Organix — Collect Implicit Organizational Knowledge
Through a Gaming Platform

Valter Miguel Gaspar Nepomuceno

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Professor José Luís Brinquete Borbinha

Examination Committee

Chairperson: Professor José Carlos Alves Pereira Monteiro
Supervisor: Professor José Luís Brinquete Borbinha
Member of the Committee: Professor Daniel Jorge Viegas Gonçalves

June 2014
Acknowledgments

The fulfilment of this dissertation was empowered by the efforts and commitment of a group of people that supported me in this project. For that I would like to thank:

- Professor José Luís Borbinha, coordinator of this dissertation, for the guidance in the process of conceptual decision making and for the critical view over the accomplished work;

- Cláudia Ribeiro, involved in serious games investigation in INESC-ID, for providing research material regarding this subject, and orientating the development of prototypes for the application;

- Dra. Carla Sousa, for proposing this dissertation and providing initial input concerning the scope of this project;

- To my sister and my mother, for their unlimited and adamant support in the most uncertain moments, not only for the last months but also throughout my whole academic life; and for the unconditional patience and comprehension when dealing with my unavailability;

- To my friends and colleagues, who motivated and advised me during the toughest moments, and kept me away from the computer when I needed the most.

"Things outside you are projections of what's inside you, and what's inside you is a projection of what's outside. So when you step into the labyrinth outside you, at the same time you're stepping into the labyrinth inside."

— Haruki Murakami
Resumo

A Gestão de Processos de Negócio envolve avaliações aos processos de negócio atuais de uma empresa (as-is), para delinear estratégias que otimizam os processos e melhoram a sua eficiência (to-be). Uma fase deste ciclo consiste na elicitação de processos de negócio, que pode ser alcançada usando diferentes técnicas, dirigidas por business analysts. Contudo, muitas dessas abordagens suscitam complicações que podem invalidar os resultados da elicitação. A solução corrente para este cenário consiste na combinação de técnicas que reúnem a maior quantidade de conhecimento de negócio possível, apesar de nem sempre se alcançarem outputs precisos.

Esta dissertação demonstra a possibilidade de criar uma solução desenhada como uma plataforma de jogo sério que cativa os jogadores, que acontece serem os participantes dos processos a serem elicitados, a contribuírem com conhecimento de negócio em relação a processos de negócio em que estão envolvidos nas suas atividades profissionais. A plataforma tenciona recolher conhecimento implícito de negócio dos participantes, de um modo sistemático e automatizado, para que o resultado do jogo consista em modelos de processo de negócio obtidos da elicitação realizada no jogo. Posteriormente, estes modelos podem ser usados para tomada de decisão relativamente ao que deve ser alterado nos processos para expandir o valor de negócio da organização.

**Palavras-chave:** Processos de Negócio, Modelação de Processos, Conhecimento de Negócio, Cenários Organizacionais, Jogos Sérios, Plataforma de Jogo
Abstract

The practice of Business Process Management includes assessments to current business processes of a company (as-is), in order to delineate strategies to optimise processes and improve their efficiency (to-be). One phase of this cycle consists in the elicitation of business processes, which can be achieved by using different techniques, directed by business analysts. However, many of these approaches bring out complications that may invalidate the results of the elicitation. The current solution to this scenario consists in the combination of techniques to gather as much business knowledge as possible, despite not always achieving accurate outputs.

This dissertation demonstrates the possibility of creating a solution designed as a serious game platform that engages players, who happen to be business participants in the processes to be elicited, to contribute with business knowledge regarding the business processes they are involved in their professional activities. The platform intends to gather implicit business knowledge from participants, in a systematic and automated way, so the outcome of the game consists in business process models obtained from the elicitation accomplished in the game. Posteriorly, these models can be used for decision making on what should be modified in processes to expand the business value of the organisation.

**Keywords:** Business Processes, Process Modelling, Business Knowledge, Organisational Scenarios, Serious Games, Gaming Platform
Contents

Acknowledgments .......................................................... iii
Resumo ................................................................. iv
Abstract ................................................................. v
List of Tables ........................................................... ix
List of Figures ......................................................... xi

1 Introduction .............................................................. 1
  1.1 Objectives ......................................................... 2
  1.2 Dissertation Structure ........................................... 2

2 State of the Art ......................................................... 4
  2.1 Business Process in the Scope of Enterprise Information Systems ................................. 4
    2.1.1 Zachman Framework ......................................... 5
    2.1.2 Cornwell Enterprise Architecture Maturity Dashboard ........................................... 8
    2.1.3 Multi-Perspective Enterprise Modelling ................................................................. 9
    2.1.4 Enterprise Resource Planning II ................................................................. 10
  2.2 Business Process Modelling ........................................ 11
    2.2.1 Equivalence of Activities ..................................... 12
    2.2.2 Flowchart ..................................................... 13
    2.2.3 Dataflow ..................................................... 13
    2.2.4 UML .......................................................... 13
    2.2.5 BPMN ......................................................... 14
    2.2.6 EPC .......................................................... 15
    2.2.7 IDEF3 ........................................................ 16
    2.2.8 Other potentially relevant references ................................................................. 16
  2.3 Serious Games ...................................................... 19
  2.4 Game Technology .................................................. 21

3 Problem Analysis ....................................................... 24
  3.1 Business Actors ................................................... 24
  3.2 Tasks ............................................................... 25
  3.3 Process Elicitation Techniques ................................... 25
List of Tables

2.1 Comparison of researched game engines ........................................ 23
4.1 Entities of domain model ....................................................... 37
4.2 Types of entities of domain model ........................................... 37
4.3 Properties of process states ................................................... 38
5.1 Achievements ................................................................. 42
5.2 Medals ................................................................. 42
5.3 Strategic objectives .......................................................... 44
5.4 Alignment of objectives and game rules .................................... 45
5.5 Bonuses and penalties of players ............................................. 46
5.6 Scoring system ............................................................. 48
5.7 Abbreviations of scoring system ............................................. 49
7.1 Fulfillment of requirements .................................................. 64
7.2 Test case 1 process data ...................................................... 64
7.3 Test case 1 results .......................................................... 65
7.4 Test case 2 process data ...................................................... 65
7.5 Test case 2 results .......................................................... 65
7.6 Test case 3 process data ...................................................... 65
7.7 Test case 3 results .......................................................... 66
7.8 Alignment of participants and activities in process “Triage of Patients” 67
7.9 Alignment of participants and activities in process “Finalise Dissertation (IST)” 68
7.10 Frequency table for testers’s gender ....................................... 69
7.11 Frequency table for tester’s age ............................................. 69
7.12 Individual information of participants ..................................... 70
7.13 Game data relating achievements with players ......................... 72
7.14 Game data relating medals with players ................................... 72
7.15 Game data relating questionnaires’ answers with players ............ 73
7.16 Game data of final consensus set ......................................... 73
List of Figures

2.1 Zachman Framework schema .................................................. 5
2.2 Cornwell Enterprise Architecture Maturity Dashboard .................. 9
2.3 Business Motivation Model .................................................. 18
2.4 Microsoft Flight Simulator .................................................. 20
2.5 iThink ................................................................. 20

4.1 Proposed solution use case diagram ....................................... 31
4.2 Pool graphic representation ................................................ 32
4.3 Event graphic representation ............................................. 33
4.4 Activity graphic representation ......................................... 33
4.5 Flow graphic representation ............................................. 34
4.6 Domain model of proposed solution ..................................... 36
4.7 Process state diagram .................................................... 38

5.1 Modified Poisson Distribution ............................................. 46
5.2 Final Consensus .......................................................... 49
5.3 Process creation screen ................................................... 51
5.4 Home screen ............................................................. 51
5.5 Version creation screen ................................................... 52
5.6 Duplicate marking screen ................................................. 52
5.7 Final scores screen ....................................................... 53
5.8 Game context definition screen ....................................... 53
5.9 Keyboard shortcuts ........................................................ 54

6.1 Game architecture diagram .............................................. 57
6.2 Data persistency implementation ....................................... 59
6.3 Client-server interactions upon client request. ......................... 60
6.4 Client-server interactions without previous client request. .......... 61

7.1 Representation of process “Triage of Patients” .......................... 67
7.2 Representation of process “Finalise Dissertation” ....................... 68
7.3 Box plot for number of activities and votes ............................ 70
Chapter 1

Introduction

Nowadays, due to the constant growth and innovation of industries, companies face many challenges in the global marketplace. To outrun competitors, it is mandatory that organisations keep improving their efficiency in bringing to market the most competitive products and services possible. For that reason, projects of Business Process Management are put into action, so that performance of business processes of a company can be managed and improved.

One phase of the BPM lifecycle consists in modelling the business processes. Typically, business analysts are responsible for the identification of existing processes, collection of business knowledge within the scope of the company’s processes, and transformation of this knowledge into structured business process models. Posteriorly, these process representations are used for improvement of effectiveness and efficiency of the business processes of the organisation.

Although the range of existing approaches for process elicitation is wide, not always some of these techniques work properly, because many may arise complications that invalidate the result of the elicitation process. Additionally, projects of this kind are usually time-consuming, and therefore costly, given that teams of business analysts are allocated to examine process documentation, approach business participants with individual or group techniques, filter and consolidate the contributed business knowledge, and finally build business process models.

In order to streamline projects of business process elicitation, it would be important to explore the possibility of creating a solution that would tackle the main obstacles that emerge from current elicitation techniques. Some of these problems are related to communication flaws between process participants and business analysts, or even contributed ambiguous information. Consequently, the proposed solution should enforce that no intermediaries are used to gather implicit knowledge about business processes. Thus, financial costs are significantly reduced for both professional services firms and companies whose business processes need redesign, because business analysts can be allocated to perform activities that follow the business process modelling phase.

The use of serious games in tasks such as learning, training, simulating and informing, among others, has been growing over time. A serious game takes advantage of the entertainment of games, and engages players in performing assignments that are very often related to collaboratively solve problems.
Disregarding the use of less dynamic tools in favour of serious games allow players to engage in a specific set of goals, under certain constraints and rules defined by the game designer, whose objective is to encourage players into solving the problem introduced by the game.

The research work of this dissertation aims at investigating the possibility of creating a solution, as described previously, designed as a gaming platform that would take advantage of capturing player’s attention, encouraging them to collaboratively elicit business processes of the organisation they act as business participants.

1.1 Objectives

The great challenge of this dissertation subsists in designing a solution that would in a systematic and automated way engage all business participants in contributing business knowledge, each providing a local view of the business processes he or she is involved. This analysis involves a review of the state of the art in business process modelling, serious games and game technology. Afterwards, it is presented a rationalisation regarding which approaches are more useful in diminishing the problems that differentiate present elicitation techniques.

It is fundamental that the solution includes a process notation, which allows participants to be able to communicate their implicit knowledge of business processes in a structured way, so the contributions can be assembled into explicit process representations. It is intended that the notation is simple and easy to use, to accommodate the professional diversity of business participants.

Furthermore, business participants will be able to cooperate with each other, by suggesting modifications to the built business process models. Subsequently, participants are able to decide consensually on which process representations should be elected to best represent the business processes performed by the organisation. The election should be based on the degree of accurateness, precision and completeness that the representation possesses, compared to how the process is performed in real life.

