management Group's Organization Structure Metamodel[2]. However, the specification uses them as a placeholder for *parts of the organisation* to which the strategic plans are linked. The specification also acknowledges that an organisation's inner organisation units can affect other units (The Object Management Group, 2015). No further definitions regarding other elements of the organisation structure (such as roles) are provided.

A relevant aspect of BMM is that, unlike i* and ArchiMate, it presents a modelling procedure, which, under the SME approach, refers to the process perspective. In Figure 5.6, the modelling process for BMM is depicted. Please note that the process describes an outside-in perspective to strategy, similar to the one adopted by software organisations. For this approach, external entities on the organisation's strategy are modelled as *influencers*, which could be *internal* or *external*, being *regulations* a specific type of the latter.

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

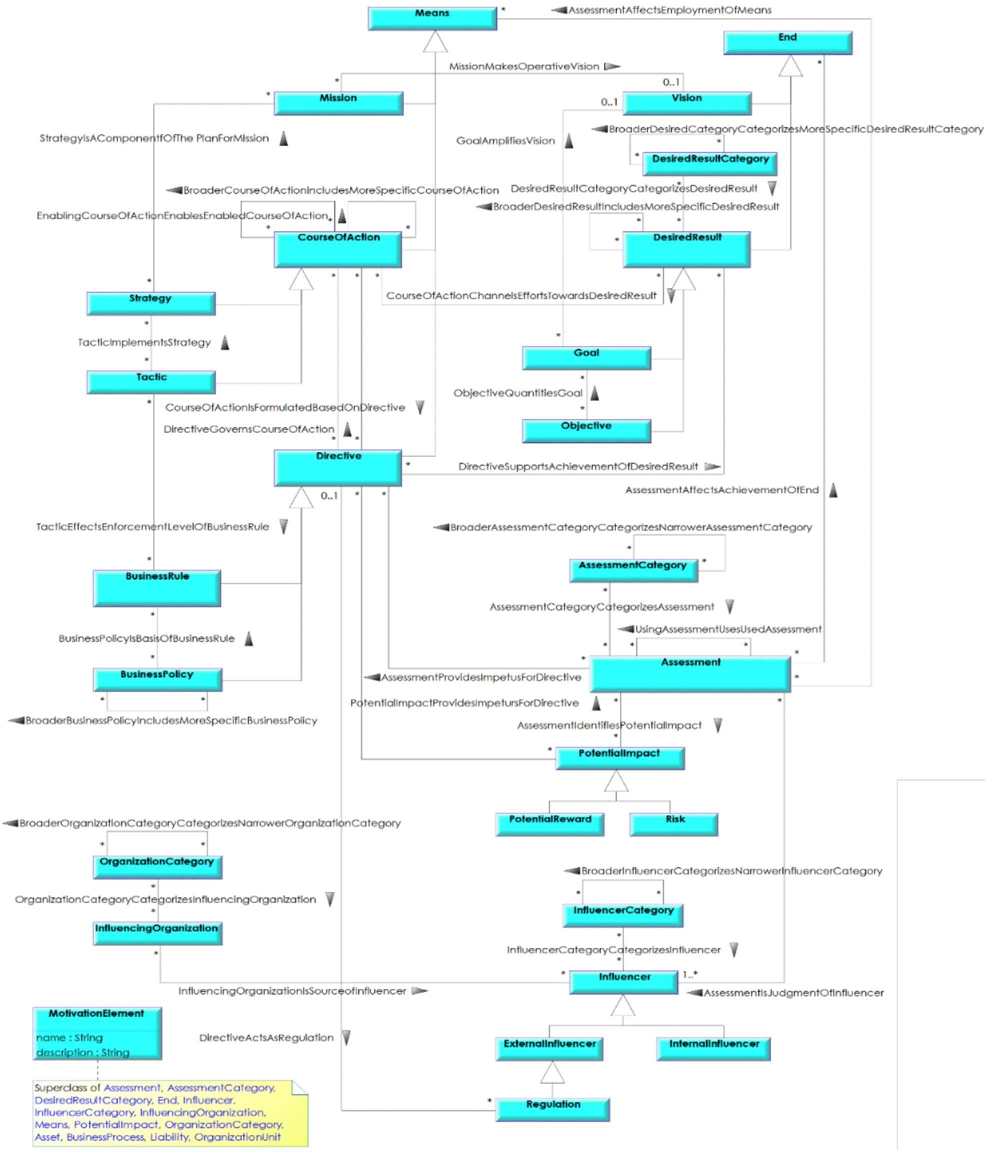


Figure 5.5: Metamodel for Business Motivation Model (The Object Management Group, 2015)

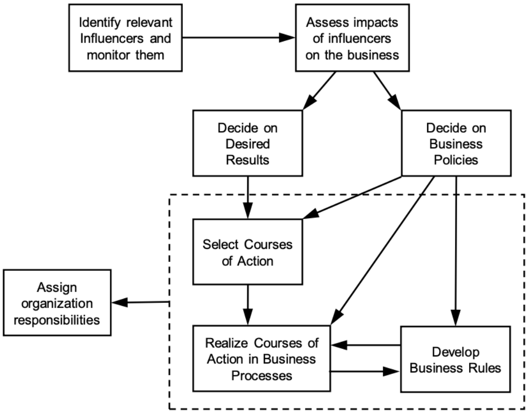


Figure 5.6: Modelling process for Business Motivation Model (The Object Management Group, 2015)

i (iStar 2.0)*

Finally, i* metamodel for its most recent version, iStar v2.0 (Dalpiaz, Franch, and Horkoff, 2016) is depicted in Figure 5.7.

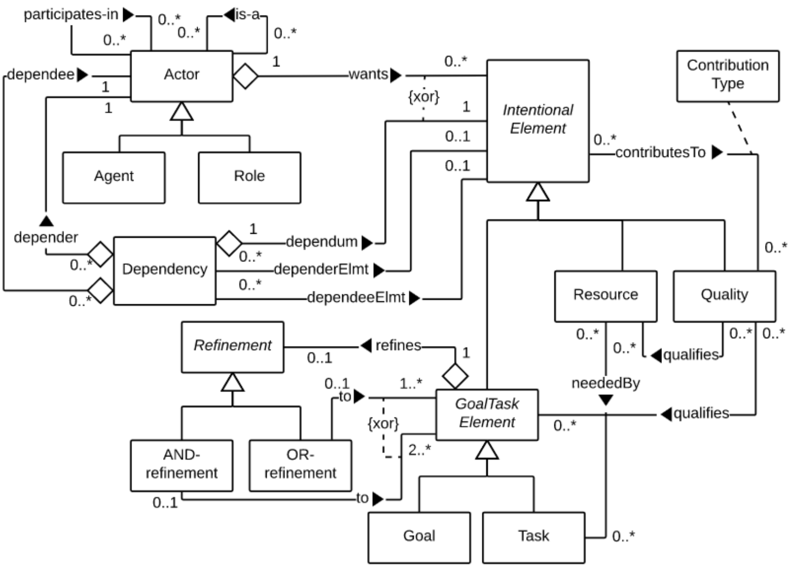


Figure 5.7: Metamodel for iStar (from (Dalpiaz, Franch, and Horkoff, 2016))

Regarding business strategy ends and means, i* provides four intentional that can have been applied in organisational modelling. Concerning organisational ends, *goals* represent a state of affairs that the actor wants to achieve, with clear-cut criteria of achievement (Dalpiaz, Franch, and Horkoff, 2016), which is similar to the definition in ArchiMate and BMM. No more specific concepts like BMM's *objectives* or ArchiMate's *outcomes* are provided. Regarding the organisational ends, *tasks* are actions to be executed, usually towards achieving a goal, while *resources* are physical or informational entities required to perform a task. A fourth element that can be associated with the organisational ends or means (goals or tasks) are *qualities*, which regard attributes for which an actor requires specific levels of achievement. However, unlike BMM's objectives, quality levels are not necessarily expressed quantitatively. The goals and tasks can be *refined* from high level to more specific elements.

Concerning the organisational structure, i* presents three types of social actors. *Actors* are entities aiming to achieve goals through their know-how and collaborating with other actors. There are two types of actors: *Roles*, which characterises a behaviour in a specific context, and *Agents*, an actor with concrete manifestations. Agents are usually employed for modelling organisations and organisation units. The *participates-in* relationship supports representing the organisational structure in terms of agents participating in other agents and roles participating in agents.

Finally, the *dependencies* between the social actors represents how actors collaborate to achieve their goals.

Summary of concepts in the method base

Table 5.2 summarises the available concepts for expressing business strategy's ends and means and the organisational structure. It is worth noting that ArchiMate and BMM present concepts with different abstraction levels for representing the organisational ends: *goals* and *outcomes* in ArchiMate, *goals* and *objectives* in BMM; in turn, i* only defines *goals*. Similarly, for the organisational means related to actions, ArchiMate offers the *course of action* and *value stream* concepts, BMM has *strategy* and *tactic*. At the same time, i* has only the *task* concept. ArchiMate and i* include the elements needed to perform the actions (resources, capability), while BMM considers them out of the scope of the definition of the business strategy.

Regarding the approaches to model business strategy, just the one by BMM provides a modelling procedure which reflects the outside-in perspective of software organisations. Finally, regarding organisation structure, ArchiMate lacks a specific concept to model organisation units, though the business actor concept is recommended since business collaboration has a limited time scope (The Open Group, 2022b). On the other hand, BMM does not include organisational structure in its concepts, though it acknowledges the relationship between the definitions of organisation units acting as influencers of business strategy. Finally, i* offers a complete approach to represent organisation units (modelled as agents) and roles. However, the concept *agent* could also be applied to several other real-world entities besides organisation units.

Table 5.2: Summary of business strategy and organisational structure concepts and approaches from the source methods.

	Business Strategy			
	Ends	***Means***	***Strategic Approach***	**Organisational Structure**
ArchiMate	goal, outcome	course of action, value stream; capability, resource	-	business role, business actor, business collaboration
BMM	goal, objective	strategy, tactic	outside-in	influencing organisation
iStar	goal	task, resource	-	agent, role

5.4 Method Requirements

From the social context's design goals, we specify the requirements for the organisational modelling method. In Figure 5.8, we present the requirements map (Rolland, 2007), following the SME methodology (Henderson-Sellers, Ralyté, et al., 2014). The requirements are specified under the following rationale: the *intentions* of the method, depicted as ellipses, regard conceptual modelling of business strategy and organisational structure knowledge. The *strategies* to achieve the method's intentions, depicted as arrows, regard business strategy techniques to analyse the domain.

The method starts with the *external actor analysis* strategy, which aims to identify the external actors affecting the organisation. This permits achieving the first intention of the method, which is *external influence modelling*, which aims to represent the information needed to describe what is happening in the organisation's environment and what business goals are affected. This intention addresses an outside-in approach to strategy, addressing the design goal DG3 from Table 5.1.

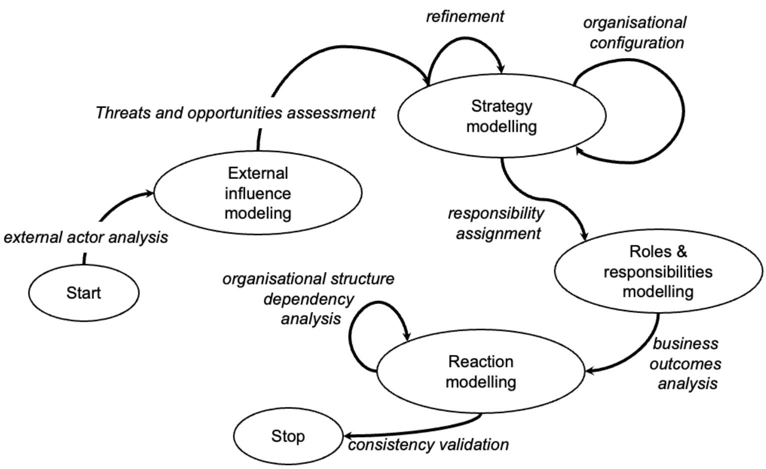


Figure 5.8: Requirements for the organisational modelling method.

Then, a *threats and opportunities assessment* must be performed so the business strategy to react to external elements can be defined. *Strategy modelling* regards representing the strategic actions that can be decomposed into more specific levels through *refinement*; also and jointly deals with representing the organisational structure that will deploy the strategy, which is achieved by the *organisational configuration* strategy. This intention addresses modelling business strategy as a plan and jointly considers the organisational structure needed to deploy the strategy, managing the design goals DG1 and DG2 from Table 5.1.

The *responsibility assignment* strategy allows identifying who is responsible for achieving specific results of the strategy, which are represented in the *Roles & responsibilities modelling intention.* This intention also supports modelling business strategy and organisational structure in its more detailed elements (objectives and roles), also realising DG1 and DG2 from Table 5.1.

Finally, the method should support *reaction modelling* to represent how the organisation is supposed to affect its environment once the strategy is implemented. This is achieved by the *business outcomes analysis* strategy. Similarly, the new dependencies in the organisational structure must be identified, which is achieved by the strategy *organisation structure dependencies analysis.* The application of the method finishes with an *integrity validation* to ensure the organisational model is complete and correct. This intention closes the outside-in approach to strategy, addressing the design goal DG3.

Regarding the design goal DG4 concerning the lightweight approach of the method, it is addressed by selecting the bare minimum number of method chunks to satisfy the method's requirements.

5.5 Selection and assembly of method chunks

In order to fulfil the modelling intentions of the requirements map in Figure 5.8, we carefully selected concepts from the existing modelling methods presented in Section 5.3.3. It is worth noting that we are considering each concept in the current modelling methods as a method chunk since it autonomously defines a product (a piece of information of the modelled domain) and how to produce it (how to model it). We apply the SME approach of *assembly by integration*; this is, we connect overlapping concepts between the existing modelling methods to form a new one.

Next, we present the rationale behind the method chunk (i.e., concepts) selection and then the assembled method from the product and process perspective. The assembled method is named *LiteStrat*, being *lite* and *strat* short and informal names for lightweight and strategy from software and gaming domains, respectively.

5.5.1 Method chunks selection

For each of the intentions in the requirements map of the method depicted in Figure 5.8, we selected a set of concepts from the source methods presented in Section 5.3.3: i* (Yu, 2011b), ArchiMate (The Open Group, 2013) and Business Motivation Model (BMM) (The Object Management Group, 2015).

The *External influence modelling* intention addresses modelling entities outside the organisation that affect the organisation. We selected the *actor* concept, which is common to i* and Archimate, for representing entities in the organisation's environment. The effect of actors over the organisation is represented by an *influence* relationship, which is found in Archimate and BMM. The organisation itself is represented by the concept *organisation unit*, which is not explicitly defined in any of the source methods. ArchiMate states that the actor concept could serve this purpose (The Open Group, 2013), while i*'s actors and agents have been used for it (Yu, 1995); on the other hand, BMM specification (The Object Management Group, 2015) references organisation units as a concept from organisational structure model by OMG which was not officially released. Given the ontological relevance of the organisation unit concept (Santos Jr, Almeida, and G. Guizzardi, 2013), we opted to differentiate it from i* and ArchiMate's actors.

For the *Strategy modelling* intention, we selected the four concepts from BMM, as they provide different abstraction levels for desired results (*goals* and the more specific (*objectives*) and courses of action (*strategies* and *tactics*), which matches the GOST approach to define a strategic plan (Horwath, 2014). Also, BMM explicitly states that these specifications are meant to be realised by business processes, *providing an integration point suitable for a model-driven context*. These features are not present in the other referenced frameworks. We considered i*'s *refinement* relationship to connect more abstract elements with more specific ones.

For the *Role & responsibility modelling* intention, we aimed to represent the assignment of the strategy elements (previously modelled) to organisational actors. For representing organisational actors, we reuse the *organisation unit* concept and select the *role* concept, shared by ArchiMate and i*. To connect the strategy elements with the organisational actors, we selected the *assignment* relationship from ArchiMate, in a similar way as i* actor's inner intentional elements are assigned.

Finally, for the *Reaction modelling* intention, we aimed to represent the effects of the organisation on its environment and the effects between the inner organisation units. We reuse the *actor* concept to represent the external entities affected by the organisation and the *influence* relationship to connect them. Similarly, we use the *influence* relationship to connect *organisation units* to represent the effects among them.

5.5.2 LiteStrat's product perspective

The assembly of the concepts selected from the existing modelling methods is presented in Figure 5.9, in the form of the method's metamodel. In Table 5.3 and Table 5.4, we describe the concepts and relationships, respectively. For each of them, we present the source method from which they were selected and the proposed notation.

The proposed notation is mostly based on i*. Actors and roles preserve the i* notation, while we use i*'s agent notation for organisation units. We also keep the notation for goals, while for more specific objectives, we use the same symbol, adding vertical lines to state the difference. For strategies, we use i*'s notation for tasks and add two straight vertical lines for tactics. Please note that we acknowledge the importance of designing a user-centred notation since it is a factor affecting the method's understandability (Lindland, Sindre, and Solvberg, 1994); however, this research problem is outside the scope of this thesis.

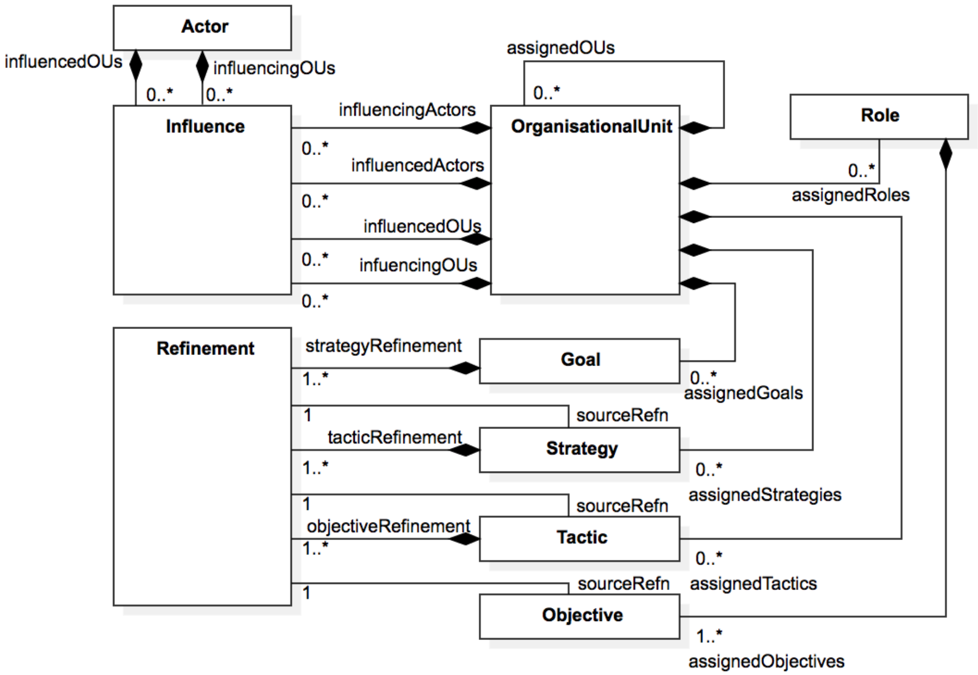


Figure 5.9: LiteStrat metamodel.

5.5.3 LiteStrat's process perspective

Following the SME approach, we detail the guidelines for assembling the concepts selected in the previous section. As described in Section 1.5, we present the procedure in terms of three types of guidelines: Intention Selection Guideline (ISG), Strategy Selection Guideline (SSG), and Intention Achievement Guideline (IAG). Please note that IAGs are actually conceptual modelling guidelines, while ISG and SSG guidelines refer to analysis techniques from the business strategy domain, which enable analysts to collect and determine what domain information must be modelled.

Table 5.3: Constructs of the LiteStrat method.

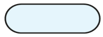
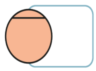
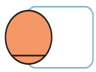
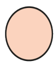
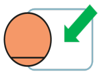
Constructs	**Domain Concept**	**Definition**	**Source Method**	**Notation**
Goal	Organisational goal	A statement about a state or condition of the organisation to be achieved by appropriate strategies.	BMM, ArchiMate, i*	
Objective	Organisational objective	A statement of an attainable, time-targeted, and measurable target that the enterprise seeks to meet to achieve its goals.	BMM	
Strategy	Strategic action	Represents a high-level action towards the achievement of a goal.	BMM, ArchiMate	
Tactic	Tactical action	Represents more concrete actions towards the implementation of a strategy.	BMM	
Organisation Unit	Organisation unit	Represents a group of social actors working together to achieve a goal. It could represent the organisation under analysis and its sub-units.	ArchiMate, i*	
Role	Organisational role	Represents abstractions of well-defined behaviors in the organisational context.	ArchiMate, i*	
Actor	Agent	Represents entities that are external to the organisation and whose behaviour affects or is affected by the organisation.	ArchiMate, i*	

Table 5.4: Relationships of the LiteStrat method.

Relationships	**Domain Concept**	**Definition**	**Source Method**	**Notation**
Refinement	Refinement	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).	i*	
Assignment	Organisational unit assignment	Is a structural relationship that represents the allocation of responsibility, the performance of behaviour, or execution.	Archimate	
Influence	Influence	Is a dependency relationship that describes an action or behaviour of the source element (actor or organisation unit) that affects the goals of the target element (an actor or organisation unit).	Archimate	

In Table 5.5, we present LiteStrat's IAGs following the structure recommended by SME (Henderson-Sellers and Ralyté, 2010) in terms of an ID for the guideline, the section describing the source intention, the target intention, and the strategy, and the guideline description. We added a fourth column to specify the product elements added by achieving each intention in terms of LiteStrat's metamodel concepts presented in the previous section.

In Table 5.6, we present the intention selection guidelines (ISG), which guide the method's users to choose how to move from one intention of the method to the next.

Finally, in Table 5.7, we define the strategy selection guidelines (SSGs), which guide the method's users to choose between the IAGs in Table 5.5 for moving from one intention to the chosen according to the ISG in Table 5.6.

Table 5.5: Intention achievement guidelines for the LiteStrat method.

ID	Section	Guideline description	Related Products
IAG1	<Start, External influence modelling, *external actor analysis*>	This guideline proposes to identify actors who are external to the organisation and whose influence affects the organisation's goals.	actor, organisation unit, *influence*
IAG2	<External influence modelling, Strategy modelling, *threats and opportunities assessment*>	This guideline proposes to identify the goals that the organisation wants to meet under the influence of the external influence and to refine them into strategies to achieve them.	goal, strategy, *refinement*
IAG3	<Strategy modelling, Strategy modelling, *refinement*>	This guideline proposes to refine the strategies into more specific tactics and goals into measurable objectives related to such tactics.	tactic, strategy, *refinement*
IAG4	<Strategy modelling, Strategy modelling, *organisational configuration*>	This guideline proposes to identify, add, merge or split the organisational units needed to deploy the strategy and assign them tactics and objectives.	organisation unit, assignment
IAG5	<Strategy modelling, Roles & responsibilities modelling, *responsibility assignment*>	This guideline proposes to identify or define new organisational roles that will be responsible for achieving or tracking the achievement of the objectives, and assign those objectives to the corresponding roles.	role, *assignment*
IAG6	<Roles & responsibilities modelling, Reaction Modelling, *business outcome analysis*>	This guideline proposes to identify the actors in the organisation's environment that are expected to be influenced by the outcomes of the strategy. Also, other actors in the organisation's environment that could influence the strategy's successful implementation are identified.	actor, *influence*
IAG7	<Reaction Modelling, Reaction Modelling, *organisational structure dependency analysis*>	This guideline proposes to identify how the tactics implemented by each organisation unit influence other organisation units.	*influence*
IAG8	<Reaction Modelling, Stop, *integrity validation*>	This guideline proposes to check if the modelling elements have been correctly assembled according to the metamodel and the modelling guidelines.	Intention selection and strategy selection guidelines.

Table 5.6: LiteStrat's intention selection guidelines.

ID	Intention	Guideline Description
ISG1	I_0: Start	Progress to intention *I_1: External Influence modelling* if the following preconditions are satisfied: - At least one actor offering a business opportunity or threat for the organisation's goals has been identified. - The opportunity or threat affects the organisation's software products.
ISG2	I_1: External Influence modelling	Progress to intention *I_2: Strategy modelling* if the following preconditions are satisfied: - At least one organisational goal has been set in the context of the influence of the actors in the organisation's environment.
ISG3	I_2: Strategy modelling	Progress to intention *I_3: Roles & Responsibilities modelling* if the following preconditions are satisfied: - At least one strategy has been defined for achieving each goal. - At least one tactic has been defined to implement each strategy. - At least one objective has been defined for each tactic. - All the tactics and objectives have been assigned to at least one organisation unit.
ISG3	I_3: Roles & Responsibilities modelling	Progress to intention *I_4: Reaction modelling* if the following preconditions are satisfied: - All the objectives have been assigned to a role.
ISG4	I_4: Reaction modelling	Progress to intention *I_5: Stop* if one of the following preconditions is satisfied: - At least one actor in the organisation's environment meant to be influenced by the strategy are modelled. - All the influences among the organisation units are modelled.

Table 5.7: LiteStrat's strategy selection guidelines.

ID	**<Source intention, target intention>**	**Guideline description**
SSG1	<Start, External influence modelling>	Select IAG1 if there is a strategic scenario to be assessed which could affect the organisation's software product(s)
SSG2	<External influence modelling, Strategy modelling>	Select IAG2 if the actor(s) and influence(s) over the organisation have been modelled.
SSG3	<Strategy modelling, Strategy modelling>	Select IAG3 if the goals and strategies have been modelled. Select IAG4 if the tactics and objectives have been modelled.
SSG4	<Strategy modelling, Roles & responsibilities modelling>	Select IAG5 if the organisation units have been modelled and tactics and objectives have been assigned to them.
SSG5	<Roles & responsibilities modelling, Reaction modelling>	Select IAG6 if the roles for the organisation units have been modelled and objectives have been assigned to them.
SSG6	<Reaction modelling, Reaction modelling>	Select IAG7 if the influences of the organisation towards the actors in the organisation's environment have been modelled.
SSG7	<Reaction modelling, Stop>	Select IAG8 if the influences among organisation units have been modelled.

5.6 Application example

We illustrate the application of LiteStrat through the Example 5.1, which presents a business strategy scenario. The LiteStrat model representing the scenario is depicted in Figure 5.10. Below, we describe how the IAGs were applied to model the strategic scenario.

Example 5.1: Strategic scenario for S-Learn.

S-Learn is the leading 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's goals, but the content quality of S-Learn is far superior and could mitigate the threat. After this assessment, S-Learn executives decided that the company must maintain the sales 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 the Courses Squad (a multidisciplinary organisation unit specialised in producing and delivering courses) to adapt the online enrolment process and improve the security of the learning platform. The UX Designer must ensure that the enrolment process takes 5 minutes on average, and the Content Producer must ensure that 80% of the produced content implements DRM protection. (4) As a high drop-off is expected, the Sales area must increase the enrolment by 200%.

External influence modelling. By applying the guideline IAG1, the following elements are modelled: an actor (EManager), the organisation under analysis (S-Learn), and the influence (free courses).

Strategy modelling. From the application of IAG2, an organisational goal under external influence is defined (maintain sales projection for the next quarter). The strategy to achieve this goal is also modelled (offer a 30-day trial). Following the IAG3 guideline, the strategy's tactics are defined (improve the enrolment process, improve the content protection and sales process), and objectives are modelled for each tactic (enrolment in 5 minutes and DRM for 80% of content an enrolment increased by 200%, respectively). Following the

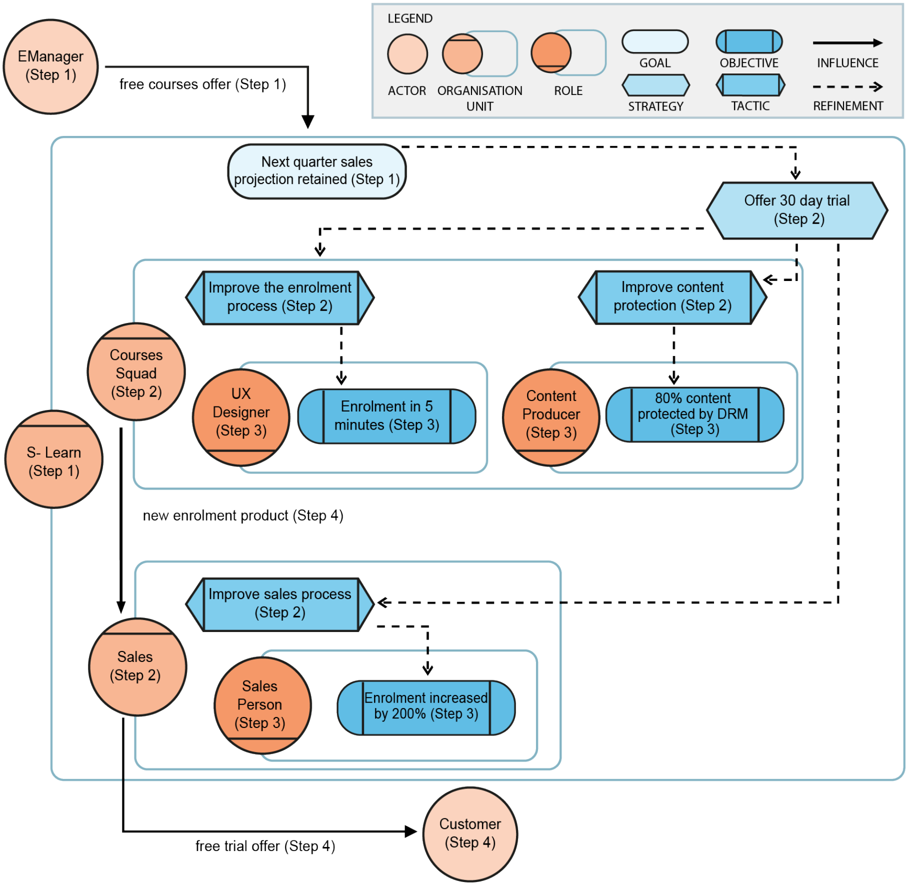


Figure 5.10: LiteStrat application example.

IAG4 guideline, the organisation units implementing the strategy are identified (Courses Squad and Sales department), and tactics are assigned to them.

Roles & responsibilities modelling. The application of IAG5 results in modelling the organisational roles for each organisation unit (UX Designer, Content Producer for the Contents Squad, and Sales Person for the Sales department), and objectives are assigned to them.

Reaction modelling. Following IAG6, the expected influence of the strategy and the target actors are modelled (free trial offer for Customers). The application of IAG7 results in modelling the influences between the organisational units; this is, the new enrolment feature produced by the Contents Squad influences the Sales department). Finally, integrity checking is performed according to IAG8.

5.7 Tool support

The tool support for the LiteStrat method is conceptually based on the ADOxx meta²model introduced in (Karagiannis et al., 2016). The metamodel is implemented by the ADOxx development toolkit (OMiLAB, 2023a). We adapted LiteStrat's conceptual metamodel (Figure 5.9) to the specific LiteStrat metamodel for ADOxx, presented in Figure 5.11.

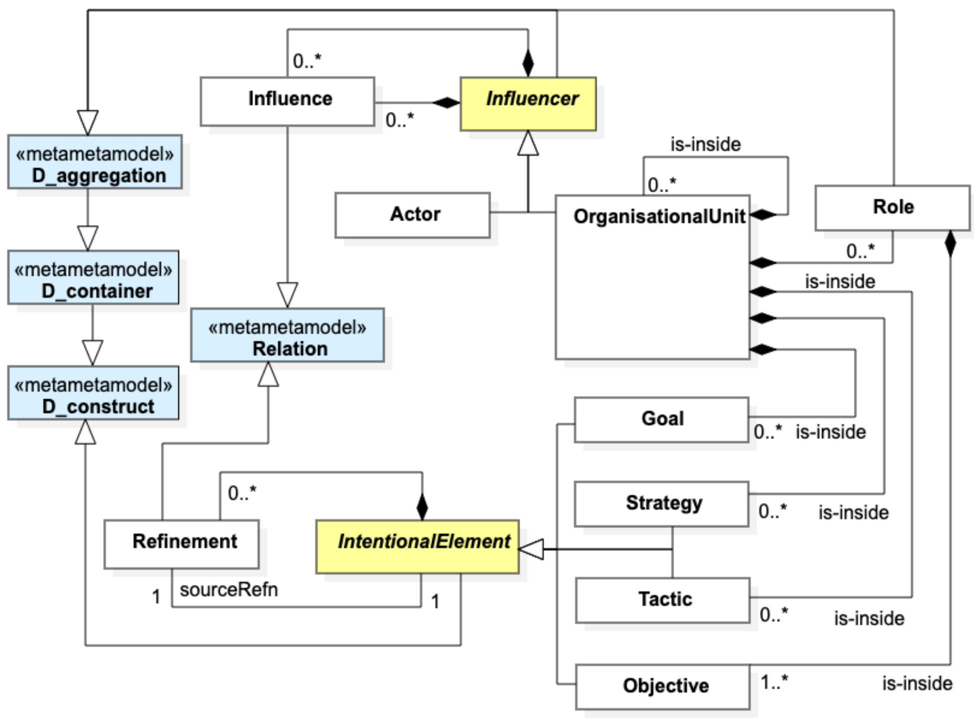


Figure 5.11: LiteStrat metamodel adaptation for its implementation in ADOxx.

The ADOxx meta^{2}model classes from which the LiteStrat constructs inherit are coloured in pale blue and stereotyped as *metametamodel* in Fig. Figure 5.11. As ADOxx *Relation* class only allows to defining a single 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 generalises 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 the actor and organisational unit constructs, and to allow them to relate using the relationship (*Influence*). Both abstractions are coloured in yellow and are considered abstract classes, as they have no graphical representations in the prototype.

The adaptation of the LiteStrat metamodel allows the simplification of 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. We implemented a "Validate Model" menu option using ADOxx scripting features to support IAG8 regarding checking the integrity constraints of the model. The integrity constraints validated by the tool are detailed in Table 5.8.

In Figure 5.12, we present a screenshot of the LiteStrat Supporting Tool prototype. This prototype is publicly available on the ADOxx Developer Community website (OMiLAB, 2023b).

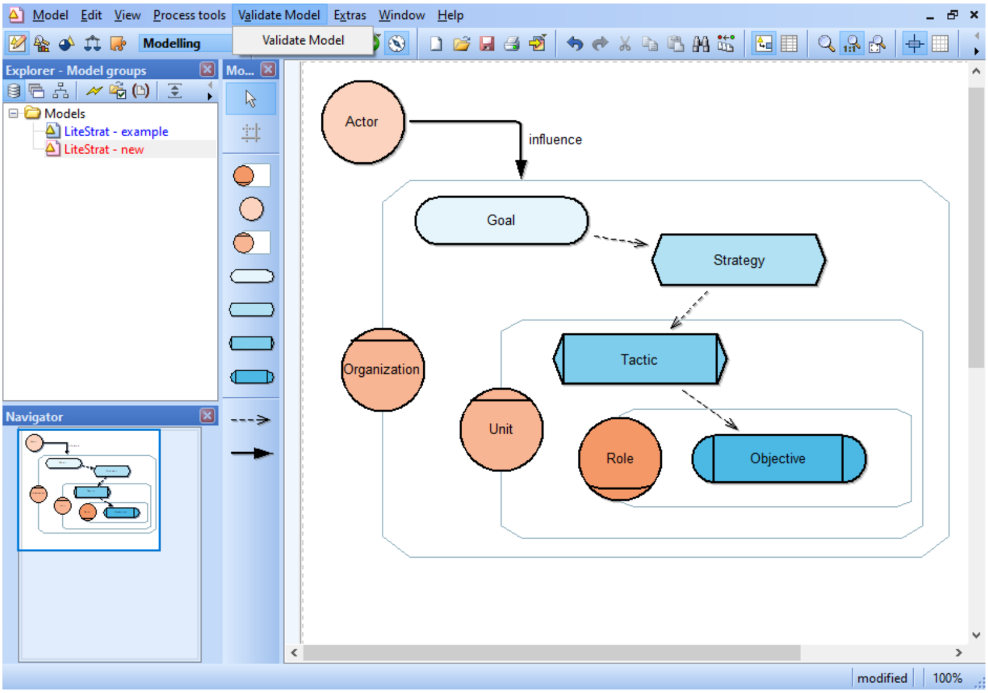


Figure 5.12: Screenshot from the LiteStrat supporting tool.

Table 5.8: LiteStrat's integrity constraints

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 associated goals.
IC3	At least one organisation unit must have one or more associated strategies.
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 another organisation unit.
IC6	Roles can not be inside an organisation unit that is not inside another organisation unit.
IC7	At least one organisation unit must have one or more roles inside.
IC8	At least one organisation unit must be inside another organisation unit.
IC9	The model must have at least one actor being influenced by an organisation unit.
IC10	Goals must not be refined from other elements and must be refined only by strategies.
IC11	Strategies must only be refinements of goals and be only refined by tactics.
IC12	Tactics must only be refinements of strategies and be only refined by objectives.
IC13	Objectives must only be refinements of tactics and can not be refined.
IC14	Strategies, tactics, and objectives must be refined for other elements.
IC15	Actors can not have elements inside of them nor be inside another element.
IC16	All Actors must be influenced by some organisational unit or influence an organisational unit.
IC17	Actors can not be influenced by other actors or influence other actors.
IC18	All roles must have at least one objective.
IC19	All objectives must be assigned to roles.
IC20	Roles can only contain objectives.

5.8 Summary

Chapter 5: Design of an organisational modelling method

- In this chapter, we presented the design of LiteStrat, addressing the design problem of how to model organisational information relevant for strategic alignment **(DP1)**.
- We discussed the organisational domain information relevant to the software organisation's strategic alignment approach as the social context for LiteStrat.
- We reviewed the knowledge context for the new method in terms of the existing modelling methods for organisational modelling: i*, as the current organisational modelling method of the existing model-driven development method; ArchiMate, as the most comprehensive enterprise modelling framework, and Business Motivation Model (BMM), which inspired the inclusion of business motivation and strategy elements into ArchiMate's version 3.0.
- From the social context, we extracted design goals for representing organisational domain information and specified the method requirements to fulfil such goals. We designed LiteStrat by selecting concepts of the method base and providing the guidelines for their assembly.
- To assess the feasibility of implementing the modelling method, we implemented a supporting tool using the ADOxx development environment. We adapted LiteStrat's metamodel to the conceptual architecture of the environment. The developed tool supports applying the modelling method and checking its integrity constraints.

Chapter 6

Design of a method for modelling strategically aligned business processes

6.1 Motivation

This chapter presents the design of a method for addressing the second improvement goal for the baseline MDD method identified in Chapter 4. This improvement goal regards using organisational model information to design modular business processes. To achieve this, we designed a method that follows up LiteStrat toward achieving alignment between business strategy and business processes.

Stra2Bis (a short name for *Strategy-to-Business*) is a situational business process modelling method that assembles two methods: LiteStrat, for modelling business strategy, as presented in the previous chapter, and Communication Analysis (España, González, and Pastor, 2009), the business process modelling method of the existing model-driven development method. As for LiteStrat's design, we adopt the Design Science (Wieringa, 2014) research methodology and Situational Method Engineering (Henderson-Sellers, Ralyté, et al., 2014) (SME) as the design method.

From a design science perspective, Stra2Bis and LiteStrat share the same motivation from the social context: to include organisational information on business strategy and organisational structure in the software development process as software organisations do. Regarding the knowledge context, Stra2Bis focuses on aligning business strategy and business processes, which has been broadly studied in model-driven engineering research. In the rest of the section, we elicit the design goals from the social context, review the knowledge context regarding model-driven initiatives for strategic alignment, and Communication Analysis characteristics that enable its integration as a method part for designing Stra2Bis.

This chapter presents the design of Stra2Bis, addressing the design problem of integrating organisational information into the baseline model-driven software production method **(DP2)**. In Section 6.2, we present the social and knowledge context for the method. Section 6.3 details the requirements for the method, while Section 6.4 describes the selection and assembly of method chunks from the method base. Section 6.5 shows an example of the application of Stra2Bis. In Section 6.6, we present the application of a focus group with software engineering professionals as an exploratory evaluation of the proposal. Section 6.7 describes the tool for supporting the method, and finally Section 6.8 summarises the chapter.

6.2 Social and knowledge context for the method

Strat2Bis has the same social context as LiteStrat, detailed in Section 5.1. From the social context, we propose four design goals, presented in Table 6.1. These goals are specified in the method's requirements in Section 6.3.

6.2.1 Knowledge context for the business process modelling method: strategic alignment

A particular concern in model-driven research is the alignment with high-level business definitions, namely strategic alignment. Model-driven initiatives for strategic alignment aim to address concerns which are specific to information systems development, such as assessing the effect on the organisational goals of supporting particular business process activities (R. Guizzardi and Reis, 2015), assessing whether business processes set requirements for collect-

Table 6.1: Design goals for the Stra2Bis method.

	Design Goal	**Description**
DG1	Business strategy modelling	Include business strategy elements that motivate the software development endeavour
DG2	Organisational structure mapping	Align the business process design with the organisational structure configuration that will enable the business strategy
DG3	Inform strategy performance	Include business process activities to report the status of the business strategy implementation.

ing data to verify the achievement of business goals (Kraiem et al., 2014), and prioritise software development activities for the incremental development of a software system (Insfrán et al., 2017). This focus on specific software development endeavours of model-driven approaches is different from the strategic alignment perspective of enterprise architecture frameworks such as Zachman's framework (Zachman, 1987), TOGAF (The Open Group, 2018), ARIS (Santos Jr, Almeida, and G. Guizzardi, 2010), and ArchiMate (The Open Group, 2022e), which aim to have a comprehensive view of the whole organisation's business and technology strategy. It is also different from business frameworks such as Balanced Scorecards, which provide tools to track and measure the organisation's performance towards the strategic goals, or Strategy Maps that help visualise the organisations' strategic objectives and the cause-and-effect relationships between them (Kaplan, 2009).

However, business knowledge is extensive, and different models must be used jointly to represent it, such as business value (Gordijn and Akkermans, 2003), processes (Rosing et al., 2015; España, González, and Pastor, 2009; Buhr and Casselman, 1995), goals (Yu, 2011a; Dardenne, Van Lamsweerde, and Fickas, 1993; Bresciani et al., 2004), and strategy (Rolland, 2007) models, among many others. Having multiple models to represent business knowledge sets the challenge of making them consistent with each other. In an MDA context, consistency among CIM-level models is critical since missing or inconsistent business information would produce information system models not aligned with the business needs. Metamodel mapping techniques help to design consistent models by defining model-to-model checking or transformation rules. Particularly, strategic alignment initiatives aim to align goal and business process models to ensure the organisation's operation is consistent with the business goals. (Gröner et al., 2014; R. Guizzardi and Reis, 2015; Sousa and Prado Leite, 2014).

We set a knowledge context similar to the strategic alignment initiatives that traditionally have employed goal models to align business processes in two main concerns: 1. They focus on alignment requirements for a specific software development endeavour rather than aligning the whole organisation's systems as in enterprise architecture approaches, and 2. Rely on model-driven techniques such as metamodel mappings to provide alignment.

Goal modelling languages have been used, for instance, to analyse whether business process activities (modelled using BPMN) support organisational goals (modelled with TROPOS) (R. Guizzardi and Reis, 2015), or to study how business processes constraint business goals (modelled using KAOS)(Nagel, Gerth, Engels, et al., 2013). The Goal-Oriented Requirements Language (GRL) has been combined with Use Case Maps to model strategically aligned processes in the last two decades (Amyot et al., 2022) and also to prioritise business processes (Insfrán et al., 2017). MAP models (that define goals and the strategies to achieve them) have been mapped directly to the business processes elements that operationalise them (Kraiem et al., 2014) and also served to analyse the purpose behind the creation, modification, and deletion of business process elements (Vara, Sánchez, and Pastor, 2008). I* models have been used for transforming social dependencies into interactions at the process level (Ruiz, Costal, et al., 2015), validating the consistency of the process interactions (Gröner et al., 2014), and checking whether the business processes have the elements needed to collect information to verify the goal achievement (Sousa and Prado Leite, 2014).

Besides goal modelling, other initiatives have combined frameworks addressing business strategy concerns. Business plans (modelled in Business Motivation Model (The Object Management Group, 2015)) have been used jointly with i* to add intentionality to the process of enterprise architecture construction (Yu, Strohmaier, and Deng, 2006). Business value models (modelled using the e3Value method) have been used for generating performance requirements for an enterprise architecture (Engelsman, Gordijn, et al., 2021). In (Bērziša, Bravos, T. Gonzalez, et al., 2015), organisational capabilities, modelled at the enterprise architecture level, are the starting point for the model-driven development of context-adapting software systems.

The above initiatives show that integrating modelling methods is a powerful tool for strategic alignment. These strategic alignment initiatives help business analysts elicit requirements for a software development endeavour when having multiple stakeholders with competing goals in complex organisations. However, as presented in the social context in Section 1.3, the alignment problem is focused on something other than stakeholders and their goals. In software organisations, the complexity is handled through reconfiguring the organisation structure (Karim and Capron, 2016) into small independent organisation units designed around business strategy. The above methods have arguably yet to address this issue, as studied in Chapter 2, which is the gap covered by Stra2Bis.

6.2.2 Communication analysis as a method part

Communication Analysis (CA) was initially introduced in (España, González, and Pastor, 2009). Later, it was integrated with the OO-Method in (España, 2011). In the later work, CA's platform independent metamodel was introduced. The metamodel supports CA's communicative events diagram constructs, the specification of message structures, and the specification of communicative events. Moreover, it introduced some organisational modelling concepts that put business processes in the organisational context. Figure 6.1 presents an extract from CA's metamodel, including organisational concepts (in pale green) and the concepts related to communicative event diagrams (in white).

From an SME perspective, the overlapping between the organisational modelling constructs of CA's metamodel and LiteStrat's metamodel presents an integration opportunity (Henderson-Sellers, Ralyté, et al., 2014). In Table 6.2, we present the analysis of the overlapping concepts between CA and LiteStrat and discuss their potential for methodological integration.

6.3 Method requirements

Stra2Bis' requirements are inferred from the need to enable software organisations' approach to software design in MDA-based software development methods. In Figure 6.2 presents the requirements map for the method.

The first intention is to *model business strategy*, which is achieved through *modelling the strategic scenario* that drives the software development endeavour. Stra2Bis considers LiteStrat as the method part for achieving this intention. This intention addresses the design goal DG1.

Once the strategic scenario is modelled, the second intention is to *model business processes*. This intention is achieved by a set of *alignment-driven transformations* to generate an initial structure (namely scaffold) of the business process model. The transformations aim to mirror the organisational struc-

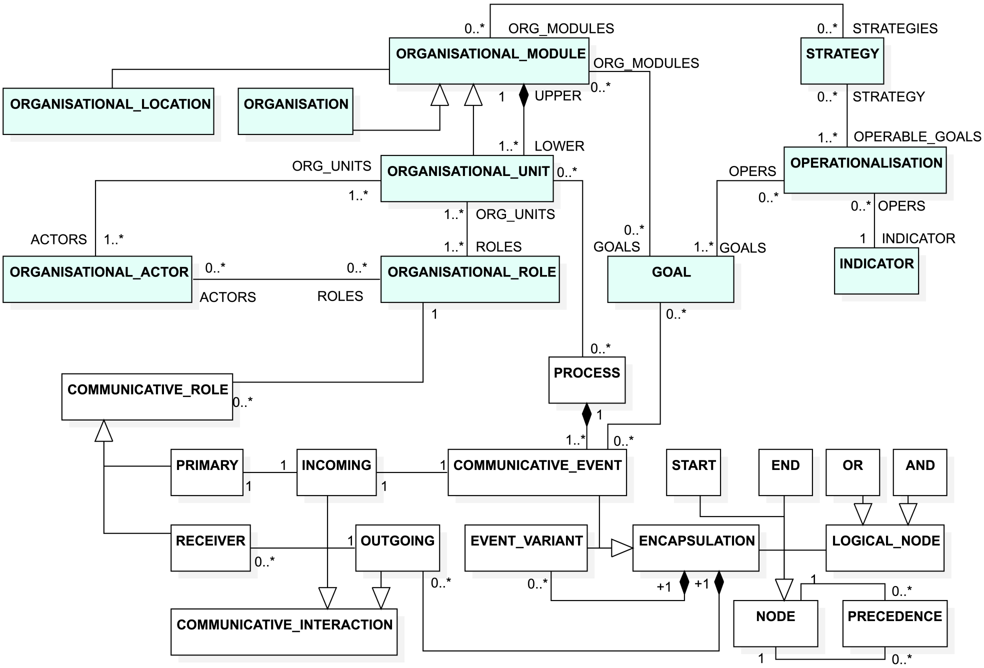


Figure 6.1: Organisational and communicative event diagram concepts from the Communication Analysis metamodel (from (España, 2011)).

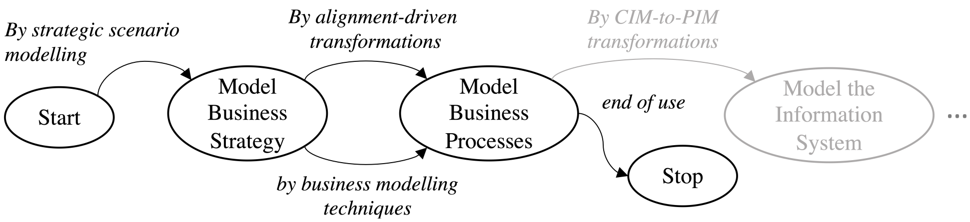


Figure 6.2: Stra2Bis requirements map.

ture and strategic objectives of the business strategy model to design business processes. The transformations address design goals DG2 and DG3.