Another objective of this dissertation is to provide mechanisms that allow business analysts and consultants to export the process knowledge contributed by real world business participants, such that process models can be used for posterior decision making on what modifications can be done to processes to improve their effectiveness and efficiency.

1.2 Dissertation Structure

This dissertation is structured in 8 chapters, and the work composed by every chapter starts with high-level conceptualisations, progressively evolving to a concretisation of a solution. The content of each chapter is outlined in the following paragraphs:

- **Chapter 1 - Introduction**: Presents a context and motivation for this research work, delineates the objectives to be accomplished, and summarises the structure of this dissertation.
• **Chapter 2 - State of the Art**: A review of relevant state of the art methodologies and techniques is made, regarding the areas of business process modelling, serious games and game technology.

• **Chapter 3 - Problem Analysis**: An assessment of the mechanisms used to elicit business processes is performed, along with an evaluation of the substantial complications that may come from these approaches.

• **Chapter 4 - Proposed Solution**: Encloses a blueprint of a solution designed to tackle problems that may arise from business process elicitation, along with a specification of the desired behaviour for such system, and a definition of boundary for the domain in which it operates.

• **Chapter 5 - Game Design**: Consolidates an explicit description of the game parts that belong to the scope of this project, giving a clear picture of the gameplay, mechanics, interfaces, etc.

• **Chapter 6 - Game Implementation**: Exposes the most influential decisions made during the implementation of the proposed solution.

• **Chapter 7 - Evaluation**: An evaluation of the application is made, concerning the consummation of the functional and non-functional requirements prioritised primarily.

• **Chapter 8 - Conclusions**: The conclusions reached along the development of this work are presented, achieved objectives within the scope of this dissertation are acknowledged and proposals of future work are brought out.
Chapter 2

State of the Art

In this chapter it is addressed an examination and discussion of existing research work regarding the areas of business process modelling, serious games and game technology. The analysis focus in approaches and techniques relevant in the recognition of the problem and design of a solution.

Primarily, a study on conceptual frameworks is made, assessing organised structures commonly used in business process modelling. Afterwards an overview on process modelling is traced, along with techniques and notations used for process representation.

The second part of this chapter focus in research work and related projects in the area of serious games. An examination of the importance of serious games in the game industry is traced, followed by a research on existing projects that are relevant to the scope of this dissertation.

The last part presents an assessment of 7 game engines well-known to the game development market. Then, a comparison between the described game engines is made, evidencing the advantages and drawbacks of each one. Subsequently, it is identified the most needed features in the scope of this dissertation, so a game engine can be chosen for the development of the solution.

2.1 Business Process in the Scope of Enterprise Information Systems

Business process architectures are expected to be aligned with the overall enterprise architecture of an organisation, for which conceptual frameworks have been developed over time. A conceptual framework can be described as an organised and structured set of concepts regarding a certain domain. In the application area of business process modelling, the more relevant conceptual frameworks for the scope of this project are the Zachman Framework, MEMO and ERP II. Additionally, the Cornwell Enterprise Architecture Maturity Dashboard can be also very useful in the evaluation of completeness and depth of a business process representation.
### 2.1.1 Zachman Framework

The Zachman Framework is a logical structure for classifying and organising the relevant descriptive artefacts or models of an enterprise that will aid in the process of building a complete, detailed and accurate representation of an organisation [1]. Posteriorly, this representation is of the utmost importance for an organisation, because not only provides a broad and complete view of the multiple dimensions of an organisation, it also depicts that information in different perspectives, that are to be intelligible to different audiences.

![Zachman Framework schema](image)

**Figure 2.1: Zachman Framework schema [2].**

The framework was derived through observation of descriptive representations of various physical objects like airplanes, buildings, ships, computers, etc. in which it was observed that the design artefacts (the descriptive representations, the product descriptions, the engineering documentation) of complex products can be classified by the audience for which the artefact was constructed as well as classified by the content or subject focus of the artefact [3]. Given the fact that the framework is a normalised schema, it is also a good conceptual tool that is commonly used nowadays to perform a thorough analysis in enterprises, or parts of it. Using Zachman Framework as an instrument in the definition of an Enterprise Architecture has empirically proven over the years to be very successful, when adequately used.

To fully understand how the framework works, it is essential that one fully understands the internal structure of the schema and the particularities and uniqueness of each cell. When observing the schema
depicted in Figure 1, while moving vertically through a column of the framework, the focus is set on a single part of the organisation through the different viewpoints of each different stakeholder. When moving horizontally through a row, the focus is set on a single viewpoint of one stakeholder through the different parts that constitute the organisation.

**Perspectives of the Framework**

The first and vertical dimension of the framework is named Perspective, representing a manifold of views of the whole organisation regarding different stakeholders. Each Perspective will contain the descriptive representations under certain constraints that will guarantee that all information present in that Perspective is intelligible to the corresponding stakeholder. So, in a single row, it can be communicated to a particular stakeholder, all the different representations that constitute a comprehensive depiction of the organisation as a whole. If a stakeholder finds hard to understand the descriptive representations of the corresponding perspective, then the representations are probably not coherent, complete or detailed enough.

The Perspectives are designated as follows:

- **The Owner’s Perspective (row 1)** — The client of the end product. The descriptive representations that are present in this Perspective refer the usage traits of the product and reflect the utility of the product and how the owner will use it.

- **The Designer’s Perspective (row 2)** — The intermediary between what is desirable and what is physically and technically possible. These descriptive representations reflect the system and logical constraints for the design of the product.

- **The Builder’s Perspective (row 3)** — The employer of some technical capacity for producing the end product. In this Perspective, the descriptive representations reflect physical or technological constraints of using the organisation’s technology to achieve the end product.

- **A Scope Perspective (row 4)** — The context that situates the organisation. Consists of a list of relevant artefacts that must be taken into account by other descriptive representations of the remaining perspectives.

- **An Out-of-Context Perspective (row 5)** — A detailed description that disassociates the parts of the complex object for manufacturing purposes. The artefacts present in this Perspective will support the transformation from the media of the design of the product to the media of the final product.

The last row (row 6) of the Zachman Framework represents the physical manifestation of the end product itself. Despite this row is not technically Architecture, because it is not a representation, but a concretisation, it is still useful to be completely filled in the framework because it completes the Enterprise Architecture.
Abstractions of the Framework

The second and horizontal dimension of the framework is named Abstraction, representing a manifold of focal points or descriptions of part of the system for all the different stakeholders. Each Abstraction will contain the descriptive representations of the organisation under a certain focus in a way that it is comprehensible to each stakeholder, guaranteeing that their needs, goals and expectations are aligned and can be easily traced along the other different stakeholder views. So, in a single column, it can be communicated to all the different stakeholders, all the representations of the organisation restricted to a certain subdomain (the What, How, Where, Who, When or Why).

The Abstractions are designated as follows:

- **What (column 1)** — The Data Description corresponds to the depiction of the composition of the end product, using Data (or Thing) Models.

- **How (column 2)** — The Functional Description corresponds to the functional specification of the Enterprise, using Process (or Function) Models.

- **Where (column 3)** — The Spatial Description communicates the location of the components relative to each other, using Network (or Logistics) Models.

- **Who (column 4)** — The Operational Description specifies the manuals and operating instructions of the enterprise and who is assigned to each activity, using Workflow (or People) Models.

- **When (column 5)** — The Timing Description presents when certain activities take place relative to one another, using Time (or Dynamics) Models.

- **Why (column 6)** — The Motivation Description exposes the reason why the organisation’s activities happen, using Motivation Models.

Each of these Abstractions will answer one of the six fundamental questions regarding the process of organisational scenarios definition, essential to model business processes within an enterprise.

Framework Properties

There is a reason for why the Zachman Framework has been used for years, which is related to the fact that due to the way it is built, disposes of properties such as primitiveness, comprehensiveness and normalisation, that ensure that the representation of the organisation is complete, intelligible and portrays the way the enterprise deals with its business with sufficient level of detail.

Each of the six rows and columns of the schema are primitive in the sense that each Perspective and each Abstraction, respectively, differ and vary independently from all the others of the same dimension, and all of them must be present so the business representation is as accurate and complete as possible. Consequently, each of the cells of the schema also have the property of primitiveness, meaning that the complete set of cells present in the Zachman Framework is the minimum set to fully represent an organisation, without any redundancy. Saying that the complete set of cells in the schema is comprehensive
means that no additional Perspectives or Abstractions need to be added, since it is the minimum set. The framework is normalised because each Perspective and each Abstraction is unique, consequently, each cell in the schema is also unique.

In order not to violate the properties of the schema, some rules have to be followed so these properties remain intact. The most important are [2]:

- An artefact can only be placed in one cell. The fact that the framework is normalised means that every cell is unique, so, placing an artefact in more than one cell would mean that there would be redundancy in the representation.

- All cells must be complete. Any cell left incomplete will mean that there are assumptions regarding the business, and that can lead to risk and possible restructuring.

- The structure of the framework should remain as it is. This means no addition or deletion of rows or columns. The reason why the schema should remain intact was already aforementioned in the beginning of this section. The name of the rows and columns should not be changed either, because doing this may lead to changes in the basic logic structure of the framework, possibly violating its beneficial properties.

Bottom line idea is that the Zachman Framework has proven to be empirically successful, being used in various fields of Engineering over the years. This shows that the framework can be used for any kind of organisation and business process with no need to readjust the structure of the schema to fit a particular organisation, because it already does.

### 2.1.2 Cornwell Enterprise Architecture Maturity Dashboard

In Section 2.1.1 *Zachman Framework*, it was introduced a conceptual tool, whose main goal is to describe and represent an organisation in its wholeness. In this section, it is described a method to measure the completeness and maturity of the Enterprise Architecture (EA) so an overall quality measure of the representation can be estimated.

One of the main advantages of the Cornwell Enterprise Architecture Maturity Dashboard is that it provides an overall impression of the strengths and weaknesses of an organisation’s representation. This model communicates to top management the strengths and weaknesses of the representation, and where the focus and EA efforts should be put in.

The auditing of architectural products leads to two measures for each cell of the Zachman Framework: completeness and maturity. The completeness of a view (focus in one conceptual part of the organisation through the prism of one stakeholder) is directly related to the sufficient breadth and depth of the artefacts that cover the scope of that cell. Maturity is a measure applicable to the representation models present in the measured cell. The author refers to three different kinds of discrete levels of Maturity, that vary from the lowest level, corresponding to rudimental diagrams, to the highest level, corresponding to models that can be integrated with other models, creating a synergetic descriptive representation.
The Maturity Dashboard can be visually depicted using vertical bars for each work stream in each view. The completeness and maturity are represented by percentage and colour of the bars, respectively, for each workflow in each view. The percentage represents the breadth and depth of the architectural artefacts, as in the colour represents the maturity of the models, ranging from red, representing mostly “dumb” artefacts, to yellow, representing mostly “smart” models, and finally green, which represents fully integrated architectural products.

An example is represented in Figure 2. The architectural artefacts describing the domain that covers the Technology Model and Function are 60% complete, which means that the view has a medium level of completeness of its models, and the corresponding bar has a green colour, which means that the present models are fully integrated with other models in different views.

Figure 2.2: Cornwell Enterprise Architecture Maturity Dashboard [4].

2.1.3 Multi-Perspective Enterprise Modelling

The main purpose for enterprise modelling is to provide an abstract representation that describes the most important aspects of an organisation, such as structure, processes, information and resources. This representation supplies different views on an organisation to facilitate the communication of the enterprise architecture among stakeholders with different professional backgrounds.

"MEMO (Multi-Perspective Enterprise MOdelling) is a method for enterprise modelling that offers a set of specialized visual modelling languages together with a process model as well as techniques and heuristics to support problem specific analysis and design." [5]

The MEMO conceptual framework fits generic abstractions of the way organisations deal with its business. The framework consists of three Perspectives, such as Strategy, Organisation and Information System, and four Aspects, such as Structure, Process, Resources and Goals, assembled in a two dimensional schema. A part of the organisation can be communicated and discussed by choosing one Perspective upon a particular Aspect.

When focusing in multiple Aspects of the framework, a correspondence to particular models is established (e.g. An Information System model describes the structure and process of the information
system, and its associated resources and business goals).

To enhance the completeness, integration and traceability of the framework, three specific modelling languages are defined in order to better represent the three models: the Strategy Model, the Organisation Model and lastly the Information System Model. The language used for the modelling of Strategy is named MEMO-SML and incorporates common concepts used for strategic planning. The Organisation Model can be described using the modelling language MEMO-OrgML, useful to represent a company’s organisation. Finally, the Information System Model can be defined using the object oriented modelling language MEMO-OML, allowing the specification of the technological infrastructure of the company regarding information systems.

The groups of concepts used by MEMO-SML to describe the goals and strategies of a company are Roles, Resources, Markets and Analysis. Roles are used to integrate the relevant actors that contribute to the business process. Resources represent multiple assets within the scope of the organisation. Markets include concepts that are important to strategic planning. Analysis offers a variety of concepts to aid the modeller in the definition and redesign of the company’s strategy. Some of these concepts are related to concepts of the other two modelling languages.

The MEMO-OrgML integrates the key concepts ProcessType, ProcessUse, ContextOfProcessUse, InputSpec, OutputSpec and Event, used to model a business process. A ProcessType contains two specialisations: BasicProcessType and ComplexProcessType, being the latter composed of multiple ProcessType. ProcessUse is useful in order to distinguish between many existences of the same ProcessType, aggregated in the same ComplexProcessType. To distinguish identical ProcessUse aggregated to different existences of ProcessType, the ContextOfProcessUse is assigned to every ProcessUse. InputSpec and OutputSpec serve as containers for information assets regarding Events. The MEMO-OML contains four categories of concepts: Semantic, used to define features of an object model concerning properties of the real world; Organisation, containing concepts used to reduce the complexity of a project by exercising adjustments in the aggregation of concepts; Resources associates artefacts that compose an information system; Management accommodates information that is valuable in the process of maintenance of instantiated object models.

The way MEMO is built, using different specific modelling languages for different foci in the conceptual framework, sharing common concepts among the different languages and enabling the cohesive assembly of the models over different aspects, allows for more intuitive, broad and deep representations of an organisation.

2.1.4 Enterprise Resource Planning II

Enterprise Resource Planning is a method used for the planning of a company, that incorporates internal and external management information concerning an organisation and uses it in the management of accounting, supply chain, manufacturing, sales, customer relationship, among others [6]. The demand for ERP systems increased considerably at the end of the 1990s due to multiple companies trying to solve the Y2K problem. However, after that, the ERP market stagnated, because it was believed that
ERP was no longer suitable to satisfy the new requirements of companies emerging at the time. ERP II is an evolution of ERP and was introduced in order to fulfil the new requirements of the next-generation enterprises. The GartnerGroup, creator of the concept ERP II, describes it as a business strategy that creates value amongst customers and owners of a company, by assisting collaborative, operational and financial processes, within the context of enterprise and inter-enterprise relationships.

The conceptual framework of ERP II aims to combine enterprise systems' concepts, resulting in a taxonomy and generic representation, suitable for corporate-wide enterprise systems. The ERP II conceptual framework relies on four groups of components [7]. The core components include the application framework and the integrated database (implementations may differ from company to company). The central component is the ERP itself, which is composed by its traditional modules, such as manufacturing, sales, human resources, financials, etc. The business process management is the base for ERP II systems, which are built considering the process as the central model entity. The corporate components support management decision regarding relations and corporate issues such as management of supply chain, customer and supplier relationship, product lifecycle, employee lifecycle and corporate performance. The collaborative components intend to establish the integration between the ERP II system and external actors, covering Business-to-Consumer (B2C), Business-to-Business (B2B), Business-to-Employee (B2E) and also Enterprise Application Integration (EAI).

This generic conceptual framework is considered to be the first step to achieve a tool that can be used to fully evaluate the completeness of an ERP II plan towards an organisation.

2.2 Business Process Modelling

The work and effort of every company is driven by its business goals, and the accomplishment of these objectives involves delivering value to a target customer, either by providing products or services. A business process consists of a structured set of activities within the scope of an organisation, which describes their logical order and dependencies, focused in one or more business goals, to produce, naturally, a desired result [8]. Other authors consider a business process as market-centred descriptions of the business activities of a company, encompassing both material and information processes [9], or even as a representation of the flow of work and information along the business [10].

In order to amplify the effectiveness and efficiency of its business, an organisation must assess its business processes. The most important reasons for modelling a business process include [11]: communicating and training the key mechanisms of the business process to stakeholders or new participants in the business; aiding the creation of suitable information systems that will support the business; improving the business carried out by the company, allowing the analysis, reengineering and improvement of its various processes, taking advantage of the conscious and informed decision making provided by the performed assessment; experimenting new business concepts and innovations, allowing the possibility of evaluating and estimating the profit, costs, risks and benefits of the implementation of the business changes under analysis; identifying portions of the business which are best suited to be executed by outside suppliers, being the models used as specification in the outsourcing process.
When considering a process language to represent and model the business of a company, a set of properties inherent to the language must be taken into consideration so the representation fully depicts the flow of work, information and resources of the process, as accurately as in reality. Some properties that are of the utmost importance in a process language are [12]: completeness, every part of the business process should be represented and reached, otherwise, the language is not capable of representing the business process in its wholeness; clearness, a process model must be capable of communicating the process to stakeholders over different perspectives, and it must do it in clear and intuitive way to every stakeholder, even if they come from different professional backgrounds; ease of use, the learning curve of the language should be as easy as possible, so it is not required a great effort for a person who does not have great experience with the language, to communicate the process using it; no redundancy, the language should be complete, as aforementioned, but it cannot contain any redundancy or ambiguity, otherwise it will invalidate other properties, such as clearness.

Reviews and explanations of a wide variety of business process languages, notations and techniques can be found in research work of various authors, but the most common are the following: Flow Chart, Data Flow, UML, BPMN, EPC and IDEF3. Additionally, related work on Enterprise Ontology and Business Motivation Model is also explored. More methodologies on business process modelling can be found throughout the research work in this area; however, they are not as relevant for the scope of this paper as the ones listed.

2.2.1 Equivalence of Activities

In the area of business process modelling, it is a common problem to get different blueprints of an organisation when they are built from different teams of process modellers. One additional inconvenience in this problem is that since different blueprints of the enterprise can compose a complete and thorough representation of the organisation, it can be quite hard to maintain the multiple process blueprints in adequate and functional repositories. Facing this problem with no sustainable solution to solve it, a research work was set on motion to find a suitable set of principles, in order to adequately identify equivalent blueprints of a given business process, and to appropriately generate multiple blueprints from a canonical process model, which will serve the purpose of providing different views of the processes towards professionals with different perspectives of the business [13].

According to the authors of the research work, two business process blueprints are considered to be equivalent if they can be described using the same properties, even if the processes have different names. This postulation leads to the first principle of equivalence of activities:

1. An activity A is dimensional equivalent to another activity A', both belonging to the same process, when each of these two activities have equal dimensions \( \{ D[n, i]; \ldots; D[n, i] \} \).

The process’s dimensions refer to the six dimensions of the Zachman Framework (See Section 2.1.1 Zachman Framework). Since the representation of a business process depicted under the Zachman Framework can be recursive, being composed of a hierarchical structure of representation, another principle arises in order to assess the equivalence of processes:
2. A process $P$ is dimensional equivalent to another process $P'$ when $P$ transitive closure $T$ is equal to $P'$ transitive closure: $T(P) = T(P')$. This means that every children of the recursive structure of both processes must have equivalent dimensions.

To analyse a real world use case, aiming at the assessment of processes’ equivalence within the business of a given organisation, two teams were assigned to blueprint its business processes. Another conclusion was obtained: the detection of partial equivalence between processes should not be discarded for being useless, because constitutes useful information in particular business scopes. As an example, while focusing in a logistics perspective of the business, the equivalence of the dimensions Where, What and When is sufficient to draw beneficial conclusions in the context of this part of the organisation, as in the unique equivalence of just one dimension, such as the Who, can be useful for the human resources department.

2.2.2 Flowchart

The flowchart technique can be very useful when dealing with processes that require a high level of detail. A flowchart is a structured graphical representation of a work or manufacturing process, whose symbols are used to represent operations, data, flow direction and equipment [8]. It is not possible to divide activities into tasks or more fine-grained activities, so, since activities and tasks are not distinguishable, the readability of the representation can sometimes be quite difficult. Also, there is no way of assigning responsibilities to activities in the chart, making it hard to relate activities to departments (or sections) of the organisation.

2.2.3 Dataflow

The dataflow technique, described in the same article by Aguilar-Sáven, consists of diagrams representing the flow of information with respect to a business process. These diagrams also communicate the relationships between processes, and the relationships between processes and the participants. This type of representation allows the specification of the process at the logical level, depicting its functionality. Although this technique does not portray the flow of resources, such as materials, it allows the breakdown of processes into sub-processes or tasks, in order to provide a more fine-grained level of detail of the representation.

2.2.4 UML

UML or Unified Modelling Language is a general purpose modelling language, commonly used as a standard modelling language in the application area of software systems. Even if it was not conceived with that purpose, UML has also been used to model processes, using either its notations for behaviour (such as activity diagrams) or extended profiles. An example is the extension with business concepts such as Resources, Goals, Processes and Rules [11].
Resources consist of objects that are part of the input/output of the business, namely people, materials, information and products, being these objects well structured, relating to each other. Goals represent the desired outcomes and purposes of the business, and can be broken down into sub-goals, in which resources can be assigned to. Processes consist of activities performed in the business context, which affect and transform relevant resources to the business. Rules constrain certain aspects of processes, define the flow of business and structure resources, establishing relationships between them. Activity diagrams can represent sequential and parallel activities, the objects used, produced and transformed by a given activity, the assigned responsibilities to each activity and the relationships between them [14].

It is understandable that considering the aforementioned features of this type of representation, activity diagrams comprehend characteristics that are important to business process modelling. Additionally, activity diagrams can be focused under certain perspectives, named views, particularly business vision, business process, business structural and business behavioural views, consisting respectively in a focus under the overall vision of the business, the relevant activities to the business and the produced value, the structural organisation of the business resources and the behaviour of each resource that is part of the business.