The scaffold is completed through existing *business process modelling techniques*. In the context of the existing model-driven development method, Stra2Bis assembles LiteStrat with Communication Analysis to exploit the capabilities of CA for generating the conceptual schema of the information system

Table 6.2: Analysis of overlapping concepts between Communication Analysis (CA) and LiteStrat (LS).

Overlapping Concepts	Integration Analysis
CA.ORGANISATION_UNIT, LS.ORGANISATION_UNIT	The concepts are the same; however, the relationships with other constructs reveal CA has the CA.ORGANISATION concept for defining the organisation under analysis and the CA.ORGANISATION_MODULE generalisation for both the organisation and its organisational units. On the other hand, LiteStrat manages the organisation and its units through the only concept of LS.ORGANISATON_UNIT. Besides the differences, CA.ORGANISATION_UNIT is associated with CA.PROCESS construct, which gives an integration point for LS.ORGANISATION_UNIT and CA.PROCESS.
CA.ORGANISATIONAL_ROLE, LS.ROLE	The concepts do not have the same name but might be equivalent since LiteStrat does not use the *organisational* prefix for any of its concepts. CA.ORGANISATION_ROLE is associated with CA.COMMUNICATIVE_ROLE construct, which gives an integration point for LS.ROLE and CA.COMMUNICATIVE_ROLE.
CA.GOAL, LS.GOAL	The name of the concepts is the same. CA.GOAL is connected with CA.COMMUNICATIVE_EVENT so it might provide an integration point for LS.GOAL. However, it is not clear that CA.GOAL is closer to the operation than CA.OPERATIONALISATION, so we think that a proper integration point could be LS.OBJECTIVE and CA.COMMUNICATIVE_EVENT, since LiteStrat's objectives are closer to the operation than goals.
CA.STRATEGY, LS.STRATEGY	The name of the concepts is the same. However, the relationship of CA.STRATEGY with CA.OPERATIONALISATION with the role *OPERABLE_GOALS* suggest it might be a more concrete strategic action, similar to LS.TACTIC, though it is not clear. Nevertheless, CA.STRATEGY is not connected to any business process constructs, so it does not provide a sound integration point.

(España, 2011), as presented in Chapter 1. However, Stra2Bis' scope is limited to generating the scaffold of business process models.

6.4 Selection and assembly of method chunks

Unlike LiteStrat, which assembled concepts from existing modelling methods, Stra2Bis assembles two full modelling methods: Litestrat and Communication Analysis. Next, we present the rationale behind the assemble design and detail the guidelines for assembling them from the product and process perspective.

6.4.1 Method chunks selection

The requirements presented in Figure 6.2 are addressed by assembling existing methods. As presented in Chapter 5, the intention *Model Business Strategy* is supported by LiteStrat. The intention *Model Business Processes* is supported in the existing model-driven-method by Communication Analysis, as presented in Chapter 1.

From an SME perspective, the assembly of LiteStrat and Communication Analysis follows an *assembly-based approach by association strategy* (Henderson-Sellers, Ralyté, et al., 2014) since a new method is created by positioning the existing method chunks and providing guidelines for their association. Stra2Bis does not define new product elements for the existing method parts but integration guidelines. Next, we detail Stra2Bis' process perspective, which details the guidelines, and then the product perspective, detailing the relationships between the metamodels of LiteStrat and Communication Analysis.

6.4.2 Stra2Bis' process perspective

Stra2Bis' process perspective addresses the guidelines to assemble the methods, particularly the set of alignment-driven transformation guidelines to go from Model Business Strategy to Model Business Processes. The Intention Achievement Guidelines (IAG) for the whole method are detailed in Table 6.3; however, motivation, definition and effects of the transformation guidelines are specified in the Definitions 6.1, 6.2, and 6.3,

Definition 6.1: Guideline 1 - Organisation units' independence

Design a single business process for each organisation unit.

Motivation: This guideline is based on the research by Forsgren et al. (Forsgren, Humbpotifle, and Kim, 2018), who found that the coupling between teams has been reported as a hindering factor for efficient software development. The problem addressed is that teams with multiple business processes or business processes addressed by multiple teams increase the need for communication and collaboration between teams, and, in the same way, the software design replicates the coupling. The solution proposed is to design processes that are as independent as possible for each team.

Transformation description: For each organisation unit belonging to the overall organisation in the business strategy model, create a new process in the business process model. Add a start event with the unit's name to the new process to make the process visible in the model.

Effects on the business process model: Modelling a strategic scenario helps the analysts to reflect on designing separate processes for orders and delivery management. Failing to do this will traduce assigning the business processes and supporting software components to an organisational unit (i.e., development team) that will not be autonomous to manage their requirements at the process level and thus to design and evolve the information system. The generated elements in the business process model reflect the ideal separation of processes. The analyst should assess whether this separation is feasible considering the actual context of the problem.

Definition 6.2: Guideline 2 - Managed strategic dependencies

Design the interactions between business processes to manage the organisation units' strategic dependencies.

Motivation: This guideline is based on the need for managing and reducing the dependencies among development teams to foster their autonomy, which is a practice followed by operational models such as the Spotify Model (Atlassian, 2021) and also contributes to the design of autonomous teams (Forsgren, Humbpotifle, and Kim, 2018; Highsmith, Luu, and Robinson, 2019). Another motivation is the Domain-Driven Design approach (E. Evans and E. J. Evans, 2004), which states that the integration between different business contexts must be carefully designed at the information system level. The problem addressed is that new strategic scenarios could introduce new dependencies among units, which, if overlooked, could hinder the efficiency of the software delivery. The solution approach is to ensure that these dependencies are considered for designing business processes.

Transformation description: For each influence dependency between organisation units in the business strategy model, add events to the source and target organisation units' processes to handle the dependency. In the source unit's process, add an event to provide the information to satisfy the dependency and a receiver actor representing the target organisation units' process. Similarly, add an event and a primary actor to the target unit's process to receive information about the dependency from an actor representing the source organisation unit.

Effects on the business process: The strategic scenario helps the analyst design the interface between the orders management area and the delivery cell based on strategic criteria. Since the delivery cell is affected by the requests of the order management area, the cell must provide a well-defined way to manage these requests at the process level, and the order management area must also consider this mechanism in its process. Failing to do this could result in designing ad hoc interoperability mechanisms at the process and system levels. The guideline assumes that the information needed for the process interaction is already known; otherwise, the analyst can add a primary actor to provide the required information.

Definition 6.3: Guideline 3 - Strategic objectives measurement:

Design business process elements to collect data to measure strategic objectives.

Motivation: This guideline is based on the practice of a shared measurement of the success of strategic initiatives, which is enforced by frameworks for digital transformation such as EDGE (Highsmith, Luu, and Robinson, 2019) and Objectives and Key Results (OKR) (Doerr, 2018). The problem addressed is to consider in advance requirements to measure and share the status of strategic objectives in order to enable the assessment and continuous adjustment of the business strategy. The solution approach ensures that the strategic objectives are considered in business process design.

Transformation description: For each business strategy objective, add an event to their respective organisation unit's process to collect information about the objective's status. Add a receiver actor following the name of the objective's role.

Effects of the business process: Mapping the strategic scenario helps the analyst consider specifying requirements to measure consumer growth and satisfaction with the delivery service. Failing to consider these requirements may require adding them later on demand of top executives, which may harm the system design and performance. Similarly to guideline 2, the transformation does not generate a primary actor to provide the information. It will not be needed if the information is already in the system; otherwise, the analyst can add a primary actor according to the problem domain.

6.4.3 Stra2Bis' product perspective

In Figure 6.3, we detail how the guidelines associate the metamodels of LiteStrat and Communication Analysis (CA). The relationships for Guidelines 1, 2, and 3 are coloured in green, orange, and yellow, respectively.

Regarding Guideline 1, represented in green in Figure 6.3, LiteStrat's *organisation unit* concept is associated with the CA's *process* abstract concept, with the semantic intention of asserting that an organisational unit owns one business process. LiteStrat's *organisation unit* is also associated with CA's *Start* in a 1-to-1 relationship, meaning the production of such modelling element, as recommended by the guideline.

The associations introduced by Guideline 2, depicted in orange in Figure 6.3, connect LiteStrat's *organisation unit* as *primary* and *receiver* communicative roles in CA. This means that organisation units which depend on each other should engage in a communicative interaction to operationalise such dependency. The dependency itself, a LiteStrat's *influence*, is connected through a *communicative event* in CA, which also implies defining *incoming* and *outgoing* communicative interactions.

Finally, Guideline 3 introduces the associations depicted in yellow in Figure 6.3. In this case, LiteStrat's *role*, who has assigned an *objective*, is associated with CA's *receiver* communicative role since it is expected to be informed from the status of such objective. LiteStrat's *objective* is associated with a CA's *communicative event* to report the objective status and with a CA's *outgoing* communicative interaction to connect the event with the previously generated *receiver* role.

Please note that none of the guidelines specifies are no connections with CA's *message structures*; however, it is expected that the method user specifies message structures for the generated communicative interactions.

Table 6.3: Intention achievement guidelines for the LiteStrat method.

ID	**Section**	**Guideline description**	**Related Products**
IAG1	<Start, Model business strategy, *strategic scenario modelling*>	This guideline proposes to use LiteStrat to model business strategy, which procedure follows the strategic scenario modelling approach.	business strategy model
IAG2	< Model business strategy, Model business processes, *alignment-driven transformations*>	this guideline proposes to perform three model-to-model transformations to generate the scaffold of the business process model. The input for the transformation is a business strategy model. The transformations are specified in the Definition 6.1, Definition 6.2, and Definition 6.3.	Business Process Model (scaffold)
IAG3	<Model business strategy, Model business processes, *business process modelling techniques*>	This guideline proposes to complete the business process modelling scaffold by using existing business process modelling techniques, in particular, following the Communication Analysis method. The modeller must consider the strategic scenario represented in the business strategy model for redesigning the business processes.	Business process model
IAG4	<Model business processes, Model the information system, *by systematic derivation*>	This guideline proposes to continue with the existing model-driven developing method by deriving the information system model from the Communication Analysis Model. The derivation must be performed following the guidelines by España in (España, 2011).	Information system model
IAG5	<Model business processes, Stop, *end of use*>	This guideline proposes to stop using the method after business process modelling, so other manual or automatic approaches can take as input the strategically aligned business process model.	-

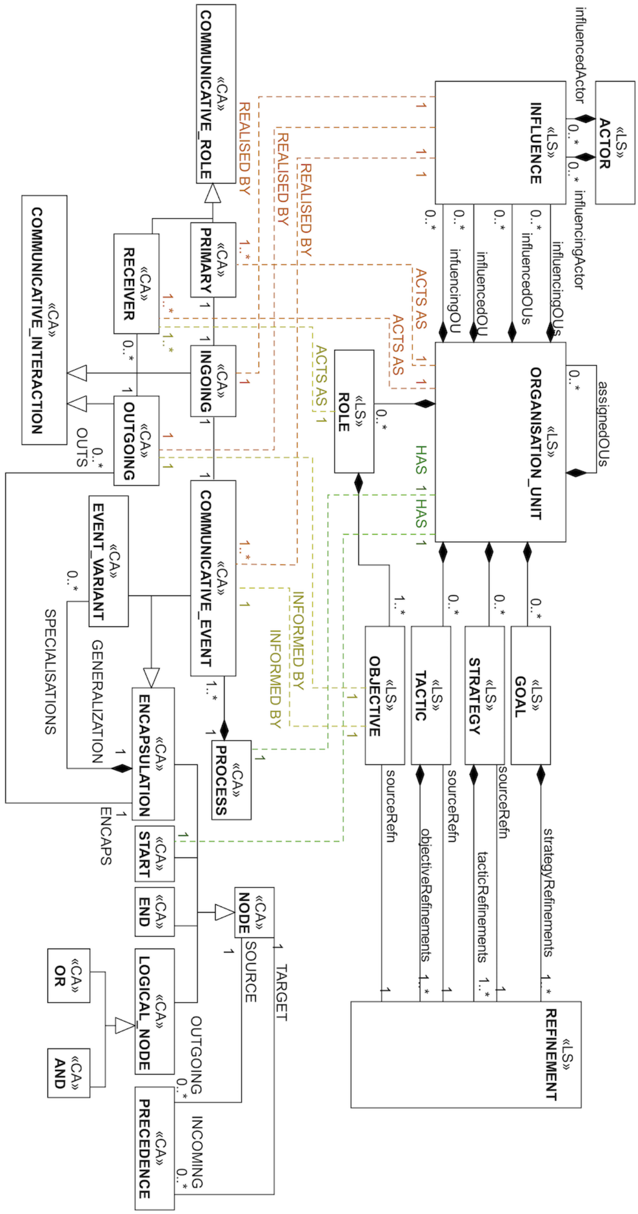


Figure 6.3: Metamodel mappings for LiteStrat (LS stereotype) (Noel, Panach, and Pastor, 2021b) and a simplified Communication Analysis (CA stereotype) (España, 2011) metamodels.

6.5 Application example

To describe the application of Stra2Bis, we first present the current situation of the organisation processes and systems, then the strategic scenario driving the software development endeavour, and then the alignment-driven transformations for generating the business process model scaffold. We also discuss the implications for the design of the information system.

6.5.1 *Current situation*

The current situation is described in the Example 6.1, and depicted in 6.4.

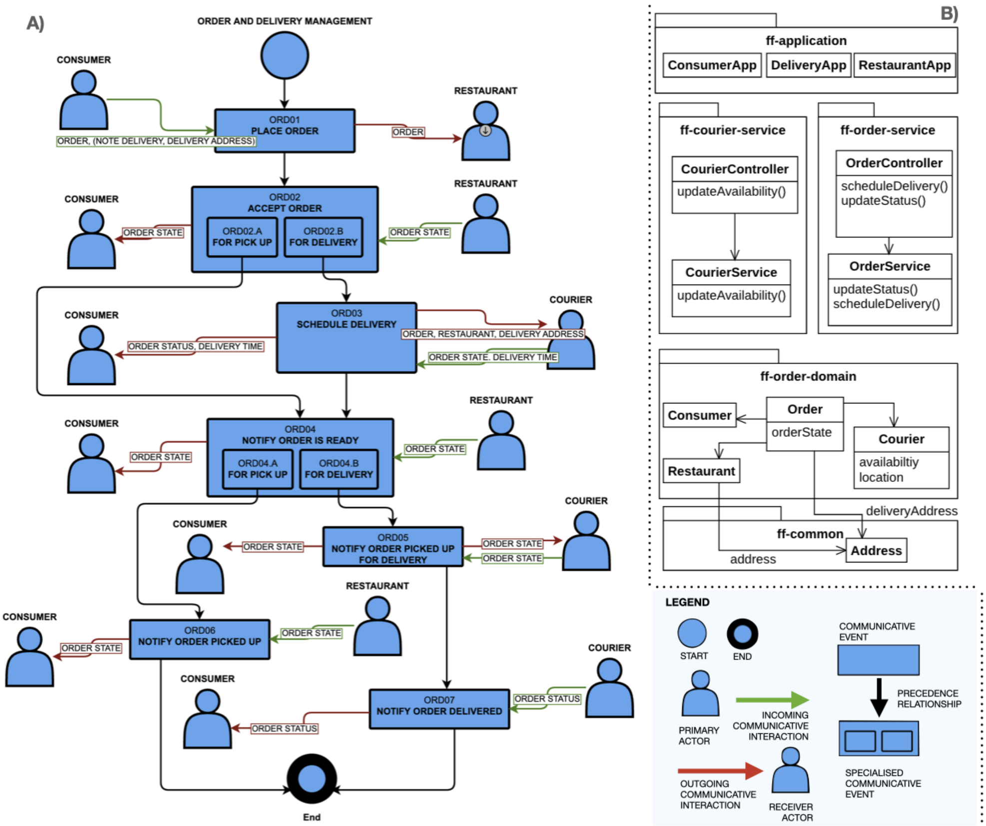


Figure 6.4: Current situation models: A) Business process model. B) Class diagram of the Information System.

Example 6.1: Current situation for F-FOOD

F-FOOD is a software-as-a-service company that allows ***consumers*** *to order food from* ***restaurants****, for pickup or for delivery. After the restaurant confirms an order, the delivery orders are scheduled to the closest available* ***courier****. F-FOOD has grown exponentially since its foundation, and most of its software development efforts have focused on mobile applications. However, the back end is still a monolithic application. Figure 6.4.A depicts the current business process model, while Figure 6.4.B shows the components of the information system.*

6.5.2 Model business strategy

The strategic scenario is described in Example 6.2. We numbered the statements in the strategic scenario according to their representation in the LiteStrat model, depicted in Figure 6.5.

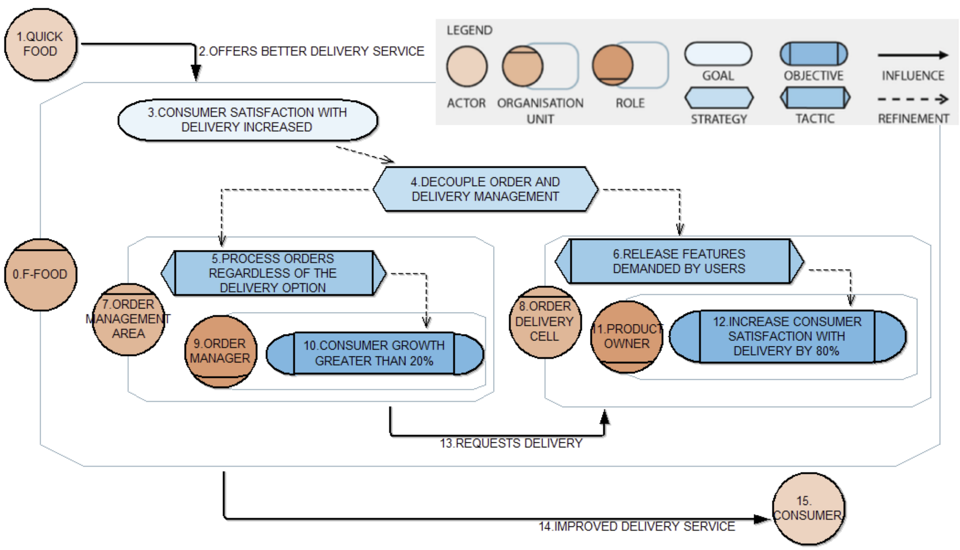


Figure 6.5: Business strategy model depicting the strategic scenario.

Example 6.2: Strategic scenario for F-FOOD

In the last quarter, the growth of consumers in F-FOOD (0) has decreased. F-FOOD discovers that a new competitor, QUICKFOOD (1), has a better order delivery service (2). Consumers claim that the F-FOOD app lacks several features for delivery tracking and has a slow response when placing delivery orders. F-FOOD discovers that the Order Management Area (7) constantly gives a lower priority to new delivery features and optimisations, favouring the order management functionality. F-FOOD management has decided that consumer satisfaction with the delivery is the top strategic goal for the next quarter (3). To achieve this goal, the strategy is to decouple the delivery service as an independent service (4) owned by a new cross-disciplinary team called Order Delivery Cell (8) that is meant to release all the features demanded by the customers (6). The Product Owner (11) will track the objective of increasing consumer satisfaction with delivery by 80% (12). The Order Management Area will have a leaner order processing, regardless of their delivery option (5) and will depend on the Order Delivery Cell to deliver the orders (13). New consumers are expected to increase by 20% (10), which will be tracked by the Order Manager (9). The implementation of the strategy seeks to offer an improved delivery service (14) for consumers (15).

6.5.3 Alignment-driven transformations

Next, we detail the application of each of the three guidelines, taking as input the Litestrat model in Figure 6.5 in order to produce the business process models depicted in Figure 6.6.

Guideline 1 - Organisation units' independence:

In the business strategy model in Fig. Figure 6.5, *Order Management Area* and the new *Order Delivery Cell* units originate the *Order Management* and *Delivery Management* processes depicted as green start nodes in Figure 6.6. The start nodes are named following the names of their respective organisational units. The guideline proposes designing an independent business process for the delivery service; otherwise, the new team would still be coupled to the Order Management Area process. Although the example specifically regards the

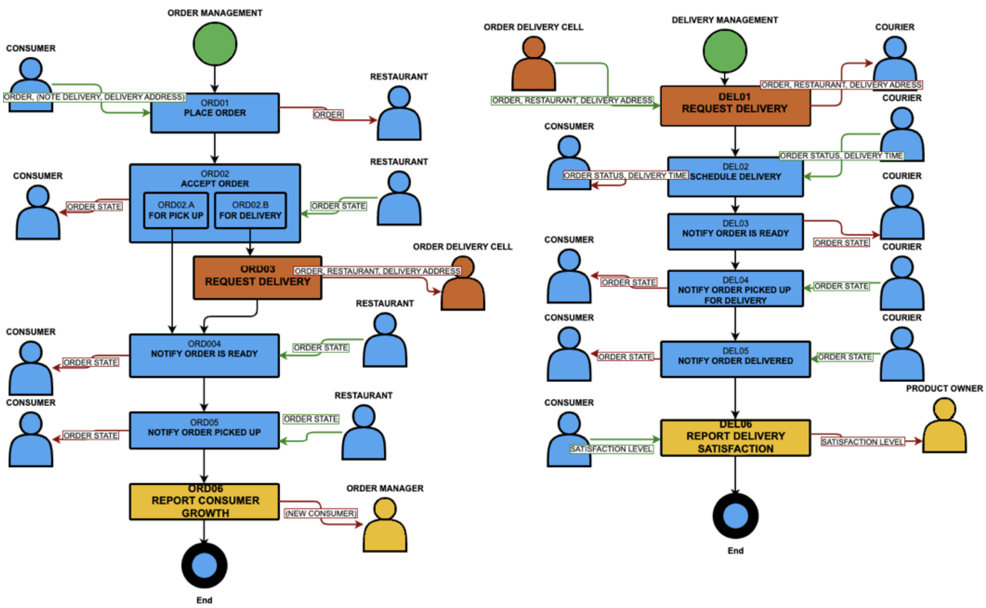


Figure 6.6: Business process model resulting from applying the alignment-driven transformation.

split of an existing unit, the guideline is also helpful in analysing the creation, fusion, or hiring of external teams for tackling new business opportunities.

Guideline 2 - Managed strategic dependencies:

The influence relationship *16.Requests Delivery* from the organisation unit *Order Management Area* to the *Order Delivery Cell* in Figure 6.5 is mapped as the events depicted in orange in Figure 6.6: an event to perform the influencing behaviour (16.Requests Delivery), and an event to address the influence (DEL01-Handle Delivery Request). A new actor is introduced to handle the dependency, representing the target organisation unit of the dependency (Order Delivery Cell). The names of the events and actors follow the strategy diagram, but the analyst can change them according to the domain information.

Guideline 3 - Strategic objectives measurement:

In the strategy diagram in Figure 6.5, the objectives *10.Consumer growth greater than 20%* of the organisation unit *Order Management Area* is mapped to the event *ORD06.Report Consumer Growth* in Figure 6.6, depicted in yellow. Similarly, the objective *12.Increase consumer satisfaction with delivery by 80%* is mapped to the *DEL06-Report Delivery Satisfaction* event. In both cases, the receiver actors are the roles assigned to the objectives in the strategy diagram (Order Manager and Product Owner).

6.5.4 Effects on the PIM level in an MDA Context.

The guidelines are expected to affect the information system model at the PIM level. Although the integration of the business process and the information system models is not part of this work (but has already been proposed in (España, 2011)), we exemplify in Figure 6.7 the effects of the guidelines on the initial information system model presented in Figure 6.4.B.

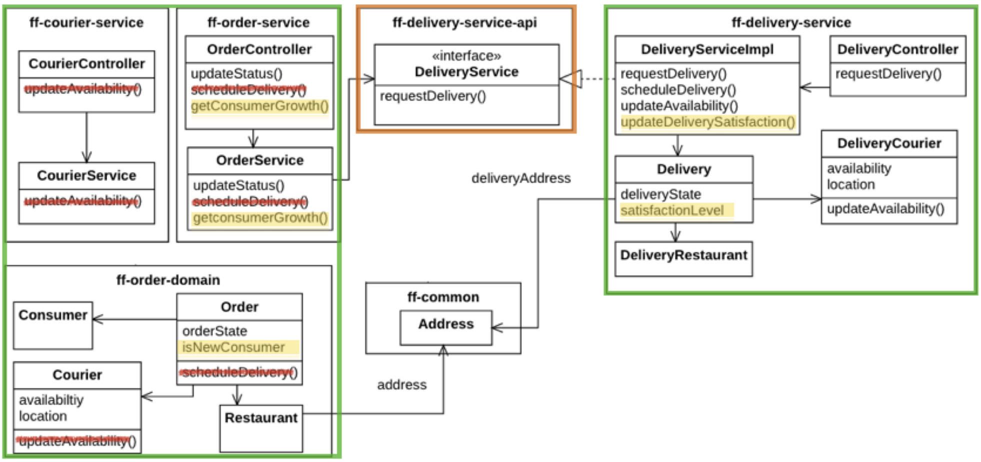


Figure 6.7: Re-designed class diagram for the information system model.

Regarding Guideline 1, since the two organisation units *Order Management Area* and *Order Delivery Cell* had their separated business processes *Order Management* and *Order Delivery Management*, the `Delivery` domain class and services must be disentangled in a different component. Figure 6.7 shows the components for both processes in green. The new component `ff-deliver-service` supports the Order Delivery Process. Some services are removed from the initial order management components (see Figure 6.4.B). The changes mainly consist of removing the delivery-related services that were initially located in the `ff-courier-service`, `ff-order-service` and `ff-order-domain` components and moving them to the new `ff-deliver-service` component.

Regarding Guideline 2, the interaction between the processes is mapped as an interface `ff-deliver-service-api` depicted in orange in Figure 6.7. The interface is implemented by the component supporting the delivery process `ff-delivery-service`. It allows the initial order management system to request the services that were moved to the new `ff-delivery-service`.

Finally, the effects of Guideline 3 are mapped into services and attributes to update the values for the strategic objectives collected through the process. As highlighted in yellow in Figure 6.7, the Order class has a new attribute `isNewConsumer` to identify whether the order is from a new consumer. This helps track the objective *10.Consumer growth greater than 20%* objective initially defined in the strategy model in Figure 6.5. Similarly, the `Delivery` class has the attribute `satisfactionLevel` of the objective *12.Increase consumer satisfaction with delivery by 80%.*

6.6 Exploratory evaluation

We conducted an exploratory evaluation through a focus group since this technique is suitable for the *"initial evaluation of potential solutions, based on the practitioner or user feedback"* (Kontio, Lehtola, and Bragge, 2004). The focus group was carried out after the design of LiteStrat and before the formal definition and implementation of the Stra2Bis guidelines to validate their relevance from the practitioners' point of view.

The research question was, *what information from the business strategy model is valuable for designing business processes?*. The goal is to find whether practitioners' insights and experience match the Stra2Bis guidelines in terms of the information traceable from business strategy to business process and to the information system model. We wanted to contrast opinions from practitioners working in traditional consulting service companies (CSC) and in Software-as-a-Service companies (SaaS), whose main value offer is based on software. The participants were five volunteers with a technical leader or scrum master role, with between four and nine years of experience. Participants S1 and S3 work in CSC, and participants S2, S4, and S5 work in SaaS company.

The activity had two parts of 30 minutes each. In the first part, we presented the working example described in Example 6.1. Then, the current business process and system structure were presented through a CA communicative event diagram and a UML class diagram, depicted in Figure 6.4.A and Figure 6.4.B respectively.

From Figure 6.4 and asked, *what information would be useful for redesigning business processes and why?*. The participants shared and agreed on a set of statements that the moderator publicly wrote down. In the second part, we presented the Stra2Bis guidelines and the models from Figure 6.6 and asked the participants to comment on their usefulness and drawbacks. The analysis method was based on pattern-matching (Kontio, Lehtola, and Bragge, 2004) the participant's ideas from the first part of the focus group with the guidelines and then looking for explanations in the discussion of the second part.

Insights for Guideline 1: In the first part, the respondents did not identify the organisation units as an important source of information for the business process design. After seeing the redesigned process and the guideline 1, all the participants agreed that independent units must have independent processes. All respondents recalled difficulties when business processes and software code of different units were entangled. Respondent S2, from a SaaS, stated that *"it is important for us to have an independent business flow because each cell can take the challenges and opportunities of their own process"*.

Insights for Guideline 2: In the first part, all the respondents identified as relevant the dependency among the organisation units. S1 and S2 agreed that *"the dependency must be clear in the business process flow"*. All the participants agreed on the value of the guideline for defining the dependency at the process level. It is worth noting that respondents S1 and S3, from CSCs, claimed that sometimes the flow interactions were not well defined by *business people*, requiring *"several meetings between teams to define the flow"* (S1). On the other hand, S2, from a SaaS, declared that her unit was *"designed with a well-defined contract with other organisation units"* and never had this kind of problem.

Insights for Guideline 3: In the first part, just S1 identified as valuable the objectives and linked them with OKR, one of the frameworks on which the guideline is based on (Doerr, 2018). In the second part, all the respondents valued measuring strategic objectives in the business process. Participants S4 and S5 commented *"we have code written to measure the NPS"*[1]. However, for the rest of the participants, the effect on the software product was different to what we presented in Section 6.5.4. The participants stated that the measure-

[1] Net Promoter Score, https://hbr.org/2003/12/the-one-number-you-need-to-grow

ment for checking objectives is solved using external tools such as Hotjar[2] (for measuring customer satisfaction) or Google Analytics.

6.7 Tool support

Similarly to LiteStrat, Stra2bis' implementation is supported by ADOxx. We modelled CA's communicative event diagram concepts using ADOxx meta²model (Karagiannis et al., 2016), as depicted in Figure 6.8.

As shown in the metamodel, this is a simplified version of CA that does not include specialised communicative events but an exclusive gateway to represent specific preceded by the same event. The attributes shown for some metamodel classes support the traceability links to LiteStrat constructs. The data type of these attributes is *INTERREF* or *Inter-Model Reference*, which ADOxx provides to connect elements between different models. We used inter-model reference to trace the CA modelling elements to the LiteStrat modelling elements that generated them.

The inter-model reference attributes are defined as follows. For Guideline 1, the *CAStart.Objective* attribute traces the start node of a process with the organisational unit that owns such process. For Guideline 2, *CAActor. OrganisationUnitInDependency* traces the Actor with the organisation unit which generated it. For Guideline 3, *CAActor.ReportingRole* and *CAActor. InformedOrganisationUnit* trace the role that reports an objective's status and the organisation unit informed of such objective, while *CACommunicativeEvent. Objective* traces the strategic objective for which the event reports its status.

Similarly, we updated LiteStrat's metamodel to provide traceability, as depicted in Figure 6.9. Similarly, the inter-reference metamodel attributes provide traceability from the LiteStrat source elements to the elements generated in the CA diagrams.

The inter-reference model attributes support the transformation guidelines as follows. Guideline 1 is supported by *OrganisatinalUnit. BusinessProcesses*, which connects the Organisational Unit to its owned business processes. Guideline 2 is supported by *OrganisatinalUnit.InDependencyEvents* and *OrganisatinalUnit.OutDependencyEvents* that connect the organisational unit with the communicative events on which they interact with other organisational units which it depends on or depends on the organisational unit, respectively. Fi-

[2]https://www.hotjar.com/

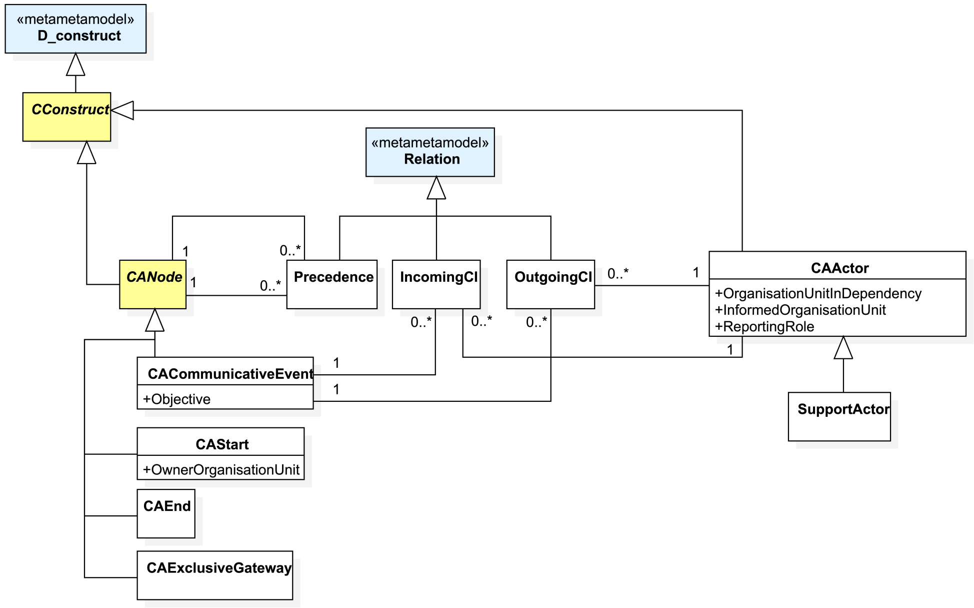


Figure 6.8: Metamodel for Communication Analysis' communicative event diagrams using the ADOxx meta^{2}model

nally, Guideline 3 is supported by *Objective. CommunicativeEvent*, to trace the objective to the communicative event which reports its status.

We implemented the three transformation guidelines using ADOxx script[3]. The implementation takes as input a LiteStrat diagram and generates the scaffold of a business process modelled in Communicative Analysis, linking the generated elements through the inter-reference model attributes. The implementation of the guidelines is detailed on Appendix A.2. In Figure 6.10 and Figure 6.11, we depict an example of the automatic transformation where the LiteStrat model in Figure 6.10 generates, among other constructs, the communicative interaction depicted in Figure 6.11. Following Guideline 3, an objective assigned to a role in LiteStrat is transformed into a communicative event where the role reports the status to its organisation unit. The specification of the modelling element shown in Figure 6.11 depicts a navigable inter-reference model link to the LiteStrat source element, in this case, an organisation unit.

[3]https://www.adoxx.org/live/scripting-language-adoscript

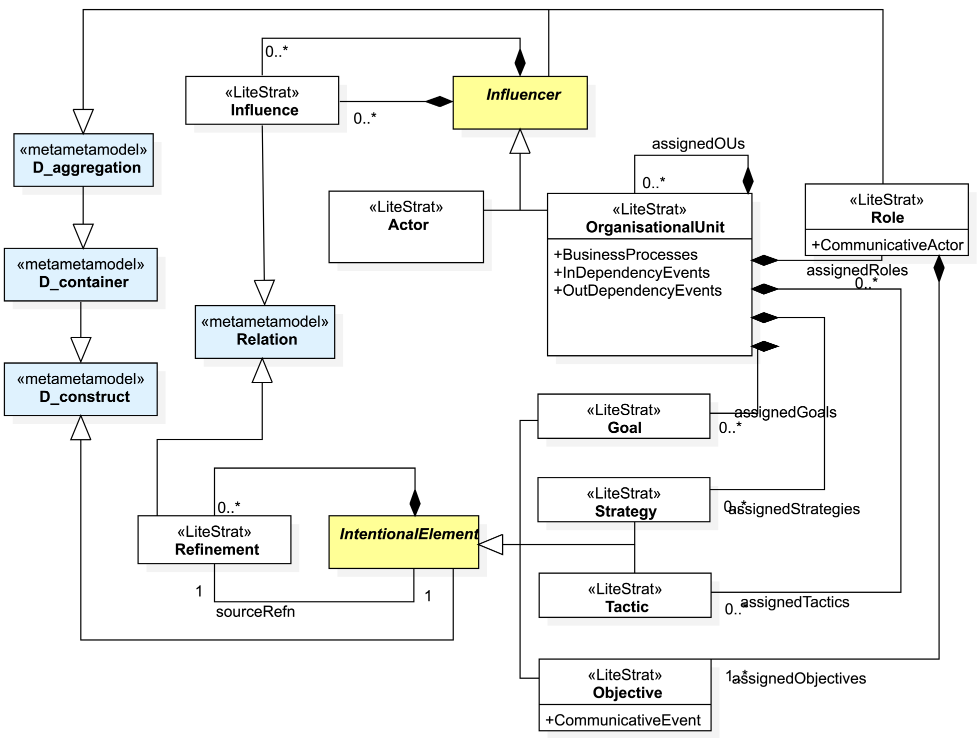


Figure 6.9: Metamodel update for LiteStrat including inter-reference model attributes for traceability

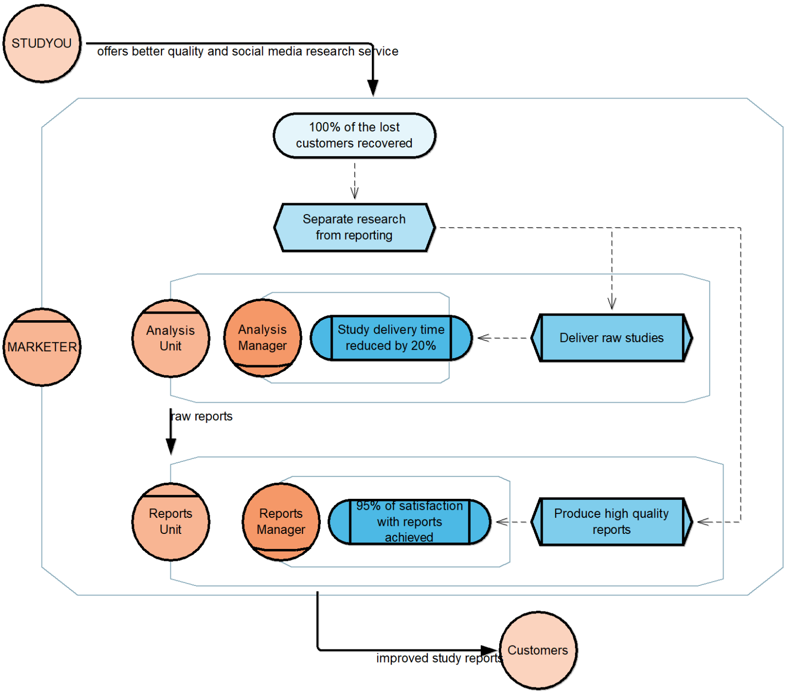


Figure 6.10: LiteStrat example model implemented in ADOxx

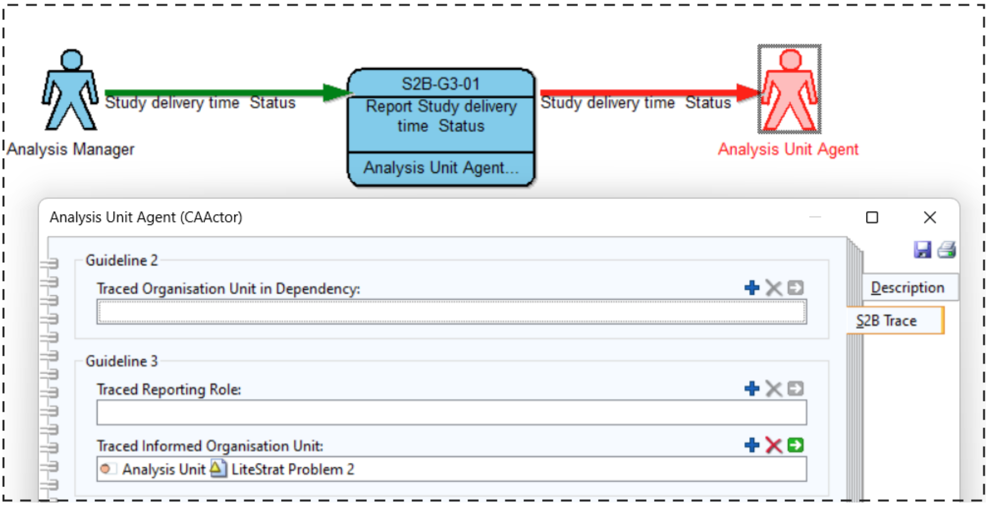


Figure 6.11: Detail from the generated Communication Analysis model, showing the specification of a communication analysis actor traced to a LiteStrat's organisation unit.

6.8 Summary

Chapter 6: Design of Stra2Bis

- In this chapter, we presented the design of Stra2Bis, a method for modelling strategically aligned business process models, addressing the design problem of how to integrate organisational information into the baseline model-driven software production method **(DP2)**
- Stra2Bis requirements are inferred from the need to enable the software-centric organisations' approach to strategic alignment based on business structure and well-defined strategic objectives.
- Following the situational method engineering approach, we studied the overlapping concepts of LiteStrat and Communication Analysis metamodels and proposed an integration by association.
- Three transformation guidelines were designed to generate an initial structure of a business process model (or scaffold) from the business strategy model based on the requirements for enabling strategic alignment in software-centric organisations.
- We implemented a supporting tool using the ADOxx development environment, which supports modelling with LiteStrat and the automatic generation of the business process model scaffold in Communication Analysis, providing bidirectional traceability between the source and target models.

Part III

Treatment Validation

Chapter 7

Validation of LiteStrat

7.1 Motivation

As presented in Chapter 5, Litestrat was designed as an alternative to i* for the baseline model-driven development method to integrate organisational information relevant to strategic alignment. In order to support the improvement claims that guided treatment designed in Chapter 5, scientific evidence must be produced. According to Design Science (Wieringa, 2014), an *empirical cycle* allows answering the scientific knowledge questions about the effects of the proposed artefacts when applied to their problem context.

The validation activities presented in this chapter address the research question *KQ3: Does LiteStrat improve the representation of information relevant for strategic alignment?*. From the experimental results, we aim to develop a theory to explain and generalise the improvements produced by applying the proposed artefact to the situational context for which they were designed. We performed an empirical cycle for validating LiteStrat, the organisational modelling method, in the context of model-driven development for organisational alignment.

The research method selected for the validation activities is Statistical Difference-Making Experiments (Wieringa, 2014), since it allows objectively establishing causal relationships on the effects of the artefacts—moreover, experiments' replicability helps to collect evidence from more and more varied method users incrementally. In particular, we follow the guidelines by Wohlin et al. (Wohlin et al., 2012) for designing the experiments and assessing the validity threats.

As presented in Chapter 1, LiteStrat is proposed as an alternative to i* (Yu, 2011a) under a situational context of model-driven development for organisational alignment; the research goal of the validation is to find out whether LiteStrat better represents organisation information relevant for the strategic alignment of the existing model-driven method with respect to i*. In the following subsections, we present the validation of LiteStrat, through a family of experiments that assess whether the proposal meets the requirements that guided LiteStrat's design, presented in Section 5.4. The experiment is composed of an initial experiment (namely baseline validation) conducted with undergraduate students and two replications conducted with master's degree students.

This chapter detail the validation of LiteStrat, addressing the knowledge question of whether LiteStrat, the proposed organisational modelling method, improves the representation of relevant information for strategic alignment **(KQ3)**. The rest of the chapter continues as follows: Section 7.2.3 describes the experimental design based on a targeted literature review on experimental comparisons of modelling methods. Section 7.3 presents the results of the baseline experiment, while Section 7.4 details the design changes for the replications with respect to the initial experimental design. Section 7.5 and Section 7.6 present the results of the first and second replications, respectively. In Section 7.7, the aggregated results for the family of experiments are described, which are discussed in Section 7.8. The analysis of the threats to validity is detailed in Section 7.9, and finally, Section 7.10 summarises the chapter.

7.2 Experiment design

7.2.1 Research goal and scope

The knowledge question that guides the research is whether LiteStrat, the proposed organisational modelling method, improves the representation of information relevant for strategic alignment when compared with i*, the method used in the existing model-driven development method for organisational modelling. In Definition 7.1, we scope the experiment using the template provided by Basili and Rombach (Basili and Rombach, 1988) and recommended by Wohlin et al. (Wohlin et al., 2012) and comment on the definitions below.

Definition 7.1: LiteStrat validation definition

Analyse *the LiteStrat and i* methods*
for the purpose of *evaluation*
with respect to *their semantic quality*
from the point of view of the *researcher*
in the context of *bachelor and master students modelling a strategic scenario.*

The first two elements of the definition are straightforward: the *objects of study* are the modelling methods, i* and LiteStrat, and the *purpose* is to evaluate them, this is, comparing their effects. Similarly, as the aim of the experiment is to validate the design of an artefact, the results must be observed from the *point of view* of the researcher, which would be comparable to SME's method engineer role. The *experimental context* is given by the constraints of resources of the study; though subjects should ideally be software engineers, students have been considered as a valid simplification for novel approaches (Falessi et al., 2018).

On the other hand, defining the *quality perspective*, i.e., semantic and pragmatic quality, requires considering a conceptual framework suitable for the artefact and the problem context. Since the context of the artefacts is model-driven development for organisational alignment, two quality perspectives must be addressed: how well the methods serve to capture domain information which is relevant for organisation alignment (analysed in Section 1, and how well this information serves to model-driven development. In both cases, we aim to empirically study the quality of the methods by assessing the quality of the models produced by the methods.

Regarding the first quality perspective, we refer to the model quality framework by Lindland et al. (Lindland, Sindre, and Solvberg, 1994). The framework identifies three quality concerns: 1. syntactic quality, which regards how well the model corresponds to the language; 2. semantic quality, which addresses how well the model corresponds to the domain; and 3. pragmatic quality, concerning how well the model corresponds to its audience interpretation. According to Lindland, semantic quality regards how accurately a domain is represented in the model and how complete these models are, which matches the study's intent.

Concerning the second quality perspective, it is necessary to assess whether i* or LiteStrat models serve better to convey business strategy information to other models through model transformations. A conceptual framework for defining quality for models in a model-driven engineering context is provided by (Mohagheghi, 2008). Some quality attributes match Lindland's semantic quality approach, such as models' correctness and completeness, while others assess the consistency of model-to-model transformations and models' comprehensibility. Since model-to-model transformation techniques are not the object of study and model readers' point of view is out of the scope of the experiment, we will consider semantic quality as the only quality perspective of the experiment.

In summary, we consider semantic quality as the primary quality perspective for the experiment. However, other quality-related measurements are proposed to assess the treatments' unintended side effects as recommended in (Wohlin et al., 2012) and detailed in Section 7.2.3.

7.2.2 Related works on semantic comparison of modelling methods

In order to design the experiment based on previous studies, we reviewed the existing experimental comparisons of modelling methods, languages, tools, or notations with respect to the semantic quality of the resulting models. The main outcome regards what metrics and measurement procedures are applied in studies with similar goals and scope.

Inspired by Petersen's guidelines (Petersen, Vakkalanka, and Kuzniarz, 2015), we conducted a targeted literature review. We searched for experimental comparisons of modelling methods in Web of Science, using the following search string: *(TS=((experiment* OR empiric*) AND (comparis* or evaluation or assess* or 'versus' or 'vs') AND model* AND conceptual and (language* or method* or tool*) AND (<modelling method>))*, where the modelling methods considered were organisational, strategy, goal, business process, and class models. The resulting 224 papers were filtered applying the following inclusion (IC) and exclusion criteria (EC): (IC1) Studies must refer to experiments and not to other empirical approaches; (IC2) The object of the study must be modelling languages, methods, notations, or modelling tools, and not to other objects such as algorithms or transformation techniques; (EC1) The design of the activity must focus on creating the models; (EC2) At least one of the assessed variables must regard to how well the method represents the problem domain. Thirty-two articles were selected after the title and abstract review, and eight were finally selected after reviewing the full article.

We performed a backward snowballing (Wohlin, 2014) looking for relevant referenced studies, adding five more articles. From the resulting thirteen studies, we examined the compared methods, languages or tools, the number of participants and whether they are professionals or students. Our primary focus was to analyse the metrics and measurement procedures for assessing quality and the conclusions of the experimental comparisons. Table 7.1 presents the main findings. Next, we summarise and classify our findings according to their approach to quality measurement.

Information retrieval approach

The information retrieval (IR) approach is based on the work of Frakes and Baeza-Yates (Frakes and Baeza-Yates, 1992). It is focused on measuring models' completeness (containment of all the information) and correctness (conformance with the modelled domain) through precision and recall metrics. Precision is the number of correctly modelled elements compared to an ideal model (the oracle) out of the total number of modelled elements. Recall refers to the number of correctly modelled elements out of the total elements in the oracle. Different precision and recall metrics are usually defined for every construct of the modelling language.

Abrahao et al. applied IR for assessing two goal-modelling languages (i* and their proposal, value@GRL) for incremental software development (Abrahão et al., 2018). The researchers summarised the quality in a single metric (F-measure) as the harmonic mean of the precision and recall metrics for the quality evaluation. Significant results favoured value@GRL over i*. The study was replicated and reported as a family of experiments in (Abrahao et al., 2019), preserving the experimental design and involving 184 subjects. The experimental replications confirmed the initial results. The study by Jesus Souza et al. (Jesus Souza et al., 2016) compares two modelling techniques for Dynamic Software Product Lines: the Context-aware Feature Model and the Tropos Goal Model. Significant precision differences were found.

Scanniello and Erra (Scanniello and Erra, 2014) also applied the IR approach to study the quality of models produced by two requirements engineering modelling approaches: their proposal, Think-Pair-Square, and the Face-to-Face approach. The research presents two replications of the same study involving 36 bachelor and master students, and no significant differences in model quality were found.

From the above works, just Scanniello and Erra (Scanniello and Erra, 2014) discuss that creating an oracle model could introduce bias because the oracle model is just one of many possibilities to represent a problem description, which was mitigated by pair-reviewing the oracles.

Semantic quality inspection approach

Semantic Quality Inspection is the name we give to the expert review of models produced by subjects. We found two types of metrics: a score based on a grading scheme and a simple count of errors or hits.