2.2.5 BPMN

The Business Process Modelling Notation, or most popularly known as the acronym BPMN, was created with the intention of providing a mechanism that could create business process representations easily communicated to a wide variety of business participants, without losing the ability to do it in a comprehensive and intuitive way [15].

The symbol objects that compose this notation can be organised in four different categories: Flow Objects, Connecting Objects, Swimlanes and Artefacts. Within the Flow Objects category, the symbols Event, Activity and Gateway can be used. The symbol Event is a relevant occurrence to the business, affecting the workflow, and is characterised by a trigger and a result. Events can be differentiated as Start Events, Intermediate Events and End Events, as they occur in the beginning, middle or end of the business process, respectively. Activities are used to represent work performed by the organisation, and can either be atomic, named Tasks, or composed, named Sub-Processes. A Gateway is considered a business decision, representing the divergence or convergence of the flow of the business process.

The Connecting Objects category is used to link together the various symbols of a BPMN diagram, and contains the connectors Sequence Flow, Message Flow and Association. These represent the order of execution of activities, the flow of messages between business participants and the association between objects, including the communication of inputs and outputs of a given activity, respectively.

Swimlanes are used to assign responsibility to functional capabilities, either by associating a business participant to a process, using a Pool, or by organising different activities inside a Pool, making use of a sub-partition named Lane.

In order to properly build the context of process representations, the usage of Artefacts is essential. Within this category, Data Objects communicate the required or produced data of activities. Groups
are used to organise diagram objects. Annotations are used to further inform the reader about certain aspects of the representation that cannot be provided by any other symbol of the diagram.

There are two types of business process models within BPMN, Collaborative B2B Processes and Internal Business Processes. Collaborative B2B processes serve to apprise the reader about the relations and interplay between business entities, namely the course of activities and message exchange between different organizational units involved in the business, not focusing in the private behaviour of the process, as in Internal Business Processes focus mainly in the inner working behaviour, leaving aside any public features that regard business processes.

One of the great advantages of using BPMN is related to the fact that it is considered by the business process modelling community, a standard process modelling notation, reaching a broad target public, that are familiar to the language, allowing a faster and more intuitive progress when learning about a given representation in this notation.

2.2.6 EPC

Event-driven Process Chain, or EPC, is a type of flow graph used to blueprint business process workflows, considered the main benefits of this graphical notation to be its simplicity and the easiness to understand of its graphical symbols [16]. In the 1990s, simultaneously with the evolution of the ARIS Process Platform, which serves as an integrated toolset for designing, implementing and controlling business processes for this notation, the EPC has been extended with a set of symbols that aid in the definition and representation of a business process model [17].

The various symbols and representations that compose the notation include Event, Function, Organisation Unit, Information, Material, Resource Object, Logical Connector, Logical Relationship, Control Flow, Information Flow, Organisation Unit Assignment and Process Path. An Event is a passive element that depicts the working circumstances of a function or a process, being required that an EPC graph starts and ends with a Start Event and an End Event, respectively. A Function is an active element that models tasks or activities regarding business, representing state transformation. A Function can be further detailed as another EPC graph, creating hierarchical structures within the scope of the main graph. An Organisation Unit, including the entity Process Owner, serves to assign responsibility of specific functionality to certain business processes. Information, Material and Resource Object consist of real world objects such as business entities, serving as input or output data, used as a basis for Function, or produced by Function, respectively.

Logical Connectors are used to depict logical relationships between elements of the graph such as events or functions, manipulating the control flow of the business within the model. Logical Relationships are composed of three different types, such as Branch/Merge, Fork/Join and Or, portraying exclusive business path decisions (just one path is activated), total concurrent business path decisions (all paths are activated), or multiple business path decisions (one or more paths are activated), respectively. Control Flow and Information Flow lay out the connections and relationships amongst events and functions, process paths or logical connectors, and between functions and input or output data,
respectively. Organisation Unit Assignment serves the purpose of representing the connection between an organisation unit and the function under its responsibility. Finally, a Process Path is used to reveal connections between processes.

2.2.7 IDEF3

IDEF3, with its official name Integrated DEFinition for Process Description Capture Mode, is a business process modelling method aiming at capturing the description of a scenario-driven process flow in an explicit way. One of the main drivers when developing the IDEF3 method was the need to distinguish between what the description of the system under analysis is supposed to do and the representation used to predict what the system will do. The method further captures precedence and causality relationships between situations and events in an intuitive way for experienced professionals in the area of business processes [18]. This method conforms two modelling modes, the Process Flow Description (PFD), and the Object State Transition Description (OSTD), characterising the business work within the organisation and outlining an object's allowable transitions in a business process, in that order [8].

A PFD is composed by the structures Unit of Behaviour, Elaboration, Referent, Junction and Link. A Unit of Behaviour represents a function, activity or process, and an Elaboration, which describes entities that participate in the business and their relations, can be associated to it. Units of behaviour can be broken down into other units of behaviour, creating a hierarchical structure. A Referent serves to graphically communicate information or direct the control flow of the diagram. Junctions and Links are used to create relations between units of behaviour. Junctions characterise the behaviour and process flow, namely, expressing synchronous and asynchronous behaviour amongst units of behaviour, as in links serve to express temporal precedence, object flows and relations.

An OSTD is composed by the elements Object State and State-Transition Arc. For an object state to be defined, the existence of the object in that state needs to be validated considering facts and constraints that need to be true, being characterised by entry and exit conditions, which are facts and conditions as well. Just like in usual directed graphs, arcs unveil the possible transitions between object states of the representation.

2.2.8 Other potentially relevant references

Enterprise Ontology

Ontology is, according to J. Dietz, a formal and explicit specification of a shared conceptualisation. It serves as a basis for the communication and understanding of a part of the world, more specifically in the enterprise domain, insight about the deep structure of an organisation, leaving aside information regarding the implementation of the system. This specification is communicated among a community of people that may have different professional backgrounds. The Enterprise Ontology should embody the properties of coherence, comprehensiveness, consistency and conciseness. This means that the different aspect models of the ontology must, when composed, make a whole, and they must also at all moments, be free from contradictions. The ontology must not contain superfluous material, and should
demonstrate the essence of the enterprise and its structure [19]. Enterprise Ontology organises a frame of knowledge, or at least, a set of terms related to the knowledge frame, like a dictionary or a glossary, although working in the reverse way [16].

DEMO, standing for Design and Engineering Methodology for Organisations, is a technique used for the design, engineering and implementation of organisations. It differs from other process modelling languages such as BPMN, UML, EPC and IDEF3 because these languages are method-independent, which implies that DEMO offers mechanisms to evaluate and assess the consistency and completeness of business processes [20]. DEMO is based on four core axioms that constitute the phi/psi-theory: Operation, Transaction, Composition and Distinction.

The Operation axiom concerns the characterisation of acts that can be performed by actors. These acts can be either production acts or coordination acts, the former, contributing to the accomplishment of goals and achievement of purposes, and the latter implying the compliance with mutual commitments about production acts. The Transaction axiom affirms that production and coordination acts occur in transactions. The Composition axiom states that every transaction is either confined in other transaction, or it is a customer transaction, or it is a self-activating transaction. The Distinction axiom is related to the role played by people when composing an enterprise [19].

C. Pereira also explains the four aspect models of DEMO, which are different perspectives upon the same model. The four models constitute the ontological knowledge of the organisation. The Construction Model defines the actor roles, information banks and transaction types, also evidencing connections between actor roles and information resources. The model is decomposed in two models, the Interaction Model and the Interstriction Model, which show active and passive relationships between actor roles, respectively. The Process Model identifies the transaction pattern of every transaction present in the Construction Model and shows the relationships between them. The Action Model explains the action rules that actors use in their activities. Each agendum type may have multiple rules, grouped based on the actor roles. The State Model identifies the entity and fact types of the production world and the associated rules and restrictions.

**Business Motivation Model**

Business Motivation Model (BMM) consists of a structured schema used to communicate and manage business plans, according to the notion of motivation, which is essential to map the organisation’s goals to the business strategies that drive and direct business processes. One of the main benefits of this model is that it evidences traceability traits between chosen approaches for business activity and the reasons for why the business is directed that way, and vice versa. Enterprise’s aspirations are of paramount importance to motivation related work, for they represent the vision of the company, as in action plans are used to realise them. The core concepts of the BMM are the Ends and Means. Ends is a term that refers generally to aspiration concepts, such as Vision, Goal or Objective. Means can be used to refer to action plan concepts, such as Mission, Strategy or Tactic [21].

Besides the company’s mission and vision, the enterprise must also take into consideration other concepts, such as Influencers. An Influencer can be an Internal Influencer, acting within the company
as a Strength or a Weakness, or an External Influencer, acting into the company as an Opportunity or Threat. Evaluating the leverage and impact that an Influencer has in the company is a very important task that will aid in the enhancement of the business plan development. Therefore, Assessment is a concept that integrates the identification and evaluation of the impact of Influencers on Means and Ends of an organisation. It allows the measurement of leverage of internal and external business elements, leading to the delineation of Directives, which can be Business Policies or Business Rules, to guide the enterprise's Courses of Action and direct it towards the desired results. Business Rules define Courses of Action at operational level, further providing specific solutions when a failure in a Course of Action occurs, as well as resolutions to conflicts that may arise between Ends.

The OMG organisation explains the key ideas of the Business Motivation Model, which can be useful to understand and take advantage of, in order to maximise the benefits of using the model [22]. Business Perspectives are the foundation for all the elements of the model. By developing business models for the elements of business plans before design of the system, a mapping between system solutions and business intents is achieved. Motivations set in motion business processes of a company and lead to the application of Business Rules, therefore these should easily evidence the business motivations that conduct them. Since people drive the motivation of an enterprise, the model should also link the business decisions to who has elaborated and enforced them, and based on which Assessments on Influencers. Reaction to Change is a fundamental assumption stating that business processes of an enterprise are executed based on how the company decides to react to change. The model presents bilateral traceability regarding reaction to change: forward, mapping the impact of influence to the specification of business processes, business rules and organisation responsibilities; and backward, mapping business motivations to business plans. Separation of Concerns favours simplicity in the model, by distinguishing:
Ends and Means, representing what a company wants to achieve as opposed to the business decisions taken to accomplish these goals; changes caused by Influencers and Assessments of their impact, and posterior analysis of the leverage of such changes; Assessments and the decisions based on them, as can be traced in Business Policies and Ends; and lastly, Courses of Action and Business Policies, based on the management decisions of resources and capabilities, as opposed to Directives that enable and restrict Courses of Action. Decomposition consists in the idea of defining and applying the BMM in different Organisation Units of the company. Metrics are measures of performance that are defined in the model as Objectives. These should be based on potential risks and benefits identified in Assessments.

### 2.3 Serious Games

The concept of serious games is quite vast, ranging from a manifold of definitions, considering different authors, who describe the concept under various perspectives or focuses. Some of the most relevant definitions will be stated. A serious game is about taking advantage of the power of computer games to encourage and engage players in a specific goal, or set of goals, being commonly associated to knowledge learning or skills improvement [23]. A serious game is considered a mental challenge, played under specific game rules, using entertainment for public or private purposes, these being training, communicating, learning, or even others [24]. A serious game consists of a voluntary activity, within the context of a simulated, conceptual or imaginary world created purposely for the game, to capture the player's attention, being the game played in a specific place and time, conducted under defined rules, and possibly leading to the emergence of social groups of players [25]. The concept definition that will serve as a basis for the development of this project is a combination of the aforementioned concept descriptions. A serious game is a game which main focus and purpose is beyond pure entertainment of the player, being used in a variety of tasks such as learning, training, simulating, informing, advertising, among others. Using a game instead of other more static tools, allow the players to engage in a specific set of goals under certain constraints and rules, defined by the game designer.

The involvement of serious games brings many advantages and benefits, such as the development of a variety of skills, including analytical, spatial and strategic skills or learning and recollection capabilities. Furthermore, serious games also improve problem recognition and solving, decision making and social skills, such as collaboration and shared decision making [26]. In the same article, T. Susi et al., also lists the benefits of serious games in the corporate application area as improvements in people skills (teamwork and general enhancement of performance within the company), job-specific skills (know-how regarding software or hardware needed for the job position), organisation skills (resources and time management), communication skills (including cooperation and coordination with co-workers) and strategy skills (goals definition and proceedings to reach them).

The main application areas of serious games are as extensive as the own definition itself, and can vary from education, healthcare, corporate, simulators, military, government, virtual reality, etc. Some of the existing projects include: CyberCIEGE, a game sponsored by the U.S. Navy used as a training tool to teach network security concepts; Democracy, a government simulation game in which the player
assumes the leadership role, such as president or prime minister of a democratic government; Intelligym, a game-like training program developed to improve cognitive performance of professional athletes; Microsoft Flight Simulator, the longest running and most complete home flight simulator available on the market (see Figure 2.4); NoviCraft HRD, a game that supports the learning of business customers to construct shared understanding between different people in changing contexts; EteRNA, a game in the area of biological science, designed to engage players in solving puzzles related to the folding of RNA molecules; and Houthoff Buruma The Game, a game designed to challenge the skills of their players in the area of corporate acquisitions, taking into account the client’s and shareholders’ objectives.

A study was made on the present existing serious games in the context of software engineering education [27], which reveals that great part of them assigns the player roles such as project manager and involve decision making in the areas of processes, business life cycles, resources’ management, company-customer relationships, etc. These decision making actions can then be evaluated and analysed on how they leverage the project success. The company PIXELEarning describes a few projects in the area of serious business games [23]: LearningBeans, a manufacturing application developed by them that allows the player to learn through a highly detailed scenario including the tasks of manufacturing, sales, marketing, human resources, finance, production and distribution; Virtual-U, designed to promote a better understanding of the management process in American universities; at last, Better Business Choices, a simulation game developed by British Telecom that features the social and environmental management of a business, being the role of corporate CEO assigned to the player.

Other gamming environments were developed in order to aid in the requirement elicitation process, such as iThink [28], a web-based gamming environment that allows and encourages its players, in a collaborative way, to bring out and generate new requirements regarding the desired system, and also to analyse, evaluate and comment on the previously existing requirements, as a way to strengthen the requirement elicitation process, and make the elicitation itself as complete and relevant as possible (see Figure 2.5). ImPROVE [29] is a 3D gamming platform aiming at improving the mechanism of modelling a business process and promoting organisational learning and change. Its simulation features allow the visualization and exami-
nation of the impact of the parameterised values and costs in the business. The context of the simulated environment is a Portuguese emergency unit hospital, based on the specific implementation of a previously existing health care triage system, the Manchester triage system. The ImPROVE platform relies on a methodology named Time-Driven Activity-Based Costing [30], which aims at defining processes as a composition of activities. By calculating the cost of resources involved in an activity and the time to perform a given task, the cost of a particular activity can be estimated and obtained by the multiplication of the time by the unit cost. By exploring and analysing the existing gamming platforms that assist and help in the corporate area, more specifically in the delineation and definition of a model of the business process of a given organisation, one can see that the results in projects previously executed have been quite positive, revealing that a gamming platform might be a strong solution and a useful tool to be used within an organisation, by the employees that contribute somehow to the business, to model it, for posterior analysis and informed and beneficial change.

In the interest of developing a gamming application, to follow a set of principles is considered to be a very beneficial task, so the development process goes in the right direction. Schell lists the four basic elements of a gaming application as Story, Mechanics, Aesthetics and Technology, and defends that all the four elements are connected and dependent of each other. Consequently, every gaming platform can be described taking into account the definition and explanation of these four elements [31]. Story corresponds to the sequence of events that arise until the goal of the game is achieved, being this sequence linear and static, or branched and dynamic Mechanics defines the procedures and rules of the game, therefore shaping the flow of the game play. It also defines the game goal(s) and how players can and cannot achieve it/them. Aesthetics complement the Story and Mechanics elements by providing appearance, sound and animation to the game, being the responsible element to form the look and feel of the game. Finally, Technology encompasses the materials and interactions that make the game possible.

2.4 Game Technology

The video games industry is currently valued as a $70+ billion business, and due to the fast growth of this market and worldwide demand, the evolution and development of improved software and powerful hardware have been growing accordingly [32]. Modern computer games use technology from an extended number of computer science areas, such as graphics, artificial intelligence, networking, algorithms, parallelism, programming languages, etc. [33]. In order to reduce complexity, improve the overall quality of games and promote the outstanding evolution of this industry, cutting edge game engines have been developed to overcome the problems associated with the ceaseless expansion of the video game’s business.

A game engine is a software system designed specifically for the creation and development of games that either run in game consoles or desktop operating systems, providing a functional core that assist in the development of the game application, reducing costs, time to market, complexity and unnecessary work in base functionality, common to every game. The main concern of present game engines is to
provide a collection of modules of available functionality and features that do not directly specify the
game's behaviour or environment. These modules make the job of game developers much easier by
providing features that cover multiple areas in computer science technology, including: input handling,
output (such as 3D rendering, 2D drawing and sound), generic dynamics for game worlds, networking,
artificial intelligence, animation, scripting, etc.

The power of a game engine can be also strengthened by resources, usually organised in asset
stores, that offer valuable assets for game developers, like 3D models, textures and materials, audio,
complete projects for learning, editor extensions, scripts, particle systems, among others. The fact that
game developers using these game engines are spared of doing the dull programming of the generic
base needed for every game, give them more time to think about the desired particularities and higher
level aspects of the application in development, and work in the particular features of the game.

A brief search on the existing and available game engines reveals that the number of results is
considerably generous. Nevertheless, to decide which game engine should be used to develop a certain
game, a popular one with numerous years in the market should be preferred, because that may indicate
that the game engine will be more bug free, more communities willing to help users will be available and
more documentation and helpful material will be accessible. In the beginning of a game's development
lifecycle, the developer or team of developers must also think of a list of priority features that a game
engine must deliver, so the tools and features required to develop the game meet the features and
functionality offered by the game engine.

The Unreal Engine, developed by Epic Games, the same creators of Unreal Tournament, and the
Quake Engine, developed by id Software, the same creators of the namesake game, constitute two of the
most used game engines in scientific research [34]. IdTech, developed by id Software, serves the same
purpose of the two aforementioned game engines. Their primary focus is in 3D first-person shooter
games, but they have been used through time in the development of other genres, such as stealth or
RPGs. Although they are very practical tools to research on artificial intelligence and learning, they are
not that practical to use if the game in development aims at an environment different than a first person
shooter. CryEngine, developed by Crytek, is a game engine exceptionally focused in high quality and
high fidelity 3D rendering, providing a broad variety of tools that help the user bringing realism to their
games.

Unity 3D is a cross-platform engine developed by Unity Technologies. Some of the available features
include multi platform assisted development, which allows a developer to transport its game into different
platforms (namely Windows, Linux, Mac, Android, iOS, Unity Web Player, Adobe Flash Player, PS3,
Xbox360 and Nintendo Wii) with no difficulty; networked multiplayer assisted features, offering a variety
of networking approaches and methods of network connection to facilitate the developer in the task
of creating a game than can be played between more than one player at the same time in the same
context; integrated Unity editor, allowing coupled development and debugging tools to assist in the
product development lifecycle; external script editor integration, making more flexible the integration
of scripts that will model and control the game flow and mechanics; rapid iteration with "play" mode,
allowing a fast visualisation of the progress of the game without the need of recompilation of the project
to evaluate modifications, and last but not least, Unity 3D offers an easy learning curve, granting the
developers a fast progress and evolution in theirs skills when using the game engine.

IW Engine is a game engine developed by Infinity Ward, which is based on idTech 3 and is fa-
mous for the development of acclaimed games such as Call of Duty or Quantum of Solace. Source
Engine is a game engine developed by Valve Corporation, in active development since the release of
Counter-Strike, in 2004. Although it was designed to accommodate first-person shooters, it has also
been professionally used to create puzzles, real-time strategy games and MMORPGs, evidencing its
great flexibility. It provides features that enable high quality rendering with high performance, supplying
scalable multiprocessor support. RAGE is a game engine developed by the RAGE Technology Group,
whose creation was driven to facilitate the development of games aimed to run in Microsoft Windows,
PS3, Nintendo Wii and Xbox360.

<table>
<thead>
<tr>
<th>Feature</th>
<th>UE</th>
<th>idT</th>
<th>CE</th>
<th>U3D</th>
<th>IW</th>
<th>SE</th>
<th>RG</th>
</tr>
</thead>
</table>

Table 2.1: Comparison of features among researched game engines.

Table 2.1 portrays a visual comparison between all the game engines listed and described above,
considering different features, tools and characteristics of each game engine. According to the matrix,
given a certain feature of a particular game engine, if the value of the cell is ![Symbol], ![Symbol] or ![Symbol], then the
feature is fully implemented, partially implemented, or not implemented at all, respectively.
Chapter 3

Problem Analysis

After reviewing multiple business process modelling techniques in Chapter 2. *State of the Art*, relevant in the context of the proposed goal of this dissertation, it is now made an assessment on how business processes are elicited within the context of an organisation.

Starting with the highlighted stakeholders, main business actors implicated in process elicitation are acknowledged. Afterwards, it is elaborated an analysis and description of the available techniques used for process elicitation nowadays. Lastly, substantial limitations and complications that come from the described elicitation approaches are understood, in order to delineate strategies to diminish them and accomplish a solution that enhances projects whose intention is to define the business process models of a company.

3.1 Business Actors

With the purpose of obtaining a local view of the business processes performed by a certain organisation, multiple business actors are involved in process elicitation. An outline of the most important business actors in the problem being analysed is presented as follows:

- **Business participants** involved in the company’s processes.
  Employees who are directly involved in the business processes performed by an organisation possess the type of knowledge that facilitates the task of defining business process models. Consequently, to accomplish a realistic vision of how the processes are executed, the involvement of all process participants in the elicitation of business processes is of the utmost importance.

- **Administrators** regulating process elicitation.
  Projects that regard process elicitation regularly concern the top management of a company, because they are commonly directed at obtaining business process models so they can be perfected afterwards, increasing the efficiency of processes. Although administrators are not usually involved in the elicitation process itself, the initiative to start new projects of this kind is often theirs. Therefore, it is of their best interest that the elicitation process follows some guidelines, aimed at achieving certain goals (*e.g.* process improvement, learning, training, etc.).
• **Consultants** performing decision making.