Regarding grading schemes, Kabeli and Shoval (Kabeli and Shoval, 2005) compared two analysis specification methodologies to assess the correctness of the resulting model: their proposal, Functional and Object-Oriented Methodology (FOOM) and Object-Process Methodology (POOM), with a total of 156 participants. The grading scales defined different error points depending on the severity of the error. Significant differences in quality were found favouring FOOM. The same approach is applied by Peleg and Dori (Peleg and Dori, 2000) for comparing their modelling method proposals, Object-Process Methodology (OPM) and Object Modelling Technique (OMT), in order to assess their correctness. The experiment involved 86 participants. The grading scheme has 38 items, each of which can score from 0 to 1, with minor, medium, and major errors with 0.25, 0.5, and 0.75 points, respectively. Of the 38 items, 8 presented significant differences favouring OPM and two favouring OMT. The authors refer their selection of the grading scheme to a previous work by Shoval and Shiran (Shoval and Shiran, 1997), who compare two data modelling techniques: Extended Entity-Relationship (EER) and Object-Oriented (OO). The grading scheme had nine items, finding significant differences for two items, favouring EER.

Concerning the assessment by counting errors or hits, Trkman et al. (Trkman et al., 2019) compared the effects in user stories' identification of two business process specification formats: business process models and use cases. The researchers counted the correctly identified stories, having as subjects 75 undergraduate students; no significant differences were found. Saputri and Lee (Saputri and Lee, 2020) compared the effectiveness of two sustainability requirements engineering (RE) methods: the traditional RE approach and their proposal, ENSURE, in a study that involved 18 experienced subjects. Effectiveness was measured by counting the number of correctly identified stakeholders, requirements, features, and the percentage of solved conflicts, with significant results favouring their proposal. Thabet et al. (Thabet et al., 2021) compared two tools that support the Business Process-Risk Management-Integrated Method (BPRIM): a multi-view tool and a diagram-oriented one. The researchers assessed the correctness of the models produced by 41 subjects by counting the errors against the source specification. The authors counted the number of modelling errors (wrong use of constructs) and semantic errors (when the subjects miss elements of the underlying business process model). The results for the two measurements showed significant differences favouring the multi-view tool. Ionita et al. (Ionita et al., 2015) studied the effects of tangible modelling on eliciting domain knowledge. Tangible and computer-aided designed models were assessed by counting errors, considering three types of errors: (1) placing an element where none was expected; (2) a missing element where one was expected; and (3) using a wrong concept to represent an element. The tangible group finished 52% faster and had 50% of the errors than the control group.

As discussed in some of the studies, the semantic quality inspection approach introduces subjectivity, which is usually mitigated by the collaborative definition of the grading scales and error types.

Other approaches for model quality measurement

Another approach for comparing models' quality is applying the semiotic approach using the SEQUAL framework by Krogstie (Krogstie, 2002). In (Matulevičius and Heymans, 2007a), Matulevičius and Heymans applied it to compare the quality of two goal-modelling languages using a 5-point Likert scale for 15 questions derived from the quality attributes proposed in SEQUAL. Another approach is to directly count the number of mismatches of the models produced by the subjects and an ideal model (gold standard), which is followed by Ibriwesh et al. (Ibriwesh et al., 2017) to compare three data documentation perspectives as inputs for use-case modelling.

Efficiency and satisfaction measurement

As summarised in Table 7.1, besides the quality of the resulting models, most experimental comparisons also measure the users' efficiency and satisfaction, usually to assess if there are unintended effects of new modelling languages or tools. Efficiency is mostly measured by the time needed to complete a modelling task (Jesus Souza et al., 2016; Scanniello and Erra, 2014; Kabeli and Shoval, 2005; Shoval and Shiran, 1997; Saputri and Lee, 2020; Thabet et al., 2021; Ionita et al., 2015; Ibriwesh et al., 2017). With regard to user satisfaction, in (Ionita et al., 2015), the authors considered the usability questionnaire proposed by Lewis (Lewis, 1991), while Saputri et al. defined an ad-hoc instrument of 27 Likert scale questions (Saputri and Lee, 2020) for assessing practicality and usability. Abrahao et al. (Abrahão et al., 2018; Abrahao et al., 2019) measured the modellers' satisfaction by the perceived ease of use and perceived usefulness, based on the Technology Acceptance Model by Davis (F. Davis, 1989). Matulevicius and Heymans (Matulevičius and Heymans, 2007a) designed a questionnaire based on the pragmatic quality attributes of the SEQUAL framework.

Summary

In summary, the comparison of the models' quality is mainly assessed in terms of correctness (also named accuracy) and completeness and the inspection of models against the original specification over a grading scheme is one of the most used measurement approaches. Other variables for assessing undesired side effects are user efficiency and user satisfaction, measured in terms of users' perception of ease of use and usefulness of the modelling methods.

Table 7.1: Summary of related works for experimental comparisons of modelling methods.

Study	Method, Language or Tool	Participants	Variables & Metrics	Results
Kabeli and Shoval, 2005	FOOM, POOM	156 undergraduate (UG) students	Correctness, Comprehension, Efficiency, Preference	FOOM specifications are more correct, comprehensible, and preferred by the users.
Ibriwesh et al., 2017	Use Case	103 UG students	Correctness, Time, Perceived Ease of Use	ERD specifications are easier to understand, more helpful, and less time-consuming for use case modelling.
Thabet et al., 2021	BPRIM diagram, multi-view tool	41 bachelor, M.Sc., and Ph.D. students	Correctness, Usability, Efficiency	Multi-view modelling is more usable and efficient, and produces more correct models.
Trkman et al., 2019	Business Process Models, Use Cases	75 UG students	Accuracy, Dependency understandability, Integration understandability	BPMN specifications produce better understanding of the execution-order dependencies. No differences in Accuracy and Integration Understandability were found.
Ionita et al., 2015	Tangible modelling, Computer-aided modelling	8 UG students	Correctness, Learnability, Efficiency, Satisfaction	The results suggest that Tangible Modelling is faster, easier to learn, more efficient, and produces more correct models and more satisfaction to the users.
Saputri and Lee, 2020	ENSURE, Traditional RE	18 postdoc and professional subjects	Effectiveness, Efficiency, Practicability	ENSURE is more effective, and efficient and is perceived to be more practical than a traditional RE approach.
Abrahao et al., 2019	i*, value@GRL	100 M.Sc. students, 64 UG students, 20 professionals	Model Quality, Efficiency, Productivity. Perceived ease of Use, Productivity	Value@GRL is more efficient, productive, and is perceived to be more useful by users. Also, value@GRL produces models with better quality.
Jesus Souza et al., 2016	Context-aware Feature Model, Tropos	10 Ph.D. and M.Sc. students	Effectiveness, Efficiency	Context-aware Feature Model is more effective than Tropos Goal Model considering precision, but no differences in recall were found.
Shoval and Shiran, 1997	Entity-Relationship, Object Oriented	44 UG students	Correctness, Time, Preference	EER yields to better modelling quality regarding unary and ternary relationships, is less time-consuming, and is preferred by users.
Peleg and Dori, 2000	OPM, OMT	86 UG students	Correctness, Comprehension	OPM produces specifications with higher quality and is more comprehensible.
Abrahão et al., 2018	i*, value@GRL	40 M.Sc. students, 28 software engineers, 12 analysts	Correctness, Completeness, Efficiency, Perceived ease of Use, Productivity	Value@GRL produces models of higher quality, and users are more productive. Also, value@GRL is perceived to be easier to use
Matulevičius and Heymans, 2007a	i*, KAOS	19 graduate students	Model Quality, Language Quality	The KAOS language quality is higher than i*. i* models have higher quality than KAOS models.
Scanniello and Erra, 2014	Think-Pair-Square, Face to Face	36 UG and M.Sc. students	Correctness, Completeness, Time	Face-to-Face interaction is less time consuming, and no significant differences in quality were observed.

7.2.3 Experimental design

Research questions and hypothesis formulation

As commented in the definition of the experiment, the quality perspective regards semantic quality. Based on the conclusions of the literature review (which are consistent with the framework by Lindland et al. (Lindland, Sindre, and Solvberg, 1994)), the research questions will address the quality of models' accuracy and completeness. As also found in the literature review, we consider studying the side effects of the methods on the quality of the modelling process, for which we follow the method evaluation model by Moody et al. (Moody, 2003). This model focuses on assessing the method users' performance and their perception of its usefulness, ease of use, and intention to keep using it. Therefore, the research questions regarding the modelling process are referred to modellers' efficiency and perceived satisfaction. The research questions derived from the defined quality goals and the associated null hypotheses are detailed below.

- **RQ1: Is modelling accuracy affected by the modelling method used?** Following Lindland's definition of semantic quality, we adhere to the definition of semantic accuracy in ISO25012[1] as *"the closeness of the data values to a set of values defined in a domain considered semantically correct"*. The null hypothesis associated with this question is H_{a0} - There are no differences in accuracy between LiteStrat and i* for business strategy modelling.
- **RQ2: Is modelling completeness affected by the modelling method used?** According to Lindland, a model is complete if it *contains all the statements about the domain that are correct and relevant* (Lindland, Sindre, and Solvberg, 1994). The null hypothesis to assert this question is H_{c0} - There are no differences in completeness between LiteStrat and i* for business strategy modelling.
- **RQ3: Is modellers' satisfaction affected by the modelling method used?** Following the model in (Lindland, Sindre, and Solvberg, 1994), we consider the modeller's satisfaction as the subjective perception of the method and the intention to use it for business strategy modelling. The null hypothesis formulated from these definitions is H_{s0}- There are no differences in modellers' satisfaction when using LiteStrat or i* for business strategy modelling.

[1] https://iso25000.com/index.php/en/iso-25000-standards/iso-25012

- **RQ4: Is modellers' efficiency affected by the modelling method used?** Efficiency, another component of the model in (Moody, 2003), is defined as the effort required to apply a method. The null hypothesis to address this question is H_{e0} - There are no differences in modellers' efficiency when using LiteStrat or i* for business strategy modelling.

Factors and treatments

The factor under study is the modelling method, which has two treatments: i* and LiteStrat. Both treatments are applied to the representation of a business strategy case (the problem), which textually describes an organisation's strategy in offering new products or services to its customers, harnessing opportunities, or mitigating environmental risks. The modelling methods are applied without tool support. Subjects using LiteStrat must use the language's constructs and relationships and follow the modelling procedure prescribed by the method. Subjects using i* must use the constructs and relationships of iStar 2.0 (Dalpiaz, Franch, and Horkoff, 2016), but the modelling procedure depends entirely on the subjects' criteria. A second factor that might affect the observed phenomena is the problem to be modelled. The problem is considered a blocking variable since we are not interested in analysing differences between problems; however, we must ensure that the results are independent of the problems. This blocking variable has two levels: Problem 1 and Problem 2, which are further detailed below.

Response variables and metrics

To assess accuracy and completeness (RQ1 and RQ2), we will follow the approach of what we named *semantic inspections* in Section 7.2.2 . The approach is to review the models designed by the subjects and inspect whether they accurately and completely represent the given problem. To do this, we divided the experimental problem's sentences that present strategic information (namely *statements*) to construct the grading schemes for each problem. For a more detailed analysis, we classified the statements into four types:

1. Motivation Statements, which describe the organisation's higher-level goals and the elements from the organisation's environment that drive such goals.

2. Action Statements, which describe what the organisation is willing to do to achieve its goals

3. Roles and Responsibilities Statements, that quantitatively define the desired ends and their assignment to organisational roles
4. Outcome Statements, that describe the effects of the strategy among the organisation's parts and on its environment.

To address RQ1, we define the *accuracy* response variable, meaning how well a statement in the problem description is represented in the conceptual model. Each statement in the experimental problem is checked and rated according to a three-level grading scale: 2 accuracy points if the whole statement is represented with the appropriate constructs and relationships provided by the language; 1 accuracy point if the statement is partially represented with the appropriate constructs, i.e., some constructs can be misused; and 0 accuracy points, if the statement is misrepresented or missing. It is worth noting that, for both languages, every semantically valid combination of their concepts and relationships is considered a correct representation. We define five accuracy metrics:

- **Motivation Accuracy (MA)**, the sum of accuracy points for all the motivation statements.
- **Actions Accuracy (AA)**, the sum of accuracy points for all the action statements.
- **Role-Responsibility Accuracy (RRA)**, the sum of accuracy points for all the role-responsibility statements.
- **Outcome Accuracy (OA)**, the sum of accuracy points for all the outcome statements.
- **Total Accuracy (TA)**, the sum of all the above metrics: `TA = MA + AA + RRA + OA`

Taking into account the grading scale described above and the number of statements in each experimental problem (detailed in the definition of the experimental problems), in Table 7.2 details the maximum scores for each variable and each experimental problem. The grading scheme and the detailed evaluation of models are available in the experimental package (Noel, Panach, and Pastor, 2021a). In Appendix B, we included the grading scheme, semantic inspection guidelines for each problem, and two reference models used during the assessment. We also included two models produced by the experimental subjects as examples.

Table 7.2: Maximum scores for accuracy metrics for each experimental problem.

	Max. Accuracy Score	
Accuracy Metric	**Problem 1**	**Problem 2**
EIA	6	6
SA	6	6
RRA	8	8
RA	4	4
TA	24	24

For RQ2, we define the completeness response variable as the degree to which all concepts in a statement are represented in the model. For instance, the statement 'the organisation x must achieve y' will be complete if the actor organisation x and the intention y are modelled and somehow related, using any of the constructs and relationships of the languages. It is worth noting that a statement can be complete, but it will not be accurate if the concepts and relationships used are not semantically valid. Similarly, we define a grading scale of two, one, and zero completeness points for complete, incomplete, and non-modelled statements, respectively, for the accuracy variable. We also define five metrics associated with subsets of statements:

- **Motivation Completeness (MC)**, the sum of completeness points for all the motivation statements.
- **Actions Completeness (AC)**, the sum of completeness points for all the action statements.
- **Role-Responsibility Completeness (RRC)**, the sum of completeness points for all the role-responsibility statements.
- **Outcome Completeness (OC)**, the sum of completeness points for all the outcome statements.
- **Total Completeness (TC)**, the sum of all the above metrics: TC = MC + AC + RRC + OC

Since the completeness metrics have the same grading scale as accuracy metrics, the maximum values for MC, AC, RRC, OC, and TC are 6, 6, 8, 4, and 24, respectively, for both experimental problems.

For RQ3, we define the user satisfaction variable, addressed by the evaluation model proposed by Moody (Moody, 2003). It defines three metrics: **Perceived Ease of Use (PEU)**, **Perceived Usefulness (PU)**, and **Intention to Use**

(IU). The model proposes a survey that consists of 16 questions expressed on a 5-point Likert scale, which represents the degree of satisfaction from totally disagree (1 point) to totally agree (5 points). The instrument provides six questions to measure PEU, whose scale is [6-30] (adding the results of the six PEU questions), eight questions to measure PU, whose scale is [8-40], and two questions for IU, whose scale is [2-10].

Finally, to assess RQ4, we define the efficiency variable, measured as the **time (t)** needed to perform the business modelling tasks. Time is measured in minutes and was self-reported by the subjects. We checked that the reported times were consistent from when the subjects received the business strategy case to when they submitted the finished models.

We summarise the variables and metrics for each research question in Table 7.3.

Table 7.3: Research questions, hypotheses, variables, and metrics.

RQs	Hypotheses	Response Variables	Metrics
RQ1	Ha0	accuracy	TA, EIA, SA, RRA, RA
RQ2	Hc0	completeness	TC, EIC, SC, RRC, RC
RQ3	Hs0	satisfaction	PEU, PU, IU
RQ4	He0	efficiency	time

Measuring semantic quality

In order to ensure the objective measurement of quality, both the experimental problems and the measurement procedure must not favour any of the modelling methods. This implies, on the one hand, that motivation, action, role and responsibility, and outcome statements can be modelled with i* and LiteStrat, and on the other hand, that the semantic inspection of the models considers the differences between the two languages. Tables 7.4 and 7.5 details the differences and similarities between LiteStrat and i* constructs and relationships, respectively.

In the semantic inspection, we aim to assess the conceptualisations that subjects can perform with their given modelling methods, evaluating whether they are accurate and complete. Our approach is to provide a natural language description of the domain on which statements of different types are seeded. Subjects are expected to conceptualise them using their respective methods. In the semantic quality inspection, we examine the models produced by the subjects and search for modelling constructs and relationships that could rep-

Table 7.4: Differences between i* and LiteStrat constructs.

I* Construct	LiteStrat Construct	Comment
Actor Types		
Actor	Actor	In i*, Actors represent any type of intentional actor. In LiteStrat, Actors represent intentional actors that are outside the organisation and whose intention cannot be known.
Agent	Organisation Unit	In i*, Agents represent an actor with concrete physical manifestation, such as an individual, organisation or department. In LiteStrat, Organisation Units represent the same elements except for individuals.
Role	Role	Both are abstract characterisations of the behaviour of a social actor within some context.
Actor Association Links		
Participates-in	Participates-in	While i* does not restrict the types of actors that can be linked, LiteStrat defines that only organisation units and roles can participate in organisation units.
Is-A	–	While i* does not restrict the types of actors that can be linked, LiteStrat defines that only organisation units and roles can participate in organisation units.
Intentional Elements		
Goal	Goal	While both represent a desired state of affairs of any type of actor, in LiteStrat, it is reserved just for Organisation Units.
	Objective	Is a LiteStrat's specification of measurable goals that is reserved for Roles.
Task	Strategy	While i* defines tasks as actions that an actor wants to be executed usually with the purpose of achieving some goal, LiteStrat separates these actions into high-level actions (Strategies) and specific actions (Tactics). Strategies represent an explicit high-level action towards the achievement of a goal.
	Tactic	Represents concrete actions towards the implementation of a strategy
Quality	–	LiteStrat does not support the quality construct, since it is expected that objectives could serve to represent measurable desired levels of quality regarding the business strategy.
Resource	–	Resource modelling is out of the scope of LiteStrat.

Table 7.5: Differences between i* and LiteStrat relationships.

I* Construct	LiteStrat Construct	Comment
Actor Types		
Social Dependencies		
Goal	Objective assignment	In i*, any type of actor can socially-depend on any other type of actor to achieve its goals. In LiteStrat, only Organisation Units can depend on roles to achieve objectives through objective assignment.
Quality	–	Not supported in LiteStrat.
Task	Tactic Assignment	In i*, any type of actor can socially-depend on any other type of actor to achieve its goals in performing an action. In LiteStrat, only Organisation Units can depend on other Organisation Units to implement tactics.
Resource	–	Not supported in LiteStrat.
–	Influence	In LiteStrat, Actors or Organisation Units can behave in a way that affects other Organisation Units or Actors, but not necessarily with an intention to affect them. LiteStrat proposes the Influence construct to represent this relationship.
Intentional Element Links		
Refinement	Refinement	In i*, it is a hierarchical link between goals or tasks. In LiteStrat, it is also a hierarchical link, but the hierarchy is prescribed by the modelling procedure going from goals to strategies to tactics and then to objectives.
Needed by	–	Not supported in LiteStrat.
Contribution	–	Not supported in LiteStrat.
Qualification	–	Not supported in LiteStrat.

resent the statements. To illustrate this, please consider Example 7.1. In Table 7.6, we provide examples of how the motivation, action, role and responsibility, and outcome statements can be modelled using the i* and LiteStrat constructs. Please note that for i*, a statement can be completely and accurately represented using other semantically valid constructs.

Example 7.1: Strategic scenario for Real-Estate Co.

Real-Estate Co. is a house renting company that has detected the need of abroad customers to rent a house remotely. Under this scenario, the company sets the strategic goal of increasing its customer base and defines its main strategy to provide a virtual tour for the houses. Besides other specific actions needed to implement this strategy, the company defines that the Rentals Team must implement the virtual showroom by allowing users to navigate 360º pictures in the company's app. The success of the strategy will be measured in terms of a specific objective: to rent 20 houses in the first three months since the feature is delivered. Rentals Team's Product Owner is responsible for tracking and reporting this objective. The new virtual showroom feature is expected to be delivered to abroad customers by the Rentals Team.

The i* and LiteStrat examples provided are accurate and complete representations of the statements. Lacking a construct or relationship would result in an incomplete and inaccurate representation; while representing all the elements but using non-semantically equivalent constructs (e.g., modelling a role with an i* agent or with an organisation unit in LiteStrat) would result in a complete by inaccurate representation. Table 7.7 presents the grading scheme for accuracy and completeness.

Experimental problems

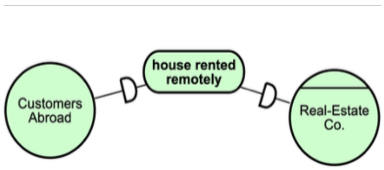
We designed two business strategy cases, describing a scenario where the organisations must define a strategy to address external factors. The cases detail the ends and means of the organisation and the organisational structure needed to deploy the strategy. Problem 1 describes a telecommunications company reacting to a new competitor with an improved service. Problem 2 describes an insurance company aiming to exploit a new business opportunity from a regulation change. The experimental problems were paragraphs written in prose, in natural language, and the statements relevant to business strategy modelling

Table 7.6: Examples of representations for motivation, action, role and responsibility, and outcome statements in i* and LiteStrat.

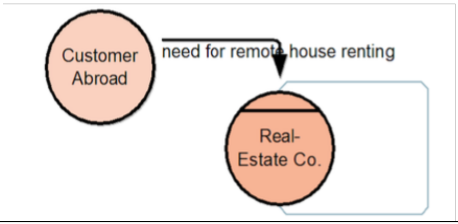
Motivation Statement

"...has detected the need of abroad customers to remotely rent a house"

i* example



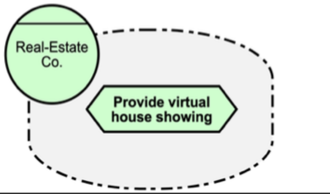
LiteStrat example



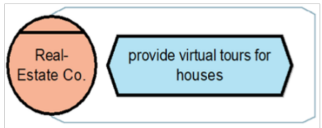
Action Statement

"...its main strategy to provide virtual tour for the houses."

i* example



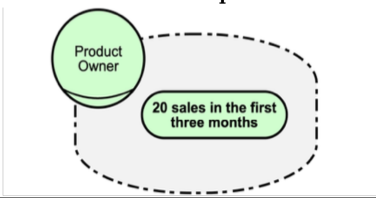
LiteStrat example



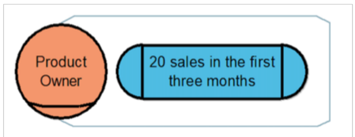
Role and Responsibility Statement

"...to rent 20 houses in the first three months since the feature is delivered. The responsible of tracking and reporting this objective is Rentals Team's Product Owner."

i* example



LiteStrat



Outcome Statement

"The new virtual showroom feature is expected to abroad customers by the Rentals Team."

i* example

LiteStrat example

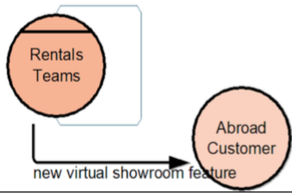


Table 7.7: Accuracy and completeness grading schemes.

Accuracy Grading Scheme (per statement)		
Accuracy Level	***Points***	***Description***
Achieved	2	The model represents all the domain elements of the statement using the proposed or semantically equivalent i*/LiteStrat constructs.
Partially Achieved	1	The model represents most of the domain elements of the statement using the proposed or semantically equivalent i*/LiteStrat constructs.
Unachieved	0	The model represents most of the domain elements of the statement using constructs different to the proposed i*/LiteStrat constructs which are not semantically equivalent.
Completeness Grading Scheme (per statement)		
Completeness Level	***Points***	***Description***
Achieved	2	The model represents all the domain elements regardless of the i*/LiteStrat constructs used.
Partially Achieved	1	The model represents most of the domain elements of the statement, regardless the i*/LiteStrat constructs used.
Unachieved	0	The model fails to represent most of the domain elements of the statements.

were embedded in the narrative. Hence, the subjects had to identify them from the text. The style and vocabulary used in the problems are based on business strategy cases from Forbes[2] and McKinsey[3]. To illustrate the style of the problem descriptions, an extract from problem 1 is given below. Example 7.2 and 7.3 present the textual descriptions for problems 1 and 2, respectively.

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

[3]https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/planning-in-an-agile-organization

Example 7.2: Problem 1: Telecommunications Company

A mobile phone service provider, "BTR" has 25% market share and is the third largest national mobile phone provider. BTR's top management notices the entry of a new competitor, "WOW", which has a very aggressive advertising campaign. This campaign highlights the shortcomings of its competitors and has special offers for the youth segment. Although WOW is new in the country, it is a brand of a multinational firm with wide dominance in other countries of the continent. BTR's top management is concerned about how this may impact its goal of remaining the third-largest mobile provider in the country.

One of BTR's strengths is that the largest percentage of its customers are professionals aged 40 and over, so the impact of WOW's youth campaign is low. However, a weakness is that BTR has one of the highest complaint rates among telephone providers regarding erroneous charges and monthly billing problems, and WOW's aggressive campaign explicitly mentions clear charges as one of its advantages. On the other hand, BTR has a mobile application (app) used by about 95% of its customers. The current monthly billing process usually requires much extra effort to be completed on time, partly because some steps (e.g. validation of charges) are done manually, and there is no time left to verify that they have been carried out correctly.

With this, the BTR concludes that there is a risk of losing customers due to the poor quality of the billing process in general. Given this situation, its goal is to retain customers by offering a quality billing and collection service, so it seeks to mitigate the weakness with the following strategies: 1. Billing with zero errors and 2. The Billing Area will implement the strategies. For the first strategy, two tactics associated with the billing process have been defined: reduce the time it takes to complete the billing process, to implement the second tactic, which is adding a set of quality control activities to the process. For the second strategy, it is decided that the customer's charges for each service usage should be automatically validated and published through the app.

To implement the optimisation of billing, the objective has been defined as reducing the duration of the billing process by three days, with the Head of the Billing Area being responsible for achieving this; in addition, the new role of Quality Manager is assigned the responsibility of controlling and correcting billing in the three days left by optimisation. On the other hand, for the tactic "Integrate quality control into the invoicing process", the objective is that at least 25% of the invoices should be audited, and the Quality Manager is responsible for ensuring this happens. Finally, for the tactic "Automate the collection validation process", the objective is defined as being that collections must be validated no later than 12 hours after they are made and that after this, their publication in the app must be instantaneous, with the current collection validation manager being responsible for this.

As a result, the organisation seeks to re-engage its existing customers through improved billing service and a new service with daily billing details in the app. The billing improvement is delivered directly by the Billing Area, while the Marketing Area will be used to inform customers about the new app service.

Example 7.3: Problem 2: Insurance Company

The insurance company "Short Life" is the country's leader in motor insurance. Its clients highlight the high quality of its executives' attention, the clarity of its statements, and the broad coverage of its products. While Short Life does not offer other types of insurance to date because of the high cost of validating them with the Superintendency of Insurance, a new regulation from the Superintendency has determined that it is possible to add home theft insurance to auto insurance.

Short Life's senior management is anxious to take advantage of this opportunity. One of Short Life's weaknesses is the cost of advertising new products, which it usually outsources to an external company. Despite this, Short Life has decided to make it a goal to offer and advertise home burglary insurance without using a third party. In order to achieve this goal, it has decided to create a new Insurance Promotion Department, which will work in tandem with the Insurance Design Department, which is responsible for creating new insurance products. It has also decided to involve the Customer Service Department to personally contact existing customers to inform them about the new product. The goal for the Head of the Customer Service Department is to reach 70% of customers by telephone within the first week of a new product launch.

The company states that the Promotion Department should be involved in the design process of new insurance products from day one, and the Product Design Department should provide information from customer and competitor studies obtained during design. The Promotion Department must meet the objective of having the advertising campaign in place at least two weeks in advance of the product launch, which will be the responsibility of the Chief Publicist; in addition, its advertising campaigns must reach at least 20% of the market that is not yet a Short Life customer, which must be met by the Chief Publicist. The Insurance Design Department should ensure that customer and competitor research is carried out within a maximum of 20 working days, which is the responsibility of the Market Analyst.

With the above decisions, it is expected that Short Life will be able to offer its existing customers a new Home Burglary Insurance service through the Customer Service Department, and through the new Advertising Area, to potential new customers.

The two problems have an equivalent complexity, with the same number of relevant statements for business strategy modelling. For each problem, we designed a semantic assessment guideline, which details the statements seeded in the text and proposes combinations of constructs to represent them in i* and LiteStrat. In Table B.1 and Table B.2 in the Appendix B, we detail semantic inspection guidelines for Problem 1 and Problem 2, respectively. In the same Appendix, Figures B.1, B.2, B.3, and B.4 present LiteStrat and i* accurate and complete solution examples for Problem 1 and Problem 2.

Experimental design

The experiment has a 2x2 factorial design (Wohlin et al., 2012). The factor corresponding to the modelling method is the treatment, with two levels (LiteStrat and i*), and the blocking variable Problem also has two levels (Problem 1 and Problem 2). Four experimental groups were defined due to the combination of the two factors and problems. The subjects are randomly assigned to the groups.

Data analysis

The models produced by the participants were reviewed using the grading scheme and inspection guidelines presented in the Appendix. The scores for each model and each statement are available in the experimental package (Noel, Panach, and Pastor, 2021a).

We performed a univariate General Linear Model (GLM) analysis for each variable to analyse the interaction treatment*problem, thereby checking if a problem works better for a specific treatment or, conversely, if the blocked design was successful. The assumptions needed to apply GLM were verified for all the variables using the Shapiro-Wilk test for the normality of the residuals and the Levene test for variance homogeneity. The analysis was performed using SPSS version 25. As Section 3 shows, one of the variables did not fulfil the normality assumption. Even though GLM is robust to normality deviation, we opted to apply the $1/\sqrt{X}$ transformation to data that are not normally distributed for the AC metric.

The results from the GLM are considered significant when the p-value is less than .05. The effect sizes are calculated for those metrics with significant results to analyse the magnitude of the differences using the partial $\eta 2$ generated by SPSS; a value lower than .01 is considered to be a small effect, a value between .01 and .06 is associated with a moderate effect size, and higher than .14 is considered a large effect size. We also calculated the statistical power, which is the probability of rejecting the null hypothesis when it is false. Dyba et al. (Dybå, Kampenes, and Sjøberg, 2006) report values greater than 0.39 for medium power and greater than 0.63 for large power.

7.3 Baseline validation

The first experiment was carried out in September 2020, during the COVID-19 pandemic. The following sections detail the experimental setting and results.

7.3.1 Experimental Setting

Experimental Subjects

Twenty-eight undergraduate students from a third-year Requirements Engineering course at the Universidad de Valparaíso in Chile participated in the baseline experiment. They all had conceptual modelling backgrounds from a Software Engineering Foundations course covering a subset of UML models (use cases, classes, state, sequence, and deployment diagrams). The course also introduced organisational goals and their relationship with requirements, although no modelling languages were introduced for this topic. None of the subjects had professional experience in software projects.

All subjects completed an informed consent form. Since participants were taking a course that included training in the two methods compared in the experiment, the study was considered non-interventional, and therefore, no ethics committee approval was required.

Discarding the subjects that dropped out, the final distribution of subjects is summarised in Table 7.8.

Table 7.8: Experimental design and distribution of subjects

	LiteStrat	**i***
Problem 1	7	7
Problem 2	6	8

Experimental procedure

The experiment was performed in a remote, online context using ZOOM due to the COVID-19 pandemic. Initially, 30 subjects voluntarily participated in the activity. The procedure had two stages: the training session and the experimental activity. Each stage lasted one hour.

The training session was performed just before the experimental activity, divided into two parts: a 20-minute presentation introducing business strategy definitions and concepts and a second 40-minute presentation addressing the modelling method. Both parts were taught by the instructor of the requirements engineering course, who is not involved in this thesis or in the design of LiteStrat. The first part was the same for all the subjects and covered general business strategy and organisation structure concepts. The subjects were randomly assigned to LiteStrat or i* training sessions in the second part. The training for the i* group was performed first, and then the second group received the LiteStrat training.

Besides presenting the framework for the i* group, the training considered examples of how to use i* constructs to represent business strategy elements relevant to the experiment, based on the proposal presented in (Noel, Ruiz, et al., 2021). For the LiteStrat group, we presented the modelling language and procedure described in Chapter 5.

The experiment took one hour and consisted of three steps: completing the informed consent form, performing the modelling activity, and completing the post-test survey. The informed consent form and the post-test questionnaire were implemented in Google Forms, while the modelling activity was performed using pen and paper. During the modelling activity, the subjects accessed the experimental problems (randomly assigned and balanced between groups) and the training materials. The subjects uploaded pictures of their models in the last question of the post-test questionnaire. All the subjects in i* group completed the activity and the surveys, and two from the LiteStrat group dropped out. No models were discarded for lack of picture quality.

7.3.2 Results

Below, we detail the results for the research questions stated in Section 7.2.3.

Research Question 1: Accuracy

To answer RQ1, we examined the Total Accuracy (TA) metric as well as the metrics for each requirement type: Motivation Accuracy (MA), Actions Accuracy (AA), Role and Responsibility Accuracy (RRA), and Outcome Accuracy (OA). Figure 7.1 shows the box plot for these metrics.

As can be observed, except for OA, LiteStrat shows better results for all accuracy metrics; the more pronounced differences can be identified in TA and MA. The data analysis results for the accuracy metrics are detailed in Table 7.9. The Treatment column shows the probability value (p-value) for the effect

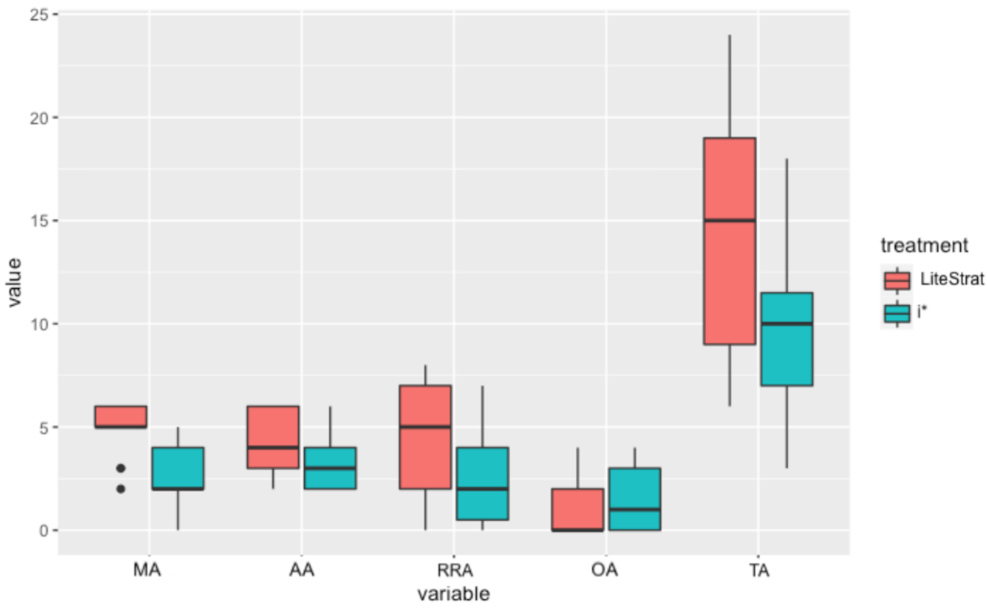


Figure 7.1: Box plot for the accuracy variables.

of the treatment, which is our main interest and is presented in bold text. Statistically significant results for the treatment effect are marked with (**). The Interaction column shows the p-value of the Treatment*Problem interaction. The table also shows the means for LiteStrat and i*, the effect size (just for significant differences), and the statistical power.

Table 7.9: Data analysis for accuracy variables.

	Treatment	Interaction	Mean	Effect Size	Power
TA	****.014**	.409	LS: 14.697 i*: 9.696	.227	.721
MA	****.000**	.173	LS: 4.879 i*: 2.527	.412	.976
AA	**.117**	.788	LS: 4.036 i*: 3.179	-	.346
RRA	**.051**	.125	LS: 4.488 i*: 2.473	-	.506
OA	**.636**	.488	LS: 1.214 i*: 1.518	-	.075

Significant differences were found for Total Accuracy ($p = .014$). The difference favours LiteStrat over i*, with a mean of 14.697 score points versus 9.696 points. The size of this effect can be described as large ($es = .227$), and the statistical power is high ($p = .721$). This result means that LiteStrat outperforms i* in the accuracy for describing the business strategy domain, and this difference has practical significance, i.e., the effect is large enough to be meaningful in the real world.

Concerning the analysis by requirement types, the MA variable also presents significant results ($p < 0.001$), favouring LiteStrat over i*, with a mean of 5.095 score points versus 2.661, respectively. The effect size is large ($es =.412$), and the statistical power is high ($p = .976$). This means that using LiteStrat is associated with better modelling of the external factors that trigger the strategy and a better specification of the overall goal that drives the organisational change. This difference has practical significance.

We did not find significant differences in the treatment effect for the rest of the metrics related to other requirement types, even though the statistical power is moderate for the AA and RRA metrics ($p > .39$). None of the metrics showed significant effects caused by the treatment*problem interaction.

From the above results, we can state that the null hypothesis H_{a0} can be rejected for the Total Accuracy (TA) and Motivation Accuracy (MA) metrics favouring LiteStrat over i*. For the Actions Accuracy (AA), Role and Responsibility Accuracy (RRA), and Outcome Accuracy (OA) metrics, hypothesis H_{a0} can not be rejected. We can also state that the results are independent of the problems since treatment*problem interactions are not significant.

Research Question 2: Completeness

We conducted the same analysis presented in the previous section for the five metrics for completeness: Total Completeness (TC), Motivation Completeness (MC), Actions Completeness (AC), Role and Responsibility Completeness (RRC), and Outcome Completeness (OC). We present the box plot for the five metrics in Figure 7.2. As can be observed, most of the metrics favour LiteStrat over i*, except for OC. For TC, the results for i* seem to be more spread out than LiteStrat results, showing two mild outliers.

Table 7.10 details the data analysis results for the completeness metrics. The treatment had significant effects on the TC, MC, and AC metrics. For TC, the p-value is .015, favouring LiteStrat over i* with a significant difference. The

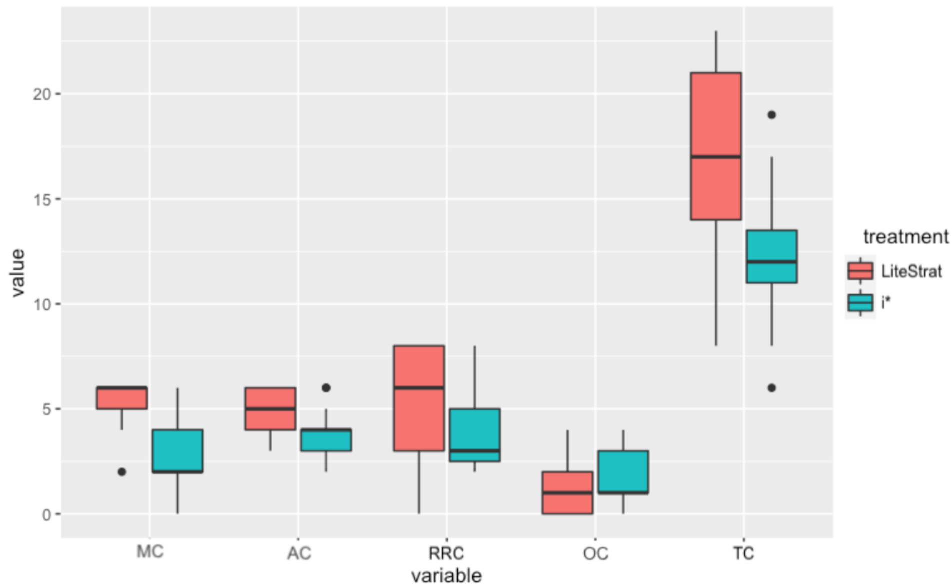


Figure 7.2: Box plot for the completeness variables.

effect size is large (es = .222), while its statistical power is also high (pw = .710). For MC, the treatment is significant (p < .0001, favouring LiteStrat over i*. This effect size is large (es = .426), and the statistical power is high (pw = .981). For AC, the difference is significant (p = .046), also favouring LiteStrat over i*. The effect is large (es = .164), and the power is moderate (pw = .548). None of the metrics showed significant effects for the treatment*problem interaction (p > .05 in column Interaction).

We did not find differences for the RRC and OC metrics, and the statistical power is low (pw < .39). From these results, we conclude that LiteStrat outperforms i* in terms of completeness of the business strategy models, and this difference has practical implications, given its large effect size. Following these results, the null hypothesis H_{c0} can be rejected for the Total Completeness (TC), Motivation Completeness (MC), and Actions Completeness (AC) metrics. For the Role and Responsibility Completeness (RRC) and Outcome Completeness (OC) metrics, H_{c0} cannot be rejected. We found no evidence that the experimental problems affected the results.

Table 7.10: Data analysis results for completeness variables.

	Treatment	**Interaction**	**Mean**	**Effect Size**	**Power**
TC	****.015**	.208	LS: 16.381 i*: 12.268	.222	.710
MC	****.000**	.153	LS: 5.095 i*: 2.661	.426	.981
AC	****.046**	.296	LS: 4.821 i*: 3.884	.164	.548
RRC	**.178**	.414	LS: 5.179 i*: 3.929	-	.265
OC	**.372**	.519	LS: 1.286 i*: 1.795	-	.141

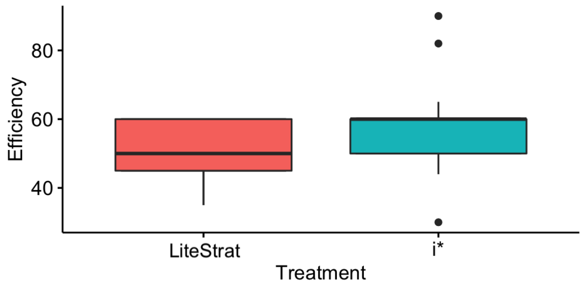


Figure 7.3: Box plot for efficiency.

Research Question 3: Efficiency

As shown in Figure 7.3, the medians for LiteStrat show a higher efficiency for i*, as the top LiteStrat values are equal to the i* median. It is important to note that efficiency is measured in minutes, so lower values mean more efficiency. Note that all LiteStrat users completed the tasks in less time than the median of i* users.

The data analysis results for efficiency are shown in Table 7.11. The means show a tendency of the subjects to need less time to complete the modelling task using LiteStrat than using i* (50.214 minutes v/s 58.232 minutes, respectively). However, the difference is not statistically significant (p = .102). No significant interaction effects caused by the problem or by the treatment*problem interaction were found. The statistical power is low (pw < .39)

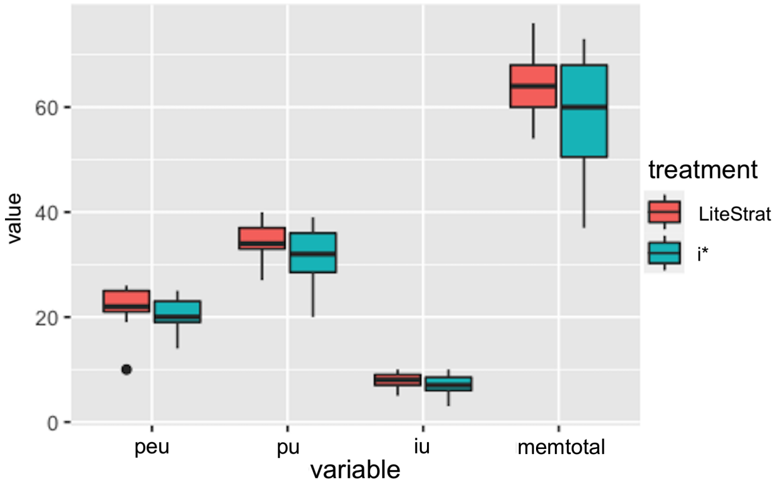


Figure 7.4: Box plot for users' satisfaction variables.

Thus, it is not possible to reject the null hypothesis H_{e0}, which means that differences were found in the effort needed to complete the modelling task using LiteStrat or i*. Also, the low statistical power does not allow us to suggest that there are no differences between the two treatments.

Table 7.11: Data analysis results for the efficiency variable.

	Treatment	Interaction	Means	Effect Size	Power
Time	.102	.336	LS: 50.214 i*: 58.232	-	.372

Research question 4: Users' satisfaction

The user satisfaction survey followed the Method Evaluation Model, which defines three metrics: perceived ease of use (PEU), perceived usefulness (PU), and intention to use (IU). In Figure 7.4, we present a box plot of the three metrics.

The data analysis results for efficiency are shown in Table 7.12. For the four metrics, no significant differences were found ($p > .05$), and the statistical power is low ($pw < .39$). No significant differences were found for the treatment and treatment*problem interaction.

Table 7.12: Data analysis for user's satisfaction variables.

	Treatment	Interaction	Means	Effect Size	Power
PEU	.489	.489	LS: 21.500 i*: 20.500	-	.104
PU	.387	.154	LS: 34.548 i*: 31.705	-	.293
IU	.276	.198	LS: 8.012 i*: 7.250	-	.188

We did not find differences in perceived ease of use, usefulness, or intention to use. So, the null hypothesis H_{s0} cannot be rejected.

7.4 Design of the experiment replications

We performed two replications of LiteStrat's validation experiment in November 2020 and November 2021. Both replications share the same design but differ from the original experimental design. In the sections below, we present the experimental design concerning the baseline and the considerations for analysing the replications.

7.4.1 Changes in the experimental design

Table 7.13 summarises the differences between the baseline experiment and the two replications, following the template used by Panach et al. in (Panach, Dieste, et al., 2018). Below, we comment on the elements that were changed from the original design.

Operationalisation. Regarding the operationalisation of the experiment, no further changes are considered. The *factor* under study is the same (the modelling method), having two levels (i* and LiteStrat). However, the experimental problem, which was considered a blocking variable in the baseline experiment, is not included in this replication (see the Protocol differences). The *treatment definitions, transmission and instructions* were not changed, and the same training materials and experimental guidance were considered for the replications. The replications were also performed using pen and paper, so there were no changes in *treatment resources*. The *response variables*, *metrics*, and *measurement procedure* were also unchanged.

Population. Concerning the population, we decided to perform the comparison with more experienced and business-focused *subjects*, for which we

Table 7.13: Summary of differences across experiment designs

		Baseline	Replication 1	Replication 2
Operationalisation	**Factor**	Modelling method		
	Treatment Definition	i* and LiteStrat		
	Treatment Transmission	Real Estate Company		
	Treatment Instructions	1 hour of training + 1 hour of practice of i* and LiteStrat		
	Treatment Application Procedure	Subjects apply one of the treatments in 1 hour of application		
	Treatment Resources	Pen and paper (no tool support) for the two methods		
	Metrics	Total accuracy, motivation accuracy, action accuracy, outcome accuracy, total completeness, motivation completeness, action completeness, outcome completeness, time, perceived ease of use, perceived usefulness, intention to use		
	Response Variables	Accuracy, completeness, efficiency, satisfaction		
	Measurement Procedure	Semantic inspection of models, time and satisfaction questionnaire		
Population	**Subjects**	28 third-year computer science bachelor students	17 master in business process management and technology students	19 master in business process management and technology students
	Properties of Experimental Objects	Business strategy scenarios redacted according to business case studies.		
Protocol	**Experimental Objects**	Problem 1 (Telco), Problem 2 (Inssurance Company)	Problem 1 (Telco)	
	Experimental Design	2x2 factorial design	One factor with two treatments	
	Guides	i* 2.0 language + modelling procedure, LiteStrat language + modelling procedure		
	Measurement Instruments	Semantic inspection guidelines for i* and LiteStrat, Satisfaction Questionnaire		
	Data Analysis Techniques	GLM univariate with two factors (treatment and problem)	GLM univariate with one factor (treatment)	
Experimenters	**Designer**	Researchers from UPV, U Valencia		
	Trainer	Teacher from U Valparaíso	Teacher from U Granada	
	Monitor	Researcher from UPV		
	Measurer	Researchers from UPV, U Valencia		
	Analyst	Researchers from UPV, U Valencia		

recruited students from the Business Process Management and Technology Students at Universidad de Granada. For the first replication in November of 2020, 24 students were recruited, while 22 subjects participated in the second replication in November of 2021. The subjects were balanced and randomly assigned to the i* and LiteStrat groups in both cases. After checking the completion of the experimental tasks, 17 and 19 subjects were considered for the

2020 and 2021 replications, respectively. On the other hand, the *Properties of Experimental Objects* remain unchanged.

Protocol. As for the experimental protocol elements, we decided to change the design regarding the number of *experimental problems.* Since the results from the baseline experiment did not show significant effects of the problem or the interaction treatment*problem on the dependent variables, we decided to use just one of the problems (Problem 1, from the domain of Telecommunications Company). With this, the *experimental design* changed from 2x2 factorial design to one factor with two treatments design, according to Wohlin's classification (Wohlin et al., 2012). This also implied a change in the *data analysis techniques*, from a GLM univariate with a factor (the modelling method) and the blocking factor (the problem), to a GLM univariate with one factor (the modelling method). On the other hand, the experiment's *guides* and measurement instruments (including the semantic inspection guidelines detailed in the Appendix B) were also reused.

Experimenters. Finally, with respect to the experimenters, the roles of *designer*, *monitor*, *measurer*, and *analysts* were performed by the same researchers, while the *training* was performed by an instructor of the Universidad de Granada's master program.

7.4.2 Analysis of the replications

Since we opted for using just one experimental problem, some considerations regarding the analysis of the replication's collected data. In order to apply the univariate General Linear Model (GLM) analysis for each of the experimental variables, the assumptions of normality and variance homogeneity of the data are checked with the Shapiro-Wilk and Levene tests, respectively. If some variables fail to fulfil the requirements for a GLM analysis, we opt for a non-parametric approach, applying the Mann-Whitney-Wilcoxon test, the non-parametric alternative to compare two independent samples (Wohlin et al., 2012).

The results from both the GLM tests are considered significant when the p-value is less than .05. For the parametric approach, the effect size and statistical power are calculated in SPSS in a similar way to the baseline experiment, considering $\eta 2$ with values lower than .01 as a small effect, a value between .01 and .06 a moderate effect size, and higher than .14 a large effect size. As in the baseline experiment, we take into account the statistical power following the guidelines by Dyba et al. (Dybå, Kampenes, and Sjøberg, 2006), considering values greater than 0.39 as medium power and greater than 0.63 as large power. As for the non-parametric approach, the results of the Mann-Whitney-Wilcoxon test are significant when the p-value is less than .05. The effect size is calculated using R, in particular, the function `wilcox_effsize`[4] from the package `rstatix`, version 0.7.0. The effect size is considered small under .3, medium under .5 and large from .5 and on. The estimated power is calculated with the function `pwr.2p.test`[5] from the package `pwr`, version 1.3-0, using as a parameter the effect size previously calculated. The power is considered low below .5, moderate below .8 and high for values equal to or greater than .8.

Finally, we opted for transforming the scores of all the accuracy, completeness, and satisfaction metrics to percentages with respect to their maximum score (previously presented in Table 7.2 to simplify the comparison among metrics.

7.5 First replication results

A total of twenty-four volunteers initially committed their participation to the first replication. Three subjects did not submit the assignment. After an initial review of the submitted models, the data from other three subjects were discarded since the uploaded models were unreadable. The resulting sample comprises six subjects applying i*, and 11 subjects applying LiteStrat. The results for each research question are detailed in the following subsections.

7.5.1 Research question 1: Accuracy

In Figure 7.5, we present a box plot describing the data for the Total Accuracy (TA), Motivation Accuracy (MA), Action Accuracy (AA), Roles and Responsibility Accuracy (RRA), and Outcome Accuracy (OA) variables. As can be seen, except for AA, LiteStrat outperforms i* median and average values. Some outliers are reported for variables MA and AA, though we did not find problems in the data collection procedure that could justify the exclusion of these data points. Please note that for the TA variable, most of the LiteStrat models scored above the 50% of accuracy points, unlike i* models.

Regarding hypotheses testing, although the variance assumptions were confirmed for all the variables, the normality assumption for three metrics (MA, AA, and OA) was nor confirmed by the Shapiro-Wilk test. We opted for a non-parametric approach and applied the Mann-Whitney test and their respective

[4] https://www.rdocumentation.org/packages/rstatix/versions/0.7.0/topics/wilcox_effsize
[5] https://www.rdocumentation.org/packages/pwr/versions/1.3-0/topics/pwr.2p.test

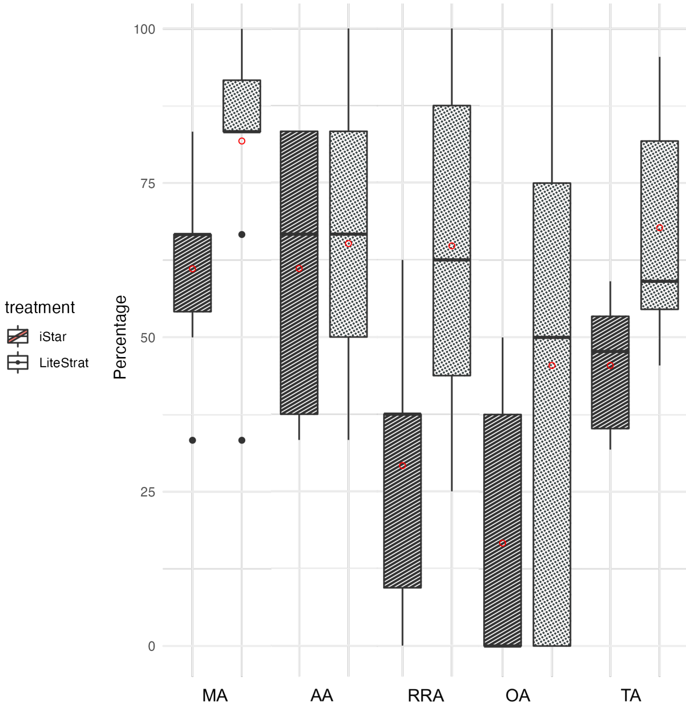


Figure 7.5: Box plot for the accuracy variables for the first replication.

effect size and power estimation procedures. The results of the statistical tests are shown in Table 7.14.

Significant differences were found for Total Accuracy (TA) favouring LiteStrat over i* ($p = .023$). The effect size is large (.564), and the power is low (.377). Regarding the metrics per type of statement, significant differences were also found for MA and RRA metrics ($p = .024$ and $p = .032$ respectively), both with large effect sizes (.561, .532) and low estimated power (.373).

From the results, we reject null hypothesis H_{a0} for the TA, MA and RRA metrics, meaning that LiteStrat outperformed i* in accuracy, mostly due to more accurate modelling of the elements that motivate the software development endeavour (MA) and the assignation of well-defined intentions to specific organisational roles (RRA). Given the low power and high effect size, these differences can have practical significance and are meaningful in the real world.

Table 7.14: Data analysis for accuracy variables for replication 1.

	Treatment	Means	Effect Size	Power
TA	****.023**	LS: 67.768 i*: 45.455	.564	.377
MA	****.024**	LS: 81.816 i*: 61.112	.561	.373
AA	.835	LS: 65.151 i*: 61.108	.063	.054
RRA	****.032**	LS: 64.773 i*: 29.167	.532	.342
OA	.173	LS: 45.455 i*: 16.667	.344	.171

7.5.2 Research question 2: Completeness

The box plots for the data regarding the Total Completeness (TC), Motivation Completeness (MC), Action Completeness (AA), Role and Responsibility Completeness (RRC) and Outcome Completeness (OC) metrics are presented in Figure 7.6. As shown, median and average values for LiteStrat outperform i* for all the metrics; however, most LiteStrat and i* models score over 50% of the completeness points. Some outliers were found for the MC, AC and TC metrics, though no reasons for excluding them were identified.

Since the normality assumption was not fulfilled for the MC, AC, and OC metrics according to the Shapiro-Wilk test, we opted for the non-parametric approach. The results for the Mann-Whitney test are presented in Table 7.15. Significant differences were found for the TC metric ($p = .003$), with a large effect size (.722) and a moderate statistical power (.557). For metrics grouping types of statements, significant differences were found for the RRC metric ($p = .042$) with a large effect size (.505) and a low power (.314).

Given the above results, we reject the null hypothesis H_{c0} for the TC and RRC metrics, meaning that LiteStrat outperformed i* in the completeness of the produced models, especially for modelling the assignment of well-defined needs to specific roles (RRC). These differences have practical significance and can be perceived in the real world.

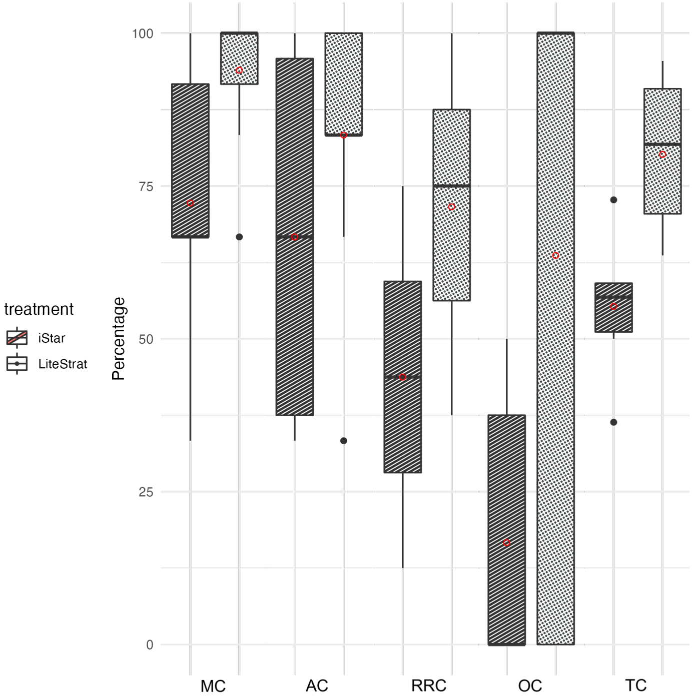


Figure 7.6: Box plot for the completeness variables for the first replication.

7.5.3 Research question 3: Efficiency

concerning the efficiency variable, the box plot for the time metric is shown in Figure 7.7. It is worth noting that higher values for time mean a worse performance. As can be seen, LiteStrat outperforms i*, having most of its values below 50 minutes.

Since the normality and homogeneity of variances assumptions are held for the time metric, we applied the parametric approach. The results are detailed in Table 7.16. No statistically significant differences were found for the time (p = .169), so we fail to reject the null hypothesis H_{e0}. Please note that the test's statistical power is low according to (Dybå, Kampenes, and Sjøberg, 2006) (.302).

Table 7.15: Data analysis for completeness variables for replication 1.

	Treatment	Means	Effect Size	Power
TC	****.003**	LS: 80.165 i*: 55.303	.722	.557
MC	.054	LS: 93.939 i*: 72.223	.482	.290
AC	.399	LS: 83.332 i*: 66.665	.217	.097
RRC	****.042**	LS: 71.591 i*: 43.75	.505	.314
OC	.068	LS: 63.636 i*: 16.667	.456	.264

Table 7.16: Data analysis for the efficiency variable for replication 1.

	Treatment	Means	Effect Size	Power
time	.169	LS: 48.455 i*: 54.667	.122	.302

7.5.4 Research question 4: Users' satisfaction

With respect to the satisfaction perceived by the participants, Figure 7.8 displays the box plot for the Perceived Ease of Use (PEU), Perceived Usefulness (PU) and Intention to Use (IU) metrics. As can be seen, though median and average values seem to favour LiteStrat over i* for PEU, average values for PEU and IU do not show a clear tendency towards any of the treatments.

Since the normality and variance homogeneity assumptions are verified for PEU, PU and IU, we applied the parametric comparison for the two samples. The results for the GLM analysis are shown in Table 7.17. No significant differences were found for the PEU, PU and IU metrics (p = .218, p = .950, p = .947). For the three metrics, the tests have low statistical power (.249, .050, and .051, respectively.). With the above results, we fail to reject the null Hypothesis H_{s0} for the three variables, meaning no differences were found.

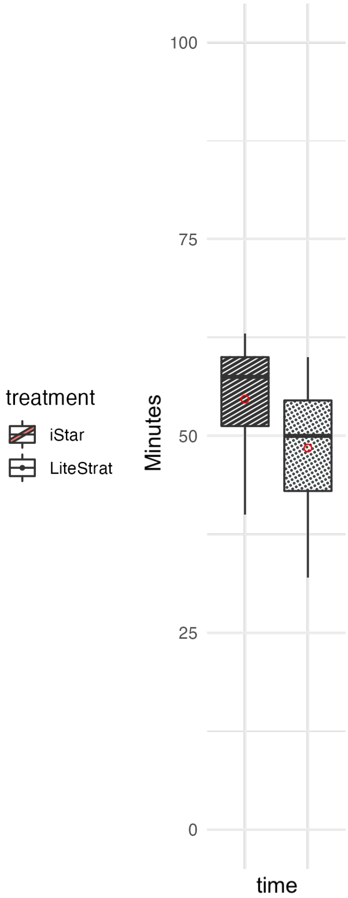


Figure 7.7: Box plot for the efficiency variable for the first replication.

7.6 Second replication results

A total of twenty-two volunteers started the training activities, and all of them completed the assignment. After reviewing the submitted models, the data from two subjects were discarded since the uploaded models were unreadable. The resulting sample is nine subjects applying i*, and 11 subjects applying LiteStrat. The results for each research question are detailed in the following subsections.

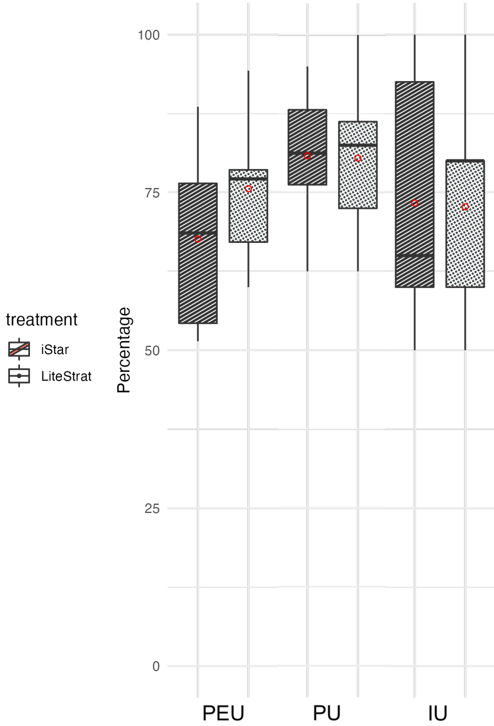


Figure 7.8: Box plot for the satisfaction variable for the first replication.

7.6.1 Research question 1: Accuracy

Figure 7.9 presents a box plot for the Total Accuracy (TA), Motivation Accuracy (MA), Action Accuracy (AA), Roles and Responsibility Accuracy (RRA), and Outcome Accuracy (OA) metrics. The median and average values for TA, MA, and RRA metrics seem to favour LiteStrat over i*. Unlike i*, most of the TA values scores for LiteStrat are over 50% of the total points.

The Shapiro-Wilk normality test rejected the normality assumption for metrics MA, AA, RRA and OA, so we opted for a non-parametric approach. The results for the Mann-Whitney tests and the effect size and power analysis are detailed in Table 7.18. Significant differences for the TA metric were found ($p < .005$), with a high large effect size (.676) and a medium power (.549). Similarly, MA showed significant differences ($p < .005$) with a high large effect size (.810) and a medium power (.704). No differences were found for the rest of the accuracy metrics.

Table 7.17: Data analysis for satisfaction variables for replication 1.

	Treatment	Means	Effect Size	Power
PEU	.218	LS: 75.584 i*: 67.62	.099	.249
PU	.950	LS: 80.455 i*: 80.833	.000	.050
IU	.947	LS: 72.727 i*: 73.333	.000	.051

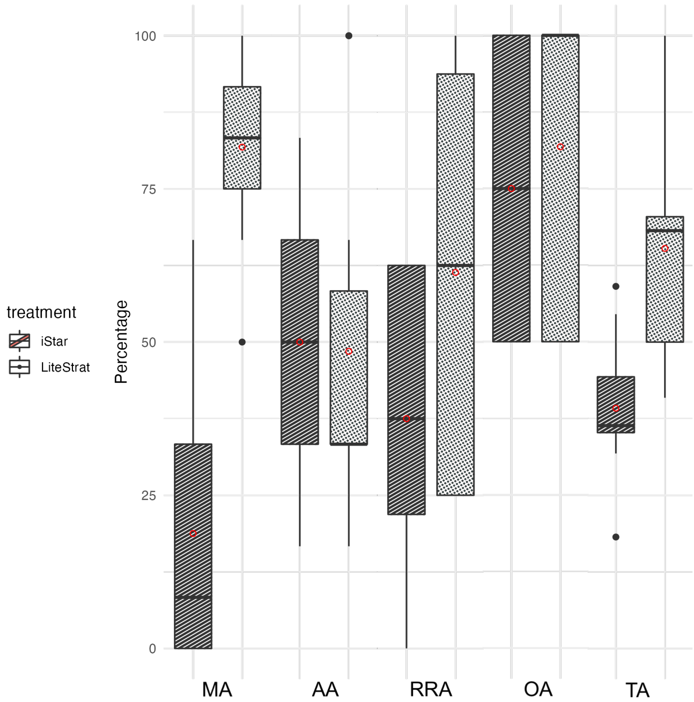


Figure 7.9: Box plot for the accuracy variables for the second replication.

Hence, the null hypothesis H_{a0} can be rejected for the TA and MA metrics. This means that the users of Litestrat produce more accurate models than the users of i*, especially regarding the strategic information that motivates the software development endeavour. The differences have practical significance and can be noticed in the real world.

Table 7.18: Data analysis for accuracy variables for replication 2.

	Treatment	Means	Effect Size	Power
TA	****.004**	LS: 65.288 i*: 39.204	.676	.549
MA	****.000**	LS: 81.817 i*: 18.75	.810	.704
AA	.703	LS: 48.485 i*: 50	.097	.060
RRA	.118	LS: 61.364 i*: 37.5	.368	.206
OA	.596	LS: 81.818 i*: 75	.133	.069

7.6.2 Research question 2: Completeness

As seen in the box plots in Figure 7.10, the average and median values for the TC and MC metrics seem to clearly favour LiteStrat over i*, while for RRC and OC metrics, the difference seems less important. It is worth noting that for the AC metric, the i* average value outperforms LiteStrat.

Since, according to the Shapiro-Wilk test, the normality assumption does not hold for the MC, AC, RRC and OC metrics, we applied the Mann-Whitney test and its corresponding effect size and power analysis procedures. The results are shown in Table 7.19. We found statistical significance for the differences in the TC metric ($p = .013$), with a large effect size (.581) and a low power (.433). Similarly, MC also presented significant differences ($p < .005$) with a large effect size (.846) and a medium power (.742). No statistically significant differences were found for the AC, RRC and OC metrics.

The above results allow us to reject the null hypothesis H_{c0} for the TC and MC metrics, meaning that subjects using LiteStrat produce more complete models than subjects using i*, mainly because they represent most information regarding the strategic motivation for the software development endeavour. The result has practical significance.

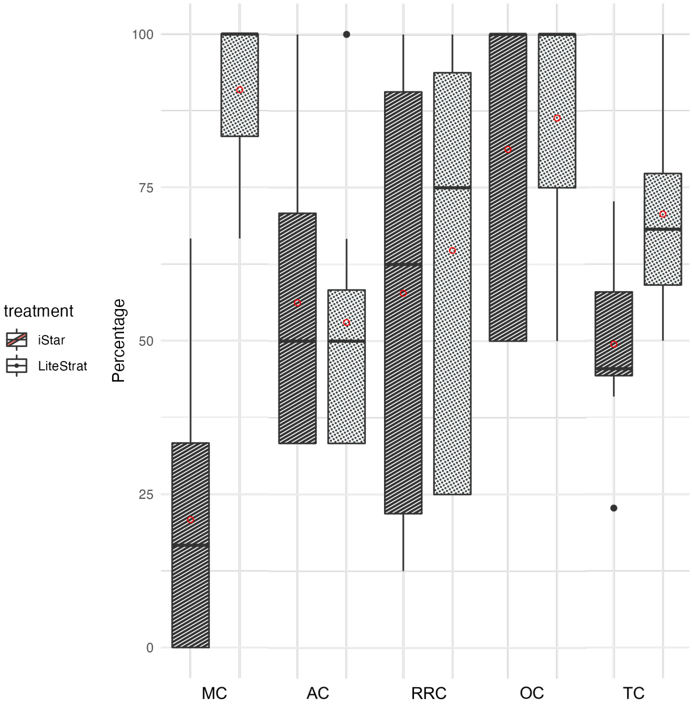


Figure 7.10: Box plot for the completeness variables for the second replication.

Table 7.19: Data analysis for completeness variables for replication 2.

	Treatment	Means	Effect Size	Power
TC	****.013**	LS: 70.661 i*: 49.431	.581	.433
MC	****.000**	LS: 90.91 i*: 20.833	.846	.742
AC	.761	LS: 53.029 i*: 56.249	.080	.057
RRC	.704	LS: 64.773 i*: 57.813	.097	.060
OC	.682	LS: 86.364 i*: 81.25	.106	.062

7.6.3 Research question 3: Efficiency

The box plot in Figure 7.11 describes the time metric, showing a better performance of the LiteStrat users (i.e., they complete the task in less time). An outlier is detected for the LiteStrat group, but we did not find reasons for excluding this data point from the analysis.

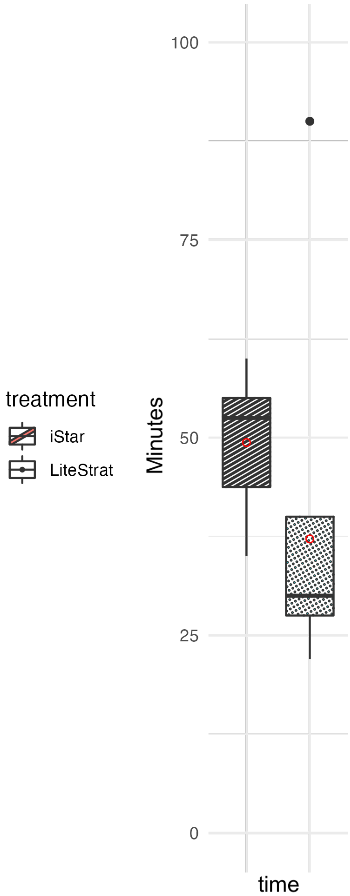


Figure 7.11: Box plot for the efficiency variable for the second replication.

Since the samples were not normally distributed according to the Shapiro-Wilk test, we applied the Mann-Whitney test. The results are presented Table 7.20. We found significant differences for the time metric ($p = .012$), with a large effect size (.583) and a low estimated statistical power (.435). With this result, we reject the null hypothesis H_{e0}, meaning that LiteStrat users were more efficient than i* users in the modelling task. This result has practical significance.

Table 7.20: Data analysis for the efficiency variable for replication 2.

	Treatment	Means	Effect Size	Power
time	****.012**	LS: 37.182 i*: 49.375	.583	.435

7.6.4 Research question 4: Users' satisfaction

The box plots for the PEU, PU, and IU metrics are shown in Figure 7.12. All three variables show better average and median values for the LiteStrat group; however, the differences do not seem relevant. It is worth noting that for both i* and LiteStrat methods, the score for the intention to use metric is beyond 75%.

The three metrics comply with the normality and homogeneity of variances assumptions, so we performed the parametric analysis, detailed in Table 7.21. Significant differences were found for the PEU metric ($p < .005$), with a large effect size (.407) and a high statistical power (.925). No significant differences were found for the rest of the metrics.

Table 7.21: Data analysis for satisfaction variables for replication 2.

	Treatment	Means	Effect Size	Power
PEU	****.000**	LS: 85.715 i*: 69.285	.407	.925
PU	.27	LS: 82.955 i*: 77.188	.071	.205
IU	.57	LS: 78.182 i*: 75	.019	.089

We reject the null hypothesis H_{s0} for the PEU metric, meaning that subjects perceive LiteStrat as easier to use than i*, for the specific task of modelling organisational information.

Table 7.22 summarises the p-values for all the variables and metrics across all the replications, highlighting significant differences and the favoured treatment.

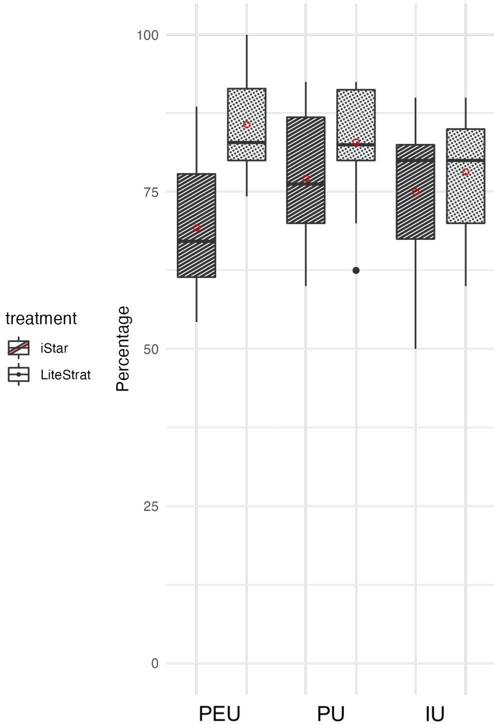


Figure 7.12: Box plot for the satisfaction variable for the second replication.

7.7 Aggregation results

In this section, we present the aggregated results of the baseline experiment with the two replications as a family of experiments. Following the approach by (Panach, Pastor, and Juristo, 2021), we consider the replication as a context variable, having the values 0 for the basement experiment, 1 for the first replication, and 2 for the second replication. Since the experimental problem was considered a blocking variable only for the baseline experiment, it is excluded from the design for the aggregation. Hence, the experimental groups are detailed in Table 7.23.

Similarly to the baseline, we performed a univariate General Linear Model (GLM) analysis for all the metrics to analyse the effect of the treatment and the combined effect of the treatment and the replication (treatment*replication).

Table 7.22: Summary of p-values and treatment for all the metrics accord the family of experiments (Tx: Treatment).

		Baseline		Replication 1		Replication 2	
Variable	**Met.**	**p**	**Tx**	**p**	**Tx**	**p**	**Tx**
Accuracy	TA	**.014	LiteStrat	**.023	LiteStrat	**.004	LiteStrat
Accuracy	MA	**.000	LiteStrat	**.024	LiteStrat	**.000	LiteStrat
Accuracy	AA	.117	LiteStrat	.835	LiteStrat	.703	i*
Accuracy	RRA	.051	LiteStrat	**.032	LiteStrat	.118	LiteStrat
Accuracy	OA	.636	i*	.173	LiteStrat	.596	LiteStrat
Completeness	TC	**.015	LiteStrat	**.003	LiteStrat	**.013	LiteStrat
Completeness	MC	**.000	LiteStrat	.054	LiteStrat	**.000	LiteStrat
Completeness	AC	**.046	LiteStrat	.399	LiteStrat	.761	i*
Completeness	RRC	.178	LiteStrat	**.042	LiteStrat	.704	LiteStrat
Completeness	OC	.372	i*	.068	LiteStrat	.682	LiteStrat
Efficiency	time	.102	LiteStrat	.169	LiteStrat	**.012	LiteStrat
Satisfaction	PEU	.489	LiteStrat	.218	LiteStrat	**.000	LiteStrat
Satisfaction	PU	.387	LiteStrat	.950	i*	.27	LiteStrat
Satisfaction	IU	.276	LiteStrat	.947	i*	.57	LiteStrat

Table 7.23: Experimental design and distribution of subjects for the aggregation of results.

	LiteStrat	**i***
Baseline (base)	13	15
First Replication (rep1)	11	6
Second Replication (rep2)	11	8

Consistently, the normality and variance homogeneity are tested with the Shapiro-Wilk and Levene's tests, respectively.

In the following subsections, we present the analysis results for the aggregated data.

7.7.1 Research question 1: Accuracy

In Figure 7.13, we describe the aggregated data for the metrics Total Accuracy (TA), Motivation Accuracy (MA), Action Accuracy (AA), Roles and Responsibility Accuracy (RRA), and Outcome Accuracy (OA). For MA, RRA and TA metrics, there is a tendency for LiteStrat to outperform i*, while AA and OA metrics show similar results for the two treatments.

The results of the GLM analysis are shown in Table 7.24. Significant results were found for the effect of the treatment for TA ($p < .005$), with a large effect

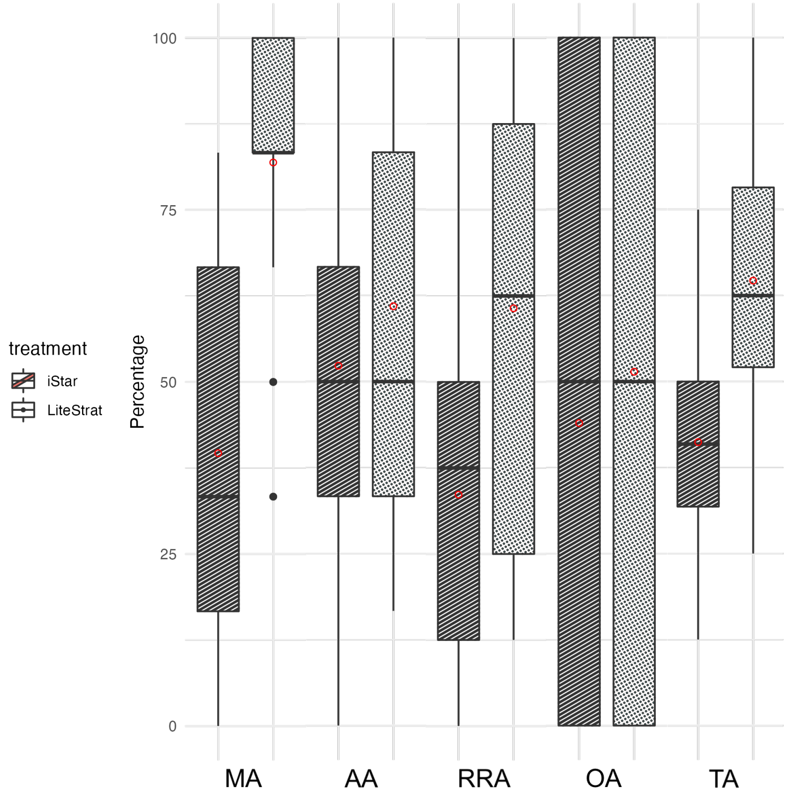


Figure 7.13: Box plot for the accuracy metrics for the family of experiments.

size (.285) and a high power (.997). Similarly, the MA and RRA metrics showed significant effects for the treatment ($p < .005$ and $p = .001$, respectively), with a large effect size (.464 and .172, respectively) and a high estimated statistical power (1.00 and .927) respectively. On the other hand, significant effects for the interaction Treatment*Replication were found for the MA metric ($p = .025$).

With the above results, *we reject the null hypothesis H_{a0} for the TA, MA, and RRA metrics.* This means the LiteStrat users produce more accurate models, particularly for representing the information regarding the strategic motivation for the software development endeavour (MA) and the assignment of specific responsibilities to organisational roles (RRA). Given its large effect size, these differences have practical significance, so they are noticeable in the real world. Regarding the representation of information about strategic actions and the

Table 7.24: Data analysis for accuracy variables for the family of experiments.

	Treatment	Treatment* Replication	Means	Effect Size	Power
TA	****.000**	.897	LS: 64.675 i*: 41.196	.285	.997
MA	****.000**	****.025**	LS: 81.903 i*: 39.655	.464	1.000
AA	.329	.283	LS: 60.952 i*: 52.298	.016	.163
RRA	****.001**	.890	LS: 60.714 i*: 33.621	.172	.927
OA	.323	.284	LS: 51.429 i*: 43.966	.017	.165

outcomes of the strategy (AA and OA metrics, respectively), we cannot reject the null hypothesis. However, the mean values also tend to favour LiteStrat over i*.

We did not find significant effects of the combination of the treatments and the replications, except for the metric MA. This result means that the treatment LiteStrat was more accurate in some replications than i*. As shown in the profile plot in Figure 7.14, Replication 2 presents the more significant differences between the LiteStrat and i*.

7.7.2 Research question 2: Completeness

Figure 7.15 shows the box plot for the Total Completeness (TC), Motivation Completeness (MC), Action Completeness (AC), Role and Responsibility Completeness (RRC), and Outcome Complexity (OC) metrics. As for the accuracy metrics, a tendency favouring LiteStrat over i* is shown for the TC, MC, and RRC metrics.

Table Table 7.25 shows the statistical test results for the completeness metrics. We found significant differences for the TC metric ($p < .005$), with a large effect size (.307) and a high power (.999). The MC and RRC metrics also showed significant differences ($p < .005$ and $p = .004$, respectively); for MC, the effect size is large (.501) and the estimated power high (1.000), while for RRC, the effect size is medium (.068) and the power, medium (.526). Significant effects for the combination Treatment*Replication were found for the MC and OC metrics ($p = .009$ and $p = .035$, respectively).

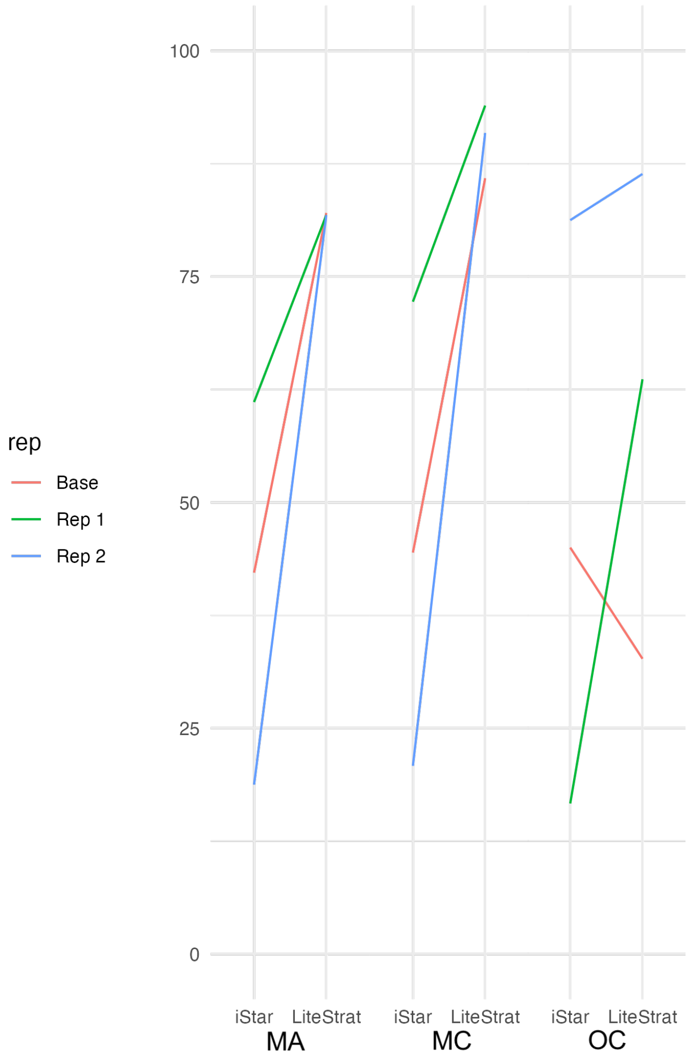


Figure 7.14: Profile plot for Motivation Accuracy (MA), Motivation Completeness (MC) and Outcome Completeness (OC) metrics for the Treatment*Replication effect.

The above results guide us to *reject the null hypothesis* H_{c0} *for the TC, MC, and RRC metrics*, meaning that LiteStrat users produced more complete models than i* users, mainly regarding the motivation behind the software develop-

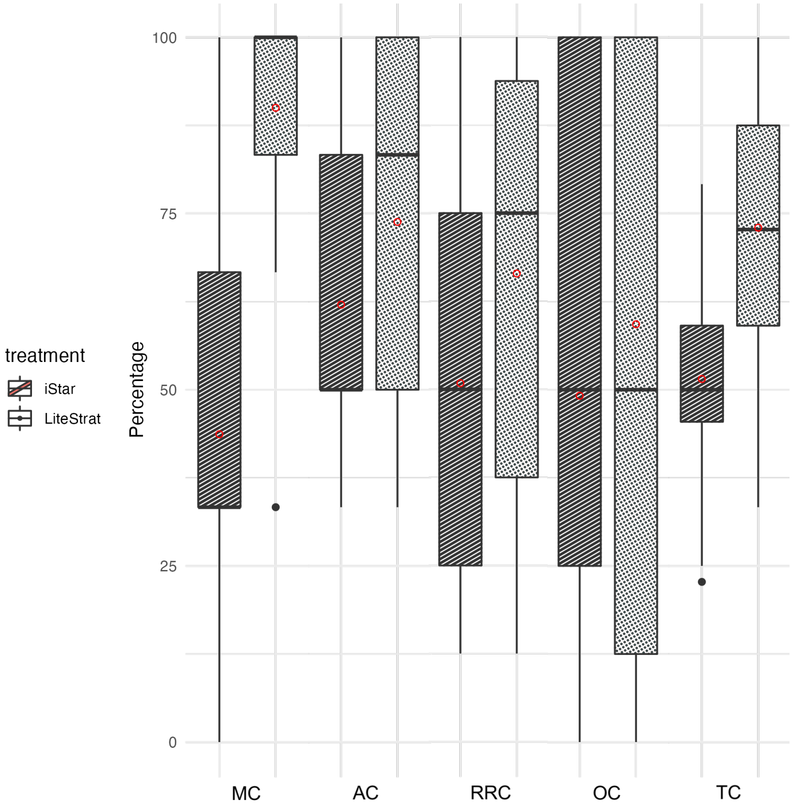


Figure 7.15: Box plot for the completeness metrics for the family of experiments.

ment endeavour (MC) and the representation of responsibilities and its assignment to organisational roles (RRC). We failed to reject the null hypothesis for the AC and OC, meaning that we did not find differences regarding modelling strategic actions and the outcome of the strategy. The differences in completeness have practical significance and can be noticed in the real world, particularly regarding the strategic motivation for the software development endeavour.

We also found significant effects of the combination of the treatments and the replications for the MC and OC metrics. This result means that in some replications, the treatment LiteStrat was more accurate for modelling the motivation and outcomes of the business strategy than in others. As shown in the profile plot in Figure 7.14, Similar to accuracy, the MC metric shows more

Table 7.25: Data analysis for completeness variables for the family of experiments.

	Treatment	Treatment* Replication	Means	Effect Size	Power
TC	****.000**	.782	LS: 72.997 i*: 51.515	.307	.999
MC	****.000**	****.009**	LS: 90 i*: 43.678	.501	1.000
AC	.076	.245	LS: 73.809 i*: 62.068	.053	.428
RRC	****.044**	.587	LS: 66.429 i*: 50.862	.068	.526
OC	.157	****.035**	LS: 59.286 i*: 49.138	.034	.292

pronounced differences between the treatments for Replication 2. On the other hand, the OC metric presents more important differences between the treatments in Replication 1.

7.7.3 Research Question 3: Efficiency

The box plot in Figure 7.16 suggests that LiteStrat users finished the modelling task in less time than i* users: most of them finished the task in under 50 minutes, while most of i* users finished after 50 minutes.

The results of the GLM analysis are presented in Table 7.26, showing significant differences ($p = .008$), with a medium effect size (.114) and a high estimated statistical power (.767). No effects of the combination Treatment*Replication were found. This means that LiteStrat users are more efficient than i* users, though the differences might be partially evident in the real world.

Table 7.26: Data analysis for the efficiency variables for the family of experiments.

	Treatment	Treatment* Replication	Means	Effect Size	Power
time	****.008**	.766	LS: 45.571 i*: 55.138	.114	.767

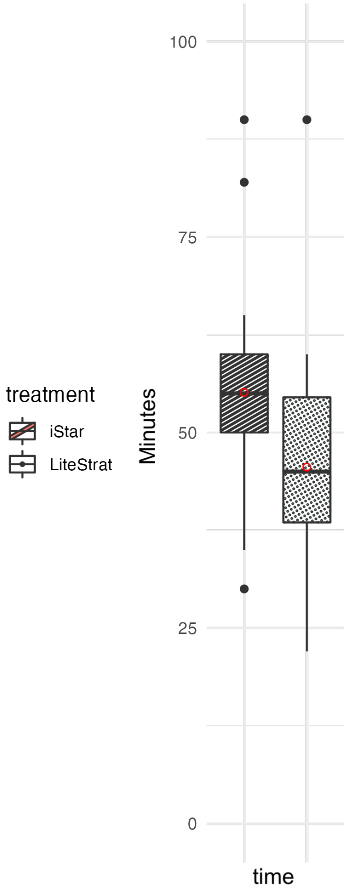


Figure 7.16: Box plot for the efficiency variable for the family of experiments.

7.7.4 Research question 4: Users' satisfaction

In Figure 7.17, we present the box plots for the Perceived Ease of Use (PEU), Perceived Usefulness (PU) and Intention to Use (IU) metrics. The median and average values show a tendency favouring LiteStrat over i*, for the three metrics.

The GLM analysis results, shown in Table 7.27, show significant differences for the PEU metric ($p = .002$) with a large effect size (.149) and a high power (.881). We were not able to find statistically significant differences for the PU and IU metrics.

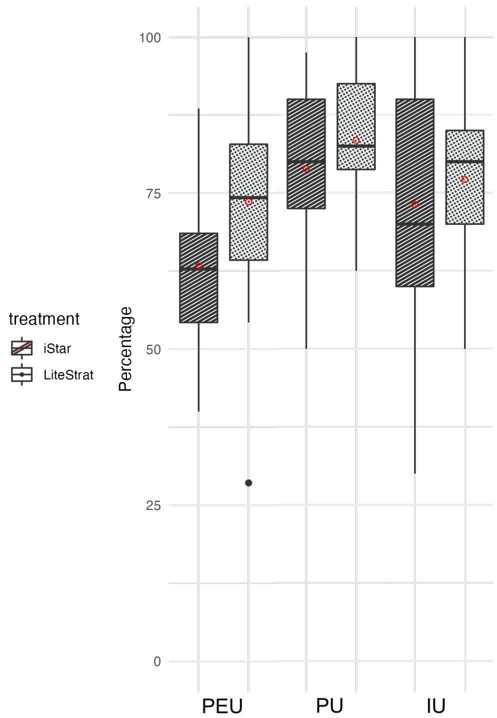


Figure 7.17: Box plot for the satisfaction variable for the family of experiments.

Table 7.27: Data analysis for satisfaction variables for the family of experiments.

	Treatment	Treatment* Replication	Means	Effect Size	Power
PEU	****.002**	.148	LS: 73.633 i*: 63.349	.149	.881
PU	.177	.583	LS: 83.429 i*: 78.879	.031	.269
IU	.417	.701	LS: 77.143 i*: 73.103	.011	.127

We reject the null hypothesis H_{s0} for the PEU metric, meaning that LiteStrat users perceived the method as easier to use with respect to i* users. Given the large effect size, these differences have practical significance and can be noticed in real-world settings.

7.8 Discussion

7.8.1 Overall results

The aggregated results show that LiteStrat produced more accurate and complete models than i* ($p = .004$ and $p = .013$ for Total Accuracy and Total Completeness metrics, respectively) for the specific purpose of modelling business strategy and organisational structure elements which are relevant for model-driven development. Furthermore, the accuracy and completeness differences favouring LiteStrat were statistically significant for the baseline experiment and the two replications.

Even though the definition of a structured modelling procedure could hinder modellers' efficiency and satisfaction, the results show that LiteStrat users required significantly less time than i* users to complete the modelling task ($p = .012$ for the time metric) and perceived LiteStrat as easier to use than i* ($p < .005$). Except for efficiency, all these differences have practical significance and might be observed in real-world settings.

While no significant effects of the combination of the modelling method and the replications (treatment*replication) were found for the main accuracy and completeness metrics and the satisfaction and efficiency metrics, significant interaction effects were found in metrics describing the accuracy and completeness of representing the business motivation behind the software development endeavour ($p = .025$ and $p = .009$ for Motivation Accuracy and Motivation Completeness, respectively) and for the complete representation of the outcomes of the business strategy ($p = .035$ for Outcome Complexity). Even though significant treatment*replication interactions must be discussed in further detail, they do not confound the modelling method's main effect on the organisational models' accuracy and completeness.

The overall results confirm the achievement of the design goals of LiteStrat (see chapter 5), which aimed to represent the organisational domain better, thus allowing more accurate and complete modelling. The accuracy and completeness results are consistent with the outcomes expected from using a situational method engineering approach (Henderson-Sellers and Ralyté, 2010) since LiteStrat is built upon i* concepts and adds organisational domain-specific constructs, helping to reduce the construct deficit (Rosemann, Green, and Indulska, 2004) of applying a general modelling framework as i* to a specific purpose.

As for the effects on user efficiency, LiteStrat's thorough modelling procedure could have required more time to be applied; however, LiteStrat outperformed i*, at least for the aggregated results. We believe that the baseline experiment and the replications individually did not show significant differences due to low statistical power, which has improved for the family of experiments (.767). Similarly, users' satisfaction regarding the perceived ease of use favoured LiteStrat just in the aggregated results due to the low statistical power of the baseline, and replications studied individually. We believe that LiteStrat's guided modelling procedure explains efficiency and satisfaction results since it helped users know "what to do" throughout the modelling task. On the other hand, i* users could have felt confused with its more free modelling approach, having no guidance to choose what construct to use, given its more general approach.

The efficiency and satisfaction results are consistent with the initial experiment by Abrahão et al. (Abrahão et al., 2018), which compared i* with value@GRL, a method for the specific purpose of modelling the value of the system to be developed. In this initial experiment, the researchers found significant differences in users' perception of ease of use and productivity. However, while LiteStrat effects on efficiency and ease of use are significant in the family of the experiments rather than in the individual experiments, value@GRL effects seem to be dimmed in the results of the family of experiments (Abrahao et al., 2019).

The above results pose LiteStrat as a valid method for modelling organisational information relevant to software development, suitable to be selected under the situation of organisations adopting model-driven approaches supporting organisational alignment for software development. Though one of the study limitations is the participation of undergraduate and master students and not software engineering professionals, this is a valid simplification for testing novel approaches in the research domain (Falessi et al., 2018). Another limitation regards the nature of the modelling problems; such textual descriptions of a strategic scenario could not represent real-world artefacts. However, LiteStrat adopters could use the method and modelling procedure to elicit the same organisational information from stakeholder meetings.

In order to study the differences between LiteStrat and i* in representing the organisational domain more in depth, the following subsections present the analysis of accuracy and completeness metrics regarding motivation, strategic actions, roles and responsibilities, and outcomes statements. We comment on the implications and limitations of the results for the overarching research aim, i.e., integrating organisational information into the model-driven development method.

7.8.2 Accuracy and completeness

Total accuracy (TA) and total completeness (TC) metrics show significant differences, favouring LiteStrat over i*. Since TA and TC are the sums of metrics focused on specific parts of the organisational domain, we discuss the results for the metrics composing TA and TC below, which showed significant differences.

Motivation accuracy and completeness metrics

Motivation Accuracy (MA) showed significant differences favouring LiteStrat ($p < .005$), with a large effect size (.464). Similarly, Motivation Completeness (MC) showed significant differences ($p < .005$) with a large effect size (.501) and high statistical power (1.00), also favouring LiteStrat over i*.

A first factor that could explain the results is the *influence* relationship that exists in LiteStrat but not in i*, as shown in Table 7.4. Although i* users were taught to use goal or task dependencies to connect external actors with the organisation, the construct seems to be not considered by the subjects. Since influence does not imply an intentional action from the external actor to the organisation, it could be more appropriate than the i* relationship (social dependency). For example, a competitor that offers a new product does not socially depend on the organisation under analysis; it is more accurate to model that the competitor influences the organisation. The influence relationship might reduce the construct deficit (Rosemann, Green, and Indulska, 2004) of i* and, therefore, improve the ontological completeness of LiteStrat over i*.

Role-Responsibility accuracy and completeness metrics

With significant differences favouring LiteStrat ($p = .001$) and large effect size (.172), the subjects more accurately modelled the assignment of responsibilities to roles with LiteStrat than with i*. Regarding completeness, Litestrat significantly outperformed i* ($p = .004$), with a medium effect size (.068) and moderate power (.526). Even though the difference between high-level and concrete ends was equally taught in training, LiteStrat users seem to have better support to model it. While in i*, subjects work with the goal construct, in LiteStrat, subjects work with goals and objectives, as detailed in Table 7.4. Having two different constructs could better support working with two levels of abstraction, which is consistent with the notion of construct deficit commented on above. Another possible factor is using LiteStrat's modelling guidelines, which guide users to connect objectives and roles (Noel, Panach, and Pastor, 2021b). Method engineering literature (Henderson-Sellers, Ralyté, et al., 2014; Brinkkemper, 1996) has widely supported the benefits of using guidelines.

Accurately modelling the assignment of responsibilities is of great interest in an MDD context since model transformation and alignment frameworks have exploited them as an integration point for goal and business process models (Ruiz, Costal, et al., 2015; Sousa and Prado Leite, 2014). Typically, the assignment of responsibilities is modelled as dependencies between actors, which are transformed into (parts of) business process models that realise the collaboration of such actors to fulfil the dependency. We explored the models produced by the subjects in the baseline experiment, looking for different representations of such statements. We found that i* users modelled these statements in six different ways, while LiteStrat users employed three different representations, as shown in Table 7.28. Figure 7.18 depicts the frequency of the different representations in LiteStrat and i*, using the A to F labels from Table 7.28. Although the more frequent representations in i* and LiteStrat could be helpful in an MDD context, when applying existing model-driven techniques, more information might be lost from i* models (even when they can be considered accurate). It is worth noting that in LiteStrat, only the first representation, "objective inside a role," is considered accurate.

In conclusion, the results suggest that LiteStrat models could better support users to more accurately model strategic elements that can serve as integration points with other models at the CIM level.

7.8.3 Summary

We compared LiteStrat and i* to look for differences in the accuracy (H_{a0}) and completeness (H_{c0}) of the models, and users' efficiency (H_{e0}) and satisfaction (H_{s0}). Considering that LiteStrat was specifically designed to represent business strategy, while i* has a broader scope, the expected results were more accurate and complete models when using LiteStrat. On the other hand, i* would perform better on users' efficiency and satisfaction given its non-restrictive modelling procedure.