After the conception of the business process models and consequent validation of their accuracy and legitimacy, they can be used by business consultants in the process of decision making in order to perfect the business activities performed by the organisation.

### 3.2 Tasks

Considering the description of business actors provided in the previous section, the following tasks portray how business process elicitation is normally carried out:

- **Define Business Scope**: Administrators reach a consensus regarding the business scope of the business process elicitation, choosing which processes are to be modelled.

- **Contribute with Business Knowledge**: Business participants contribute with business knowledge, providing their local view of the business processes they are involved.

- **Perform Elicitation Techniques**: Consultants perform numerous elicitation techniques and approaches, to collect the necessary implicit and explicit business knowledge to create business process models.

- **Build Business Process Representations**: Consultants put together and consolidate the gathered knowledge about the business processes, building the process models afterwards.

### 3.3 Process Elicitation Techniques

Within the field of business process management, there exists many techniques that business analysts have the possibility to choose, according to which ones they think will be most successful in eliciting business processes, such as:

- **Interview techniques**
  Teams of business analysts, aimed at extracting the largest amount of relevant business knowledge as possible, conduct interviews to business participants. This kind of technique might also include interactions such as storytelling activities or role-playing games.

- **Analysis of the existing documentation**
  Business analysts study the documentation available on the business processes performed by the company, which may include schedules, stakeholders’ identification, procedure descriptions, etc.

- **Direct observation of business actors**
  Business analysts examine detailedly the business participants performing their daily professional activities, enclosed in the business processes to be elicited.
• **Brainstorming sessions on process modelling**

Right after the necessary business knowledge about processes is gathered, a team of modellers is united to collaboratively interpret the information and build business process models from the resultant conclusions they made consensually.

• **Use of graphic languages**

Since business knowledge present in process models is graphic for nature, modellers choose one or more well-known graphic notations to represent business processes in a structured way.

Other types of approaches are also commonly used, such as: a) focus groups, leveraging group interaction to bring out ideas about business processes; b) surveys, contributing with information via questions (physically or virtually) to sample groups of business participants; c) storyboarding, consisting in group meetings that focus in the construction of high level models that depict business processes; etc.

However, these techniques are detailed no further because of their similar technical nature to other aforementioned described methods.

### 3.4 Substantial Complications

Although the aforementioned approaches have been used in enterprise environments for quite a while, they are not perfect and present some flaws that may invalidate the results of the process elicitation. Some problems that may arise from the practice of these methods are detailed below:

1. **Difficulties in communication**

Main issues which complicate process elicitation are directly related to gaps in communication between interviewer and interviewed, in case interview techniques are used.

Sometimes the business domain of processes is so complex that business analysts struggle to fully grasp the tacit knowledge being transmitted, making it difficult to formulate adequate questions to the business participant because of the lack of common vocabulary. These cases generally happen with businesses that use very specific terminologies (e.g. insurers, law firms, etc.). Furthermore, the interviewed may even conceal relevant information, considering it to be too evident to expose, despite the fact that it might not be in the perspective of the interviewer.

Additionally, due to the subjectivity of the oral and written language, respondents may even provide ambiguous information without noticing. In this situation, it may be troublesome for business analysts to detect these incongruities and filter the correct information.

2. **Lack of explicit information**

As aforementioned, one of the available approaches used in process elicitation is the analysis of existing and relevant information concerning the performance of business activities. In case there is a lack of documentation or even a complete absence of processes’ explicit information, the business analyst has to rely on other techniques that may not contribute with the same adequate result.
3. **Restriction imposed by skills**

While choosing process elicitation techniques, the business analyst may pick one approach that although is based on his/her expertise, may not be the correct approach to collect the necessary business knowledge. On the other hand, if the chosen approach is the most appropriate one, the business analyst may lack experience in practicing it. Consequently, the resultant business process representation built by the business analyst may be restricted by the lack of proficiency of professional skills.

4. **Time-consuming construction of representations**

The most common scenario in business process elicitation is that the way it is carried out consists in the combination of diverse techniques performed by a team of business analysts and modellers. Accordingly, the development of these approaches results in very time-consuming projects, allocating a large amount of human resources, and consequently leading to high costs to the company whose process management is being exercised.

### 3.5 Conclusions

Considering the elaborated analysis on how process elicitation is commonly performed, it is noticeable that a more productive solution can be complementarily used with the techniques presented in Section 3.3 *Process Elicitation Techniques*, or even replace some of those.

Because of the nature of complications brought by the described techniques, an hypothetical solution must guarantee that process elicitation is made in a systematic and automated way, the business knowledge is contributed directly by business participants, and the result is a consolidation of their local view of the business processes they are part of. By accomplishing so, it is assured that difficulties in communication and restrictions imposed by consultant’s skills are excluded from the scenario, because there is only one actor responsible for building business process models, the business participants.

The next chapter contemplates a formalisation of a proposition of system that meets the description given above. Thus, a concretisation of a solution of this kind puts aside some of the problems listed in section 3.4 *Substantial Complications*, with the possibility of integration of this system in enterprise environment as a possible approach in business process elicitation.
Chapter 4

Proposed Solution

In the previous chapter it was analysed the main complications regarding how business process elicitation is accomplished nowadays. The solution detailed in this chapter aims at diminishing some of these problems by using state of the art techniques and technology, scrutinised in Chapter 2. State of the Art.

4.1 Overview

This dissertation proposes a solution based on the creation of a serious game that engages players to elicit the business processes they are involved, within the organisation they operate professionally. The game should integrate features that make possible the creation of structured business process representations. Additionally, the gaming platform must motivate players in building the most accurate and realistic representation according to their local view of the processes they act as business participants. Since some of the built representations may be more precise and realistic than others, the gameplay must not only stimulate the competitiveness between players to determine who achieves the best process representation, but also to encourage the cooperation among them, to obtain the best possible representation of business processes, considering the point of view of all participants involved. For as long as the game is being played, a consensus must be reached between the players, regarding which of the representations are elected to be portraying the elicited processes more correctly. Thus, the output of the game consists of a set of business process representations built and decided by players to be the process models that best represent how the processes are carried out in real life. Lastly, the game results may be exported in a well-known and structured format, to be finally analysed by consultants, who will use that business knowledge to perfect the current processes, instruct newcomers of the organisation, or even do decision making in respect to desired modifications to the processes.

4.2 Game Requirements

Described below are the requirements for the proposed gaming platform. The requirements are derived from the analysis made to the problem and the stipulated objectives of this project.
4.2.1 Functional Requirements

Functional requirements specify the desired functionality for the application, allowing players to accomplish their tasks, therefore satisfying the business requirements.

REQ.1 - Contribute with business knowledge

The player must be able to define organisational scenarios, which will be based on the creation of business process representations, that communicate in a manner the local view of business processes, according to real business actors, who can make a decision on building the models using different levels of detail.

REQ.2 - Graphic representation of business processes

The gaming platform must provide a graphic representation language that assists players in the representation of the business processes they are involved. Since not all players may have the theoretical and practical knowledge about business process modelling, the graphic language must be simple and easy to learn and use.

REQ.3 - Propose modifications to other players’ contributions

Posterior to the creation of business process models, given a particular business process, players other than the author of that process representation must be able to propose modifications. According to the players’ opinion, the proposed alterations will increase the accuracy of the process representation, compared to how the process is performed in real life.

REQ.4 - Rate other players’ contributions

It must be possible for players to rate the contributions other players have created during the game. Consequently, there must exist a voting process that reflects the global opinion of players regarding a certain contribution. At the end of the voting process, the result (i.e. the contribution's rate) must show on an established scale how much do players think that the contribution enriches the final representation of the processes performed within the company.

REQ.5 - Identify a process model as a duplicate of other

It must be possible for a player to mark a process model created by other player as a duplicate. In order to effectively identify a model as a duplicate, it must be possible for players to vote collaboratively on whether the representation is truly a duplicate or not.

REQ.6 - Game settings

For every instance of the game, the platform must allow for administrators to set up a bundle of configurations that will be applied to the soon to be played game. Configurations may consist of setting the game’s time length, the number of players, among others.

REQ.7 - Export business knowledge

The system must allow for administrators to export the business processes elicited in the game, to posteriorly use them for decision making. The exportation result must be implemented in a well-known
markup language to structure and store the information (e.g. XML).

**REQ.8 - Save and load game instances**

Whenever a game instance is being played, it must be possible for administrators to save the current game state. Thus, the system must also allow administrators to load a previously saved game state, so players can resume the contributions they were building.

**REQ.9 - Visualisation of game results**

The platform must allow players to visualise the final results of the game instance they played.

### 4.2.2 Non-Functional Requirements

Non-functional requirements specify different types of demands on the system, rather than specific behaviour (e.g. usability, portability, extensibility, etc). The following requirements were defined according to the considered needs of the desired application.

**REQ.10 - Multiplayer application**

The platform must allow for different players to play the same game instance simultaneously. This requirement is directly related to functional requirements **REQ.3** and **REQ.4**.

**REQ.11 - Cross-platform application**

The application must not restrict the players regarding the platform to be used to play the game. Therefore, players must be able to play the game using computers with different configurations and/or operating systems.

### 4.3 Use Cases

The following use case diagram depict the multiple interactions that users of the proposed gaming platform are allowed to perform. Each use case is then described, accompanied by the identification of corresponding requirements.

- **UC1: Contribute with Business Knowledge**
  
The actor defines organisational scenarios by depicting the business processes he or she is involved, as business process representations. These representations encompass the local view of business participants in a manner that communicates the way these actors perceive the business processes. (Requirements **REQ.1**, **REQ.2**)

- **UC2: Cooperate with other Players**
  
The actors play the gaming platform simultaneously, in a way that allows them to cooperate with each other in the contribution of business knowledge about the processes they participate. (Requirements **REQ.2**, **REQ.3**)

30
Figure 4.1: Use case diagram for the proposed solution.

- **UC3: Make Collaborative Decisions**
  The actors evaluate the business knowledge contributed by other players by classifying and categorising it in terms of quality and accuracy. Thus, reaching a consensus by making decisions in a collaborative way. (Requirement REQ.4, REQ.5)

- **UC4: View Process Representations**
  The actor visualises the resultant representations of the elicited business processes, as well as his individual contributions. (Requirements REQ.2, REQ.9)

- **UC5: Set Game Configuration**
  The actor sets a game configuration that will parameterise the game instances to be played posteriorly. (Requirement REQ.6)

- **UC6: Save & Load Game State**
  The actor saves the information concerning the game instance being played at the moment, or restores that information, so the game instance can continue to be played after it was interrupted. (Requirement REQ.8)

- **UC7: Export Business Knowledge**
  The actor exports the business knowledge specified in the platform by its players. (Requirement REQ.7)

### 4.4 Business Process Graphic Language

In order to construct a graphic notation capable of representing business processes, multiple desired features can be outlined in order to decide the most appropriate type of representation that should be
used by the platform.