Table 7.28: Different representations for roles and responsibilities in i* and LiteStrat. The subjects that not modelled roles or responsibilities are not included in the table.

i*					**Subjects**							
R&R Representation	19	27	28	29	1	3	6	8	11	16	25	26
A. goal refining a task	x				x						x	
B. goal dependency	x	x		x		x			x	x	x	x
C. softgoal qualifying a goal			x					x			x	
D. softgoal refining a goal							x					
E. softgoal qualifying a task										x	x	x
F. task											x	
LiteStrat					**Subjects**							
R&R Representation	2	4	5	7	9	10	12	14	15	18	23	24
A. objective inside role	x				x	x	x		x	x	x	
B. tactic inside role		x	x	x				x				
C. text inside role			x					x				x

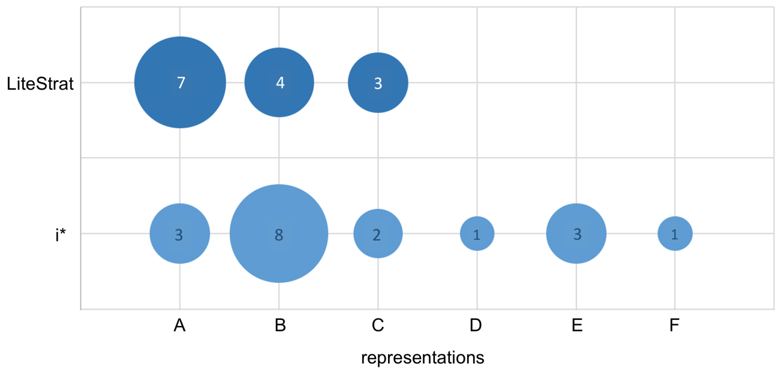


Figure 7.18: Number of subjects using different representations for role and responsibility assignment in LiteStrat and i*.

Results confirm the differences favouring LiteStrat in models' accuracy and completeness, whereas no differences in users' satisfaction and efficiency were found. We found no evidence of the effect of the experimental problems in any of the results. However, problems with different complexity or less targeted in business strategy (e.g., mixing system requirements and business strategy concepts) could produce different results, which would be a matter of further study in industrial contexts.

Regarding the results on accuracy and completeness, the solution examples in Figure B.2 in Appendix B could shed some light on the causes of LiteStrat our performing i*. As can be seen, the LiteStrat model seems to be simpler and more straightforward than the i* model, even though the two examples have the same number of domain elements, as detailed in Table B.1 and Table B.2 in Appendix B. We believe this is due to the languages' graphic representation of the refinements between goals and tasks and the participates-in links. Figure 7.19 illustrates these differences in a portion of a i* diagram (A) and a LiteStrat diagram (B). In LiteStrat, it is possible to have refinements of intentional elements across organisational units, such as "End Billing Errors" and "Reduce Billing Processing Time" in (B). At the same time, in i*, it is impossible to refine elements between actors for which a dependency link is needed, as in "Billing processing time reduced" (A). In LiteStrat, an organisation unit that participates in another is placed inside the parent organisation, and no link is needed, while in i*, the participates-in link is needed. This fact may make LiteStrat models simpler and easier to design and manipulate for the domain under study.

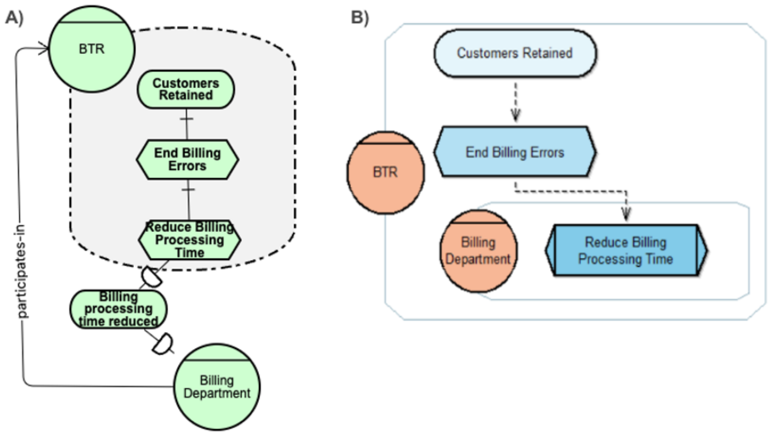


Figure 7.19: Examples of refinement and participates-in relationships modelled in A)i* and B) LiteStrat.

7.9 Threats to validity

Following Wohlin's guidelines (Wohlin et al., 2012), we present the analysis for the threats to experimental validity.

7.9.1 Conclusion validity threats

Conclusion Validity threats deal with the ability to draw the correct conclusions between the treatments and the experiment outcomes.

- Violated assumption of statistical tests: For each metric, we applied the Shapiro-Wilk test to check the normality of the residuals and the Levene test to determine variance homogeneity without finding any violations.
- Low statistical power: Concerning the baseline experiment, the sample size of the aggregated data (64 subjects) provided a high estimated statistical power for most of the metrics; however, there might be existing differences that we were not able to find, in particular Action Accuracy and Completeness (AA an AC metrics), Outcome accuracy and completeness (OA and OC) and Perceived Usefulness and Intention to Use (PU and IU). Further replications could help find statistical differences, as happened with the Perceived Ease of Use metric.
- Reliability of measures: We believe the measurement of the model's accuracy has an unavoidable level of subjectivity. To mitigate this as much as possible, we performed a detailed review of related works to address this issue. We decided to perform a semantic quality inspection like most experiments assessing modelling accuracy. The authors co-designed the grading scale, having in mind the modelling steps for both LiteStrat and i*. The solutions for the problems (see Appendix B) were proposed by the author and reviewed by the thesis co-director.
- Random heterogeneity of subjects: Working with undergraduate subjects with no industrial experience relevant to the experiment and with the same formation in conceptual modelling helped mitigate differences in experience that could impact the results.
- Random irrelevancies in the experimental setting: We designed the modelling task to be performed with paper and pencil to avoid technical differences in internet speed or power of the students' computers. This also helped us to mitigate the threat of the reliability of treatment implementation.

- Fishing: We mitigated the threat of searching for a specific result by defining a detailed protocol to rate the accuracy and completeness of the models. We also mitigated bias in the training process by asking a third party (the instructor of the requirements engineering course) to provide the training in LiteStrat and i*.

7.9.2 Internal validity threats

Internal validity threats deal with influences that can affect the independent variable without the researcher's knowledge, which are commented on below.

- Instrumentation: We used standard metrics and instruments to measure user satisfaction and effort in method experimentation: the Method Evaluation Model survey (Moody, 2003) and time. We based our design on relevant and well-documented experiments on assessing quality, presented in Section 7.2.2. The three authors participated and reviewed the grading scale and the solutions (see Appendix B).
- Selection: While working with undergraduate subjects might not represent the real population, it is considered a valid simplification in laboratory contexts for software engineering (Falessi et al., 2018).
- Interactions with selection: This threat was mitigated by randomly assigning the subjects to the experimental groups and the treatments and problems to each group.
- Diffusion or imitation of treatments: We explicitly asked the subjects not to share the training materials or knowledge. We did not identify imitations of treatments during the data analysis.

7.9.3 Construct validity threats

Construct validity threats deal with an experimental design that does not reflect the theoretical constructs under study and are commented on below.

- Inadequate preoperational explication of constructs: We carefully studied and selected the metrics and variables of other empirical studies used to assess the quality of the models, as presented in 7.2.2. We adapted it to the theoretical constructs as described in the experimental planning in Section 7.2.3.

- Mono-operation bias: This threat is related to confusing the treatment's effect with the problems' effect. We mitigated it by using two different experimental problems and verifying that the method and problem interaction did not affect the results. A possible threat regards the fact that the problems were specially designed for the experiment and were not taken from other third-party sources. We decided to opt for designing the problems to mitigate other relevant biases such as differences in complexity (that must be equivalent between the two problems), problem extension (we ensured that the problems could be modelled in the time slot given for executing the experiment), and that the relevant concepts for the MDD context (business strategy and organisational structure) were considered in the problem.

- Mono-method bias: This threat deals with having a unique measure for the effect of the treatments. We mitigated it by defining a set of metrics and a protocol to reduce the subjectivity in the scoring process.

- Restricted generalisability across constructs: This threat regards not studying other variables that can be negatively affected as a trade-off for the improvements of the new treatment. We mitigated it by assessing users' satisfaction and efficiency, finding no drawbacks at a statistically significant level.

Social threats to construct validity regarding the participants' behaviour during the experiment and their mitigation are commented on below.

- Hypothesis guessing: This concerns subjects changing their behaviour after guessing the tested hypothesis or being afraid of being evaluated. This was mitigated by not providing information about the experiment's goal or the relationship between the experimenters and the tested methods.

- Evaluation apprehension: We mitigated this threat by rewarding the subjects for participating in the training and the experiment, regardless of their performance.

- Experimenter expectancies: Since the final aim is finding out which of the methods serves better for representing business strategy for its inclusion in an MDA-based method, all the metrics were collected as objectively as possible, giving no space to the results to be biased by the expectancies of the experimenter.

7.9.4 External validity threats

Finally, external validity threats, which concern the ability to generalise the study's conclusions, have been managed in the ways commented on below.

- Interaction of selection and treatment: This threat deals with the representativeness of the experimental subjects concerning the population. We have selected undergraduate computing science students, which, as the experimental software engineering community has widely discussed (Falessi et al., 2018), is a valid simplification for testing and experimentally assessing novel techniques. However, other empirical studies are needed to transfer the technique to specific industrial settings, such as technical action research (Wieringa, 2014).

- Interaction of setting and treatment: This threat deals with the representativeness of the setting and objects of study. We have mitigated it by designing problems that, even though they are not real-world problems, are described similarly to strategic business cases in trending business magazines, such as Forbes.

- Interaction of history and treatment: The activity was performed on the same day and hour as the subjects usually have their lectures within the course content. No special events were identified previously or during the experimental activity.

7.10 Summary

Chapter 7: LiteStrat validation

- This chapter presents a family of experiments for addressing the knowledge question of whether LiteStrat improves the representation of relevant information for strategic alignment **(KQ3)**.
- We compared i* and LiteStrat in terms of the accuracy and completeness for modelling a strategic scenario and measured users' efficiency and satisfaction. We also studied how specific parts of the organisational domain are represented. The design is based on a targeted review of the literature on experimental comparisons of modelling methods.
- The aggregated results of the experiments showed statistically significant differences favouring LiteStrat over i* in the models' accuracy and completeness, especially regarding the representation of the business motivation and assigning responsibilities to roles. The differences have a large size effect, meaning that they can be noticeable in practice. Significant differences were also found, favouring LiteStrat for users' efficiency and perceived ease of use.
- We theorise that the results are explained by LiteStrat's more specific language, which helped users to more accurately represent the business strategy and organisational domain concepts. On the other hand, LiteStrat's guided modelling procedure could also explain the differences in user efficiency and satisfaction since i*'s freer approach could have hindered users' efficiency and perception of ease of use by not knowing what constructs to represent certain parts of the domain. Finally, the more straightforward approach of LiteStrat to represent relationships between actors and intentional elements could have also affected the overall results.
- The results confirm that the situational method engineering approach was successful in achieving the design requirements of LiteStrat, posing it as a valid alternative for including business strategy and organisational structure information in the baseline model-driven development method.

Chapter 8

Validation of Stra2Bis

8.1 Motivation

Similar to LiteStrat's validation, the claims on the benefits of Stra2Bis must be validated empirically. According to Section 1.5.3, the research question that guides this empirical cycle is *KQ4: Does Stra2Bis improve the design of strategically aligned business processes?* In particular, we look for improvements regarding the strategic alignment, this is, including information from organisational models with LiteStrat into the business process models.

As for LiteStrat's validation, the research method is Statistical Difference-Making Experiments (Wieringa, 2014) to establish causal relationships on the effects of the Stra2bis objectively. We also follow the guidelines by Wohlin et al. (Wohlin et al., 2012) for the experimental design. The design focuses on validating whether Stra2Bis better satisfies the requirements that drive its design, detailed in Chapter 6. In this case, since the baseline model-driven development method does not specify alternative methods for integrating LiteStrat information with the business process modelling level, we compare Stra2Bis with an unguided (or ad hoc) approach.

This chapter details the validation of Stra2Bis, addressing the knowledge question of whether Stra2Bis, the proposed method for designing strategically aligned business processes, improves the design of business processes **(KQ4)**. The rest of the chapter continues as follows: Section 8.2 describes the experimental design based on a targeted literature review on validations of methods for aligning organisational and business process models. Section 8.2.3 presents the experiment results, which are discussed in Section 8.3. The analysis of the threats to validity is detailed in Section 8.4. Finally, Section 8.5 presents the chapter summary.

8.2 Validation design

8.2.1 Research goal and scope

The goal of the experiment it to answer the question of whether Stra2Bis improves the design of strategically aligned business processes. In Definiton 8.1, we scope the experiment using the template provided by Basili and Rombach (Basili and Rombach, 1988).

Definition 8.1: Stra2Bis validation definition

Analyse *the Stra2Bis method*
for the purpose of *evaluation*
with respect to *strategic alignment of business processes*
from the point of view of the *researcher*
in the context of *bachelor students modelling re-designing business processes models.*

The *object of study* is Stra2Bis, and the *purpose* is to evaluate it, in this case, against an unguided or ad hoc approach. The results must be observed from the *point of view* of the researcher, which would be comparable to SME's method engineer role. The *experimental context* is given by the constraints of resources of the study, in this case undergraduate computer science students which is a valid simplification for novel approaches in software engineering (Falessi et al., 2018).

With respect to the *quality perspective*, we again focus on semantic quality (Lindland, Sindre, and Solvberg, 1994), this is, the quality of the produced business process models with respect to how well they represent both the business domain in terms of the business processes and in terms of the alignment of such processes with the organisational information. Besides semantic quality, we also evaluate the effects on user efficiency and satisfaction in order to identify undesired effects on these topics.

8.2.2 Related works on validation of methods for goal and business process alignment

Strategic alignment in model-driven methods has been widely studied. We focus our related works review on articles that could connect organisational and business process information by mapping elements from one level to another. Initiatives that consider goal modelling frameworks, particularly those that include social actors, are in our main interest since these constructs could help to represent organisational roles and units jointly with their goals. Hence, we review alignment works with three main concerns: whether the alignment methods are part of MDD processes, what information is mapped, and what empirical validations existent studies have. We summarise our findings in Table 8.1 and comment below.

Amyot et al. (Amyot et al., 2022) have summarised two decades of the User Requirements Notation (URN) Standard (ITU-T, 2018), which combines Use Case Maps (UCM) for process modelling and the Goal-Requirement Language (GRL) for goal modelling. The authors comment on their approach to aligning GRL and UCM models, which is achieved by defining consistency and completeness rules to check whether the goal model elements and the business process elements are traced. Although GRL would support representing organisational structure elements (such as organisation units and roles), the links between elements just state that there is a trace between elements without further semantics about the meaning of the trace. Gröner et al. (Gröner et al., 2014) proposed to combine GRL with the Business Process Model Notation (BPMN). The alignment approach is to automatically validate inconsistencies gven the propossed mappings; however, the analysis focuses on elements that are inner to the system instead of its organisational context. The proposal value@GRL by Insfran et al. (Insfrán et al., 2017) also combined GRL and BPMN to prioritise BPMN activities in incremental software development. By assigning values to the goal-level elements, the method helps to prioritise the business process elements through traceability rules. All intentional and social elements of the organisational level could be traced to process-level elements. The method also focuses on modelling the system's inner elements rather than organisational-level elements.

Other works have included i* (Yu, 2011a) for modelling social actors and their intentions. Sousa y Prado Leite (Sousa and Prado Leite, 2014) proposed GPI, which adds an intermediate modelling level to connect i* models with BPMN models. The alignment approach analyses whether the business processes have activities to produce critical resources (data) to measure key performance indicators on the organisation's goals. Although the method explicitly states that the organisational model should include the business as an actor with high-level goals, the business's organisational structure is not present in the method nor in the mappings to the business process level. The work by Ruiz et al. (Ruiz, Costal, et al., 2015) also considers i* for organisational modelling and, through a set of guidelines, map the social dependencies into interactions at the business process level. In this case, business processes are modelled using Communication Analysis (CA) (España, González, and Pastor, 2009), which focuses on the information exchange between the process actors instead of the actors' activities as in BPMN. The approach also focuses on dependencies but does not address organisational structure.

Another initiative is proposed by Guizzardi & Reis (R. Guizzardi and Reis, 2015), that combines the TROPOS language (Bresciani et al., 2004) and BPMN. The alignment approach assigns weights to goals and propagates them to the business process elements so analysts can reason about the impact of plans in achieving the goals. Similarly to other initiatives, though it is possible to model working groups or organisation units, this information is not considered in the alignment approach.

Besides the initiatives based on agent-oriented goal modelling frameworks, other proposals have considered other modelling approaches such as KAOS (Dardenne, Van Lamsweerde, and Fickas, 1993), and MAPs (Rolland, 2007). Nagel et al. (Nagel, Gerth, Engels, et al., 2013) proposed a constraint-based validation method to evaluate the consistency between the goals modelled in KAOS4SOA (a KAOS extension to consider the temporal dependencies among goals) (Nagel, Gerth, Post, et al., 2013) and the business process model; however KAOS4SOA does not specify elements that could support modelling the organisational structure. In (Kraiem et al., 2014), Kraiem et al. propose ten rules to map each MAP metamodel element to BPMN modelling elements. Though MAPs could be used for expressing organisational goals and strategies, the work shows MAP models with the same granularity as processes and, thus, not at the strategy level.

On the other hand, de la Vara et al. (Vara, Sánchez, and Pastor, 2008) proposed using MAPs to represent high-level organisational ends and means. The method first proposes to model the existing business processes; then, a MAP model representing the organisational strategies and goals is mapped to BPMN elements and describes the effect of the strategy in the business process elements (maintain or remove or create new elements). The resulting business processes are labelled, indicating whether the process tasks are automated by the information system, performed manually by users, or outside the system's scope. However, the method does not specify social actors, and no organisational structure elements are considered.

Three different validation methods were identified. Following Wieringa's classification of empirical validation methods (Wieringa, 2014), most of the studies performed single-case mechanism experiments (SCME) (Amyot et al., 2022; Insfrán et al., 2017; Kraiem et al., 2014; Nagel, Gerth, Engels, et al., 2013; Gröner et al., 2014), thus applying the method to examples or simulations. Two proposals were validated through technical action research (TAR) activities (Vara, Sánchez, and Pastor, 2008; Li, Zhou, et al., 2015), this is, applied their method in real-world problems; however, the validation method is not described in detail. Ruiz et al. (Ruiz, Costal, et al., 2015) and Guizzardi and Reis (R. Guizzardi and Reis, 2015) validated their proposals through controlled experiments with 19 and 15 subjects, respectively. The proposal was compared against the subject's criteria (or *ad hoc* approach) in both cases. Both studies focused on the validity of the alignments. However, Guizzardi and Reis calculated the ratio metrics (valid alignments per time, valid alignments per number of total alignments); Ruiz et al. measured the efficiency and completeness as separate variables. Regarding the work by Sousa et al. (Sousa and Prado Leite, 2014), the original article states that it was validated through a systematic approach, but no further information about the method and results was found.

As detailed in Table 8.1, just three initiatives are proposed in the context of MDD methods: the works by Guizzardi and Reis (R. Guizzardi and Reis, 2015), Gröner et al. (Gröner et al., 2014), and Li et al. (Li, Zhou, et al., 2015) refer to a process-oriented MDD methods, using BPMN, Business Process Definition (BPDM) (The Object Management Group, 2008), and Business Process Execution Language (WS-BPEL (OASIS, 2007)) at the CIM, PIM, and PSM levels of MDA, respectively. While these proposals combine goals and process modelling at the CIM level, the information of other organisational structure elements are traced to the PIM level.

In summary, from the above strategic alignment initiatives, we can conclude that no previous works mapped organisational structure and business strategy elements to the business process levels, and just two works presented statistical evidence of improving the default situation of subjects using their own criteria for strategic alignment. Regarding the experimental design, the two studies compare the proposals against ad hoc or unguided approaches and measure how well the alignments were performed (completeness + accuracy in (Ruiz, Costal, et al., 2014), effectiveness in (R. Guizzardi and Reis, 2015), users' efficiency, and in (Ruiz, Costal, et al., 2014), users' perceived ease of use, usefulness, and intention to use the proposal.

Table 8.1: Summary of related works for experimental comparisons of strategic alignment methods.

Article	Methods	Approach	MDD	Validation	Results	Limitations
Amyot et al., 2022	URN (GRL, UCM)	Validation rules to check typed links between modelling elements at different levels.	No	SCME (Example)	Show feasibility.	No semantics about the meaning of the traces are provided.
Gröner et al., 2014	GRL, BPMN	Validation rules between goals and activities in the business process models.	Yes	SCME (Simulation). Metric: performance.	Execution time is reasonable for realistic-size business process models.	Mappings are focused on inner system's elements rather than in organisational structure.
R. Guizzardi and Reis, 2015	TROPOS, BPMN	Assigns weights to goals and propagates to process activities to reason about the impact of plans in achieving goals.	Yes	Experiment, 15 subjects. Comparison: proposal vs ad hoc, Metrics: Efficiency, Effectiveness	Ad hoc outperforms the proposal in efficiency. Proposal outperforms in effectiveness.	No mappings on organisational structure (roles, organisation units) are specified.
Insfrán et al., 2017	GRL, BPMN	Assigns value to goals to prioritise BPMN activities for incremental software development.	No	SCME (Example)	Show feasibility.	Focus on modelling the system inner elements rather than in organisational level elements.
Kraiem et al., 2014	MAP, BPMN	Rules to map MAP elements to BPMN elements.	No	SCME (Example)	Show feasibility.	No organisational structure elements are mapped.
Li, Zhou, et al., 2015	GRL, UCM, BPMN	Rules to automatically check mappings from GRL to UCM elements and from UCM to BPMN elements.	Yes	TAR (Case Study)	Consistency and semantic compatibility were verified.	Need for an intermediate model which increases the complexity of the method.
Nagel, Gerth, Engels, et al., 2013	KAOS, KAOS4SOA	Constraint-based validation to check goals and business process consistency.	No	SCME (Example)	Show feasibility.	No organisational structure elements are mapped.
Ruiz, Costal, et al., 2015	i*, CA	Guidelines to transform intentional relationships between actors at the goal level into communicative interactions between actors at the business process level.	No	Experiment, 19 subjects. Comparison: ad hoc vs proposal. Metrics: Completeness, validity, efficiency, satisfaction.	Statistically significant improvements in completeness.	Some organisational structure elements are missing (organisation units and the belonging of roles to such units)
Sousa and Prado Leite, 2014	i*, BPMN, KPI	Introduces KPIs and an intermediate layer to specify roles' activities to achieve goals.	No	Systematic method for BPMN validations.	Show feasibility.	Organisation units information is not mapped to the business process level.
Vara, Sánchez, and Pastor, 2008	MAP, BPMN	Strategies and goals are mapped to BPMN elements to specify its effects in the business process elements.	No	TAR (Case Study)	Approach's feasibility demonstrated. Seniors questioned the improvements to their job.	No organisational structure elements are connected with the business process modelling level.

8.2.3 Experimental design

Research questions and hypothesis formulation

Considering the findings of the literature review, the experimental design is focused on studying how well the strategic alignment was performed. To approach to this inquiry, we refer the semantic quality of the produced business process models (Lindland, Sindre, and Solvberg, 1994), since we want to check whether the models are valid and complete representations of the domain. Hence, we must distinguish between the representation of two types of domain information: 1. business process models are consistent with organisational level domain information, this is, the models are aligned to business strategy, and 2. business process models actually represent the required business process logic. On the other hand, we want to assess whether the use of the S2B technique produces undesired effects on users' efficiency and satisfaction. We address these issues through the following research questions and hypotheses.

- **RQ1:** Is model quality affected by the transformation technique used? The null hypothesis is H_{q0}: *There are no differences in the quality of the models produced by the transformation of business strategy to business process models between the S2B technique and an ad hoc technique.*
- **RQ2:** Is user efficiency affected by the transformation technique used? The null hypothesis is H_{e0}: *There are no differences in user efficiency in the transformation of business strategy to business process models between the S2B technique and an ad hoc technique.*
- **RQ3:** Is user satisfaction affected by the transformation technique used? The null hypothesis is H_{s0}: *There are no differences in user satisfaction in the transformation of business strategy to business process models between the S2B technique and an ad hoc technique.*

Factors and treatments

To test the above hypotheses, the main factor is the *transformation approach*, with two treatments: an ad hoc approach, using the subject's own criteria for the model transformation, and the Stra2Bis technique, presented in Chapter 6.

Response variables and metrics

We will address H_{q0} using completeness and validity variables following Lindland's model (Lindland, Sindre, and Solvberg, 1994); the specific definition for the variables in the strategic alignment context is based on (Ruiz, Costal, et al., 2015). H_{e0} will be measured with the user efficiency variable, and H_{s0} with the user satisfaction variable following Moody's model (Moody, 2003). The definition of the four variables is detailed below.

- *Completeness*: The degree to which the resulting business process model contains a minimum set of elements that are relevant to represent the domain.
- *Validity*: The degree to which all the elements contained in the business process model should actually appear in the model in the right way.
- *User Efficiency*: The time needed by the experimental subject to design the re-engineered business process model.
- *User Satisfaction*: The perception of the experimental subject on the ease of use, usefulness, and intention to use the method applied to design the re-engineered business process model.

Given the importance of adding business strategy information to the business process models, we propose a subset of metrics that will allow us to observe the quality of business logic of the process and the business strategy elements separately. The metrics are defined below.

- *Process Completeness (PC):* Measures that the produced model completely preserves the business logic of the initial business process model regardless of whether it is re-engineered, except for those that must be removed for strategic definitions.
- *Strategy Completeness (SC):* Measures that the re-engineered business process model preserves information about strategic elements introduced in the strategic scenario.

- *Total Completeness (TC): Measures the total completeness of the produced model as the sum of the above metrics.*
- *Process Validity (PV):* Measures that the elements in the produced model are valid with respect to the source business process model except for those that are re-engineered, this is, no invalid modelling elements are introduced without explanation.
- *Strategy Validity (SV):* Measures that the new, updated, or deleted elements in the produced model are valid with respect to the definitions in the business strategy model.
- *Total Validity (TV):* Measures the total validity of the produced model as the sum of the above metrics.

To objectively measure the completeness and validity of the models, we designed grading schemes to qualify the models delivered by the subjects, which are detailed in Table C.1 and Table C.2 in the Appendix C. The researcher team peer-reviewed and agreed upon the grading schemes to avoid subjectivity in the evaluation of the models. The models produced by the experimental subjects are assessed by inspecting them and scoring them according to the grading schemes. One of the researchers scored the models, and another reviewed the scores.

Regarding the efficiency variable, we define the time metric as the time taken by a subject from the start to the end of the experimental activity. This metric is self-reported by the subjects; we also registered the starting time of the activity (common to all subjects) by automatically registering the time of submission of the form with the resulting model and verifying that the self-reported time was consistent.

Finally, for the user satisfaction variable, we define three metrics, following Moody's model: Perceived Ease of Use (PEU), Perceived Usefulness (PU) and Intention to Use (IU). These three variables are measured using the survey proposed in the evaluation model in a 5-point Likert scale (Moody, 2003).

In Table 8.2, we summarise the experiment hypotheses, the response variables, their metrics, and their ranges.

Table 8.2: Hypotheses, variables, and metrics of the experimental design.

Null Hypothesis	Variable	Metrics	Range
H_{q0}: There are no differences in the quality of the models produced by the transformation of business strategy to business process models between the S2B technique and an ad hoc technique.	Completeness	Process Completeness (PC)	0-6 points
		Strategy Completeness (SC)	0-6 points
		Total Completeness TC = PC + SC	0-12 points
	Validity	Process Validity (PV)	0-6 points
		Strategy Validity (SV)	0-6 points
		Total Validity TV = PV + SV	0-12 points
H_{e0}: There are no differences in user efficiency in the transformation of business strategy to business process models between the S2B technique and an ad hoc technique.	User Efficiency	Time	Minutes
H_{s0}: There are no differences in user satisfaction in the transformation of business strategy to business process models between the S2B technique and an ad hoc technique.	User Satisfaction	Perceived Usefulness (PU)	7-35 points
		Perceived Ease of Use (PEU)	6-30 points
		Intention to Use (IU)	2-10 points

Experimental problems

We designed two problems with the same structure, which is detailed below.

- **Current situation:** Briefly describes the current business processes of the organisation. It consists of a textual description and a communicative events diagram. The model's source file for the CA tool is also provided.
- **Strategic scenario:** Describes a strategic scenario where an external influence affects the organisational goals, so the organisation defines new goals and the strategies, tactics, and objectives to achieve those goals. Also, the organisational units and roles responsible for implementing the strategy are mentioned. The strategic scenario consists of a textual de-

scription and a LiteStrat model. The text also briefly describes changes to the business logic, which are not modelled in the LiteStrat model.

- **Instructions:** Provides information for the activity, such as the URL of the CA modelling tool and the URL of the submission form. In the second experimental task, the instructions were modified to ask the subjects to apply the Stra2Bis technique explicitly.

Problem 1, namely *TELEBANCO*, described the customer service hotline of an online bank that faces the growing dissatisfaction of their premium customers. Problem 2, namely *MARKETER*, describes the service request process of a market research agency that has to react to a new competitor with a better value offer. The problems were comparable (although not identical) in their size and complexity. Problem 1 had a current situation description of 199 words, and the business process model had five communicative events (one of them a specialised event) and five different actors. The strategic scenario of Problem 1 had 382 words, and the LiteStrat model had two actors, one goal, one strategy, three tactics, three objectives, three roles and two organisational units. Problem 2 had a current situation description of 127 words, and the business process model had four communicative events (one of them specialised) and four different actors. The strategic scenario of Problem 2 had 261 words, and the LiteStrat model had two actors, one goal, one strategy, two tactics, two objectives, two roles, and two organisational units. The two experimental problems are detailed in Appendix C and in the experimental package in (Noel, Ruiz, et al., 2022).

Experimental design

Because of the COVID-19 pandemic, the experimental setting was online and remote. The subject assisted in the training and experimental sessions via Zoom. We choose a within-subjects experimental design, i.e., the same subjects must perform the transformation using an ad hoc technique and then use the proposed technique. The differences between the techniques are measured as the differences of each subject in their application of both techniques.

We chose this design because both the ad hoc and the S2B transformations require a common background in modelling methods and process improvement, and background differences would affect the result of the transformations. This design also helps block the effect of other unwanted variables, such as the subject's experience and context conditions, that we could not control because of the experimental setting.

Since the proposed design requires that each subject applies the two transformation techniques, two different experimental problems are needed to avoid the carryover effect. To ensure that there is no effect of the problem in the result, two experimental groups are defined, which differ in the order sequence of problem-solving. The combination of the factors and the problem as the blocking variable yields the experimental design presented in Table 8.3. It is worth noting that, since knowing the S2B technique will bias the ad hoc transformation, the combination of subjects first applying S2B and then the ad hoc approach is not considered. Finally, the experimental design consists of a within-subjects factor (the transformation technique) that has two levels (*adhoc* and *S2B*) and a between-subjects factor (the order of problem-solving) that also has two levels (*P1-P2* for the group who first solved Problem 1 and then Problem 2, and *P2-P1* for the other group).

Table 8.3: Experimental design.

	Experimental Task 1	**Experimental Task 2**
Group P1-P2	ad hoc transformation + Problem 1	S2B transformation + Problem 2
Group P2-P1	ad hoc transformation + Problem 2	S2B transformation + Problem 1

Data analysis

We analysed all the response variables with a Two-Way Mixed ANOVA, using the General Linear Model (GLM) for repeated measures in SPSS version 25. This statistical test allows us to look for significant effects of the transformation technique, where each participant applies the two transformation techniques and the interaction with the order of problem-solving, where each participant belongs to one of two groups of problem-solving).

For each response variable, we assess whether there are significant differences explained by the effect of the transformation technique (*method*). We also check for interactions between the transformation technique and the problem order (*method*porder*). Finally, in order to verify that there are no effects that can be explained solely by the order or experimental problems, we assess the isolated effect of the problem order (*porder*).

There are five assumptions needed to apply the Two-way mixed ANOVA test. The assumptions and the means used for verifying them are listed below.

- **There should be no significant outliers in any cell of the design.** Outliers are visually inspected in an SPSS box plot that shows outliers (more than 1.5 box lengths away from the edge of their box) and extreme outliers (more than 3 box lengths away from the edge of their box). In case outliers are found, we examined ANOVA's residuals for values greater than ±3.

- **All the cells in the design must be normally distributed.** We will test the normality of the residuals generated by SPSS after running the GLM test.

- **The variance of your dependent variable should be equal between the groups of the between-subjects factor, referred to as the assumption of homogeneity of variances.** The homogeneity of variances is checked with Levene's test. P-value should be greater than 0.05 to verify the assumption.

  There should be homogeneity of covariances. This assumption is checked with Box's test. P-value must be higher than 0.001 to confirm the assumption.

The statistical test results are significant when the p-value is less than .05. The effect size and statistical power are calculated for those metrics with significant results to analyse the magnitude of the differences. The effect size is calculated using SPSS's partial $\eta 2$ generated by SPSS; a value lower than .01 is considered to be a small effect, a value between .01 and .06 is associated with a moderate effect size, and higher than .14 is considered a large effect size. The statistical power is the probability of rejecting the null hypothesis when it is false. Dyba et al. (Dybå, Kampenes, and Sjøberg, 2006) consider values greater than 0.39 for medium power and greater than 0.63 for a high power.

8.2.4 Experimental setting

Experimental subjects

The experimental subjects were fourth-year undergraduate students from the Informatics Engineering program at Universidad de Valparaíso in Chile, enrolled in a Requirements Engineering course. Thirty-two subjects participated in the training sessions and the first experimental task, while 21 completed the whole experimental procedure. The average age of the students is 22; six subjects were women, and 26 were men. None of the subjects had previously taken the Requirement Engineering course, and all of them passed the previous course on Software Engineering Fundamentals. None of the subjects had industrial experience in software engineering, except for those who went on an internship for two months. None of the subjects had conceptual modelling experience besides the Software Engineering Fundamentals course activities (class, state machine, and use case diagrams of UML), and none of the subjects used modelling tools at a professional level, while 25 declared having used modelling tools for academic and personal projects. The demographics are summarised in Figure 8.1.

While using students as experimental subjects might be a threat to the validity of the experiment, experts in empirical software engineering consider it a valid simplification for novel methods and techniques (Falessi et al., 2018). The subjects had to compulsorily participate in the teaching and training sessions, while their participation in the experimental tasks was elective. All the subjects that participated in the two experimental tasks were granted additional points for their grade on the Requirements Engineering course, regardless of their performance. The subjects are included in the experiment if they complete the two experimental tasks and all the teaching and training sessions. The subjects that fail the tests in sessions 1 and 2 must be excluded, as well as those taking more than two hours to solve any of the experimental tasks since they were designed to be completed in 60 minutes. Subjects must also be excluded in case the models they submitted are illegible screen captures and invalid CA tool model files.

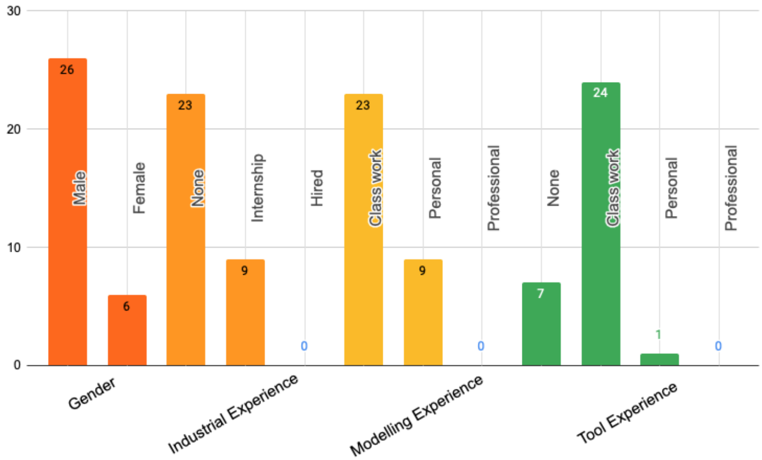


Figure 8.1: Demographics for the experimental subjects.

8.2.5 Experimental procedure

Because of the COVID-19 pandemic, the experimental setting was online and remote. The subjects participated in teaching, training, and doing experimental tasks sessions via Zoom. The teaching and training sessions were compulsory for the students as part of the first unit of a Requirements Engineering course. The experimental task sessions were not compulsory, but the students could get extra points for the assessment of the unit for their participation in the experimental tasks, regardless of their performance. The teaching, training, and experimental activities were designed to be performed during six sessions of 90 minutes each, in two sessions per week. Between the sessions, the students performed some offline tasks, mostly related to watching videos with content required for the training sessions.

The first two teaching sessions presented model-driven engineering concepts. The subjects were taught the modelling methods involved in the experiment: LiteStrat for organisational modelling and Communication Analysis (CA) for business process modelling. The subjects completed a demographics survey during these sessions and performed a test about reading LiteStrat models.

The third session covered the training in modelling business processes according to business strategy, following an ad hoc approach. Subjects applied the training task for which they received a CA model describing the current situation and a LiteStrat model describing a business strategy scenario that triggers a change in the current business process. A pilot of the submission process was performed, where subjects completed a form uploading a screen capture of the modelled business process and answered questions about their rationale for designing the model from the initial business process model and considering the business strategy model. The submitted models were assessed to verify the subject's understanding of the effect of the business strategy model in the redesign of business process models.

The first experimental task was executed in the fourth session using the ad hoc approach, thus repeating the procedure from the last training session. The subjects first signed the informed consent and then were randomly assigned to one of the two groups. The subjects received the experimental problem (a context description, a CA for the current business process model, and a LiteStrat model of the business strategy scenario) according to their group, the instructions for the activity, and the submission form URL. The submission form also included the survey to assess the perceived usefulness, ease of use and intention to use the ad hoc approach.

In the fifth session, subjects were instructed and trained in the Stra2Bis technique using the same exercise of the ad hoc training in session 3. The second experimental task was executed in the final session using the Stra2Bis technique. Again, the subjects were separated into their respective groups and were provided with a new experimental problem. The instructions and the submission form were the same as the first experimental task.

Part of the initial group of 32 subjects only participated in some of the training sessions. A total of 31 subjects participated in the first experimental task and 21 in the second experimental task. A summary of the experimental procedure and the number of participants that completed all the training activities are detailed in Figure 8.2. The experimental package, containing the training materials, forms, experimental problems, and collected data, can be found in (Noel, Ruiz, et al., 2022).

Figure 8.2: Experimental procedure.

8.2.6 Results

Next, we detail the data analysis results for each variable, providing a conclusion about the acceptance or rejection of their associated hypotheses. The data corresponds to the 21 subjects that completed the whole experimental procedure, except for four subjects discarded after the data collection procedure. In the data collection procedure, we discarded the observations of two subjects who took more than two hours to solve any of the tasks, one subject who submitted an invalid model file for the ad hoc experimental task, and one subject who failed the tests in session 2. Finally, data from 10 subjects of Group 1 (solving first Problem 1 and then Problem 2) and from 7 subjects of

Group 2 (solving first Problem 2 and then Problem 1) were considered for the analysis.

We checked the assumptions for all the variables. Next, we comment on how these violations were addressed. Regarding the no outliers assumption, the data presented outliers for some of the variables, as will be presented in the next subsections. However, we did not find outliers in the analysis of residuals for any of the variables. Regarding the normality of the cells in the design assumption, some of the cells violated these assumptions for the quality variables, both for completeness and validity. However, since Two-Way mixed ANOVA is robust to normality violations for small sample sizes ($n < 30$), we decided not to perform transformations for the sake of results clarity. The homogeneity of variances was checked using Levene's test based on the median, finding no violations for any of the metrics ($p > .05$), except for Total Validity (TV). Finally, regarding the homogeneity of covariances, we ran Box's test, finding no violations except for the Process Completeness Variable (PC), for which the test could not be calculated by SPSS since there were fewer than two non-singular cell covariance matrices.

With the above results, we decided to consider all the metrics and pay special attention when finding significant differences between the PC and TV variables.

Research question 1: Completeness

We statistically tested the metrics related to the completeness variable: Process Completeness (PC), Strategy Completeness (SC) and Total Completeness (TC). The box plots for each variable are presented in Figure 8.3, and the results for the statistical analysis are detailed in Table 8.4.

The results for the PC variable do not suggest significant differences between the problem-solving groups or for the different transformation techniques. Considering the maximum value for this metric (4 points), it seems that regardless of the problem-solving order and the transformation technique, the subjects produce models that moderately preserve the completeness of the business logic (2 to 3 points). We did not find any significant effects for the transformation technique ($p = .280$) or for the interaction of the techniques and the order of problem-solving ($p = .065$).

On the other hand, the data for the SC metric show higher results for the S2B technique in the two groups. The medians for the S2B approach were greater than half of the total points for the metric for the two groups, while just two

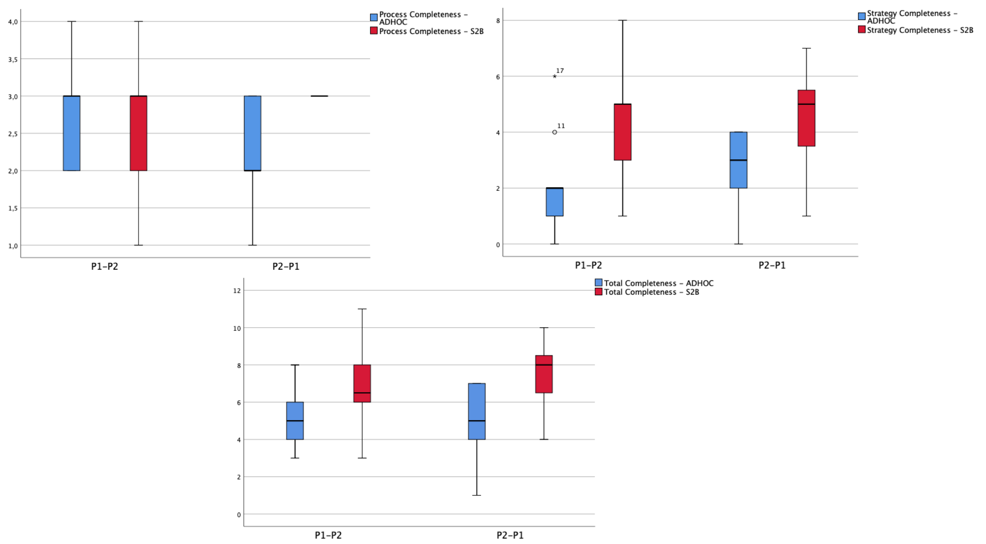


Figure 8.3: Box plot for the completeness metrics.

data points are above that value for the ad hoc approach. The results for the statistical test show significant differences in the effect of the transformation technique (p = .004), with a large effect size (partial $\eta 2$ = .433) and a high statistical power (op = .886). No significant interaction effects were found (p = 0.737).

Finally, the TC metric shows results that also favour the S2B approach in both problem-solving groups, with medians above the 8 points, from a maximum of 12. The medians of the ad hoc approach are below half of the total completeness points for the two groups. Statistically significant effects for the transformation technique were found favouring the S2B technique (p = .005), with a large effect size (partial $\eta 2$ = .424) and a high statistical power (op = .874). No significant interaction effects were found (p = 0.690).

Summarising, from the PC metric, we can state that the S2B technique did not improve the completeness of the business process models in terms of the core business process elements (actors and communicative events). However, for the SC metric, the S2B technique significantly improved the completeness of the representation of strategic elements, such as events reporting the state of the objectives, events reflecting the change of strategy envisioned in the strategic scenario, and adding new actors for representing roles supervising the correct implementation of the strategy. The TC metric shows an overall improvement of the business process model complexity explained by the S2B technique with respect to the ad hoc approach. So, we can reject the null hypothesis for the SC and the TV metrics.

Table 8.4: Data analysis results for the completeness metrics of the quality variable.

	Method	Effect Size	Observed Power	Interaction	Ad Hoc Means	S2B Means
PC	.280	.077	.183	.065	P1-P2: 2.80 P2-P1: 2.29 TOTAL: 2.59	P1-P2: 2.60 P2-P1: 3.00 TOTAL: 2.76
SC	****.004**	.433	.886	.737	P1-P2: 2.20 P2-P1: 2.71 TOTAL:2.41	P1-P2: 4.30 P2-P1: 4.43 TOTAL: 4.35
TC	****.005**	.424	.874	.690	P1-P2: 5.00 P2-P1: 5.00 TOTAL: 5.00	P1-P2: 6.90 P2-P1: 7.423 TOTAL: 7.12

Research question 2: Validity

We tested the three Validity variables: Process Validity (PV), Strategy Validity (SV), and Total Validity (TV). Figure 8.4 presents the boxplots for each variable. Table 8.5 shows the statistical results.

Table 8.5: Data analysis results for the validity metrics of the quality variable.

	Method	Effect Size	Observed Power	Interaction	Ad Hoc Means	S2B Means
PV	.511	.029	.097	.244	P1-P2: 3.20 P2-P1: 2.86 TOTAL: 3.06	P1-P2: 2.70 P2-P1: 3.00 TOTAL: 2.82
SV	****.006**	.411	.856	.065	P1-P2: 0.30 P2-P1: 1.00 TOTAL:0.59	P1-P2: 2.10 P2-P1: 1.43 TOTAL: 1.82
TV	****.021**	.307	.674	.296	P1-P2: 3.50 P2-P1: 3.86 TOTAL: 3.65	P1-P2: 4.90 P2-P1: 4.43 TOTAL: 4.71

For the PV metric, the boxplot shows no significant differences, rounding 3 points of validity score. This is confirmed by the statistical test (p = .511). The effect of the interaction is also non-significant (p = .244).

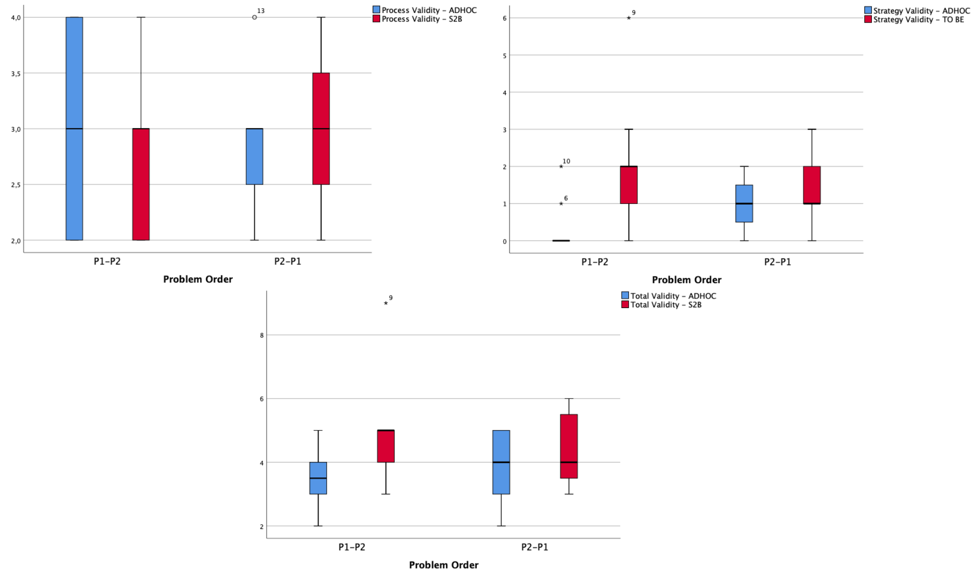


Figure 8.4: Box plot for the validity metrics.

For the SV metric, the boxplot shows a series of outliers for both transformation techniques for Group 1; however, we did not identify residuals greater than ± 3m and the outlier data points were verified as valid since they were not produced by an input or data collection error. The statistical analysis shows a significant effect of the transformation technique favouring S2B ($p = .006$), with a large effect size (partial $\eta 2 = .411$) and a high observed power ($op = .856$). No effects for the interaction were found ($p = .065$).

Finally, the boxplot for the TV variable shows a difference favouring the S2B technique for the two groups, which is confirmed by the statistical test with a significant difference ($p = .021$), with a large effect size (partial $\eta 2 = .307$ and a high observed power ($op = .674$). No significant differences in the problem order were found.

Considering the above results, the null hypothesis regarding differences with the transformation techniques can be rejected for the strategy validity and total validity variables but not for the process validity variable. This means the subjects produce equally valid models regarding the main business process logic changes, but subjects using S2B included more valid elements from the business strategy scenario.

8.2.7 Research question 3: User Efficiency

Figure 8.5 presents the box plot for the time metric. Please note that the time metric must be interpreted inversely: the higher the time, the less the efficiency. The box plot suggests that subjects that first solved Problem 1 required slightly more time to solve Problem 2 with the S2B technique, while subjects in Group 2 required more time to solve Problem 1 with the S2B approach. The statistical analysis results are shown in Table 8.6.

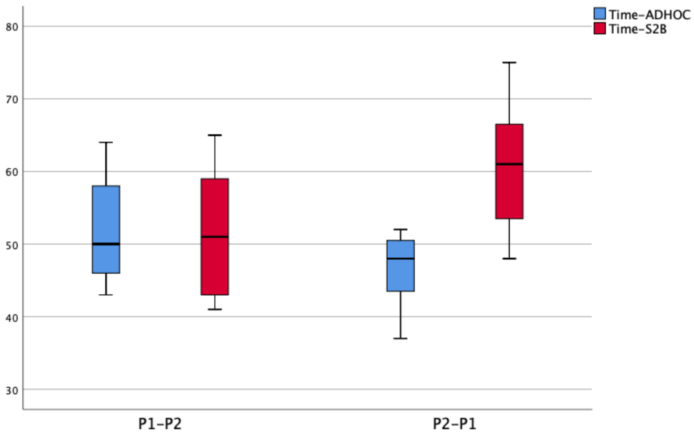


Figure 8.5: Box plot for the time metric.

Table 8.6: Data analysis results for the time variable.

	Method	**Effect Size**	**Observed Power**	**Interaction**	**Ad Hoc Means**	**S2B Means**
Time	****.016**	.329	.718	****.004**	P1-P2: 51.60 P2-P1: 46.43 TOTAL: 49.47	P1-P2: 50.00 P2-P1: 60.57 TOTAL: 54.35

The statistical analysis of the time metric showed significant effects of the transformation technique (p = 0.016) favouring the ad hoc technique. The effect size of the transformation technique is large (partial $\eta 2$ = .329), and the observed power is high (op = .718). The interaction between the transformation technique and the order of problem-solving was also significant (p = .004). The interaction effect size is also large (partial $\eta 2$ = .436), and the statistical power is high (op = .889). We found no significant effects of the problem order (p = .132).

Thus, we reject the null hypothesis H_{e0}, which means that the effort needed to complete the business process improvement task using S2B is significantly higher than the required for the ad hoc approach. Also, there was a statistically significant interaction between the transformation technique and problem-solving order. This means that the effort using the S2B technique with Problem 1 (mean = 60.57 minutes) is significantly higher than using the same technique for Problem 2 (mean = 50.00 minutes).

Research question 4: User satisfaction

We assessed the assumptions for the three metrics related to user satisfaction: Perceived Ease of Use (PEU), Perceived Usefulness (PU) and Intention to Use (IU). Figure 8.6 presents the box plot for the PEU, PU and IU metrics. The box plots do not suggest a consistent difference for the three metrics favouring one of the methods. Only PEU shows median values favouring ad hoc over S2B consistent in the two problem-solving groups (P1-P2 and P2-P1).

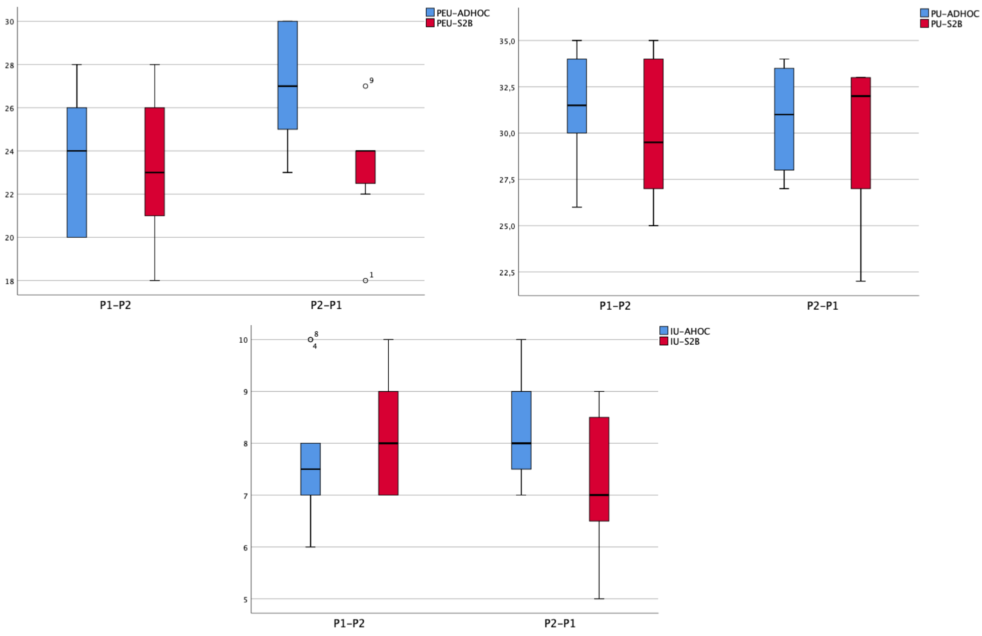


Figure 8.6: Box plot for the satisfaction metrics.

Table 8.7 shows the results of the statistical test. The analysis of the PEU metric showed significant effects of the transformation technique (p = 0.030) favouring the ad hoc technique, with a large effect size (partial $\eta 2$ = .277) and a medium power (op = .610). No significant effects of the transformation technique for the PU metric (p = .168). Also, no significant interaction effects were found (p = .924). Finally, we did not find significant effects of the transformation technique over the IU metric (p = .375). However, we found a significant effect of the interaction of the transformation technique and the order of problem-solving (p = .015), with a large effect size (partial $\eta 2$ = .334) and a high power (op = .727). The major differences are between the group that applied the S2B technique to Problem 2 (IU mean = 8.30) and the group that applied the same technique for Problem 1 (IU mean = 7.29).

Table 8.7: Data analysis results for the metrics of the user satisfaction variable.

	Method	Effect Size	Observed Power	Interaction	Ad Hoc Means	S2B Means
PEU	****.030**	.277	.610	.069	P1-P2: 23.60 P2-P1: 27.14 TOTAL: 25.06	P1-P2: 23.20 P2-P1: 23.14 TOTAL: 23.18
PU	.168	.123	.274	.924	P1-P2: 31.40 P2-P1: 30.71 TOTAL:31.12	P1-P2: 30.40 P2-P1: 29.57 TOTAL: 30.06
IU	.375	.835	.137	****.015**	P1-P2: 7.80 P2-P1: 8.29 TOTAL: 8.00	P1-P2: 8.30 P2-P1: 7.29 TOTAL: 7.88

We reject the null hypothesis H_{e0} for the perceived ease of use metric, and the since subjects gave a significantly higher score for the ad hoc approach. The null hypothesis cannot be rejected for the perceived usefulness of the method, while the intention to use is rejected since subjects have a higher intention to use the ad hoc method for solving Problem 1 in particular, according to the interaction effect.

8.3 Discussion

This section discusses the results' implications and possible causes for each dependent variable under the study.

8.3.1 H_{q0}: differences in models' quality

The results for the completeness and validity variables suggest that the null hypotheses for the quality of the produced models can be partially rejected. On the one hand, we cannot reject H_{q0} for the process completeness (PC) and process validity (PV) metrics ($p = .280$ and $.781$, respectively). As detailed in the grading scheme in Table C.1 and Table C.2 in the Appendix C, PC and PV variables capture the degree to which the re-engineered models preserve the core of the initial business process (i.e., have the same inputs, outputs, and at least one process element to produce the output) and the required changes in the business logic. We expected this result since S2B does not provide guidelines for designing the business logic; we think that the LiteStrat diagram that encompasses the textual description for the ad hoc and the S2B method could have helped all the subjects to better understand the changes in the business logic.

On the other hand, we reject H_{q0} for strategy completeness (SC) and strategy validity (SV) given the significant differences found ($p = .004$ and $.006$, respectively). This means that the models produced with S2B have a better quality regarding the preservation of strategic information in the business process model. In the context of an MDA-based development method, this is crucial for preserving functional and non-functional requirements from the CIM to the PIM level, as presented in Chapter 6. We think that this is produced by the transformation guidelines that explicitly address including business strategy information as elements of the business process. Since the effect size for SC and SV is medium, the differences have reasonably practical applications.

We reviewed the grading schemes for the strategic validity (SV) variable to explore how the guidelines helped improve the models' quality. Figure 8.7 compares the ad hoc and the S2B technique in terms of the percentage of subjects that have partially or totally modelled the elements defined by each guideline. For Guideline 1, thus, modelling and naming the business process' starting nodes for each organisation unit had the more significant improvements. There are fewer improvements for Guideline 2, i.e., the integration between different business processes. Notably, guideline 3 allowed that more than 20% of the subjects considered strategic objectives in the business processes models. We think that this could be related to the complexity of the guidelines. Guideline 1 is more straightforward than guidelines 2 and 3 since one element in the business strategy (organisation structure) is transformed into one element in the business process model (start node). In contrast, Guideline 2 produces six elements (two events, two actors, and two relationships in the business process model from a single relationship in the business strategy model). Consistently, Guideline 3 is at a medium point in complexity and improvements between Guidelines 1 and 2.

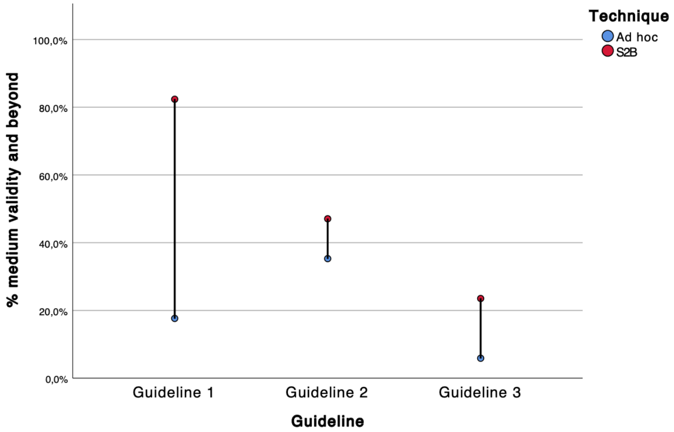


Figure 8.7: Improvements in Strategy Validity according to the grading scheme scores.

Compared to existing works, Guizzardi and Reis (R. Guizzardi and Reis, 2015) also found that subjects performed better for simpler constructs, and subjects highlighted that they would require more training to perform better. This could also be the case for the Stra2Bis guidelines, and more trained users could perform similarly by applying simple and more complex guidelines.

8.3.2 H_{e0}: differences in users' efficiency

The null hypothesis is rejected since subjects using an ad hoc approach required significantly less time than those using S2B ($p = .016$). We think this result is expected since ad hoc users did not need to review the guidelines' documentation or think in terms of source and target model elements. Considering the quality metrics' results, this could be seen as a trade-off situation where the S2B users would require more time to produce higher quality models, including business strategy information (SC and SV metrics). However, since it was the first time the subjects applied the guidelines, we believe this effect could be reduced by repeating the practice. Moreover, in an MDA context, it is crucial to ensure the quality of models, so it would be reasonable to invest in overcoming the technique's learning curve.

As commented before, in the experiment reported in (R. Guizzardi and Reis, 2015), some efficiency results favoured the ad hoc approach since subjects were quicker applying their own criteria than a systematic alignment approach.

We also found significant differences in the interaction between the transformation technique and the problem-solving order (p = .004). Subjects were more efficient in solving Problem 2 than Problem 1. We believe that this is caused by the fact that Problem 1 is slightly more complex than Problem 2; particularly, it has more transformable elements when applying the S2B guidelines. We think that this is also related to the effort needed to apply the guidelines for the first time, making the efficiency metric more sensitive to small changes in the problem's complexity.

8.3.3 H_{s0}: differences in users' satisfaction

The null hypothesis was partially rejected. In particular, it is rejected for the perceived ease of use (PEU) metric since subjects significantly perceived the ad hoc approach as easier to use than S2B (PEU's p = .030); no differences were detected regarding the perception of usefulness (PU's p = .168). We believe there is also a trade-off regarding the PEU metric, and subjects could feel less comfortable applying S2B to avoid losing strategic information. However, improving models' completeness and validity is crucial for model-driven methods, so applying the technique can be seen as an investment rather than a waste of effort.

On the other hand, intention to use (IU) showed a significant two-way effect (p = .015). The mean values suggest that both ad hoc and S2B users reported a higher intention to use for Problem 2, so this could be affected (though not significantly) by differences between the experimental problems. A possible explanation is that since the business strategy model for Problem 1 has more roles and objectives than Problem 2, the size of the task could affect the perception of the value of the transformation activity following the guidelines.

We believe that the above results could be caused by the fact that the experimental tasks are scoped just to model transformations at the CIM level, and hence, subjects cannot appreciate the results of their effort in the generated software or the system design. We believe that subjects' perceptions and intentions could change in full MDD environments.

Compared to existing works, in (Ruiz, Costal, et al., 2015) and (Domingo et al., 2020), authors report no significant differences in users' satisfaction, even though their model-driven proposals offered significant improvements. We believe this could be due to cultural aspects of software development and remarked by the fact that subjects cannot appreciate the improvements in the system model or the software code.

8.4 Threats to validity

In this section, we comment on the main threats to the experiment's validity, following Wohlin's guidelines (Wohlin et al., 2012).

Conclusion validity threats

Conclusion Validity threats deal with the relation between the treatment and its outcome. An important conclusion threat is *low statistical power*, which regards the ability to reveal a true pattern in the data. Although the tests with significant results have high observed power, a larger number of subjects could help to find differences for the test with no significant differences. However, it is difficult to commit a larger number of participants to a three-week process. It is worth noting that none of the experimental validations described in Section 8.2.3 had more than 20 subjects.

We mitigated the threat of *violated the assumption of statistical tests* by performing the test on the residuals to check for outliers and normality. We also performed Levene's and Box's tests to validate the assumptions. On the other hand, the *fishing* threat, which regards the influence of the researchers for finding specific results, was mitigated since the experimental protocol, materials, problems and data collection procedures were designed beforehand and reviewed by the authors. Similarly, the *reliability of measures* threat, concerning that the measures objectively describe the phenomenon, was mitigated by designing a grading scheme that all the authors reviewed to ensure its replicability.

Regarding the *unreliability of treatment implementation* threat, which deals with subjects performing according to their training, we ensured that all subjects similarly applied the treatment through the training sessions, working with examples and assessing their knowledge of the modelling languages and transformation guidelines.

We mitigated the threat of *random heterogeneity of subjects* by using a within-subjects approach for comparing the methods. We also surveyed subjects' backgrounds, verifying that subjects have little to no differences in modelling and professional experience.

8.4.1 Internal Validity Threats

Internal validity regards influences that can affect the causal relationship we want to observe in the experiment.

We think it might be a *history* threat related to applying the treatments in different moments since all subjects first applied the ad hoc approach and, the next week, the Stra2Bis approach. However, this design was necessary to mitigate the *maturation* threat, i.e., the effect of subjects learning through the experiment: learning and applying the Stra2Bis approach would bias the own criteria of subjects, thus making it impossible to ask them to apply an ad hoc approach.

Another source of internal validity threats is *instrumentation*, which regards biases in the instruments designed by the researchers to collect data; we mitigated it by asking the authors to review the data collection forms and the problems. We also based our assessment of perceived usefulness, ease of use, and intention to use on Moody's survey for the method evaluation model (Moody, 2003).

The *selection* threat refers to the variability between subjects. In this case, there is a threat since subjects are from a specific context (the same requirements engineering course), so further replications with subjects from other institutions could help to mitigate this issue. Regarding the *mortality* of subjects, which concerns affecting the results for subjects dropping out of the study, we mitigated it by not considering in the analysis data from subjects that did not participate in all the training and experimental activities.

8.4.2 Construct validity threats

The construct validity threats are related to generalising the experiment's results to the theory behind the experiment. We mitigated the *inadequate preoperational explication of constructs*, thus, an inadequate definition of the dependent variables and how to measure them by using grading schemes to assess the completeness and validity of the produced models.

Mono-operation bias refers to the underrepresentation of the treatments, and we mitigated it by providing two different problems. The *mono-method bias*, i.e., the under-representation of the experiment's outcomes, was mitigated by having multiple dependent variables and measurements for each variable. The *restricted generalizability across constructs* threat, related to affecting other constructs unintentionally, was mitigated by measuring model quality variables and the process efficiency and subjects' perception of the method.

8.4.3 External validity threats

Finally, external validity threats deal with limitations to the generalisation of the experimental results to real-world settings. The *interaction of selection and treatment* addresses not having subjects who are representative of the real-world. We think there is a threat since undergraduate computer science students might not represent professional business analysts. However, research on experimental software engineering considers this a valid simplification to test novel approaches (Falessi et al., 2018). *Interaction of setting and treatment* regards having a setting comparable to a real-world situation. We believe that using a business process modelling tool instead of pen and paper helped mitigate this risk.

8.5 Summary

Chapter 8: Stra2Bis validation

- This chapter presents an experiment to address the knowledge quesiton of whether Stra2Bis improves the design of business processes compared to an unguided or ad hoc approach **(KQ4)**. Through three model-to-model transformation guidelines, Stra2Bis ensures organisational level information, particularly business strategy and organisational structure, is considered in the design of business processes.
- We compared Stra2Bis and the ad hoc approach's validity and completeness of the produced business process models, regarding both the strategic alignment of the produced business process models and the representation of the business process domain information. We also measured users' efficiency and satisfaction. The design is based on a targeted literature review of experimental comparisons of modelling methods.
- The experimental results show statistically significant differences favouring Stra2Bis for strategic alignment in completeness and validity, while no differences in the representation of business process domain information were identified. On the other hand, the users' efficiency and satisfaction were negatively affected by the method, particularly in the perception of the method's ease of use.
- We theorise that the improvements are explained by Stra2Bis' explicit support to include business strategy information into business process models; since no specific guidelines on how to model business process logic are provided, no improvements on this item were found. Concerning the efficiency and satisfaction results, we think that must be considered an investment toward achieving greater quality using the existing model-driven development method.
- The results confirm that the situational method engineering approach successfully achieved the design requirements of Stra2Bis, and its claims on including organisational information for aligning business process models have been validated. In this way, Stra2Bis is a valid alternative for adding strategic alignment to the baseline model-driven development method.

Part IV

Conclusion

Chapter 9

Conclusions

9.1 Overview

This doctoral thesis presented an endeavour to include organisational information on business strategy and organisational structure into model-driven development, assembling a fully-fledged software production method from requirements to code. Within this research project, two artefacts, namely LiteStrat and Stra2Bis, have been developed, each intending to address specific challenges and advance the discipline's collective understanding.

Litestrat addresses organisational modelling and introduces innovative approaches to fill existing gaps in existing model-driven initiatives. Its development is rooted conceptually and methodologically in leading reference ontologies and modelling frameworks. It is guided by the commitment to improving the current state of including business information in model-driven development methods. Through a series of experimental validations, we showed the accuracy and efficiency of LiteStrat for organisational modelling with respect to existing goal-oriented approaches; we hope future applications to real-world problems shed light on its potential contributions to the broader academic community.

In tandem, Stra2Bis has been designed to complement the existing baseline model-driven development framework by integrating LiteStrat's organisational models with business process models, aiming to design modular processes and information systems. Developed in response to the challenges presented by software organisations for drafting the software architecture from the organisational structure and business strategy, this artefact opens new avenues for exploration in model-driven development methods.

Together, the two artefacts contribute to the evolution of the model-driven development field by including requirements from organisational information. By expanding the baseline model-driven software production method and offering practical applications, they seek to provide modest yet meaningful additions to the existing body of knowledge. Next, in Section 9.2, we detail the contributions of this thesis in terms of the knowledge and design goals achieved through the research. In Section 9.3, the impact of the thesis is reviewed in terms of the scientific publications in the specific context of the thesis. In Section 9.4, the main academic projects and outreach activities are described. Section 9.5, presents results of other side research endeavours that have been made possible thanks to the methodological and disciplinary background achieved through the PhD program. Finally, in Section 9.7, we outline the future work expanding the results of this thesis.

9.2 Contributions

Contribution 1: A Conceptual framework for including organisational information into an MDD method (Chapter 3).

Based on sound information systems theory (Steven Alter, 2013) and reference ontology frameworks (G. Guizzardi, Botti Benevides, et al., 2022), we proposed a conceptual framework for describing the organisational context relevant to eliciting requirements for designing information systems in a modular, decoupled approach. Taking into account the environment, infrastructure, and strategy elements of the organisational domain, we propose three conceptual frameworks for describing elements relevant to eliciting requirements for the information system.

Contribution 2: Identification of challenges for including organisational information into an MDD (Chapter 4).

We analysed a reference method for aligning goals and processes to identify the challenges for including organisational information into the baseline model-driven development method, considering the proposed domain conceptualisation. Using the single-case mechanism experiment as a method for abductive reasoning, we identified challenges regarding the preservation of the organisational structure to the business process level, coupling between unrelated business processes when defining strategic, cross-organisational goals, and overlapping between organisational and business level goals and actions. The findings were discussed and reviewed by experts.

Contribution 3: LiteStrat, an organisational modelling method (Chapter 5).

Using the situational method engineering approach (Henderson-Sellers, Ralyté, et al., 2014), we designed LiteStrat as a modelling method encompassing an organisational modelling language and procedure. We designed a modelling language to address the domain conceptualisation proposed in Chapter 3 by selecting and assembling method chunks from existing goal and enterprise architecture frameworks. The modelling procedure is scoped to the initiating stage of the software development life cycle, similar to goal and agent-oriented methods, where early requirements from the organisational domain are collected.

The modelling method focuses on identifying influences from the organisation's environment that drive the software development endeavour, the organisational strategy to address such influence in terms of ends and means, the organisational structure relevant to deploying the strategy, and the organisation's reaction to create a new value offer towards its environment.

Contribution 4: Stra2Bis: A modelling method for including organisational information into an MDD method (Chapter 6).

Building upon LiteStrat's capabilities for representing organisational information, we designed Stra2Bis, a method to integrate organisational information in the business process design of the baseline MDD method. Stra2Bis helps to collect requirements from organisational models to design modular, loosely coupled business processes that provide information on the performance of the organisational strategy.

Stra2Bis was designed using the situational method engineering approach to assemble LiteStrat and the business modelling method of the baseline MDD method, Communication Analysis. This way, requirements from the organisational level are considered for designing independent business processes for independent organisation units, defining their integration points and activities to measure the organisational strategy performance. The expected effect of this integration on the software modelling levels is to have independent software models that generate a loosely coupled system architecture that resembles the organisational structure.

Contribution 5: Empirical evidence on the effects of LiteStrat (Chapter 7).

We contributed empirical evidence of the accuracy and completeness improvements introduced by LiteStrat for describing the organisational domain when compared with a reference goal modelling framework as i*. Through a family of experiments, 28 undergraduate and 36 master's degree subjects participated in a baseline experiment and two replications. The experiments consistently favoured LiteStrat over i* in accuracy, completeness, efficiency, and users' perception of usefulness and ease of use.

Contribution 6: Empirical evidence on the improvements introduced by Stra2Bis (Chapter 8).

We assessed the improvements of Stra2Bis over an ad hoc approach for considering organisational information in the design of business processes. A total of 21 subjects completed the four-week experimental procedure, which yielded significant results favouring LiteStrat in the completeness and validity of the designed business processes. Nonetheless, a trade-off in efficiency and user satisfaction is needed to achieve these results, which is justified under an MDD approach.

Contribution 7: Tool Support (Chapter 5, Chapter 6).

We implemented LiteStrat and Stra2Bis using the ADOxx metamodelling framework, contributing to the digital innovation environment of the OMI-LAB organisation. The implementation of LiteStrat, described in Chapter 5 required the adaption of LiteStrat conceptual metamodel to the ADOxx framework metamodel, which helped to reason about and refine the design of LiteStrat itself, as well as its simplification for improving the usability of the method. Implementing Stra2Bis, on the other hand, required the design of a reduced version of the Communication Analysis method to assess the automatic transformation from organisational to business process models.

The current implementation of the two methods is consolidated in a single component, covering two out of the three methods of the baseline MDD method. Since the ADOxx framework provides seamless integration with other platforms through web services and REST APIs, it is possible to envisage integration with INTEGRANOVA, the OO-method's supporting tool, in the short term.

9.3 Thesis Impact

9.3.1 Publications

Journal Papers

1. *Noel, R., Panach, J. I., & Pastor, O.*

   Including business strategy in model-driven methods: an experiment. Requirements Engineering, Pages 1-30. 2023. Springer. Impact factor 2.8. H Index: 61. Scopus (SJR) Q2. JCR Q2.

2. *Noel, R., Panach, J.I. & Pastor, O.*

   Challenges for model-driven development of strategically aligned information systems. IEEE Access, volume 10, Pages 38237-38253. 2022. IEEE Inc. Impact factor 3.9. H Index: 204. Scopus (SJR) Q1. JCR Q2.

3. *Noel, R., Ruiz, M., Panach, J.I. & Pastor, O.*

   From Business Strategy to Business Processes: Empirical Validation. *Under review in Information Systems journal.*

Conference Papers

1. *Velasquez, R., Negri-Ribalta, C., Noel, R., & Pastor, O.*

   Exploring Understandability in Socio-technical Models for Data Protection Analysis: Results from a Focus Group. In: Sales, T.P., Araújo, J., Borbinha, J., Guizzardi, G. (eds) Advances in Conceptual Modeling. ER 2023 Workshops. Publication: Print ISBN 978-3-031-47111-7 Online ISBN 978-3-031-47112-4 Volume 14319, 2023. Pages 263–273. Lisbon, Portugal, November 3-9, 2023.

2. *Negri-Ribalta, C., Noel, R., Pastor, O., & Salinesi, C.*

   An Empirical Study on Socio-technical Modeling for Interdisciplinary Privacy Requirements. In Proceedings of the 29th International Conference on Cooperative Information Systems (COOPIS 2023). Core index: B. Publication: Print ISBN 978-3-031-46845-2 Online ISBN 978-3-031-46846-9 Volume 14353, 2023. Pages 137-156. Groningen, The Netherlands, October 30 - November 3, 2023.

3. *Noel, R., Panach, J. I., & Pastor, O.*

   Using Team Topologies in Model-Driven Strategic Alignment. In Anais do XXVI Congresso Ibero-Americano em Engenharia de Software (CIbSE 2023). Pages 168-175. Montevideo, Uruguay, April 24-28, 2023.

4. *Noel, R., Velasquez, R., Panach, J. I., & Pastor, O.*

   Agile development of Systems from REquirements to Code (SREC). In Proceedings of the Research Projects Exhibition Papers Presented at the 35th International Conference on Advanced Information Systems Engineering (CAiSE 2023). Pages 83-90. Zaragoza, Spain, June 12-16, 2023.

5. *Noel, R., Panach, J. I., Ruiz, M., & Pastor, O.*

   Stra2Bis: A Model-Driven Method for Aligning Business Strategy and Business Processes. In: Ralyté, J., Chakravarthy, S., Mohania, M., Jeusfeld, M.A., Karlapalem, K. (eds) Conceptual Modeling. ER 2022. Core index: A. Publication: Print ISBN 978-3-031-17994-5 Online ISBN 978-3-031-17995-2 Volume 13607, 2023. Pages 255–270. Online, October 17-20, 2022.

6. *Negri-Ribalta, C., Noel, R., Herbaut, N., Pastor, O., & Salinesi, C.* Socio-Technical Modelling for GDPR Principles: an Extension for the STS-ml. In Proceedings of the IEEE 30th International Requirements Engineering Conference Workshops (REW 2022). Publication: Print ISBN 978-1-6654-6001-9 Online ISBN 978-1-6654-6000-2, 2022. Pages 238-234. IEEE. Melbourne, Australia, August 15-19, 2022.

7. *Velasquez, R., Noel, R., Panach, J. I., & Pastor, O.* A Proposal for Measuring Understandability of Business Process Models. In Proceedings of RCIS 2022 Workshops and Research Projects Track (RCIS 2022). Barcelona, Spain, May 17-20, 2022.

8. *Noel, R., Ruiz, M., Panach, I., & Pastor, O.*

Beyond conventional model-driven development: from strategy to code. In Proceedings of the 14th International iStar Workshop (iStar 2021), co-located with 40th International Conference on Conceptual Modeling (ER 2021). St. Johns (NL), Canada, October 18, 2021.

9. *Noel, R., Panach Navarrete, J. I., Ruiz, M., & Pastor Lopez, O.*

   The LiteStrat Method: Towards Strategic Model-Driven Development. In proceedings of the he 29th International Conference on Information Systems Development (ISD2021). Core index: Not qualifies, fomerly A (2018). Publication: AIS eLibrary, 2022. Valencia, Spain, September 8-10, 2021.

10. *Noel, R.*

    From Strategy to Code: A Model-Driven Software Production Method. In Proceedings of the Doctoral Consortium Papers Presented at the 33rd International Conference on Advanced Information Systems Engineering (CAiSE 2021). Pages 79-88. Melbourne, Australia, June 28 - July 2, 2021.

11. *Noel, R., Panach, I., & Pastor, O.*

    A models-to-program information systems engineering method. In proceedings of the 2nd International Workshop on Modelling to Program. Publication: Print ISBN 978-3-030-72695-9, Online ISBN 978-3-030-72696-6, Volume 1401. Pages 162-176. Springer. Lappeenranta, Finland, March 10–12, 2020.

Book Chapters

1. *Pastor, O., Noel, R., Panach, I., & Ruiz, M.*

   The LiteStrat Modelling Method: Towards the Alignment of Strategy and Code. In Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools. Publication Print ISBN 978-3-030-93546-7, Online ISBN 978-3-030-93547-4, Volume. Pages 141-159. 2021. Cham: Springer International Publishing.

2. *Pastor, O., Noel, R., & Panach, I.*

   From Strategy to Code: Achieving Strategical Alignment in Software Development Projects through Conceptual Modelling. In Transactions on Large-Scale Data-and Knowledge-Centered Systems XLVIII: Special Issue In Memory of Univ. Prof. Dr. Roland Wagner. Publication: Print ISBN 978-3-662-63518-6, Online ISBN 978-3-662-63519-3. 2021. Pages 145-164. Berlin, Heidelberg: Springer Berlin Heidelberg.

9.4 Academic Projects and Activities

9.4.1 Project Participation

- SREC: Desarrollo ágil de sistemas desde requisitos a código (SREC). PID2021-123824OB-I00: Agencia Estatal de Investigacion, España. 2022-2025.
- DELFOS: Sistema de información para la gestión de variaciones genómicas. PDC2021-121243-I00 - Agencia Estatal de Investigacion, España. 2021-2023.
- OGMIOS: Sistema inteligente de apoyo a la toma de decisiones clínicas en medicina de precisión. INNEST/2021/57 - Agencia Estatal de Investigacion, España. 2021-2023.

9.4.2 Project proposal preparation

- SREC: Desarrollo ágil de sistemas desde requisitos a código (SREC). PID2021-123824OB-I00: Agencia Estatal de Investigacion, España. Achieved funding: 149.556 €. 2022-2025.

9.4.3 Theses and final degree project advisor

Final informatics engineering degree projects

- *Eduardo Cabrera*, "Validación empírica de un método de producción de software". En: "Empirical validation of a software production method". Informatics Engineering, Universidad de Valparaíso. 2023. Final degree project direction.
- *Jorge González*, "Desarrollo de una plataforma de modelado conceptual de sistemas de información". En: "Development of a platform for conceptual

modelling of information systems". Informatics Engineering, Universidad de Valparaíso. 2023. Final degree project direction.

- *Manuel Martinez*, "El usuario en el centro de la estrategia de organizacional: aplicación de Lean UX para la mejora del método LiteStrat". En: "The user at the centre of the organisational strategy: applying Lean UX to improve the LiteStrat method". Informatics Engineering, Universidad de Valparaíso. 2021. Final degree project direction.
- *Hong Xian*, "Diseño y desarrollo de una arquitectura distribuida para la transformación de modelos". En: "Design and development of a distributed architecture for model transformation". Informatics Engineering, Universidad de Valparaíso. 2021. Final degree project direction.
- *Emilio Miranda*, "Desarrollo de una herramienta web de modelado para Análisis Comunicacional". En: "Development of a web-based modelling tool for Communication Analysis". Informatics Engineering, Universidad de Valparaíso. 2021. Final degree project direction.

9.5 Research Collaborations

In parallel with the present research, Rene Noel collaborated with research teams in Chile. He contributed with his experience in software engineering and the knowledge gained during the PhD programme in design science methodology and experimentation to initiatives in three fields: computers and education, health informatics, and engineering education.

Publications in computers and Education

1. *Miranda, D., Noel, R., Alvarado, I., Cechinel, C., & Munoz, R.*

   Are you collaborative? A framework to evaluate non-verbal communication in indoor environments. In 2022 XVII Latin American Conference on Learning Technologies (LACLO). Publication: Print ISBN 978-1-6654-6522-9 Online ISBN 978-1-6654-6521-2. Pages 01-04. IEEE. Armenia, Colombia. October 17-21, 2022.

2. *Noel, R., Miranda, D., Cechinel, C., Riquelme, F., Primo, T. T., & Munoz, R.*

   Visualizing collaboration in teamwork: A multimodal learning analytics platform for non-verbal communication. Applied Sciences. Volume 12, Issue 15, n. 7499. 2022. MDPI. Impact factor 2.7. H Index: 101. Scopus (SJR) Q3. JCR Q2.

3. *Cornide-Reyes, H., Riquelme, F., Noel, R., Villarroel, R., Cechinel, C., Letelier, P., & Munoz, R.*

   Key skills to work with agile frameworks in software engineering: Chilean perspectives. IEEE Access, volume 9, Pages 84724-84738. 2021. IEEE inc. Impact factor 3.9. H Index: 204. Scopus (SJR) Q1. JCR Q2.

4. *Vieira, F., Cechinel, C., Ramos, V., Riquelme, F., Noel, R., Villarroel, R. & Munoz, R.*

   A learning analytics framework to analyze corporal postures in students presentations. Sensors, volume 21, issue 4, n. 1525. 2021. MDPI. Impact factor 3.9. H Index: 219. Scopus (SJR) Q2. JCR Q2.

5. *Cornide-Reyes, H., Riquelme, F., Monsalves, D., Noel, R., Cechinel, C., Villarroel, R., & Munoz, R. .*

   A multimodal real-time feedback platform based on spoken interactions for remote active learning support. Sensors, volume 20, issue 21, n 6337. 2020. Impact factor 3.9. H Index: 219. Scopus (SJR) Q2. JCR Q2.

6. *Riquelme, F., Noel, R., Cornide-Reyes, H., Geldes, G., Cechinel, C., Miranda, D., & Munoz, R.*

   Where are you? Exploring micro-location in indoor learning environments. IEEE Access, volume 8, pages 125776-125785. 2020. IEEE Inc. Impact factor 3.7. H Index: 204. Scopus (SJR) Q1. JCR Q2.

Publications in health informatics

1. *Taramasco, C., Rimassa, C., Noel, R., Bravo Storm M.L., Sánchez, C.*

   Co-design of a Mobile App for Engaging Breast Cancer Patients in Reporting Health Experiences: Qualitative Case Study. Journal of Medical Internet Research, volume 24, issue 3, n e26577. 2023 (forthcoming) . Impact factor 7.4. H Index: 178. Scopus (SJR) Q1. JCR Q1.

2. *Noel, R., Taramasco, C., & Márquez, G.*

   Standards, Processes, and Tools Used to Evaluate the Quality of Health Information Systems: Systematic Literature Review. Journal of Medical

Internet Research, volume 24, issue 3, n e26577. 2022. Impact factor 7.4. H Index: 178. Scopus (SJR) Q1. JCR Q1.

Publications in engineering education

1. *Morales, G., & Noel, R.*

   Work in Progress: Examining the Impact of a Faculty Development Program in Engineering Instructors' Teaching Practices and Perceptions on Active Learning Methodologies. In 2023 ASEE Annual Conference & Exposition. Baltimore, Maryland. June 25-28, 2023.

2. *Morales, G., Noel, R., & Campos, R.*

   Lesson Learned: Active Learning Coaching Program to Promote Faculty Development and Innovation in STEM Courses. In 2022 ASEE Annual Conference & Exposition. Minneapolis, Minnesota. August 23, 2022.

Organisation of scientific conferences

- Member of the organising committee. 35th International Conference on Advanced Information Systems Engineering June 12 - 16, 2023. Zaragoza, Spain.
- Member of the organising committee. 28th IEEE International Requirements Engineering Conference. August 31 - September 4, 2020. Online.

Program committee member

- 42nd International Conference of the Chilean Computer Science Society, SCCC 2023. October 23-26, 2023. Concepción, Chile.
- XLIX Conferencia Latinoamericana de Informática, CLEI 2023. La Paz, Bolivia.
- 1st International Conference on Complex Computational Ecosystems CCE2023. April 25–27, 2023. Baku, Azerbaijan.
- 41st International Conference of the Chilean Computer Science Society, SCCC 2022. November 21-25, 2022. Santiago, Chile.

9.6 Awards and Grants

This research project has been made possible by the support and funding from two Chilean institutions: Universidad de Valparaíso and the National Agency for Research and Development (ANID). The financial backing from these institutions has played a crucial role in facilitating the execution and successful completion of this PhD thesis. Universidad de Valparaíso granted Rene Noel a commission to pursue doctoral studies. The research plan for the PhD thesis was submitted for ANID funding and received an outstanding evaluation at the national level. Grant received: the Scholarship Program Doctorado Becas Chile, 2020-72210494.

9.7 Future works

Three main courses of action guide the future work of this research in the middle term: the implementation of LiteStrat and Stra2Bis as requirements engineering methods in a real-world context, the integration of the supporting tools of the methods that make up the whole model-driven software production method, and the evolution of LiteStrat and Stra2Bis considering the latest advances on using organisational information as input for the software development process. In the long term, we expect to use LiteStrat and Stra2Bis as cornerstones for advising organisations on the early requirements engineering stages as a service of a forthcoming multidisciplinary research centre at Universidad de Valparaíso.

9.7.1 Implementation of Litestrat and Stra2Bis in a real-world context

With regard to the implementation of LiteStrat and Stra2Bis in a real-world context, we are currently in the first stages of applying them for managing the business processes in the domain of genomics variant interpretation in the context of the SREC: PID2021-123824OB-I00 research project. We aim to apply the technical action research methodology for technical problem-solving and assessing the effects of LiteStrat and Stra2Bis.

Regarding technical problem-solving activities, the implementation aims to support the process and software architecture design of the genomics variant interpretation technologies developed by the genomics group at the PROS Research Centre. The genomics group is looking forward to automating and ensuring the scalability of the OGMIOS platform that offers genomics variant reports. By applying LiteStrat, we expect to identify the optimal strategy for addressing the massive quantity of new research on genomics variants and the optimal organisational structure to deploy such a strategy. On the other hand, Stra2Bis will help design business processes that are independent and collaborative in a loosely coupled way to foster the independence of interdisciplinary research groups.

We expect that implementation results will provide insights into the strengths, limitations, and improvement opportunities for LiteStrat and Stra2Bis, helping their continuous refinement, as well as empirical evidence on their effect on process and software design.

9.7.2 Tool support integration

Concerning the tool support, we aim to extend the current implementation of LiteStrat and Stra2Bis in ADOxx. There are two further works headed to evolving the tool. Firstly, we aim to integrate the current implementation with INTEGRANOVA by means of the systematic derivation of the class diagram from Communication Analysis models. The systematic derivation is currently implemented as a web service, and future work is headed towards using the output file produced by this service as an input model for INTEGRANOVA. This way, the model-driven software production method will be totally supported without requiring manual model integration.

Secondly, we aim to generate the scaffold of business processes from LiteStrat models using BPMN choreographies. Since BPMN choreographies have been used for deriving microservices architectures, we would be able to exploit existing methods and tools to generate the code of the information system as a set of small, loosely coupled services which are aligned to the organisational structure modelled in LiteStrat.

Finally, regarding the evolution of the method, we envision two primary research efforts. Firstly, we aim to observe, assess, and eventually integrate conceptual improvements introduced by the software industry on using organisational information to drive the software development process. For instance, recently, we proposed an extension for LiteStrat to enable Team Topologies, a recent approach for characterising organisational units (i.e., software development teams) as whether they contribute to the main business outcomes, provide methodological support to overcome obstacles or build internal software products to improve software development efficiency.

The second research effort for evolving the LiteStrat and Stra2Bis will be focused on the systematic derivation of microservices from communication Analysis models, aiming to mirror the organisational and business process structure to the software components and services architecture. We look forward to new additions in this line to be integrated into LiteStrat and Stra2Bis.

9.7.3 Creation of a Research Centre at Universidad de Valparaíso

As part of Rene Noel's return plan to the Universidad de Valparaíso in Chile, there is the project of creating a research centre as a joint effort with researchers on informatics applications in business, educational, and health domains. Within this centre, it is expected to create a research group devoted to knowledge generation on model-driven approaches for software development. The activity of this centre will focus on designing requirements engineering and software production methods and technology to help organisations develop or request software solutions.

We expect to use the methods designed in this thesis as well as the methods designed by PROS research Centre at Universitat Politècnica de València and the future participation in joint international projects.

Part V

Appendixes

Appendix A

ADOxx Implementation Listings

A.1 LiteStrat Integrity Constraint Validations

```
SEND "GET_ACTIVE_MODEL" to:"Modeling" answer:modelid
IF (modelid = "")
{
   CC "AdoScript" ERRORBOX "Open a model first!"
   EXIT
}

SET sCurrentModel:(modelid)

SET sErrors:"Model Validations:"

###################################
#      INTEGRITY CONSTRAINTS        #
###################################

####################################
#IC1 - The model must have at least one actor influencing an
    Organisation unit.
#IC9 - The model must have at least actor being influenced by an
    Organisation unit.
```

```

# get the current model

# get all connectors of the model

SET nIC1Flag:0
SET nIC9Flag:0


# for each connector, test if from is an acor and out is an organizational unit
CC "Core" GET_ALL_CONNECTORS modelid:(VAL sCurrentModel)

FOR id in:(objids)
{
   CC "Core" GET_CONNECTOR_ENDPOINTS objid:(VAL id)
            # now get the classes of the connected instances
            CC "Core"  GET_CLASS_ID objid:(fromobjid)
            SET fromclassid:(classid)
            CC "Core" GET_CLASS_ID objid:(toobjid)
            SET toclassid:(classid)

            # now get the names of the classes
            CC "Core" GET_CLASS_NAME classid:(fromclassid)
            SET fromclassname:(classname)
            CC "Core" GET_CLASS_NAME classid:(toclassid)
            SET toclassname:(classname)

            #check if from is actor an to is ou
            IF (fromclassname = "Actor" AND toclassname = "OrganizationalUnit") {
                SET nIC1Flag:1
            }

            IF (fromclassname = "OrganizationalUnit" AND toclassname = "Actor") {
                SET nIC9Flag:1
            }
}

IF (nIC1Flag = 0) {
    SET sErrors:(sErrors + "\n- The model must have at least one Actor influencing an Organisation Unit. (IC1)")
}

IF (nIC9Flag = 0) {
    SET sErrors:(sErrors + "\n- The model must have at least actor being influenced by an Organisation unit. (IC9)")
}

#####################################
#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.
#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.


SET nIC2Flag:0
SET nIC3Flag:0
SET nIC4Flag:0
SET nIC7Flag:0
SET nIC8Flag:0




#Get all organizational units
CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel) classname:"OrganizationalUnit"
SET uniOids:(objids)

#look for elements inside units
FOR uniid in:(uniOids)
{
    #get unit name to query model
    CC "Core" GET_ATTR_VAL objid:(VAL uniid) attrname:"Name" as-string
    SET sUnitName:(val)

    #get unit contents
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sUnitName + "\":\"OrganizationalUnit\"}<-\"Is inside\") OR ({\""+ sUnitName + "\":\"OrganizationalUnit\"}->\"Is inside\")") modelid:(VAL sCurrentModel)
    SET uniContOids:(objids)

    #check unit contents classes
    FOR uniContid in:(uniContOids) {
        CC "Core" GET_CLASS_ID objid:(VAL uniContid)
        CC "Core" GET_CLASS_NAME classid:(classid)

        IF (classname = "Goal") {
            SET nIC2Flag:1
        }
        ELSIF (classname = "Strategy") {
            SET nIC3Flag:1
        }
        ELSIF (classname = "Tactic") {
            SET nIC4Flag:1
        }
```

```
        ELSIF (classname = "Role") {
            SET nIC7Flag:1
        }
        ELSIF (classname = "OrganizationalUnit") {
            SET nIC8Flag:1
        }
    }
}

IF (nIC2Flag = 0) {
    SET sErrors:(sErrors + "\n- At least one Organisation unit must have one or more goals associated.(IC2)")
}

IF (nIC3Flag = 0) {
    SET sErrors:(sErrors + "\n- At least one Organisation unit must have one or more strategies associated.(IC3)")
}

IF (nIC4Flag = 0) {
    SET sErrors:(sErrors + "\n- At least one Organisation unit must have one or more tactics inside.(IC4)")
}

IF (nIC7Flag = 0) {
    SET sErrors:(sErrors + "\n- At least one Organisation unit must have one or more roles inside.(IC7)")
}

IF (nIC8Flag = 0) {
     SET sErrors:(sErrors + "\n- At least one Organisation unit must be inside other Organisation unit.(IC8)")
}

#####################################
#IC5 - Tactics can not be inside an Organisation unit that is not inside other Organisation unit.
SET nIC5Flag:0
SET nAIC5_TFlag:0

##Get all tactics
CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel) classname:"Tactic"
SET tacticOids:(objids)

FOR tacticOid in:(tacticOids) {
    ##Get tactic's name
    CC "Core" GET_ATTR_VAL objid:(VAL tacticOid) attrname:"Name" as-string
    SET sTacticName:(val)

    ##Get tactic's parents
```

```
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sTacticName + "\":\"Tactic\"}->\"Is inside\")") modelid:(VAL sCurrentModel)

    ##Validate IC5 - If there are two organizational units that are parents, the constraint is satisfied
    SET tacticParOids:(objids)
    SET nParentUnits:0

    FOR tacticParentOid in:(tacticParOids) {
        #get Parent's classname
        CC "Core" GET_CLASS_ID objid:(VAL tacticParentOid)
        CC "Core" GET_CLASS_NAME classid:(classid)
        SET tacticParentClassName:(classname)
        IF (tacticParentClassName = "OrganizationalUnit") {
            SET nParentUnits:(nParentUnits + 1)
        }
    }

    #verify if there are two units as parents
    IF (nParentUnits < 2) {
        SET nIC5Flag:(nIC5Flag + 1)
    }
}

IF (nIC5Flag > 0) {
         SET sErrors:(sErrors + "\n- Tactics can not be inside an Organisation unit that is not inside other Organisation unit. (IC5)")
}


#####################################
#IC6 - Roles can not be inside an Organisation unit that is not inside other Organisation unit.
SET nIC6Flag:0

##Get all roles
CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel) classname:"Role"
SET roleOids:(objids)

FOR roleOid in:(roleOids) {
    ##Get role's name
    CC "Core" GET_ATTR_VAL objid:(VAL roleOid) attrname:"Name" as-string
    SET sRoleName:(val)

    ##Get role's parents
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sRoleName + "\":\"Role\"}->\"Is inside\")") modelid:(VAL sCurrentModel)

    ##If there are two organizational units that are parents, the constraint is satisfied
```

```
    SET roleParOids:(objids)
    SET nParentUnits:0

    FOR roleParentOid in:(roleParOids) {
        #get Parent's classname
        CC "Core" GET_CLASS_ID objid:(VAL roleParentOid)
        CC "Core" GET_CLASS_NAME classid:(classid)
        SET roleParentClassName:(classname)
        IF (roleParentClassName = "OrganizationalUnit") {
            SET nParentUnits:(nParentUnits + 1)
        }
    }

    #verify if there are two units as parents
    IF (nParentUnits < 2) {
        SET nIC6Flag:(nIC6Flag + 1)
    }
}


IF (nIC6Flag > 0) {
    SET sErrors:(sErrors + "\n- Roles can not be inside an
    Organisation unit that is not inside other Organisation unit.(IC6
    )")

}


#####################################
#IC11 - All objectives must be inside roles

SET nIC11Flag:1

##Get all objectives
CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel)
    classname:"Objective"
SET objtvOids:(objids)

IF (LEN objtvOids > 0) {
    SET nIC11Flag:0
}



FOR objtvOid in:(objtvOids) {
    ##Get objective's name
    CC "Core" GET_ATTR_VAL objid:(VAL objtvOid) attrname:"Name" as-
    string
    SET sObjtvName:(val)

    ##Get objectives's parents
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sObjtvName + "\":\"
    Objective\"}->\"Is inside\")") modelid:(VAL sCurrentModel)
```

```

    # If there is at least one parent that is a role, constraint is satisfied
    SET objtvParOids:(objids)
    SET nParentUnits:0

    FOR objtParentOid in:(objtvParOids) {
        #get Parent's classname
        CC "Core" GET_CLASS_ID objid:(VAL objtParentOid)
        CC "Core" GET_CLASS_NAME classid:(classid)

        SET objtvParentClassName:(classname)

        IF (objtvParentClassName = "Role") {
            SET nIC11Flag:1
        }
    }
}


IF (nIC11Flag = 0) {
    SET sErrors:(sErrors + "\n- All objectives must be inside roles.(IC11)")
}

####################################
#IC10 - All roles must have at least one objective
#IC12 - Roles can only contain objectives

##Get all roles
CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel) classname:"Role"
SET rolepOids:(objids)

SET nIC10Flag:1
SET nIC12Flag:1

IF (LEN rolepOids > 0) {
    SET nIC10Flag:1
}

FOR rolepOid in:(rolepOids) {
    ##Get roles's name
    CC "Core" GET_ATTR_VAL objid:(VAL rolepOid) attrname:"Name" as-string
    SET sRolepName:(val)

    ##Get roles's childs
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sRolepName + "\":\"Role\"}<-\"Is inside\")") modelid:(VAL sCurrentModel)

    SET roleChildsOids:(objids)
    SET nChildObjts:0
```

```
    SET nChildOthers:0

    IF (LEN roleChildsOids > 0) {
        SET nIC12Flag:1
    }

    FOR roleChildOid in:(roleChildsOids) {
        #get Parent's classname
        CC "Core" GET_CLASS_ID objid:(VAL roleChildOid)
        CC "Core" GET_CLASS_NAME classid:(classid)

        SET roleChildClassName:(classname)

        IF (roleChildClassName = "Objective") {
            SET nChildObjts:(nChildObjts+1)
        }
        ELSE {
            SET nChildOthers:(nChildOthers+1)
        }

    }
    IF (nChildObjts = 0) {
        SET nIC10Flag:0
    }
     IF (nChildOthers > 0) {
        SET nIC12Flag:0
    }

}
IF (nIC10Flag = 0) {
    SET sErrors:(sErrors + "\n- All roles must have at least one
    objective.(IC10)")
}
IF (nIC12Flag = 0) {
    SET sErrors:(sErrors + "\n- Roles can only contain objectives (
    IC12)")
}



####################################
#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.


##Get all connectors
CC "Core" GET_ALL_CONNECTORS modelid:(VAL sCurrentModel)
```

```

SET nAIC1_1Flag:1
SET nAIC1_2Flag:1
SET nAIC2_1Flag:1
SET nAIC2_2Flag:1
SET nAIC3_1Flag:1
SET nAIC3_2Flag:1
SET nAIC4_1Flag:1
SET nAIC4_2Flag:1

FOR id in:(objids)
{
    #select refinement connectors
    CC "Core" GET_CLASS_ID objid:(VAL id)
    CC "Core" GET_CLASS_NAME classid:(classid)
    SET connClassName:(classname)

    IF (connClassName = "Refinement") {

        CC "Core" GET_CONNECTOR_ENDPOINTS objid:(VAL id)
        # now get the classes of the connected instances
        CC "Core"  GET_CLASS_ID objid:(fromobjid)
        SET fromclassid:(classid)
        CC "Core" GET_CLASS_ID objid:(toobjid)
        SET toclassid:(classid)

        # now get the names of the classes
        CC "Core" GET_CLASS_NAME classid:(fromclassid)
        SET fromclassname:(classname)
        CC "Core" GET_CLASS_NAME classid:(toclassid)
        SET toclassname:(classname)

        #Validate AIC1 - Goals must not be refined from other elements
        IF (toclassname = "Goal") {
            SET nAIC1_1Flag:0
        }
        #Validate AIC1 - Goals must be refined only by strategies
        IF (fromclassname = "Goal" AND toclassname <> "Strategy") {
            SET nAIC1_2Flag:0
        }

        #Validate AIC2 - Strategies can only be refinements of goals
        IF (toclassname = "Strategy" AND fromclassname <> "Goal") {
            SET nAIC2_1Flag:0
        }
        #Validate AIC2 - Strategies can only be refined by Tactics.
        IF (fromclassname = "Strategy" AND toclassname <> "Tactic") {
            SET nAIC2_2Flag:0
        }

        #Validate #AIC3 - Tactics must only be refinements of Strategies.
```

```
        IF (toclassname = "Tactic" AND fromclassname <> "Strategy") {
            SET nAIC3_1Flag:0
        }

        #Validate #AIC3 - Tactics must be only refined by Objectives.
        IF (fromclassname = "Tactic" AND toclassname <> "Objective") {
            SET nAIC3_2Flag:0
        }

        #Validate AIC4 - Objectives must only be refinements of Tactics
        IF (toclassname = "Objective" AND fromclassname <> "Tactic") {
            SET nAIC4_1Flag:0
        }
        #Validate AIC4 - Objectives can not be refined.
        IF (fromclassname = "Objective") {
            SET nAIC4_2Flag:0
        }

    }

}

IF (nAIC1_1Flag = 0) {
    SET sErrors:(sErrors + "\n- Goals must not be refined from other elements.(AIC1)")
}
IF (nAIC1_2Flag = 0) {
        SET sErrors:(sErrors + "\n- Goals must be refined only by strategies.(AIC1)")
}
IF (nAIC2_1Flag = 0) {
    SET sErrors:(sErrors + "\n- Strategies can only be refinements of goals.(AIC2)")
}
IF (nAIC2_2Flag = 0) {
    SET sErrors:(sErrors + "\n- Strategies can only be refined by Tactics.(AIC2)")
}

IF (nAIC3_1Flag = 0) {
    SET sErrors:(sErrors + "\n- Tactics must only be refinements of Strategies.(AIC3)")
}
IF (nAIC3_2Flag = 0) {
    SET sErrors:(sErrors + "\n- Tactics must be only refined by Objectives.(AIC3)")
}

IF (nAIC4_1Flag = 0) {
```

```
    SET sErrors:(sErrors + "\n- Objectives must only be refinements of Tactics.(AIC4)")
}

IF (nAIC4_2Flag = 0) {
    SET sErrors:(sErrors + "\n- Objectives can not be refined.(AIC4)")
}

####################################
#AIC5 - Strategies, Tactics, and Objectives must be refinements.


##get Val strategies
SET nAIC5_1Flag:1

CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel) classname:"Strategy"
SET strategyOids:(objids)



FOR strategyOid in:(strategyOids) {
    ##Get strtegy's name
    CC "Core" GET_ATTR_VAL objid:(VAL strategyOid) attrname:"Name" as -string
    SET sStrategyName:(val)

    ##Get strategy's childs
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sStrategyName + "\":\"Strategy\"}<-\"Refinement\")") modelid:(VAL sCurrentModel)

    IF (LEN objids = 0) {
        SET nAIC5_1Flag:0
    }
}


##get Val Tactics
SET nAIC5_2Flag:1

CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel) classname:"Tactic"
SET tacticOids:(objids)


FOR tacticOid in:(tacticOids) {
    ##Get tactic's name
    CC "Core" GET_ATTR_VAL objid:(VAL tacticOid) attrname:"Name" as-string
    SET sTacticName:(val)

    ##Get tactic's childs
```

```
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sTacticName + "\":\"
    Tactic\"}<-\"Refinement\")") modelid:(VAL sCurrentModel)

    IF (LEN objids = 0) {
        SET nAIC5_2Flag:0
    }
}

##get Val Objectives
SET nAIC5_3Flag:1

CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel)
    classname:"Objective"
SET objectiveOids:(objids)


FOR objectiveOid in:(objectiveOids) {
    ##Get objective's name
    CC "Core" GET_ATTR_VAL objid:(VAL objectiveOid) attrname:"Name"
    as-string
    SET sObjectiveName:(val)

    ##Get objective's childs
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sObjectiveName + "
    \":\"Objective\"}<-\"Refinement\")") modelid:(VAL sCurrentModel)

    IF (LEN objids = 0) {
        SET nAIC5_3Flag:0
    }
}

IF (nAIC5_1Flag = 0) {
    SET sErrors:(sErrors + "\n- Strategies must be refinements.(AIC5)
    ")
}

IF (nAIC5_2Flag = 0) {
    SET sErrors:(sErrors + "\n- Tactics must be refinements.(AIC5)")
}

IF (nAIC5_3Flag = 0) {
SET sErrors:(sErrors + "\n- Objective must be refinements.(AIC5)")
}


#*************************************************
#AIC6 - Actors can not have elements inside of them nor be inside
    another element.
#AIC7 - All Actors must be influenced by some organizational unit or
    influence a organizational unit.
#AIC8 - Actors can not ne influenced by other actors or influenfe
    other actors