Since the target player is the employee of the company that decides to elicit its business processes, players will vary not only in the knowledge of business process modelling, but also in education, level of skills when dealing with computers and software, and also in their ability to solve problems. To accommodate the diversity of target players, the process language domain should be as simple as possible, because it is not certain that the player has any technical knowledge about business processes. Along with simplicity of domain, the language should also be easy to use in a general way.

However, despite the language’s simplicity, the graphic notation must represent comprehensively, accurately and detailedly the business processes of any organisation, without domain restrictions jeopardising the business process representations.

In order to elaborate a process notation that has the capacity to represent business processes and is easily translated to other BPM languages, the notation is based on primary elements of the languages BPMN and UML.

### 4.4.1 Pool and Lane

Pools group participants in a business process, which contains only one single pool. Pools can be broken down into Lanes, which are sub-partitions of a pool. To each lane is assigned a specific entity or role, encompassing activities that are performed by the entity or role in question. Pools and lanes consist of containers of language primitives aimed at specifying the business process in question. Therefore, one primitive must belong to one and only one lane, and consequently, to one and only one pool.

![Figure 4.2: Graphic representation of a pool with two lanes.](image)

If there only exists one lane in a pool and no business participant is assigned to it, the pool contains a default lane, whose scope is more general.

**Rules:**

1. Each pool must contain one and only one start and end events.

2. Each process must have one and only one pool. Each pool must have at least one default lane (with no participant), or one or more lanes, each assigned to business participants. Every assigned participant may either be an individual participant or a department of the organisation.
4.4.2 Event

Start, Merge and End events may be grouped into a set responsible for the control of the process flow and are graphically represented by a circular shape (see Figure 4.3). Start activities are coloured green and contain a white horizontal arrow. Merge activities are coloured blue and contain a white symbol of two arrows merging together. End activities are coloured red and contain a white diagonal cross.

![Event Types](image)

Figure 4.3: Graphic representation of events in charge of controlling the process flow.

Start events mark the beginning of the sequence flow of a given process, as in end events mark the termination of execution. Merge events synchronise the execution of multiple activities. If multiple primitives are connected through incoming flows to this event, then the outgoing flows are only executed when all primitives connected are finished executing.

Rules:

1. Start events may prevail as a source to multiple outgoing flows, but accept no incoming flows.
2. End events may prevail as a target to multiple incoming flows, but accept no outgoing flows.
3. Merge events accept multiple incoming and outgoing flows.

4.4.3 Activity

There are three different kinds of activities: Work Activities, Composed Activities and Ad-Hoc Activities. Activities are represented by a round cornered rectangle and a name for the activity at the centre of the shape (see Figure 4.4).

![Activity Types](image)

Figure 4.4: Graphic representation of activities.

Work activities describe atomic units of work that cannot be broken down to a further level of business process detail. Composed activities consist in the combination of one or more work activities. Ad-hoc activities consist of a group of work activities with no required sequence relationships, and therefore, admits no sequential order.
Rules: (Work Activity)

1. Each work activity must have a name.

Rules: (Composed Activity)

1. There is only one single lane in the pool of a composed activity.

2. The only primitives that are not allowed in a composed activity are composed activities themselves and ad-hoc activities. Events, work activities and flows are allowed similarly like in a process.

Rules: (Ad-Hoc Activity)

1. There is only one single lane in the pool of an ad-hoc activity.

2. Work activities are the only allowable primitives that may be contained in an ad-hoc activity. Therefore, there are no events that control time nor sequence flows that order the executability of work activities.

4.4.4 Flow

The Flow between two primitives is represented by a solid line and arrowhead at the centre of the flow length. Sequence flows are coloured black, as opposed to information flows, which are coloured green (see Figure 4.5). The internal representation of a flow consists of an ordered pair, in which the first element of the pair embodies the beginning of the flow and the second element the end of it.

![Figure 4.5: Graphic representation of flows.](image)

Sequence flows communicate the order of execution of primitives, and may be accompanied by a condition. When two primitives are connected by a sequence flow, the second primitive is only performed when the former is completely executed, unless the flow is conditional (a boolean variable is assigned to the flow), and in that case, the second primitive is only performed when the condition assumes the value `true` and the first primitive is already finished.

Information flows acquaint the exchange of information between participants/lanes. This type of flow identify the information passed along two primitives, which may be data, business artefacts or even materials used in the business process.
Rules:

1. Each primitive must be reached through flows and must connect to a subsequent primitive, except for start and end events, whose flow begins and ends in the event itself, respectively.

2. The flow between two primitives may be created whether the two primitives belong to the same lane or belong to lanes from different participants. However, it is not possible to create a flow between primitives that belong to different processes.

3. Loop sequences can be created by choosing as source and target primitives the same primitive. Only Work, Composed and Ad-Hoc activities are allowed to have loop cycles. The execution of the loop activity will be performed for as long as the condition of the flow remains true.

4.4.5 Language Rules and Constraints

Along with the explanation of elements and delineation of rules regarding each primitive, the grammar of the process language can be summarised by the rules below, to provide a formal composition that communicate how primitives interact with each other.

1. $Process \rightarrow Pool^1 \supset Start^1, End^1, Activity^*, Flow^+$

2. $Pool \rightarrow Lane^+$

3. $Lane \rightarrow Primitive^+$

4. $Primitive \rightarrow Event | Activity$

5. $Event \rightarrow Start | End | Merge$

6. $Activity \rightarrow Work Activity | Composed Activity | Ad-Hoc Activity$

7. $Flow \rightarrow Sequence | Information$

8. $Flow \rightarrow <Start^1, Activity^+> | <Activity^+, Activity^+> | <Activity^+, End^1> | <Activity^+, Merge^1> | <Merge^1, Activity^1>\$

9. $ComposedActivity \supset Lane^1 \supset Start^1, End^1, Flow^+, Primitive^* \setminus \{Start, End\}$

10. $Ad-Hoc Activity \supset Lane^1 \supset Work Activity^+$

Elements that compose an ordered pair are represented between angle brackets, such as $<ElemA, ElemB>$. Multiplicities of language elements are represented in superscript on the right side of each element, varying from $^1$ representing only one instance of the element, $^*$ zero or more instances, $^+$ one or more instances, and finally $^{2+}$ representing two or more instances. Work Activity, Start, End and Merge are considered primary elements, since they cannot be decomposed any further.
4.5 Domain Model

This section intends to determine and confine the domain of the proposed serious game. A composition of the essential entities in the solution’s domain model is revealed, and a discussion of each entity and its role in interacting with others is presented.

A list of the existing classes in the domain model of the platform is displayed in Table 4.1, along with a brief explanation of each entity. Figure 4.6 unveils the existing entities that belong to the domain model of the desired system, focusing in the relationships and multiplicities between entities.

![Figure 4.6: Domain model for the proposed solution.](image)

<table>
<thead>
<tr>
<th>Entity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>Set of sequenced and structured activities that compose and represent the business process being elicited. Main entity of the domain.</td>
</tr>
<tr>
<td>Process Version</td>
<td>Conceptually, it is similar to Process, except that it is linked to one and only one Process. (Relationships in Figure 4.6 are hidden for reasons of simplicity.)</td>
</tr>
<tr>
<td>Vote</td>
<td>Vote of one player in polls regarding quality or duplication of a Process.</td>
</tr>
<tr>
<td>Lane</td>
<td>Encompasses Primitives under the rules established in Section 4.4. The set of Lanes of a Process must contain at least one start and one end Events.</td>
</tr>
<tr>
<td>Primitive</td>
<td>Abstraction of Activities, Events and Flows.</td>
</tr>
<tr>
<td>Flow</td>
<td>Connects two Primitives, which can either be Activities or Events.</td>
</tr>
</tbody>
</table>
### Event
Controls sequence and information Flows between Primitives (except Flows themselves), beginning and terminating in start and end Events, respectively.

### Activity
Abstraction of Work, Composed and Ad-Hoc Activities.

### Work Activity
Atomic unit of work, performed within the scope of the business process.

### Composed Activity
Sequenced and order sub-set of Work Activities and Events.

### Ad-Hoc Activity
Sub-set of Work Activities that lack sequence or order.

**Table 4.1: Description of entities of the domain model.**

<table>
<thead>
<tr>
<th>Entity</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Start</td>
<td>Establishes the starting point of the flow of the Process entity.</td>
</tr>
<tr>
<td>Event</td>
<td>End</td>
<td>Establishes the ending point of the flow of the Process entity.</td>
</tr>
<tr>
<td>Event</td>
<td>Merge</td>
<td>Multiple incoming and outgoing Flows. The outgoing flow is only reached when every execution of primitives connected by the flows are finished executing.</td>
</tr>
<tr>
<td>Flow</td>
<td>Sequence</td>
<td>Establishes precedency between two Primitives, which can be Activities or Events.</td>
</tr>
<tr>
<td>Flow</td>
<td>Information</td>
<td>Specifies the flow of information between Activities, which may belong to different Lanes.</td>
</tr>
<tr>
<td>Vote</td>
<td>Quality</td>
<td>Represents the vote of a process poll regarding the accuracy, completeness and precision of the representation.</td>
</tr>
<tr>
<td>Vote</td>
<td>Duplication</td>
<td>Represents the vote of a process poll aimed at reaching a consensual decision between players, to determine if the process in question is a duplicate of another.</td>
</tr>
</tbody>
</table>

**Table 4.2: Types of entities of the domain model.**

### Process Entity
The most important class in this project is the Process entity, because it consolidates the implicit business knowledge, captured by players, about the business processes they participate. It also has assigned game data that classifies the process scores, and is related to Process Versions, which are resemblant to processes. Figure 4.7 and Table 4.3 depict how the Process entity evolves throughout the game.

Processes whose state is Original (Published) and Version (Published) are accessible to all players, as in processes in the state Original (Draft) and Version (Draft) are only accessible to the author.
Figure 4.7: State diagram for the entity Process.

<table>
<thead>
<tr>
<th>Process State</th>
<th>Visibility</th>
<th>Editable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original (Draft)</td>
<td>Private</td>
<td>Yes</td>
<td>Representation in development.</td>
</tr>
<tr>
<td>Original (Published)</td>
<td>Public</td>
<td>No</td>
<td>Published representation.</td>
</tr>
<tr>
<td>Version (Draft)</td>
<td>Private</td>
<td>Yes</td>
<td>Representation version in development.</td>
</tr>
<tr>
<td>Version (Published)</td>
<td>Public</td>
<td>No</td>
<td>Published representation version.</td>
</tr>
<tr>
<td>B.P. Representation</td>
<td>Public</td>
<td>No</td>
<td>Collaboratively and consensually decided to be an accurate representation.</td>
</tr>
</tbody>
</table>

Table 4.3: Process states and respective properties.
Chapter 5

Game Design

Based on the designed solution introduced in Chapter 4. Proposed Solution, the purpose of this chapter is to introduce the rationale for the most important game decisions, and to detail multidisciplinary aspects that concern these choices. Starting with a brief overview through the gaming phases, it is then exposed the fundamental aspects of the game design, such as concepts, mechanics, interfaces and scoring system, among others.

5.1 Game Overview

The main purpose of each game session is to **consensually and collaboratively reach a business process model that best represents how the process is executed in enterprise environment**. Each game session is divided into the following phases:

1. **Pre-Gaming Phase**: In the first phase, administrators decide on which business process is going to be elicited by players in the game, and at the beginning of each session, once players enter the game, they are informed of which process should be elicited.