```

```
##get Val strategies
SET nAIC6Flag:1
SET nAIC7Flag:1
SET nAIC8Flag:1

CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(VAL sCurrentModel)
    classname:"Actor"

SET actorOids:(objids)

FOR actorOid in:(actorOids) {

    #Validate - AIC6
    ##Get strtegy's name
    CC "Core" GET_ATTR_VAL objid:(VAL actorOid) attrname:"Name" as-
    string
    SET sActorName:(val)

    ##Get strategy's childs
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sActorName + "\":\"
    Actor\"}<-\"Is inside\") OR ({\"" + sActorName + "\":\"Actor
    \"}->\"Is inside\")") modelid:(VAL sCurrentModel)

    IF (LEN objids > 0) {
        SET nAIC6Flag:0
    }

    #Validate - AIC7, AIC8
    #Get all influences
    CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sActorName + "\":\"
    Actor\"}<-\"Influence\") OR ({\"" + sActorName + "\":\"Actor
    \"}->\"Influence\")") modelid:(VAL sCurrentModel)
    SET influenceAgentsOids:(objids)
    #Validate AIC7
    IF (LEN influenceAgentsOids = 0) {
        SET nAIC7Flag:0
    }
    ELSE {

        FOR influenceAgentOid in:(influenceAgentsOids) {

        #Validate AIC8
        ##Get Connectors

            CC "Core" GET_CLASS_ID objid:(VAL influenceAgentOid)
            SET agentclassid:(classid)
            CC "Core" GET_CLASS_NAME classid:(agentclassid)
            SET agentclassname:(classname)

            IF (agentclassname = "Actor" ) {
                SET nAIC8Flag:0
            }
        }
```

```
    }

}

IF (nAIC6Flag = 0) {
 SET sErrors:(sErrors + "\n- Actors can not have elements inside of
    them nor be inside another element.(AIC6)")
}

IF (nAIC7Flag = 0) {
SET sErrors:(sErrors + "\n- All Actors must be influenced by some
    organizational unit or influence a organizational unit. (AI7)")
}

IF (nAIC8Flag = 0) {
SET sErrors:(sErrors + "\n- Actors can not be influenced by other
    actors or influence other actors. (AIC8)")
}


#*******************************************************
#SHOW ERRORS
#*******************************************************
IF( sErrors <> "Model Validations:") {
    CC "AdoScript" ERRORBOX(sErrors)
}
```

Listing A.1: ADOxx script for Stra2Bis transformation guidelines.

A.2 Stra2Bis Transformation Guidelines Implementation

```
######### STRA2BIS TRANSFORMATIONS %%%%%%%

###SET WORK CONTEXT: LiteStrat Model ########

SEND "GET_ACTIVE_MODEL" to:"Modeling" answer:modelid
IF (modelid = "")
{
   CC "AdoScript" ERRORBOX "Open a model first!"
   EXIT
}

#Set LiteStrat model as the model under analysis
SET sCurrentModelId:(modelid)

CC "Modeling" GET_ACT_MODEL
CC "Core" GET_MODEL_MODELTYPE modelid:(modelid)

IF ((modeltype) != "LiteStrat") {
```

```
    CC "AdoScript" INFOBOX "Please open a LiteStrat model to apply the transformation."
    BREAK
}


#Get the list of organisational units inside the organisation.
GET_INTERNAL_OUS iLiteStratModelId:(VAL sCurrentModelId) result:sInternalOUS

IF(sInternalOUS <> "") {
    #1. Init vars for creating models (just the first time)
    #select root model
    CC "CoreUI" MODEL_SELECT_BOX boxtext:"Please select the root model group:" without-models mgroup-sel title:"Stra2Bis Transformation"
    SET rootModelGroupId:(VAL mgroupids)

    #Get timestamp for naming models
    CC "Application"  GET_DATE_TIME date-format:"DD.MM.YYYY" time-format:"HH:MM:SS"
    SET newModelTimeStamp:(date+time)

    #2. Init Traceability modelgroup and model
    #SET sTraceModelId:0
    #CREATE_STOB_TRACE_MODEL mgroupids:(rootModelGroupId) result:sTraceModelId


    #3. Init Business Process Model Group
    CREATE_BP_MODELGROUP mgroupids:(rootModelGroupId) result:sBPModelGroupId


    # 3. Apply Guideline 1 For each Internal organisation Unit:

    FOR organisationUnitId_i in:(sInternalOUS) {
        IF (organisationUnitId_i != "") {
            TRANSFORM_OUS pOUid:(VAL organisationUnitId_i) pBPModelGroupId:(sBPModelGroupId) result:newCAModel_i
        }
    }

    # 3. Apply Guideline 2 For each dependency between internal organisation units:
    FOR organisationUnitId_i in:(sInternalOUS) {
        IF (organisationUnitId_i != "") {
            TRANSFORM_DEPENDENCIES pOUid:(VAL organisationUnitId_i) pBPModelGroupId:(sBPModelGroupId) result:newCAModel_i
        }
    }

```

```
    # 3. Apply Guideline 3 For each objective inside internal
    organisation units:
    FOR organisationUnitId_i in:(sInternalOUS) {
        IF (organisationUnitId_i != "") {
            TRANSFORM_OBJECTIVES pOUid:(VAL organisationUnitId_i)
    pBPModelGroupId:(sBPModelGroupId) result:newCAModel_i
        }
    }

    CC "Modeling" OPEN modelids:(sCurrentModelId)
    CC "Modeling" ACTIVATE_MODEL (sCurrentModelId)
}


CC "Application" SET_STATUS ("Stra2Bis Transformation Finished ")

#### PRPOCEDURES

PROCEDURE GET_INTERNAL_OUS iLiteStratModelId:integer result:
    reference {
    SET aInternalOUs:""

    CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(iLiteStratModelId)
    classname:"OrganizationalUnit"

    FOR organisationUnitId_i in:(objids) {

        #Get Organisation Unit's parent
        CC "Core" GET_ATTR_VAL objid:(VAL organisationUnitId_i)
    attrname:"Name" as-string
        SET sUnitName:(val)

        CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sUnitName + "
    \":\"OrganizationalUnit\"}->\"Is inside\")") modelid:(VAL
    sCurrentModelId)
        SET tOrgUnitParOids:(objids)

        SET nParentUnits:0

        FOR orgUnitOids in:(tOrgUnitParOids) {
            #get Parent's classname
            CC "Core" GET_CLASS_ID objid:(VAL orgUnitOids)
            SET sOUClassId:(classid)

            CC "Core" GET_CLASS_NAME classid:(sOUClassId)
            SET orgUnitParentClassName:(classname)
            IF (orgUnitParentClassName = "OrganizationalUnit") {
                SET nParentUnits:(nParentUnits + 1)
            }
        }

        IF (nParentUnits > 0) {
            ##Add to array of internal Organisation Units
```

```
                SET aInternalOUs:(aInternalOUs + organisationUnitId_i + " ")
            }
        }
        SET result:(aInternalOUs)
}

PROCEDURE CREATE_STOB_TRACE_MODEL mgroupids: integer result: reference {
# 2. IF there are "internal" organization units, create the Traceability Model
                CC "Core" CREATE_MODELGROUP supermgroupid:(mgroupids) mgroupname:"S2B Trace" result:(traceModelGroupId)

                #ECODE = 52: Model already exists.
                IF (ecode != 0 AND ecode != 52)
                {
                    CC "AdoScript" INFOBOX "Error!\nModelgroup could not be created. Was the name you entered unique?"
                }
                #Create Traceability Model
                CC "Core" CREATE_MODEL  modeltype:"Stra2Bis Traceability" modelname:("S2BTrace-"+newModelTimeStamp) version:"1.0" mgroups:(traceModelGroupId)
                SET result:(modelid)

}

PROCEDURE CREATE_BP_MODELGROUP mgroupids:integer result:reference {

        CC "Core" CREATE_MODELGROUP supermgroupid:(mgroupids) mgroupname:"Business Process"
        SET temp:(mgroupid)

        #ECODE = 52: Model already exists.

        IF (ecode != 0 AND ecode != 52) {
            CC "AdoScript" INFOBOX "Error!\nModelgroup could not be created. Was the name you entered unique?"
        }

        SET result:(temp)


}

###GUIDELINE 1 ########
PROCEDURE TRANSFORM_OUS pOUid:integer pBPModelGroupId:integer result:reference {

#   3.1 Create a target Communication Analysis Diagram CAModel_i
            CC "Core" GET_ATTR_VAL objid:(pOUid) attrname:"Name" as-string
```

```
        SET iUnitName:(val)

        #CC "Core" CREATE_MODEL  modeltype:"Communication Analysis" modelname:(iUnitName+" Process "+newModelTimeStamp) version:"1.0" mgroups:(pBPModelGroupId)
        CC "Core" CREATE_MODEL  modeltype:"Communication Analysis" modelname:(iUnitName+" Process") version:"1.0" mgroups:(pBPModelGroupId)
        SET newCAModel_i:(modelid)

#    3.2 Create a start node in CAModel_i.StartNode_ with the nam Organisation Unit.Name + Process

        CC "Modeling" OPEN modelids:(newCAModel_i)
        CC "Modeling" ACTIVATE_MODEL (newCAModel_i)

        CC "Core" GET_CLASS_ID classname:"CAStart"
        SET sStartNodeClassId:(classid)

        CC "Core" CREATE_OBJ modelid:(newCAModel_i) classid:(sStartNodeClassId) objname:(iUnitName+" Process")
        SET sStartNodeId:(objid)

        CC "Modeling" SET_OBJ_POS objid:(sStartNodeId) x:(10.5cm) y:(1.5cm)
        CC "Modeling" REBUILD_DRAWING_AREA

#    3.3 Reference Start Node's owner Unit

        #Add to the processes owned by the unit:
        CC "Core" GET_ATTR_ID classid:(sStartNodeClassId) attrname:"Owner Unit"
        CC "Core" ADD_INTERREF objid:(sStartNodeId) attrid:(attrid) tobjid:(pOUid) tmodelid:(sCurrentModelId)

#    3.4 Reference Unit's processes (start nod)

        CC "Modeling"  ACTIVATE_MODEL (sCurrentModelId)

        CC "Core" GET_CLASS_ID objid:(pOUid)



        #Add interref as the traced BP owned by the OU (read only, just one process)
        CC "Core" GET_ATTR_ID classid:(classid) attrname:"Traced Business Process"
        SET traceAttrid:(attrid)
        CC "Core" REMOVE_ALL_INTERREFS objid:(pOUid) attrid:(traceAttrid)
        CC "Core" ADD_INTERREF objid:(pOUid) attrid:(traceAttrid) tobjid:(sStartNodeId) tmodelid:(newCAModel_i)

```

```
        #Add interref as a BP owned by the OU
        CC "Core" GET_ATTR_ID classid:(classid) attrname:"Business
    Processes"
        CC "Core" ADD_INTERREF objid:(pOUid) attrid:(attrid) tobjid:(
    sStartNodeId) tmodelid:(newCAModel_i)

        SET result:(newCAModel_i)
}

###GUIDELINE 2 ########
PROCEDURE TRANSFORM_DEPENDENCIES pOUid:integer pBPModelGroupId:
    integer result:reference {

    #Get the influence connectors for the internal OUs where OU is
    source
    CC "Core" GET_CONNECTORS objid:(pOUid) out
    SET sOutInfluence:(objids)

    FOR influence_i in:(sOutInfluence) {

        #Get spurce OU name
        CC "Core" GET_ATTR_VAL objid:(pOUid) attrname:"Name" as-
    string
        SET sSourceOUName:(val)

        #get connector class
        CC "Core" GET_CLASS_ID objid:(VAL influence_i)
        CC "Core" GET_CLASS_NAME classid:(classid)
        SET connClassName:(classname)

        #if  connector class is influence
        IF (connClassName = "Influence") {

            #POSITION COMSTANTs
            SET ypos:(15cm)

            #Get influence name
            CC "Core" GET_ATTR_VAL objid:(VAL influence_i) attrname:"
    Name" as-string
            SET sInfluenceName:(val)

            # Get target organisation unit
            CC "Core" GET_CONNECTOR_ENDPOINTS objid:(VAL influence_i)
            SET iTargetOUid:(toobjid)

            #Target OU name
            CC "Core"  GET_ATTR_VAL objid:(iTargetOUid) attrname:"
    Name" as-string
            SET iTargetOUName:(val)

            #CC "AdoScript" INFOBOX ("FROM: " + (sSourceOUName) + "
    THROUGH: " + sInfluenceName + " TO: " + iTargetOUName)

```

```
            #In the BP model of the source OU, create an event, a secondary actor, an outgoing interaction.

            ###############################################
            ##CREATE ELEMENTS IN THE SOURCE OU BP MODEL#####
            ###############################################

            ### Get the Traced BP's of the source OU
            CC "Core" GET_INTERREF  objid:(pOUid ) attrname:"Traced Business Process" index:0
            SET sOUBPModelId:(tmodelid)
            #CC "AdoScript" INFOBOX ("sOUBPModelId" + (STR sOUBPModelId))
            CC "Modeling" ACTIVATE_MODEL (sOUBPModelId)

            ### Create the communicative event
            CC "Core" GET_CLASS_ID classname:"CommunicativeEvent"
            SET sCANodeClassId:(classid)
            CC "Core" CREATE_OBJ modelid:(sOUBPModelId) classid:(sCANodeClassId) objname:("Dispatch " + (sInfluenceName))
            SET sNewCEid:(objid)

            IF (ecode != 0 AND ecode != 52)
                {
                    CC "AdoScript" INFOBOX "Error!\nModelgroup could not be created. Was the name you entered unique?"
                }

            CC "Modeling" SET_OBJ_POS objid:(sNewCEid) x:(10.5cm) y:(ypos)
            CC "Core" SET_ATTR_VAL objid:(sNewCEid) attrname:"Code" val:("S2B-G2") ###S2B-GL2 ->"Generated by Stra2Bis transformaiton Guideline 2"


            ### Create the receiver actor
            CC "Core" GET_CLASS_ID classname:"CAActor"
            SET sActorClassId:(classid)
            CC "Core" CREATE_OBJ modelid:(sOUBPModelId) classid:(sActorClassId) objname:((iTargetOUName)+ " Agent")
            SET sNewActorid:(objid)
            CC "Modeling" SET_OBJ_POS objid:(sNewActorid) x:(20cm) y:(ypos)

            ##### Receiver actor is the support actor of the CE
            CC "Core" GET_ATTR_ID classid:(sCANodeClassId) attrname:"SupportActor"
            CC "Core" ADD_INTERREF objid:(sNewCEid) attrid:(attrid) tobjid:(sNewActorid) tmodelid:(sCurrentModelId)

            ### Create the outgoing communicative event
            CC "Core" GET_CLASS_ID classname:"OutgoingInteraction"
            SET sOutgoingClassId:(classid)
```

```
            CC "Core" CREATE_CONNECTOR modelid:(sOUBPModelId)
fromobjid:(sNewCEid) toobjid:(sNewActorid) classid:(
sOutgoingClassId)
            SET sOutgoingId:(objid)
            CC "Core" SET_ATTR_VAL objid:(sOutgoingId) attrname:"Name
" val:(sInfluenceName)

            #CC "Modeling" SET_OBJ_POS objid:(sNewActorid) x:(13cm) y
:(10.5cm)

            CC "Modeling" REBUILD_DRAWING_AREA




            ### Traceability
            #### Add the CEvent to the out dependencies of the source
 OU
            CC "Core" GET_CLASS_ID objid:(pOUid)
            CC "Core" GET_ATTR_ID classid:(classid) attrname:"Out
Dependencies Events"
            CC "Core" ADD_INTERREF objid:(pOUid) attrid:(attrid)
tobjid:(sNewCEid) tmodelid:(sCurrentModelId)

            #### Add for the CA Actor and their Organisatoin units
            CC "Core" GET_ATTR_ID classid:(sActorClassId) attrname:"
Traced Organisation Unit in Dependency"
            CC "Core" ADD_INTERREF objid:(sNewActorid) attrid:(attrid
) tobjid:(iTargetOUid) tmodelid:(sCurrentModelId)

            ################################################
            ##CREATE ELEMENTS IN THE TARGET OU BP MODEL#####
            ################################################

            ### Get the Traced BP's of the targhet OU
            CC "Core" GET_INTERREF  objid:(iTargetOUid) attrname:"
Traced Business Process" index:0
            SET tOUBPModelId:(tmodelid)
            CC "Modeling" ACTIVATE_MODEL (tOUBPModelId)

            ### Create the communicative event
            CC "Core" GET_CLASS_ID classname:"CommunicativeEvent"
            SET sCANodeClassId:(classid)
            CC "Core" CREATE_OBJ modelid:(tOUBPModelId) classid:(
sCANodeClassId) objname:("Receive " + (sInfluenceName))
            SET sNewCEid:(objid)

            IF (ecode != 0 AND ecode != 52)
                {
                    CC "AdoScript" INFOBOX "Error!\nModelgroup could
not be created. Was the name you entered unique?"
                }

```

```
            CC "Modeling" SET_OBJ_POS objid:(sNewCEid) x:(10.5cm) y:(ypos)
            CC "Core" SET_ATTR_VAL objid:(sNewCEid) attrname:"Code" val:("S2B-G2") ###S2B-GL2 ->"Generated by Stra2Bis transformaiton Guideline 2"

            ### Create the primary actor
            CC "Core" GET_CLASS_ID classname:"CAActor"
            SET sActorClassId:(classid)
            CC "Core" CREATE_OBJ modelid:(tOUBPModelId) classid:(sActorClassId) objname:((sSourceOUName)+ " Agent")
            SET sNewActorid:(objid)
            CC "Modeling" SET_OBJ_POS objid:(sNewActorid) x:(2cm) y:(ypos)

            ##### Primary actor is the support actor of the CE
            CC "Core" GET_ATTR_ID classid:(sCANodeClassId) attrname:"SupportActor"
            CC "Core" ADD_INTERREF objid:(sNewCEid) attrid:(attrid) tobjid:(sNewActorid) tmodelid:(tOUBPModelId)

            ### Create the ingoing communicative interaction
            CC "Core" GET_CLASS_ID classname:"IngoingInteraction"
            SET sIngoingingClassId:(classid)
            CC "Core" CREATE_CONNECTOR modelid:(tOUBPModelId) fromobjid:(sNewActorid) toobjid:(sNewCEid) classid:(sIngoingingClassId)
            SET sIngoingId:(objid)
            CC "Core" SET_ATTR_VAL objid:(sIngoingId) attrname:"Name" val:(sInfluenceName)

            CC "Modeling" REBUILD_DRAWING_AREA

            ### Traceability
            #### Add the CEvent to the in dependencies of the target OU
            CC "Core" GET_CLASS_ID objid:(iTargetOUid)
            CC "Core" GET_ATTR_ID classid:(classid) attrname:"In Dependencies Events"
            CC "Core" ADD_INTERREF objid:(iTargetOUid) attrid:(attrid) tobjid:(sNewCEid) tmodelid:(sCurrentModelId)


            CC "Core" GET_CLASS_ID objid:(iTargetOUid)
            CC "Core" GET_ATTR_ID classid:(classid) attrname:"In Dependencies Events"
            CC "Core" ADD_INTERREF objid:(iTargetOUid) attrid:(attrid) tobjid:(sNewCEid) tmodelid:(sCurrentModelId)


            #### Add for the CA Actor and their Organisatoin units
            CC "Core" GET_ATTR_ID classid:(sActorClassId) attrname:"Traced Organisation Unit in Dependency"
```

```
                CC "Core" ADD_INTERREF objid:(sNewActorid) attrid:(attrid
    ) tobjid:(pOUid) tmodelid:(sCurrentModelId)

            }

    }

    SET result:0
}

###GUIDELINE 3 ########
PROCEDURE TRANSFORM_OBJECTIVES pOUid:integer pBPModelGroupId:integer
    result:reference {

 ## Check if the OU has elements inside
        SET yGL3:0cm
        SET codeCounterGL3:0

        #Get spurce OU name
        CC "Core" GET_ATTR_VAL objid:(pOUid) attrname:"Name" as-
    string
        SET sSourceOUName:(val)

        ### GET ROLES INSIDE THE OU
        CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sSourceOUName + "
    \":\"OrganizationalUnit\"}<-\"Is inside\") OR ({\""+
    sSourceOUName + "\":\"OrganizationalUnit\"}->\"Is inside\")")
    modelid:(VAL sCurrentModelId)

        SET uniContOids:(objids)

        #check if OU contains Roles
        FOR uniContid_i in:(uniContOids) {
            CC "Core" GET_CLASS_ID objid:(VAL uniContid_i)
            CC "Core" GET_CLASS_NAME classid:(classid)



            IF (classname = "Role") {
                CC "Core" GET_OBJ_NAME objid:(VAL uniContid_i)
                SET sRoleId:(VAL uniContid_i)
                SET sRoleName:(objname)

                ### GET OBJECTIVES INSIDE THE ROLE
                CC "AQL" EVAL_AQL_EXPRESSION expr:("({\"" + sRoleName
     + "\":\"Role\"}<-\"Is inside\") OR ({\""+ sRoleName + "\":\"Role
    \"}->\"Is inside\")") modelid:(VAL sCurrentModelId)

                SET roleContOids:(objids)



                #check unit contents classes
```

```
                FOR roleContid_i in:(roleContOids) {


                    CC "Core" GET_CLASS_ID objid:(VAL roleContid_i)
                    CC "Core" GET_CLASS_NAME classid:(classid)
                    IF (classname = "Objective") {

                        SET sObjectiveId:(VAL roleContid_i)

                        SET yGL3:(yGL3 + 3cm)
                        SET codeCounterGL3:(codeCounterGL3+1)


                        CC "Core" GET_OBJ_NAME objid:(sObjectiveId)
                        SET sObjectiveName:(objname)

                        CC "Core" GET_ATTR_VAL objid:(sObjectiveId)
    attrname:"Indicator" as-string
                        SET sObjectiveIndicatorName:(val)

                        #Creates a communicative event (and its
    associated elements) to report the status of the objective
                        CREATE_REPORTING_EVENT roleId:(sRoleId)
    objecId:(sObjectiveId) ounitId:(pOUid) ypos:(yGL3) codeCounter:(
    codeCounterGL3) result:reportingEventId resultRoleActor:
    roleActorID resultOUActor:ouActorID


                        #Traceability:
                        ## From business to strategy
                        ### CE al objective que reporta
                        CC "Core" GET_CLASS_ID objid:(
    reportingEventId)
                        CC "Core" GET_ATTR_ID classid:(classid)
    attrname:"Traced Objective"
                        CC "Core" ADD_INTERREF objid:(
    reportingEventId) attrid:(attrid) tobjid:(sObjectiveId) tmodelid
    :(sCurrentModelId)

                        ### Actores a sus units/roles
                        CC "Core" GET_CLASS_ID objid:(roleActorID)
                        CC "Core" GET_ATTR_ID classid:(classid)
    attrname:"Traced Reporting Role"
                        CC "Core" ADD_INTERREF objid:(roleActorID)
    attrid:(attrid) tobjid:(sRoleId) tmodelid:(sCurrentModelId)

                        CC "Core" GET_CLASS_ID objid:(ouActorID)
                        CC "Core" GET_ATTR_ID classid:(classid)
    attrname:"Traced Informed Organisation Unit"
                        CC "Core" ADD_INTERREF objid:(ouActorID)
    attrid:(attrid) tobjid:(pOUid) tmodelid:(sCurrentModelId)

                        ### Role a su CA ACtor
```

```
                        CC "Core" GET_CLASS_ID objid:(sRoleId)
                        CC "Core" GET_ATTR_ID classid:(classid) attrname:"Traced Communicative Actor"
                        CC "Core" ADD_INTERREF objid:(sRoleId) attrid:(attrid) tobjid:(roleActorID) tmodelid:(sCurrentModelId)

                        ## de S a B
                        ### Objective to CE
                         CC "Core" GET_CLASS_ID objid:(sObjectiveId)
                        CC "Core" GET_ATTR_ID classid:(classid) attrname:"Traced Communicative Event"
                        CC "Core" ADD_INTERREF objid:(sObjectiveId) attrid:(attrid) tobjid:(reportingEventId) tmodelid:(sCurrentModelId)

                    }
                }

            }
        }

    SET result:0
}

  PROCEDURE CREATE_REPORTING_EVENT roleId:integer objecId:integer ounitId:integer ypos:measure codeCounter:integer result:reference resultRoleActor:reference resultOUActor:reference {

    ####Position contants

    ### Traced element names:
    CC "Core" GET_OBJ_NAME objid:(objecId)
    SET sObjectiveName:(objname)

    ### Objectives Indicastor name, for short.
    CC "Core" GET_ATTR_VAL objid:(objecId) attrname:"Indicator" as-string
    SET sObjectiveIndicatorName:(val)

    CC "Core" GET_OBJ_NAME objid:(roleId)
    SET sRoleName:(objname)

    CC "Core" GET_OBJ_NAME objid:(ounitId)
    SET sOUName:(objname)


    ### Get the Traced BP's of the source OU
    CC "Core" GET_INTERREF  objid:(ounitId) attrname:"Traced Business Process" index:0
    SET sOUBPModelId:(tmodelid)
    #CC "AdoScript" INFOBOX ("sOUBPModelId" + (STR sOUBPModelId))
    CC "Modeling" ACTIVATE_MODEL (sOUBPModelId)

```

```
    ### Create the communicative event

    CC "Core" GET_CLASS_ID classname:"CommunicativeEvent"
    SET sCANodeClassId:(classid)
    CC "Core" CREATE_OBJ modelid:(sOUBPModelId) classid:(sCANodeClassId) objname:("Report " + (sObjectiveIndicatorName) + " Status")
    SET sNewCEid:(objid)

    IF (ecode != 0 AND ecode != 52)
        {
            CC "AdoScript" INFOBOX "Error!\nModelgroup could not be created. Was the name you entered unique?"
        }

    CC "Modeling" SET_OBJ_POS objid:(sNewCEid) x:(10.5cm) y:(ypos)
    CC "Core" SET_ATTR_VAL objid:(sNewCEid) attrname:"Code" val:("S2B-G3"+"-0"+(STR codeCounter)) ###S2B-GL2 ->"Generated by Stra2Bis transformaiton Guideline 2"


    ### Create the primary actor named after the role.
    CC "Core" GET_CLASS_ID classname:"CAActor"
    SET sActorClassId:(classid)

    #### Check if there are actors that reference the same Role
    CC "Core" GET_ATTR_ID classid:(sActorClassId) attrname:"Name"
    CC "Core" GET_ALL_OBJS_WITH_ATTR_VAL modelid:(sOUBPModelId) classid:(sActorClassId) id attrid:(attrid) val:(sRoleName)
    SET rActorRolesIds:(objids)

    #### Check if there are actors, use the existing actor as Primary Actor; else, create it.
        IF (rActorRolesIds != "") {
            SET sRoleActorId:(VAL rActorRolesIds)
        }
        ELSE {
            CC "Core" CREATE_OBJ modelid:(sOUBPModelId) classid:(sActorClassId) objname:(sRoleName)
            SET sRoleActorId:(objid)
        }


    CC "Modeling" SET_OBJ_POS objid:(sRoleActorId) x:(2cm) y:(ypos)


    ### Create the receiver actor named after the organisation unit
    #### Check if there are actors that reference the same Organisation Unit
    CC "Core" GET_ALL_OBJS_WITH_ATTR_VAL modelid:(sOUBPModelId) classid:(sActorClassId) id attrid:(attrid) val:(sOUName + " Agent")
    SET rActorOUIds:(objids)
```

```

    #### Check if there are actors, use the existing actor as Primary Actor; else, create it.

        IF (rActorOUIds != "") {

            SET sOUActorId:(VAL rActorOUIds)
        }
        ELSE {
            CC "Core" CREATE_OBJ modelid:(sOUBPModelId) classid:(sActorClassId) objname:(sOUName + " Agent")
            SET sOUActorId:(objid)
        }


    CC "Modeling" SET_OBJ_POS objid:(sOUActorId) x:(20cm) y:(ypos)


    ##### Receiver actor is the support actor of the CE
    CC "Core" GET_ATTR_ID classid:(sCANodeClassId) attrname:"SupportActor"
    CC "Core" ADD_INTERREF objid:(sNewCEid) attrid:(attrid) tobjid:(sOUActorId) tmodelid:(sCurrentModelId)

    ### Create the ingoing communicative event
    CC "Core" GET_CLASS_ID classname:"IngoingInteraction"
    SET sIngoingClassId:(classid)
    CC "Core" CREATE_CONNECTOR modelid:(sOUBPModelId) fromobjid:(sRoleActorId) toobjid:(sNewCEid) classid:(sIngoingClassId)
    SET sIngoingId:(objid)
    CC "Core" SET_ATTR_VAL objid:(sIngoingId) attrname:"Name" val:((sObjectiveIndicatorName) + " Status")

    ### Create the outgoing communicative event
    CC "Core" GET_CLASS_ID classname:"OutgoingInteraction"
    SET sOutgoingClassId:(classid)
    CC "Core" CREATE_CONNECTOR modelid:(sOUBPModelId) fromobjid:(sNewCEid) toobjid:(sOUActorId) classid:(sOutgoingClassId)
    SET sOutgoingId:(objid)
    CC "Core" SET_ATTR_VAL objid:(sOutgoingId) attrname:"Name" val:((sObjectiveIndicatorName) + " Status")

    CC "Modeling" REBUILD_DRAWING_AREA

    SET result:(sNewCEid)
    SET resultRoleActor:(sRoleActorId)
    SET resultOUActor:(sOUActorId)
}
```

Listing A.2: ADOxx script for Stra2Bis transformation guidelines.

Appendix B

LiteStrat Validation Experimental Materials

Table B.1: Guidelines for the semantic inspection for Problem 1

B.1 Semantic Inspection Guidelines

Type	Id	Domain elements	i* Concepts	LiteStrat Concepts
Motivation	P1R01	1. BTR	Agent or Actor	Organisation Unit
Motivation	P1R02	2. WOW	Agent or Actor	Actor
		3. "new marketing campaign"	Dependency (2->1)	influence (1->2)
Motivation	P1R03	4. Retain customers	Goal	Goal
Action	P1R04	5. "End billing errors"	Task inside (1), refined from (4)	Strategy inside (1), refined from (4)
		6. "Detailed, transparent, and timely billing information"	Task inside (1), refined from (4)	Strategy inside (1), refined from (4)
		7. Billing Department	Agent or Actor that participates in (1)	Organisation Unit inside (1)
Action	P1R05	8. "reduce the time needed to process invoices"	Dependency (5->7)	Tactic refined from (5) inside (7)
		9. "add a set of quality control activities to the process."	Dependency (5->7)	Tactic refined from (5) inside (7)
Action	P1R06	10. Automatic Billing Validation	Dependency (6->7)	Tactic refined from (6) inside (7)
		11. Automatic Billing Publication	Dependency (6->7)	Tactic refined from (6) inside (7)
Role and Resp.	P1R07	12. Billing Manager	Role that participates in (7)	Role inside (7)
		13. Reduce the billing time by 3 days	Dependency (7->12)	Objective refined from (8) inside (12)
Role and Resp.	P1R08	14. Quality Manager	Role that participates in (7)	Role inside (7)
		15. "Check and correct billing within 3 days"	Dependency (7->14)	Objective refined from (9) inside (14)
Role and Resp.	P1R09	16. "at least 25% of bills must be audited"	Dependency (7->14)	Objective refined from (9) inside (14)
Role and Resp.	P1R10	17. Validation Manager	Role that participates in (7)	Role inside (7)
		18. "the bills must be validated no later than 12 hours after they have been made."	Dependency (7->17)	Objective refined from (10) inside (17)
		19. "the publication of the bills in the app should be instantaneous"	Dependency (7->17)	Objective refined from (11) inside (17)
Outcome	P1R11	20.Customers	Actor	Actor
		21. "re-engage your existing customers through better billing service"	Dependency (20->7)	Influence (7->7)
Outcome	P1R12	22. Marketing Area	Agent or Actor	Actor
		23."inform customers about the new service"	Dependency (22->20)	Influence (22->20)

Table B.2: Guidelines for the semantic inspection for Problem 2

Type	Id	Domain Elements	i* Concepts	LiteStrat Concepts
Motivation	P2R01	1. Short Life	Agent or Actor	Organisation Unit
Motivation	P2R02	2. Insurance Regulator	Agent or Actor	Actor
		3. "add home theft insurance to car insurance. "	Dependency (2->1)	Influence (1->2)
Motivation	P2R03	4. "advertise home burglary insurance without recourse to third parties"	Goal inside (1)	Goal inside (1)
Action	P2R04	5. Marketing Department	Agent or Actor that participates in (1)	Organisation Unit inside (1)
		6. Product Design Department	Agent or Actor that participates in (1)	Organisation Unit inside (1)
		7. Design new product	Task refined from (4) inside (1)	Strategy refined from (4) inside (1)
		8. Advertise new product	Task refined from (4) inside (1)	Strategy refined from (4) inside (1)
Action	P2R05	9. Customer Service Department	Agent or Actor that participates in (1)	Organisation Unit inside (1)
		10. "Contact existing customers to inform them of the new product"	Dependency (1->9)	Tactic refined from (8) inside (9)
Action	P2R07	11. Design product marketing	Dependency (1->5)	Tactic refined from (7) inside (6)
		12. provide information from customer and competitor studies	Dependency (1->6)	Tactic refined from (7) inside (6)
Role and Resp.	P2R06	13. Head of Customer Service Department	Role that participates in (9)	Role inside (9)
		14. "inform 70% of customers by telephone within the first week of a new product release."	Dependency (9->13)	Objective refined from (10) inside (12)
Role and Resp.	P2R08	15. Head of Marketing	Role that participates in (5)	Role inside (5)
		16. "advertising campaign two weeks in advance to product launch"	Dependency (5->15)	Objective refined from (11) inside (15)
Role and Resp.	P2R09	17. Lead Publicist	Role that participates in (5)	Role inside (5)
		18."advertising should reach at least 20% of the market that is not yet a Short Life customer".	Dependency (5->17)	Objective refined from (11) inside (16)
Role and Resp.	P2R10	19. Market Analyst	Role that participates in (6)	Role inside (6)
		20."carry out the customer and competitor study within a maximum of 20 working days"	Dependency (6->19)	Objective refined from (11) inside (19)
Outcome	P2R11	21.Customers	Agent or Actor	Actor
		22."offer the new theft insurance service"	Dependency (21->19)	Influence (19->21)
Outcome	P2R12	23.New Customers	Agent or Actor	Agent or Actor
		24."offer the new theft insurance service"	Dependency (23->19)	Influence (5->23)

B.2 Solution Examples for Experimental Problems

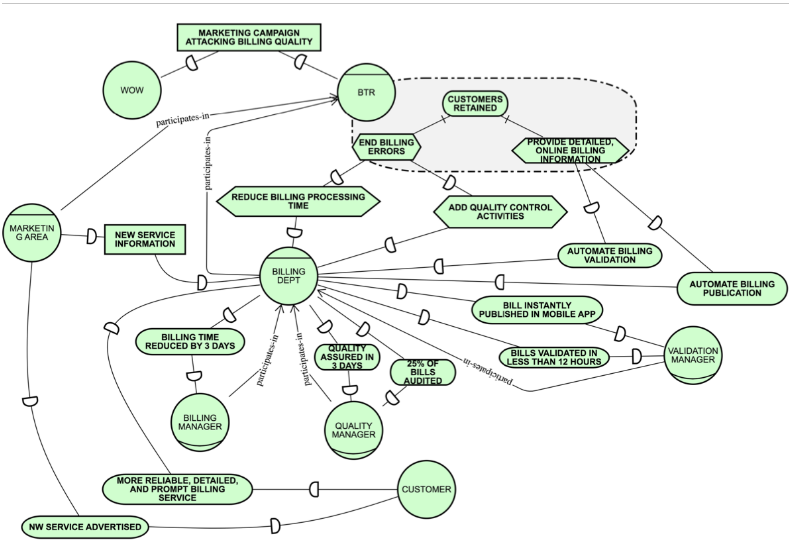


Figure B.1: Solution example for Problem 1 using i*.

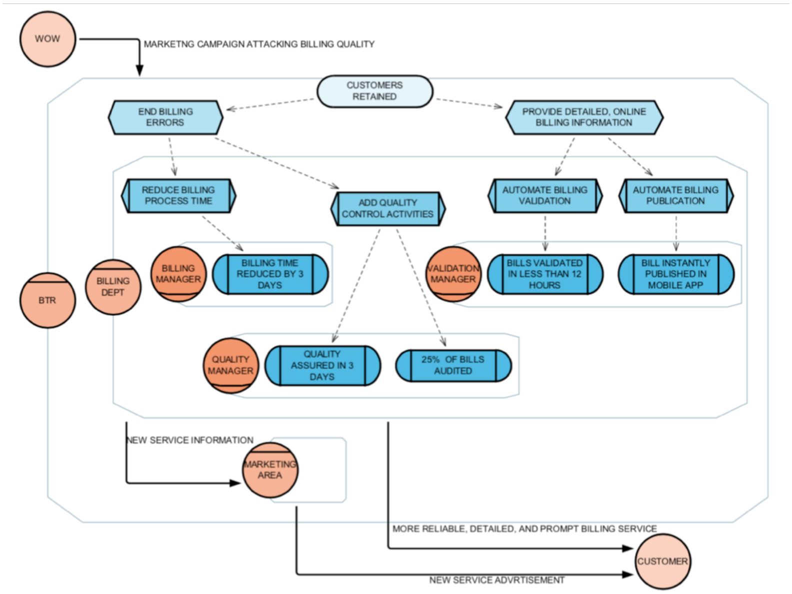


Figure B.2: Solution example for Problem 1 using LiteStrat.

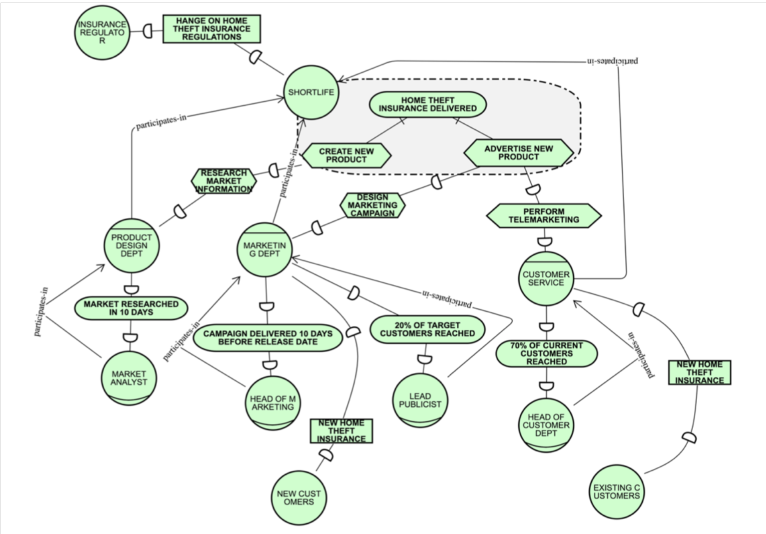


Figure B.3: Solution example for Problem 2 using i*.

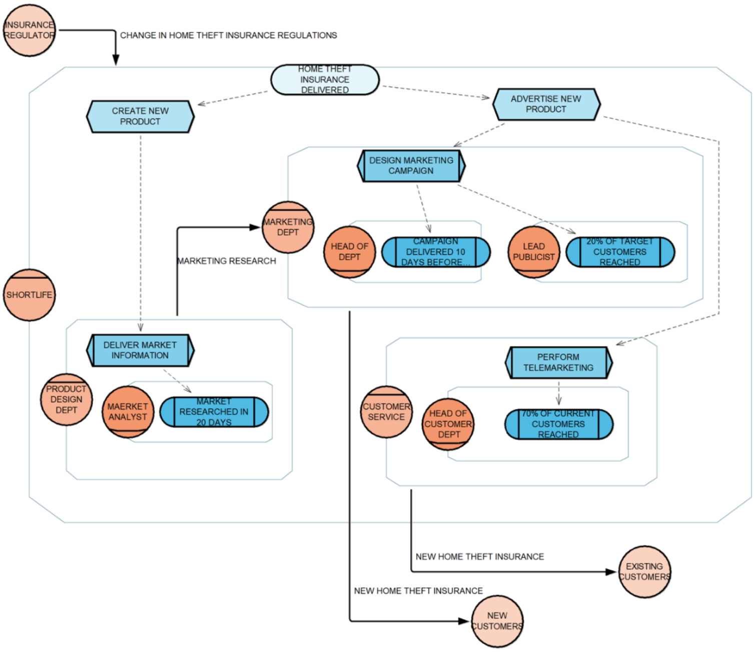


Figure B.4: Solution example for Problem 2 using LiteStrat.

Appendix C

Stra2Bis Validation Experimental Materials

C.1 Grading Schemes for Completeness and Validity Metrics

Table C.1: Grading scheme for completeness metrics.

Process Completeness (Range: 0-4)			
Criteria	**Achieved (2 points)**	**Partially Achieved (1 point)**	**Not Achieved (0 points)**
Actor Completeness	All the actors involved in the re-engineered process are modelled.	More than half of the actors involved in the re-engineered process are modelled.	Less than a half of the actors involved in the re-engineered process are not modelled.
Communicative Event Completeness	All the communicative events needed to re-engineer the process are modelled.	More than half of the communicative events needed to re-engineer the process are modelled.	Less than a half of the communicative events needed to re-engineer the process are not modelled.

Strategy Completeness (Range: 0-8)			
Criteria	**Achieved (2 points)**	**Partially Achieved (1 point)**	**Not Achieved (0 points)**
Objectives Completeness	All the strategic objectives involved in the re-engineered process have communicative events to report their status.	More than half of the strategic objectives involved in the re-engineered process have communicative events to report their status.	There are no communicative events that can be associated with reporting the status of strategic objectives.
Tactics Completeness	All the tactics of the strategy related to re-engineered processes are addressed by the creation, modification, or deletion of communicative events.	More than half of the tactics of the strategy related to re-engineered processes are addressed by the creation, modification, or deletion of communicative events.	Less than half of the tactics of the strategy related to re-engineered processes are addressed by the creation, modification, or deletion of communicative events.
Structure Completeness	There is a communicative event diagram for each organisational unit that requires to re-engineer their processes.	More than one organisational units have a communicative event for their re-engineered processes.	There is a unique process model for all the organisational units.
Role Completeness	All the roles reporting strategic objectives in the organisational model are modelled as actors in the business process model.	More than half the roles reporting strategic objectives in the organisational model are modelled as actors in the business process model.	Less than half the roles reporting strategic objectives in the organisational model are modelled as actors in the business process model.

Table C.2: Grading scheme for validity metrics.

Process Validity (Range: 0-4			
Criteria	**Achieved (2 points)**	**Partially Achieved (1 point)**	**Not Achieved (0 points)**
Core Business Logic: The re-engineered business process model preserves the core business logic.	The input, output, and processing elements of the initial business process are validly modelled in the re-engineered model.	Half or more of the input, output, and processing elements of the initial business process are validly modelled.	Less than half of the input, output, and processing elements of the initial business process are validly modelled.
Business Logic Update: The re-engineered business process model has business logic changes that are required in the strategic scenario.	All the new business logic requirements are modelled in the re-engineered business process model.	Half or more of the new business logic requirements are modelled in the re-engineered business process model.	Less than half of the new business logic requirements are modelled in the re-engineered business process model.
Strategy Validity (Range: 0-6)			
Criteria	**Achieved (2 points)**	**Partially Achieved (1 point)**	**Not Achieved (0 points)**
Guideline 1 Application: Each organisation unit has its own business process.	All the organisation units have their respective business process represented by a named process start node.	At least one of the organisation units have a business process represented by a named process start node.	None of the organisation units have a business process represented by a named process start node.
Guideline 2 Application: the dependencies between organisation units are represented as interaction between actors in the business process model.	The business process model has events to connect different processes from different units, and actors that belong to different units.	The business process model has events to communicate actors that belong to different units.	No validly interaction between actors from different organisation units are modelled.
Guideline 3 Application: the business process model has events to collect information regarding the strategic objectives.	All the objectives in the business strategy model have a event to collect information about its status in the new business process model.	At least one of the objectives in the business strategy model have a event to collect information about its status in the new business process model.	None of the objectives in the business strategy model have a event to collect information about its status in the new business process model.

Bibliography

Abrahao, Silvia, Emilio Insfran, Fernando Gonzalez-Ladron-de-Guevara, Marta Fernandez-Diego, Carlos Cano-Genoves, and Raphael Pereira de Oliveira (2019). "Assessing the effectiveness of goal-oriented modeling languages: A family of experiments". In: *Information and Software Technology* 116, p. 106171 (cit. on pp. 171, 174, 175, 221).

Abrahão, Silvia, Emilio Insfran, Fernando González-Ladrón de Guevara, Marta Fernández-Diego, Carlos Cano-Genoves, and Raphael Pereira de Oliveira (2018). "Comparing the effectiveness of goal-oriented languages: results from a controlled experiment". In: *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 1–4 (cit. on pp. 171, 174, 175, 221).

Aghina, Wouter, Karin Ahlback, Aaron De Smet, Gerald Lackey, Michael Lurie, Monica Murarka, and Christopher Handscomb (2017). *Agile Strategy*. URL: `https://www.agilebusiness.org/business-agility/fba/agile-strategy.html` (cit. on pp. 10, 14, 108).

Akhigbe, Okhaide, Daniel Amyot, Amal Ahmed Anda, Lysanne Lessard, and Daoyang Xiao (2016). "Consistency Analysis for User Requirements Notation Models." In: *iStar*, pp. 43–48 (cit. on pp. 48, 52).

Alongi, Francesco, Marcello M Bersani, Nicolò Ghielmetti, Raffaela Mirandola, and Damian A Tamburri (2022). "Event-sourced, observable software architectures: An experience report". In: *Software: Practice and Experience* 52.10, pp. 2127–2151 (cit. on p. 21).

Alter, SL (1995). "How should business professionals analyze information systems for themselves?" In: *Information System Concepts: Towards a consolidation of views*, pp. 284–299 (cit. on p. 65).

Alter, Steven (2008). "Defining information systems as work systems: implications for the IS field". In: *European journal of information systems* 17.5, pp. 448–469 (cit. on p. 65).

— (2013). "Work system theory: overview of core concepts, extensions, and challenges for the future". In: *Journal of the Association for Information Systems*, p. 72 (cit. on pp. xv, xvi, 7, 8, 46, 64–67, 70, 266).

Amyot, Daniel, Okhaide Akhigbe, Malak Baslyman, Sepideh Ghanavati, Mahdi Ghasemi, Jameleddine Hassine, Lysanne Lessard, Gunter Mussbacher, Kai Shen, and Eric Yu (2022). "Combining Goal modelling with Business Process modelling". In: *Enterprise Modelling and Information Systems Architectures (EMISAJ)* 17, pp. 2–1 (cit. on pp. 48, 52, 54, 142, 233, 235, 237).

Atlassian (2021). *The Spotify model.* `https://www.atlassian.com/agile/agile-at-scale/spotify`. (Accessed on 05/31/2021) (cit. on pp. 108, 148).

Basili, Victor R and H Dieter Rombach (1988). "The TAME project: Towards improvement-oriented software environments". In: *IEEE Transactions on software engineering* 14.6, pp. 758–773 (cit. on pp. 169, 232).

Beekun, Rafik I and William H Glick (2001). "Organization structure from a loose coupling perspective: A multidimensional approach". In: *Decision sciences* 32.2, pp. 227–250 (cit. on p. 16).

Belaunde, Mariano, Cory Casanave, Desmond DSouza, Keith Duddy, William El Kaim, Alan Kennedy, William Frank, David Frankel, Randall Hauch, Stan Hendryx, et al. (2003). *MDA Guide Version 1.0. 1* (cit. on p. 4).

Bernabé, César Henrique, Vítor E Silva Souza, Ricardo de Almeida Falbo, Renata SS Guizzardi, and Carla Silva (2019). "GORO 2.0: Evolving an ontology for goal-oriented requirements engineering". In: *Advances in Conceptual Modeling: ER 2019 Workshops FAIR, MREBA, EmpER, MoBiD, OntoCom, and ER Doctoral Symposium Papers, Salvador, Brazil, November 4–7, 2019, Proceedings 38*. Springer, pp. 169–179 (cit. on pp. 64, 77).

Bērziša, Solvita, George Bravos, Tania Gonzalez, et al. (2015). "Capability driven development: an approach to designing digital enterprises". In: *Business & Information Systems Engineering* 57.1, pp. 15–25 (cit. on p. 142).

Bharadwaj, Anandhi, Omar A El Sawy, Paul A Pavlou, and N v Venkatraman (2013). "Digital business strategy: toward a next generation of insights". In: *MIS quarterly*, pp. 471–482 (cit. on p. 13).

Bresciani, Paolo, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos (2004). "Tropos: An agent-oriented software development methodology". In: *Autonomous Agents and Multi-Agent Systems* 8.3, pp. 203–236 (cit. on pp. xvi, 42, 44, 141, 234).

Brinkkemper, Sjaak (1996). "Method engineering: engineering of information systems development methods and tools". In: *Information and software technology* 38.4, pp. 275–280 (cit. on p. 223).

Brosseau, Daniel, Sherina Ebrahim, Christopher Handscomb, and Shail Thaker (2019). *The journey to an agile organization | McKinsey*. `https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/the-journey-to-an-agile-organization`. (Accessed on 08/16/2023) (cit. on p. 12).

Brown, Alan W (2004). "Model driven architecture: Principles and practice". In: *Software and systems modeling* 3, pp. 314–327 (cit. on p. 4).

Buhr, Ray JA and Ron S Casselman (1995). *Use case maps for object-oriented systems*. USA: Prentice-Hall, Inc. (cit. on pp. 50, 52, 141).

Cano-Genoves, Carlos, Emilio Insfran, and Silvia Abrahão (2022). "A value-based goal model analysis tool". In: *Proceedings of the 25th International*

Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, pp. 32–36 (cit. on p. 49).

Chung, Lawrence, Brian A Nixon, Eric Yu, and John Mylopoulos (2012). *Non-functional requirements in software engineering*. Vol. 5. Springer Science & Business Media (cit. on p. 42).

Conway, Melvin E (1968). "How do committees invent". In: *Datamation* 14.4, pp. 28–31 (cit. on pp. xv, 12, 16, 80, 108).

Cortes-Cornax, M, Alexandru Matei, Sophie Dupuy-Chessa, Dominique Rieu, Nadine Mandran, and Emmanuel Letier (2015). "Using intentional fragments to bridge the gap between organizational and intentional levels". In: *Information and Software Technology* 58, pp. 1–19 (cit. on pp. 48, 52).

Cortes-Cornax, Mario, Alexandru Matei, Emmanuel Letier, Sophie Dupuy-Chessa, and Dominique Rieu (2012). "Intentional fragments: bridging the gap between organizational and intentional levels in business processes". In: *On the Move to Meaningful Internet Systems: OTM 2012: Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012, Rome, Italy, September 10-14, 2012. Proceedings, Part I*. Springer, pp. 110–127 (cit. on p. 54).

Dalpiaz, Fabiano, Xavier Franch, and Jennifer Horkoff (2016). "iStar 2.0 Language Guide". In: *CoRR* abs/1605.07767. arXiv: 1605.07767. URL: http://arxiv.org/abs/1605.07767 (cit. on pp. xv, xvi, 38–41, 112, 118, 119, 177).

Dardenne, Anne, Axel Van Lamsweerde, and Stephen Fickas (1993). "Goal-directed requirements acquisition". In: *Science of computer programming* 20.1-2, pp. 3–50 (cit. on pp. 43, 141, 234).

Davis, FD (1989). "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". In: *MIS Quarterly* 13.3, pp. 319–340 (cit. on p. 174).

Doerr, John (2018). *Measure what matters: How Google, Bono, and the Gates Foundation rock the world with OKRs*. Penguin (cit. on pp. 14, 107, 149, 159).

Domingo, África, Jorge Echeverría, Oscar Pastor, and Carlos Cetina (2020). "Evaluating the benefits of model-driven development". In: *International Conference on Advanced Information Systems Engineering*. Cham: Springer, pp. 353–367 (cit. on p. 258).

Dybå, Tore, Vigdis By Kampenes, and Dag I. K. Sjøberg (2006). "A systematic review of statistical power in software engineering experiments". In: *Information and Software Technology* 48.8, pp. 745–755. ISSN: 0950-5849. DOI: `http://dx.doi.org/10.1016/j.infsof.2005.08.009` (cit. on pp. 188, 198, 202, 244).

Embley, David W, Stephen W Liddle, and Oscar Pastor (2011). "Conceptual-model programming: a manifesto". In: *Handbook of Conceptual Modeling*. Springer, pp. 3–16 (cit. on p. 3).

Engelsman, Wilco, Jaap Gordijn, Timber Haaker, Marten van Sinderen, and Roel Wieringa (2021). "Quantitative Alignment of Enterprise Architectures with the Business Model". In: *International Conference on Conceptual Modeling*. Springer, pp. 189–198 (cit. on p. 142).

Engelsman, Wilco, Dick Quartel, Henk Jonkers, and Marten van Sinderen (2011). "Extending enterprise architecture modelling with business goals and requirements". In: *Enterprise Information Systems* 5.1, pp. 9–36 (cit. on p. 57).

España, Sergio (Dec. 1, 2011). "Methodological Integration of Communication Analysis into a Model-Driven Software Development Framework". PhD thesis. Valencia (Spain): Universitat Politècnica de València (cit. on pp. xvii, 4, 22, 27, 59, 60, 93, 96, 143–145, 151, 152, 157).

España, Sergio, Arturo González, and Oscar Pastor (2009). "Communication Analysis: a requirements engineering method for information systems". In: *International Conference on Advanced Information Systems Engineering*. Springer, pp. 530–545 (cit. on pp. 22, 27, 31, 49, 59, 139, 141, 143, 234).

Estrada, Hugo, Alicia Martínez Rebollar, Oscar Pastor, and John Mylopoulos (2006). "An empirical evaluation of the i* framework in a model-based software generation environment". In: *International Conference on Advanced Information Systems Engineering*. Springer, pp. 513–527 (cit. on pp. 84, 90, 96).

Evans, Eric and Eric J Evans (2004). *Domain-driven design: tackling complexity in the heart of software.* Addison-Wesley Professional (cit. on pp. 18, 107, 148).

Falessi, Davide, Natalia Juristo, Claes Wohlin, Burak Turhan, Jürgen Münch, Andreas Jedlitschka, and Markku Oivo (2018). "Empirical software engineering experts on the use of students and professionals in experiments". In: *Empirical Software Engineering* 23.1, pp. 452–489 (cit. on pp. 169, 221, 227, 229, 232, 245, 260).

Falkenberg, Eckhard D, Wolfgang Hesse, Paul Lindgreen, Bjorn E Nilsson, JL Han Oei, and Colette et al. Rolland (1998). "A framework of information system concepts". In: *International Federation for Information Processing.* Citeseer (cit. on pp. 59, 61, 63).

Fellmann, Michael, Agnes Koschmider, Ralf Laue, Andreas Schoknecht, and Arthur Vetter (2020). "Business Process Model Patterns: Classification and Repository of Published Work". In: *40 Years EMISA 2019.* Ed. by Heinrich C. Mayr, Stefanie Rinderle-Ma, and Stefan Strecker. Bonn: Gesellschaft für Informatik e.V., pp. 67–76 (cit. on p. 94).

Fonseca, Claudenir M, Daniele Porello, Giancarlo Guizzardi, João Paulo A Almeida, and Nicola Guarino (2019). "Relations in ontology-driven conceptual modeling". In: *Conceptual Modeling: 38th International Conference, ER 2019, Salvador, Brazil, November 4–7, 2019, Proceedings 38.* Springer, pp. 28–42 (cit. on p. 74).

Forsgren, Nicole, Jez Humbpotifle, and Gene Kim (2018). *Accelerate: The Science of Lean Software and DevOps Building and Scaling High Performing Technology Organizations* (cit. on pp. 5, 10–12, 15, 16, 106, 108, 109, 147, 148).

Fowler, Martin (2022). *Conway's Law.* `https://martinfowler.com/bliki/ConwaysLaw.html`. (Accessed on 04/18/2023) (cit. on p. 108).

Frakes, William B and R. Baeza-Yates (1992). *Information retrieval: Data structures & algorithms.* Prentice-Hall, Inc. (cit. on p. 171).

Giachetti, Giovanni, Francisco Valverde, and Beatriz Marín (2012). "Interoperability for model-driven development: Current state and future challenges". In: *2012 Sixth International Conference on Research Challenges in Information Science (RCIS)*. IEEE, pp. 1–10 (cit. on pp. 59, 61).

Girod, Stéphane JG and Richard Whittington (2017). "Reconfiguration, restructuring and firm performance: Dynamic capabilities and environmental dynamism". In: *Strategic Management Journal* 38.5, pp. 1121–1133 (cit. on pp. 17, 20).

Glinz, Martin (2007). "On non-functional requirements". In: *15th IEEE international requirements engineering conference (RE 2007)*. IEEE, pp. 21–26 (cit. on pp. 20, 49, 52, 55).

Gonçalves, Enyo, Jaelson Castro, João Araújo, and Tiago Heineck (Mar. 2018). "A Systematic Literature Review of iStar extensions". In: *Journal of Systems and Software* 137, pp. 1–33. ISSN: 01641212. DOI: 10.1016/j.jss.2017.11.023. (Visited on 04/08/2020) (cit. on p. 99).

Gonzalez, Arturo, Sergio Espana, and Oscar Pastor (2009). "Unity criteria for business process modelling". In: *2009 Third International Conference on Research Challenges in Information Science*. IEEE, pp. 155–164 (cit. on p. 59).

Gordijn, Jaap and JM Akkermans (2003). "Value-based requirements engineering: exploring innovative e-commerce ideas". In: *Requirements engineering* 8.2, pp. 114–134 (cit. on p. 141).

Gröner, Gerd, Mohsen Asadi, Bardia Mohabbati, Dragan Gašević, Marko Bošković, and Fernando Silva Parreiras (2014). "Validation of user intentions in process orchestration and choreography". In: *Information Systems* 43, pp. 83–99 (cit. on pp. 48, 52, 54, 55, 141, 142, 233, 235, 237).

Guizzardi, Giancarlo (2005). "Ontological foundations for structural conceptual models". In: ed. by Centre for Telematics CTIT and Information Technology (cit. on p. 70).

— (2013). "Ontology-based evaluation and design of visual conceptual modeling languages". In: *Domain Engineering: Product Lines, Languages, and Conceptual Models*, pp. 317–347 (cit. on p. 63).

Guizzardi, Giancarlo, Alessander Botti Benevides, Claudenir M Fonseca, Daniele Porello, João Paulo A Almeida, and Tiago Prince Sales (2022). "UFO: Unified foundational ontology". In: *Applied ontology* 17.1, pp. 167–210 (cit. on pp. 63, 70, 71, 73, 74, 266).

Guizzardi, Giancarlo and Gerd Wagner (2004). "A Unified Foundational Ontology and some Applications of it in Business Modeling." In: *CAiSE Workshops (3)*, pp. 129–143 (cit. on pp. 54, 64, 71, 73, 74).

Guizzardi, Renata and Ariane Nunes Reis (2015). "A method to align goals and business processes". In: *International Conference on Conceptual Modeling*. Springer, pp. 79–93 (cit. on pp. 48, 51, 140–142, 234–237, 256, 257).

Habba, Maryam, Mounia Fredj, and Samia Benabdellah Chaouni (2017). "Towards an operational alignment approach for organizations". In: *Proceedings of the 9th International Conference on Information Management and Engineering*, pp. 29–34 (cit. on p. 46).

Habba, Maryam, Mounia Fredj, and Samia Benabdellah Chaouni (2019). "Alignment between business requirement, business process, and software system: a systematic literature review". In: *Journal of Engineering* (cit. on pp. 6, 8).

Harvard Business Review (2015). *The business case for purpose*. https://pubs.opengroup.org/architecture/archimate3-doc/index.html. (Accessed on 04/13/2023) (cit. on p. 78).

Henderson, John C and Harihara Venkatraman (1999). "Strategic alignment: Leveraging information technology for transforming organizations". In: *IBM systems journal* 38.2.3, pp. 472–484 (cit. on pp. 13, 37).

Henderson-Sellers, Brian and Jolita Ralyté (2010). "Situational method engineering: state-of-the-art review". In: *Journal of Universal Computer Science* (cit. on pp. 127, 220).

Henderson-Sellers, Brian, Jolita Ralyté, Pär J. Ågerfalk, and Matti Rossi (2014). *Situational Method Engineering*. en. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-41466-4 978-3-642-41467-1. DOI: 10.1007/978-3-642-41467-1. URL: http://link.springer.com/10.1007/978-

3-642-41467-1 (visited on 06/12/2020) (cit. on pp. xv, 4, 24–27, 45, 106, 112, 120, 140, 143, 146, 223, 267).

Hevner, Alan and Samir Chatterjee (2010). "Design Science Research in Information Systems". en. In: *Design Research in Information Systems*. Vol. 22. Series Title: Integrated Series in Information Systems. Boston, MA: Springer US, pp. 9–22. ISBN: 978-1-4419-5652-1 978-1-4419-5653-8. DOI: 10.1007/978-1-4419-5653-8_2. URL: http://link.springer.com/10.1007/978-1-4419-5653-8_2 (visited on 02/15/2023) (cit. on p. 24).

Highsmith, Jim, Linda Luu, and David Robinson (2019). *EDGE: Value-Driven Digital Transformation*. Addison-Wesley Professional (cit. on pp. 14, 20, 106, 107, 109, 148, 149).

Hohpe, Gregor and Bobby Woolf (2004). *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional (cit. on p. 18).

Holbeche, Linda (2018). *The agile organization: how to build an engaged, innovative and resilient business*. 2 Edition. ISBN: 978-0-7494-8266-4 (cit. on pp. 13, 14, 21, 107).

Horita, Hiroki, Hideaki Hirayama, Takeo Hayase, Yasuyuki Tahara, and Akihiko Ohsuga (2019). "A method for goal model repair based on process mining". In: *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE, pp. 121–126 (cit. on pp. 48, 56).

Horita, Hiroki, Kozo Honda, Yuichi Sei, Hiroyuki Nakagawa, Yasuyuki Tahara, and Akihiko Ohsuga (2014). "Transformation approach from KAOS goal models to BPMN models using refinement patterns". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 1023–1024 (cit. on pp. 48, 50, 56).

Horwath, Rich (2014). *Elevate: the three disciplines of advanced strategic thinking*. John Wiley & Sons (cit. on pp. 15, 108, 123).

Hsia, Pei, Alan M Davis, and David Chenho Kung (1993). "Status report: requirements engineering". In: *IEEE software* 10.6, pp. 75–79 (cit. on p. 37).

Huang, Shuangxi and Yushun Fan (2007). "Model Driven and Service Oriented Enterprise Integration—The Method, Framework and Platform". In: *Sixth International Conference on Advanced Language Processing and Web Information Technology (ALPIT 2007)*. IEEE, pp. 504–509 (cit. on p. 7).

Ibriwesh, Iyas, Sin-Ban Ho, Ian Chai, and Chuie-Hong Tan (2017). "A controlled experiment on comparison of data perspectives for software requirements documentation". In: *Arabian Journal for Science and Engineering* 42.8, pp. 3175–3189 (cit. on pp. 173, 175).

Insfrán, Emilio, Silvia Abrahão, Raphael Pereira de Oliveira, Fernando González-Ladrón-de-Guevara, Marta Fernández-Diego, and Carlos Cano-Genoves (2017). "Specifying Value in GRL for Guiding BPMN Activities Prioritization". In: *Procedings of the International Conference on Information Systems Development* (cit. on pp. 7, 48, 49, 54, 141, 142, 233, 235, 237).

Integranova Software Solutions (2023). *Integranova Software Solutions | Integranova Software Solutions.* URL: http://www.integranova.com/es/ (visited on 06/11/2023) (cit. on p. 61).

Ionita, Dan, Roel Wieringa, Jan-Willem Bullee, and Alexandr Vasenev (2015). "Tangible modelling to elicit domain knowledge: an experiment and focus group". In: *International Conference on Conceptual Modeling*. Springer, pp. 558–565 (cit. on pp. 173–175).

ISO/IEC (2010). "ISO/IEC 25000 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE". In: DOI: citeulike-article-id:10949862 (cit. on p. 55).

ITU-T (2018). *Recommendation Z.151:User Requirements Notation (URN) - Language definition.* https://www.itu.int/rec/T-REC-Z.151-201810-I/en. (Accessed on 09/27/2023) (cit. on pp. xv, 42, 52, 233).

Jarosiński, Mirosław, Jurgita Sekliuckiene, Miklós Kozma, et al. (2023). "Born Digitals: Understanding the Sustainable Competitive Advantage Across Different Markets". In: *Artificiality and Sustainability in Entrepreneurship*, p. 41 (cit. on p. 13).

Jesus Souza, Magno Lua de, Alcemir Rodrigues Santos, Ivan Do Carmo Machado, Eduardo Santana de Almeida, and Gecynalda Soares da Silva Gomes

(2016). "Evaluating variability modeling techniques for dynamic software product lines: A controlled experiment". In: *2016 X Brazilian Symposium on Software Components, Architectures and Reuse (SBCARS)*. IEEE, pp. 1–10 (cit. on pp. 171, 173, 175).

Johannesson, Paul (2007). "The role of business models in enterprise modelling". In: *Conceptual modelling in information systems engineering*. Springer, pp. 123–140 (cit. on p. 58).

Johannesson, Paul and Erik Perjons (2014). *An Introduction to Design Science*. en. Cham: Springer International Publishing. ISBN: 978-3-319-10631-1 978-3-319-10632-8. DOI: 10.1007/978-3-319-10632-8. URL: https://link.springer.com/10.1007/978-3-319-10632-8 (visited on 02/15/2023) (cit. on p. 24).

Kabeli, Judith and Peretz Shoval (2005). "Comprehension and quality of analysis specifications—a comparison of FOOM and OPM methodologies". In: *Information and Software Technology* 47.4, pp. 271–290 (cit. on pp. 172, 173, 175).

Al-Kalbani, Jokha, Yassine Jamoussi, Naoufel Kraiem, and Zuhoor Abdullah Salim Al-Khanjari (2019). "Toward a contextual quality of service evaluation approach". In: *International Journal of Computer Applications in Technology* 59.1, pp. 82–94 (cit. on pp. 7, 48).

Kaplan, Robert S (2009). "Conceptual foundations of the balanced scorecard". In: *Handbooks of management accounting research* 3, pp. 1253–1269 (cit. on p. 141).

Karagiannis, Dimitris, Robert Andrei Buchmann, Patrik Burzynski, Ulrich Reimer, and Michael Walch (2016). "Fundamental conceptual modeling languages in OMiLAB". In: *Domain-specific conceptual modeling*. Springer, pp. 3–30 (cit. on pp. 133, 160).

Karim, Samina and Laurence Capron (2016). "Reconfiguration: Adding, redeploying, recombining, and divesting resources and business units". In: *Karim, Samina and Lawrence Capron (2016)."Reconfiguration: Adding, redeploying, recombining and divesting resources and business units." Strategic Management Journal* 37.13, E54–E62 (cit. on pp. 17, 109, 143).

Kaye, Doug (2003). *Loosely coupled: the missing pieces of Web services*. RDS Strategies LLC (cit. on p. 18).

Kettunen, Petri and Maarit Laanti (2017). "Future software organizations–agile goals and roles". In: *European Journal of Futures Research* 5.1, pp. 1–15 (cit. on pp. 10, 21).

Khalifa, Azaddin Salem (2012). "Mission, purpose, and ambition: redefining the mission statement". In: *Journal of Strategy and Management* 5.3, pp. 236–251 (cit. on p. 78).

Kirikova, Marite, Anita Finke, and Janis Grundspenkis (2010). "What is CIM: an information system perspective". In: *Advances in Databases and Information Systems: Associated Workshops and Doctoral Consortium of the 13th East European Conference, ADBIS 2009, Riga, Latvia, September 7-10, 2009. Revised Selected Papers 13*. Springer, pp. 169–176 (cit. on pp. 4, 7).

Kitsios, Fotis and Maria Kamariotou (2019). "Business strategy modelling based on enterprise architecture: a state of the art review". In: *Business Process Management Journal* 25.4, pp. 606–624 (cit. on p. 99).

Kniberg, H. and A. Ivarsson (2012). *Scaling Agile Spotify with tribes, squads, chapters & guilds*. `https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf`. (Retrieved on 08/16/2023) (cit. on pp. xv, 11, 12).

Kontio, Jyrki, Laura Lehtola, and Johanna Bragge (2004). "Using the focus group method in software engineering: obtaining practitioner and user experiences". In: *Proceedings of the 2004 International Symposium on Empirical Software Engineering* (cit. on pp. 158, 159).

Kotonya, Gerald and Ian Sommerville (1998). *Requirements engineering: processes and techniques*. Wiley Publishing (cit. on p. 49).

Kraiem, Naoufel, Houda Kaffela, Jamil Dimassi, and Zuhoor Al Khanjari (2014). "Mapping from MAP models to BPMN processes". In: *Journal of Software Engineering* 8.4, pp. 252–264 (cit. on pp. 48, 49, 56, 141, 142, 234, 235, 237).

Krogstie, John (2002). “A semiotic approach to quality in requirements specifications”. In: *Organizational Semiotics*. Springer, pp. 231–249 (cit. on p. 173).

Lapalme, James (2011). “Three schools of thought on enterprise architecture”. In: *IT professional* 14.6, pp. 37–43 (cit. on p. 38).

Larman, Craig and Bas Vodde (2016). *Large-scale scrum: More with LeSS*. Addison-Wesley Professional (cit. on p. 108).

Lewis, James R (1991). “Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ”. In: *ACM Sigchi Bulletin* 23.1, pp. 78–81 (cit. on p. 174).

Li, Zonghua and Zhengwei Ye (2021). “A Petri nets evolution method that supports BPMN model changes”. In: *Scientific Programming* 2021, pp. 1–16 (cit. on p. 48).

Li, Zonghua, Xiaofeng Zhou, Aihua Gu, and Qinfeng Li (2015). “A complete approach for CIM modelling and model formalising”. In: *Information and Software Technology* 65, pp. 39–55 (cit. on pp. 48, 50, 54, 56, 235, 237).

Lindland, Odd Ivar, Guttorm Sindre, and Arne Solvberg (1994). “Understanding quality in conceptual modeling”. In: *IEEE software* 11.2, pp. 42–49 (cit. on pp. 63, 124, 169, 176, 232, 238, 239).

Louaqad, Wahb and Mohammed El Mohajir (2014). “A holonic extension of the i* framework”. In: *Third IEEE International Colloquium in Information Science and Technology*. IEEE, pp. 7–11 (cit. on p. 58).

Lunenburg, Fred C (2012). “Organizational structure: Mintzberg’s framework”. In: *International journal of scholarly, academic, intellectual diversity* 14.1, pp. 1–8 (cit. on pp. 108, 109).

Marosin, Diana, Marc Van Zee, and Sepideh Ghanavati (2016). “Formalizing and modeling enterprise architecture (EA) principles with goal-oriented requirements language (GRL)”. In: *International Conference on Advanced Information Systems Engineering*. Springer, pp. 205–220 (cit. on p. 99).

Martin, Robert C (2000). "Design principles and design patterns". In: *Object Mentor* 1.34, p. 597 (cit. on p. 15).

Matulevičius, Raimundas and Patrick Heymans (2007a). "Comparing goal modelling languages: An experiment". In: *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer, pp. 18–32 (cit. on pp. 173–175).

— (2007b). "Visually effective goal models using KAOS". In: *Advances in Conceptual Modeling–Foundations and Applications: ER 2007 Workshops*. Springer, pp. 265–275 (cit. on pp. xvi, 43, 45).

Mellor, Stephen J, Anthony N Clark, and Takao Futagami (2003). "Guest editors' introduction: Model-driven development". In: *IEEE Software* 20.05, pp. 14–18 (cit. on p. 3).

Mintzberg, Henry (1987). "The strategy concept I: Five Ps for strategy". In: *California management review* 30.1, pp. 11–24 (cit. on pp. 13–15, 69, 107, 108).

Mishra, Mayank, Shruti Kunde, and Manoj Nambiar (2018). "Cracking the monolith: Challenges in data transitioning to cloud native architectures". In: *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, pp. 1–4 (cit. on p. 18).

Mohagheghi, Parastoo (2008). "Towards a tool-supported quality model for model-driven engineering". In: *Procedings of the 3rd Workshop on Quality in Modeling* (cit. on p. 170).

Moody, Daniel L (2003). "The method evaluation model: a theoretical model for validating information systems design methods". In: *European Conference on Information Systems 2003*, pp. 1–17 (cit. on pp. 176, 177, 179, 227, 239, 240, 259).

Muller, E. (2018). *Monitoring and Observability | the agile admin*. `https://theagileadmin.com/2018/02/16/monitoring-and-observability/`. (Accessed on 08/23/2023) (cit. on p. 21).

Mustafa, Nasser and Yvan Labiche (2017). “The need for traceability in heterogeneous systems: a systematic literature review”. In: *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 1. IEEE, pp. 305–310 (cit. on pp. 57, 59).

Nagel, Benjamin, Christian Gerth, Gregor Engels, and Jennifer Post (2013). “Ensuring consistency among business goals and business process models”. In: *2013 17th IEEE International Enterprise Distributed Object Computing Conference*. IEEE, pp. 17–26 (cit. on pp. 142, 234, 235, 237).

Nagel, Benjamin, Christian Gerth, Jennifer Post, and Gregor Engels (2013). “Kaos4SOA-Extending KAOS Models with Temporal and Logical Dependencies.” In: *CAiSE Forum*. Aachen: CEUR-WS, pp. 9–16 (cit. on p. 234).

Noel, Rene, Jose Ignacio Panach, and Oscar Pastor (2021a). *Experimental Comparison of LiteStrat vs iStar - Experimental Package*. `https://zenodo.org/record/7622995`. DOI: `10.5281/zenodo.7622995` (cit. on pp. 178, 188).

— (2021b). “The LiteStrat Method: Towards Strategic Model-Driven Development”. In: *Procedings of the International Conference on Information Systems Development*. AIS Electronic Library (cit. on pp. xvii, 31, 32, 35, 152, 223).

— (2022). “Challenges for Model-Driven Development of Strategically Aligned Information Systems”. In: *IEEE Access* 10, pp. 38237–38253 (cit. on p. 35).

— (2023). “Including business strategy in model-driven methods: an experiment”. In: *Requirements Engineering*, pp. 1–30 (cit. on p. 35).

Noel, Rene, Jose Ignacio Panach, Marcela Ruiz, and Oscar Pastor (2022). “Stra2Bis: A Model-Driven Method for Aligning Business Strategy and Business Processes”. In: *Conceptual Modeling: 41st International Conference, ER 2022, Hyderabad, India, October 17–20, 2022, Proceedings*. Springer, pp. 255–270 (cit. on pp. 31, 33, 35).

Noel, Rene, Marcela Ruiz, Jose Ignacio Panach, and Oscar Pastor (2021). “Beyond conventional model-driven development: from strategy to code”. In: *40th International Conference on Conceptual Modeling (ER2021), online,* *18-21 October 2021*. CEUR Workshop Proceedings, pp. 49–55 (cit. on pp. 35, 190).

Noel, Rene, Marcela Ruiz, Jose Ignacio Panach, and Oscar Pastor (Oct. 2022). *Empirical Validation of a Model-Driven Modelling Method for Strategic Alignment - Experimental Package*. Version 1.0. DOI: 10.5281/zenodo.7223389. URL: https://doi.org/10.5281/zenodo.7223389 (cit. on pp. 242, 247).

— (2023). "From Business Strategy to Business Processes: Empirical Validation". In: *Under Review in Information Systems* (cit. on p. 35).

Nwokeji, Joshua C, Tony Clark, and Balbir S Barn (2013). "Towards a comprehensive meta-model for kaos". In: *2013 3rd International Workshop on Model-Driven Requirements Engineering (MoDRE)*. IEEE, pp. 30–39 (cit. on p. 43).

OASIS (2007). *OASIS Web Services Business Process Execution Language (WSBPEL)*. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel. (Accessed on 09/23/2022) (cit. on p. 235).

OMiLAB (2023a). *The ADOxx Metamodelling Platform - ADOxx.org*. https://www.adoxx.org/live/home. (Accessed on 20/04/2023) (cit. on p. 133).

— (2023b). *The ADOxx Metamodelling Platform - Developer Space - ADOxx.org*. https://www.adoxx.org/live/development-spaces. (Accessed on 20/04/2023) (cit. on p. 134).

Oyetoyan, Tosin Daniel, Jean-Rémy Falleri, Jens Dietrich, and Kamil Jezek (2015). "Circular dependencies and change-proneness: An empirical study". In: *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, pp. 241–250 (cit. on p. 15).

Panach, Jose Ignacio, Oscar Dieste, Beatriz Marín, Sergio España, Sira Vegas, Oscar Pastor, and Natalia Juristo (2018). "Evaluating model-driven development claims with respect to quality: A family of experiments". In: *IEEE Transactions on Software Engineering* 47.1, pp. 130–145 (cit. on p. 196).

Panach, Jose Ignacio, Oscar Pastor, and Natalia Juristo (2021). "A Family of Experiments to Compare Two Model-Driven Development Tools vs a Traditional Development Method". In: *IEEE Transactions on Software Engineering* 48.12, pp. 4802–4817 (cit. on p. 211).

Parnas, David Lorge (1979). "Designing software for ease of extension and contraction". In: *IEEE transactions on software engineering* 2, pp. 128–138 (cit. on p. 15).

Pastor, Oscar and Juan Carlos Molina (2007). *Model-driven architecture in practice: a software production environment based on conceptual modeling.* Berlin, New York: Springer. ISBN: 978-3-540-71867-3 (cit. on pp. 23, 27, 58, 61).

Pastor, Oscar, Rene Noel, Ignacio Panach, and Marcela Ruiz (2021). "The LiteStrat Modelling Method: Towards the Alignment of Strategy and Code". In: *Domain-Specific Conceptual Modeling: Concepts, Methods and ADOxx Tools.* Springer, pp. 141–159 (cit. on p. 35).

Pastor, Oscar, Isidro Ramos Salavert, and José H Canós Cerdá (1995). "oasis v2: A class definition language". In: *International Conference on Database and Expert Systems Applications.* Springer, pp. 79–90 (cit. on p. 61).

Peleg, Mor and Dov Dori (2000). "The model multiplicity problem: experimenting with real-time specification methods". In: *IEEE Transactions on Software Engineering* 26.8, pp. 742–759 (cit. on pp. 172, 175).

Petersen, Kai, Sairam Vakkalanka, and Ludwik Kuzniarz (2015). "Guidelines for conducting systematic mapping studies in software engineering: An update". In: *Information and Software Technology* 64, pp. 1–18 (cit. on p. 170).

Raj, Vinay and Hanumanthu Bhukya (2023). "Assessing the Impact of Migration from SOA to Microservices Architecture". In: *SN Computer Science* 4.5, p. 577 (cit. on p. 18).

Raj, Vinay and Ravichandra Sadam (2021). "Evaluation of SOA-based web services and microservices architecture using complexity metrics". In: *SN Computer Science* 2, pp. 1–10 (cit. on p. 18).

Ralyté, Jolita (2013). “Situational Method Engineering in Practice: A Case Study in a Small Enterprise.” In: *Caise forum*, pp. 17–24 (cit. on pp. 26, 27).

Ramos-Merino, Mateo, Luis M. alvarez-Sabucedo, Juan M. Santos-Gago, and Francisco de Arriba-Perez (June 2019). “A Pattern Based Method for Simplifying a BPMN Process Model”. en. In: *Applied Sciences* 9.11, p. 2322. ISSN: 2076-3417. DOI: 10.3390/app9112322. URL: https://www.mdpi.com/2076-3417/9/11/2322 (visited on 10/18/2020) (cit. on p. 94).

Richardson, Chris (2018). *Microservice Architecture pattern.* https://microservices.io/patterns/microservices.html. (Accessed on 08/23/2023) (cit. on p. 21).

— (2019). *Database per service pattern.* https://microservices.io/patterns/data/database-per-service.html. (Accessed on 08/23/2023) (cit. on p. 19).

Rolland, Colette (2007). “Capturing system intentionality with maps”. In: *Conceptual modelling in Information Systems engineering.* Berlin: Springer, pp. 141–158 (cit. on pp. 26, 99, 120, 141, 234).

Rosemann, Michael, Peter Green, and Marta Indulska (2004). “A reference methodology for conducting ontological analyses”. In: *International Conference on Conceptual Modeling.* Springer, pp. 110–121 (cit. on pp. 90, 221, 222).

Rosing, Mark von, Stephen White, Fred Cummins, and Henk de Man (2015). *Business Process Model and Notation-BPMN* (cit. on pp. 6, 93, 99, 141).

Ruiz, Marcela, Dolors Costal, Sergio España, Xavier Franch, and Oscar Pastor (2014). “Integrating the goal and business process perspectives in information system analysis”. In: *International Conference on Advanced Information Systems Engineering.* Springer, pp. 332–346 (cit. on pp. 54, 236).

— (Oct. 2015). “GoBIS: An integrated framework to analyse the goal and business process perspectives in information systems”. en. In: *Information Systems* 53, pp. 330–345. ISSN: 03064379. DOI: 10.1016/j.is.2015.03.007 (cit. on pp. 4, 22, 27, 48, 49, 57, 59, 83, 91, 142, 223, 234, 235, 237, 239, 258).

Ruiz, Marcela, Sergio Espana, Oscar Pastor, and Arturo Gonzalez (May 2013). "Supporting organisational evolution by means of model-driven reengineering frameworks". en. In: *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*. Paris, France: IEEE, pp. 1–10. ISBN: 978-1-4673-2914-9 978-1-4673-2912-5. DOI: 10.1109/RCIS.2013.6577682. URL: http://ieeexplore.ieee.org/document/6577682/ (visited on 02/26/2020) (cit. on p. 94).

Salameh, Abdallah and Julian M Bass (2022). "An architecture governance approach for Agile development by tailoring the Spotify model". In: *AI & SOCIETY* 37.2, pp. 761–780 (cit. on pp. 10, 11).

Santos Jr, Paulo Sérgio, João Paulo A Almeida, and Giancarlo Guizzardi (2010). "An ontology-based semantic foundation for ARIS EPCs". In: *Proceedings of the 2010 ACM Symposium on Applied Computing*, pp. 124–130 (cit. on pp. 38, 112, 141).

— (2013). "An ontology-based analysis and semantics for organizational structure modeling in the ARIS method". In: *Information Systems* 38.5, pp. 690–708 (cit. on p. 123).

Saputri, Theresia Ratih Dewi and Seok-Won Lee (2020). "Addressing sustainability in the requirements engineering process: From elicitation to functional decomposition". In: *Journal of Software: Evolution and Process* 32.8, e2254 (cit. on pp. 172–175).

Scaled Agile, INC (2021). *SAFe 5 for Lean Enterprises*. https://www.scaledagileframe com/. (Accessed on 04/10/2021) (cit. on pp. 13, 14, 20, 106–108).

Scanniello, Giuseppe and Ugo Erra (2014). "Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments". In: *Journal of Visual Languages & Computing* 25.4, pp. 494–517 (cit. on pp. 171–173, 175).

Sebastián, Gabriel, Jose A Gallud, and Ricardo Tesoriero (2020). "Code generation using model driven architecture: A systematic mapping study". In: *Journal of Computer Languages* 56, p. 100935 (cit. on p. 58).

Shopify (2022). *What Is Organizational Structure? Definition and Guide (2023).* `https://www.shopify.com/blog/what-is-organizational-structure`. (Accessed on 08/16/2023) (cit. on p. 12).

Shoval, Peretz and Sagit Shiran (1997). "Entity-relationship and object-oriented data modeling — an experimental comparison of design quality". In: *Data & Knowledge Engineering* 21.3, pp. 297–315 (cit. on pp. 172, 173, 175).

Skelton, Matthew and Manuel Pais (2019). *Team topologies: organizing business and technology teams for fast flow.* It Revolution (cit. on pp. 10, 17, 107).

Skobtsov, Andrey V and Anna A Kalenkova (2019). "Using heuristic algorithms for fast alignment between business processes and goals". In: *2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW).* IEEE, pp. 85–91 (cit. on pp. 48, 56).

Sousa, Henrique Prado and Julio Cesar Sampaio do Prado Leite (2014). "Modeling organizational alignment". In: *International Conference on Conceptual Modeling.* Springer, pp. 407–414 (cit. on pp. 7, 48, 53, 54, 56, 141, 142, 223, 233, 235, 237).

Štefanko, Martin, Ondřej Chaloupka, Bruno Rossi, M van Sinderen, and L Maciaszek (2019). "The saga pattern in a reactive microservices environment". In: *Proc. 14th Int. Conf. Softw. Technologies (ICSOFT 2019).* SciTePress Prague, Czech Republic, pp. 483–490 (cit. on p. 21).

Sun, Zhouxuan, Jian Wang, Keqing He, Shujuan Xiang, and Dunhui Yu (2010). "A model transformation method in service-oriented domain modeling". In: *2010 21st Australian Software Engineering Conference.* IEEE, pp. 107–116 (cit. on pp. 48, 50, 54).

Thabet, Rafika, Dominik Bork, Amine Boufaied, Elyes Lamine, Ouajdi Korbaa, and Hervé Pingaud (2021). "Risk-aware business process management using multi-view modeling: method and tool". In: *Requirements Engineering*, pp. 1–27 (cit. on pp. 172, 173, 175).

The Object Management Group (2008). *Business Process Definition Metamodel Specification Version 1.0.* `https://www.omg.org/spec/BPDM/1.0`. (Accessed on 09/23/2022) (cit. on p. 235).

— (2014). *Model Driven Architecture (MDA)*. https://www.omg.org/mda/. (Accessed on 04/14/2021) (cit. on p. 3).

— (2015). *Business Motivation Model Specification Version 1.3*. URL: https://www.omg.org/spec/BMM/About-BMM/ (visited on 07/02/2020) (cit. on pp. xvi, 5, 31, 57, 58, 91, 96, 99, 112, 115–118, 122, 123, 142).

— (2017). *The Unified Modeling Language Specification Version 2.5.1*. https://www.omg.org/spec/UML/2.5.1/. (Accessed on 09/21/2022) (cit. on p. 4).

The Open Group (2013). *ArchiMate® 2.1 Specification*. https://pubs.opengroup.org/architecture/archimate2-doc. (Accessed on 11/09/2020) (cit. on pp. 58, 91, 99, 122).

— (2018). *The TOGAF® Standard, Version 9.2*. https://pubs.opengroup.org/architecture/togaf92-doc/arch/. (Accessed on 06/09/2023) (cit. on pp. 38, 96, 112, 141).

— (2022a). *ArchiMate® 3.2 Specification*. https://pubs.opengroup.org/architecture/archimate3-doc/index.html. (Accessed on 04/13/2023) (cit. on pp. 5, 37, 38, 57, 112, 113).

— (2022b). *ArchiMate® 3.2 Specification - Business Active Structure Elements*. https://pubs.opengroup.org/architecture/archimate3-doc/ch-Business-Layer.html#sec-business-Active-Structure-Elements. (Accessed on 04/13/2023) (cit. on pp. xvi, 113, 116, 120).

— (2022c). *ArchiMate® 3.2 Specification - Motivation Elements*. https://pubs.opengroup.org/architecture/archimate3-doc/ch-Motivation-Elements.html. (Accessed on 04/13/2023) (cit. on pp. xvi, 114).

— (2022d). *ArchiMate® 3.2 Specification - Strategy Layer*. https://pubs.opengroup.org/architecture/archimate3-doc/ch-Strategy-Layer.html. (Accessed on 04/13/2023) (cit. on pp. xvi, 113, 115).

— (2022e). *The ArchiMate® Enterprise Architecture Modeling Language*. https://www.opengroup.org/archimate-home. (Accessed on 04/20/2022) (cit. on pp. 31, 141).

Thönes, Johannes (2015). “Microservices”. In: *IEEE software* 32.1, pp. 116–116 (cit. on p. 18).

Thoughtworks (2016). *Inverse Conway Maneuver | Technology Radar | Thoughtworks.* `https://www.thoughtworks.com/es-es/radar/techniques/inverse-conway-maneuver`. (Accessed on 11/09/2021) (cit. on pp. 10, 12, 15, 80).

Trkman, Marina, Jan Mendling, Peter Trkman, and Marjan Krisper (2019). “Impact of the conceptual model’s representation format on identifying and understanding user stories”. In: *Information and software technology* 116, p. 106169 (cit. on pp. 172, 175).

UFES (2015a). *Ontology and Conceptual Modeling Research Group (NEMO): UFO-C Specification.* `https://ontology.com.br/ufo/ufo-c/spec/`. (Accessed on 09/11/2023) (cit. on pp. 71–74).

— (2015b). *SEON: The Software Engineering Ontology Network.* `https://dev.nemo.inf.ufes.br/seon/`. (Accessed on 09/11/2023) (cit. on pp. 71, 79, 81).

Uschold, Mike, Martin King, Stuart Moralee, and Yannis Zorgios (1998). “The Enterprise Ontology”. In: *The Knowledge Engineering Review* 13.1, pp. 31–89. DOI: `10.1017/S0269888998001088` (cit. on p. 64).

Vara, Jose Luis de la, Juan Sánchez, and Oscar Pastor (2008). “Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems”. In: *Advanced Information Systems Engineering.* Ed. by Zohra Bellahsène and Michel Léonard. Berlin, Heidelberg: Springer, pp. 213–227. ISBN: 978-3-540-69534-9 (cit. on pp. xvi, 7, 45, 46, 48, 53, 142, 234, 235, 237).

— (2013). “On the use of goal models and business process models for elicitation of system requirements”. In: *International Workshop on Business Process Modeling, Development and Support.* Springer, pp. 168–183 (cit. on pp. 48, 51, 54, 99).

Venkatraman, Nramanujam (1994). “IT-enabled business transformation: from automation to business scope redefinition”. In: *Sloan management review* 35, pp. 73–73 (cit. on p. 13).

Vial, Gregory (2021). "Understanding digital transformation: A review and a research agenda". In: *Managing Digital Transformation*, pp. 13–66 (cit. on pp. 13, 68).

Walter, Anna-Theresa (2021). "Organizational agility: ill-defined and somewhat confusing? A systematic literature review and conceptualization". In: *Management Review Quarterly* 71, pp. 343–391 (cit. on p. 14).

Wautelet, Yves (2019). "A model-driven IT governance process based on the strategic impact evaluation of services". In: *Journal of Systems and Software* 149, pp. 462–475 (cit. on p. 57).

Weinhardt, Christof, Arun Anandasivam, Benjamin Blau, Nikolay Borissov, Thomas Meinl, Wibke Michalk, and Jochen Stößer (2009). "Cloud computing–a classification, business models, and research directions". In: *Business & Information Systems Engineering* 1, pp. 391–399 (cit. on p. 13).

Wieringa, Roel J. (2014). *Design Science Methodology for Information Systems and Software Engineering.* Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-662-43838-1 978-3-662-43839-8. DOI: 10.1007/978-3-662-43839-8 (cit. on pp. 6, 21, 22, 24, 25, 28, 30, 84, 85, 105, 139, 167, 229, 231, 235).

Wohlin, Claes (2014). "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pp. 1–10 (cit. on p. 171).

Wohlin, Claes, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén (2012). *Experimentation in Software Engineering.* Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-29043-5 978-3-642-29044-2. DOI: 10.1007/978-3-642-29044-2. URL: http://link.springer.com/10.1007/978-3-642-29044-2 (visited on 03/27/2020) (cit. on pp. 168–170, 188, 198, 226, 231, 258).

Yu, Eric (1995). "Models for supporting the redesign of organizational work". In: *Proceedings of conference on Organizational computing systems*, pp. 226–236 (cit. on p. 122).

Yu, Eric (1997). “Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering”. In: *IEEE Int. Symp. on Requirements Engineering*. Ed. by IEEE, pp. 226–235 (cit. on p. 39).

— (2009). “Social Modeling and i”. In: *Conceptual modeling: Foundations and applications: Essays in honor of John Mylopoulos*. Springer Science & Business Media, pp. 99–121 (cit. on pp. 6, 38, 39).

— (2011a). “Modeling Strategic Relationships for Process Reengineering.” In: *Social Modeling for Requirements Engineering* 11.2011, pp. 66–87 (cit. on pp. 6, 27, 32, 141, 168, 233).

— ed. (2011b). *Social modeling for requirements engineering*. Cooperative information systems. Cambridge, Mass: MIT Press. ISBN: 978-0-262-24055-0 (cit. on pp. 4, 22, 31, 38, 59, 83, 122).

Yu, Eric, Markus Strohmaier, and Xiaoxue Deng (2006). “Exploring intentional modeling and analysis for enterprise architecture”. In: *2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW’06)*. IEEE, pp. 32–32 (cit. on pp. 57, 58, 142).

Zachman, John A (1987). “A framework for information systems architecture”. In: *IBM systems journal* 26.3, pp. 276–292 (cit. on pp. 37, 112, 141).