2. **Process Publication Phase**: The second phase consists in the construction and publication of business process representations. Players must use the set of language primitives and combine them under the rules and constraints defined in the previous chapter.

3. **Collaboration Phase**: In the third and last phase, players can execute multiple actions that are either related to their own processes or processes published by other players. These actions aim at enforcing multiplayer collaboration in process reviewing, rating and duplication detection. Additionally, players can also propose process modifications, in order to materialise adjustments they conclude it will improve the information assembled in the process model.

At the end of the game, according to process ratings which are a result of multiple interactions between players in the game, one or more process models are elected to be included as legit representations of the business process in real life, namely models with best classifications in the game.
5.2 Game Concepts

The following subsections aims at disclosing the most important concepts used throughout the game.

5.2.1 Process Creation

After administrators decide which process is to be elicited in the current game session, players are able to use tools provided by the platform to create the process. When a new process is created, a pool containing a default lane with a start and end events is automatically created. Given a previously created process, a player is able to create a new primitive and add it to the context of the process, as long as the result of the addition constitutes a valid operation within the language grammar defined in section 4.4.5 Language Rules and Constraints. It is also possible to create sequence flows between two primitives of a process, as long as it respects the language rules regarding this subject.

The intention of removing a primitive from a given process can also be manifested. Elements that are directly related to the primitive are eliminated from the process as well (e.g. Flows connecting target or source activities being removed).

A player is able to reposition a primitive. This action is useful to organise and group primitives within lanes of a pool. In case the player attempts to reposition the primitive to outside the process pool, the previous primitive position is restored.

5.2.2 Final Consensus

Throughout the game, multiple business process representations are built by players. However, the fact that many processes are elicited does not mean that all the models should be part of the game output, because some may be incorrect, or may even be considered duplicates of others.

The final consensus is the minimum set of published process models, with best rate of positive quality votes, which is not consensually marked as a duplicate. If a process model belongs to the final consensus, it means that players consensually elected that model as the one that best represents how the process is performed in real life, and therefore, the process author is compensated.

5.2.3 Process Versions

Whenever a process is published, it is possible for players to create alternative versions of that process, posteriorly existing two versions of the process model: the previously existing version, which is considered the original process, and the new modified version that will encompass all changes made to the original process. Thus, every original process has one author and possibly one or more modifiers.

Since multiple versions of the same process model may exist in the game, players can vote for the model they think is the most accurate, complete and detailed.

The term “process tree” is used in the game to refer to a set of processes, structured as a tree with 1 level, in which the root node of the tree consists of the original created process, and the child nodes the created process versions based on the original process.
5.2.4 Process Duplication

As aforementioned, it is required that the final set of business processes elicited in the game does not contain duplicate models. Hence, players are able to mark one process as duplicate of another, in order to exclude that model from the final set, assuming that the majority of players agree that the model in question is indeed a duplicate, by voting consensually.

The action of excluding a process from the final set begins with one player marking a process as duplicate, proposing that a previously created process that is equivalent to the one marked already exists. Whenever a player marks a process as duplicate, it is mandatory that the original process is specified. From the moment that a process gets a duplication mark, other players can perceive that the process is under evaluation for duplicate content.

At the end of the game, the platform calculates the result of the voting process, and if the number of positive votes overcome the number of negative ones, then the process is considered to be a duplicate, and therefore is excluded from the final process set.

5.2.5 Process Voting

Players are able to rate processes regarding the matter of evaluating the quality of process information and deciding if the model represents realistically the process in question, or determining if one process is a duplicate of another that was already created. The former is referred as quality voting and the latter as duplication voting.

Concerning quality voting, players can place a positive vote if they agree that the business process in question is accurately, comprehensively and detailedly represented in the game. In case the vote is negative, it means that the player agrees that the process is not complete, not accurate, or not well represented. In regard to duplication voting, players can place a positive vote if they agree that the process is a duplicate of an already existing one, or a negative vote if they believe that the original process is not equivalent to the one marked.

As the game finishes, for each process it is determined the quality and duplication (in case it was marked as a duplicate) vote rates. If the number of positive votes is at least 51% of the total number of votes, then the process is considered to be correctly represented (the greater the percentage of positive votes, the better the quality), or is permanently marked as a duplicate, respectively.

For the voting mechanism to be fair and to avoid cheats or tendencies that lead players to vote without exclusive consideration for the process content, players do not have information about the process author, and can only perceive that a process is under voting, not being able to acknowledge how many votes in favour and against there exist, until they vote. This prevents players from voting for the part that has more votes, hoping to win extra score points disregarding the process itself.

5.2.6 Achievements

Achievements consist of milestones that acknowledge the progress made by a player in the game, and are summarised in the following table:
<table>
<thead>
<tr>
<th>Achievement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Process Published</td>
<td>The player publishes his first process in the game, whether or not other players have already published processes.</td>
</tr>
<tr>
<td>Correct Own Process</td>
<td>The player proposes a modification to an original process published by himself.</td>
</tr>
<tr>
<td>Modify Other Process</td>
<td>The player proposes a modification to an original process published by another player.</td>
</tr>
<tr>
<td>Vote Processes All Players</td>
<td>The player places a vote in at least one of the published processes of all other players.</td>
</tr>
</tbody>
</table>

Table 5.1: Description of the existing achievements in the game.

5.2.7 Medals

Medals mark exceptional behaviour attained by players, and are summarised in the following table:

<table>
<thead>
<tr>
<th>Medal</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Process Published</td>
<td>The player publishes the first ever published process in the game.</td>
</tr>
<tr>
<td>Duplication Detection</td>
<td>The player marks a process as a duplicate and the vote rate indicates that other players agree.</td>
</tr>
<tr>
<td>Best Quality Content</td>
<td>The player publishes the process with best quality vote rate in the game.</td>
</tr>
<tr>
<td>Best Convergence Rate</td>
<td>The votes placed by the player meet the most the voting decisions made by other players.</td>
</tr>
</tbody>
</table>

Table 5.2: Description of the existing medals in the game.

5.3 Game Mechanics

The main goal of the player is to build a representation of the business processes of the organisation he or she belongs using the process language provided by the game, described detailedly in section 4.4 Business Process Graphic Language.

5.3.1 Player Actions

The following actions can be performed in the game by the role Player depicted in the use case diagram in Figure 4.1.

- Create new process: A player is able to create a new process model and enrich it with business knowledge regarding the process intended to be represented.
- Remove process: A player who happens to be the author of a draft or published process, is able to delete it permanently from the game.
• **Vote for quality**: A player is able to rate positively or negatively the quality of a process. Additionally, if a process is marked as duplicate, it is also possible to back up the duplicate detection, or vote against it.

• **Mark process as duplicate**: A player is able to mark a process as a duplicated version of another already existing process. To accomplish so, the player has first to select which process or version is considered the original process.

• **Vote for process duplication**: Once a process is marked as duplicate, other players are able to vote for process duplication, meaning that the player agrees that there already exists a previously created process that is equivalent to the one voted, or against process duplication, otherwise.

• **Create versions of processes**: A player is able to create and publish alternative versions of an already existing process.

5.3.2 Administrator Actions

The following actions can be performed in the game by the role Administrator depicted in the use case diagram in Figure 4.1.

• **Set game configuration**: An administrator is able to set a context configuration for games that are about to start. Thus, the game length, number of players, name of the process to be elicited and expected number of activities per process can be defined, and when players enter the game and start playing, these properties are communicated to each participant and their game context is adjusted.

• **Export state as XML**: An administrator is able to export the game data of the currently being played game into an XML format, which can then be loaded by the platform afterwards.

• **Load state from XML**: An administrator is able to load game data of a previously played game, by inserting the XML string given as output of the action export state.

• **End game before time**: It is possible for an administrator to terminate a game before elapsing the defined time length in the game context.

• **Visualize game data**: Administrators are able to visualize all data regarding players and processes (e.g. scores, statistics, process content, etc).

5.3.3 Strategic Objectives

Before specifying the game rules that regulate players’ behaviour, it is important to analyse the strategic objectives that serve as foundation for their conception. Every game rule formulated enforces at least one strategic objective. This approach ensures that every strategic objective is backed up by a set of game rules, and also that every established game rule supports the objectives of the game.
The most valuable strategic objectives identified for the scope of this dissertation are listed in the following table:

<table>
<thead>
<tr>
<th>#</th>
<th>Strategic Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>Support the creation of business processes among players.</td>
</tr>
<tr>
<td>O2</td>
<td>Embolden players to create processes as accurate as in real life.</td>
</tr>
<tr>
<td>O3</td>
<td>Enforce collaboration between players regarding process creation.</td>
</tr>
<tr>
<td>O4</td>
<td>Encourage the reviewing and voting of other players’ processes.</td>
</tr>
<tr>
<td>O5</td>
<td>Ensure that duplicated processes are detected.</td>
</tr>
<tr>
<td>O6</td>
<td>Guarantee anonymity of added game content.</td>
</tr>
<tr>
<td>O7</td>
<td>Avoid cheats or loopholes that benefit players undeservedly.</td>
</tr>
</tbody>
</table>

Table 5.3: Strategic objectives that support the foundation of game mechanics.

### 5.3.4 Game Rules

The conception of game rules directs its focus to valorise and devalue process scores, according to actions carried out by players, whose outcomes are considered to enforce or renounce the strategic objectives, respectively. The value of each process is then assigned to the player that created it. Operations that are not related to process creation follow similar mechanisms.

The following set of game rules is grouped into three macro tasks: process creation, process voting and duplicate marking. At the end of the section, an alignment between the identified strategic objectives and the game rules that support them is made in Table 5.4.

**Process/Version Creation**

- **GR1**: Players can only create processes under the process language rules defined in Section 4.4.
- **GR2**: Process drafts are only visible to the author, switching to public once it is published.
- **GR3**: Players, except the process author, can only perform actions upon a certain process if it was already published.
- **GR4**: Until the end of the game, players cannot see who the author of processes is.
- **GR5**: Authors of versions with the best rate of positive quality votes in process tree are rewarded.
- **GR6**: Players can only have access to processes published by other players once they have published at least one process of their own.

**Voting for Quality/Duplication**

- **GR7**: Players can only see the number of quality/duplication votes a process received once they voted.
- **GR8**: Players who voted for the convergent decision are rewarded.
- **GR9**: Players who voted against the convergent decision are penalized.
• GR10: Players who abstain from voting are penalized.
• GR11: Players cannot see the player who placed a vote.
• GR12: Players cannot vote for quality a process in which he is the author.
• GR13: Each player can place at most one quality vote and one duplication vote for each process.

Marking as Duplicate

• GR14: Players cannot mark process A as a duplication of process B if process A was created first than process B.
• GR15: The player who detects duplicate processes is rewarded.
• GR16: Players cannot see the player who marked a process as a duplicate of another.
• GR17: Players cannot mark a process as a duplication of itself.
• GR18: If it is consensually decided that process A is a duplicate of process B, process A can no longer be marked as a duplicate again.

<table>
<thead>
<tr>
<th></th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O7</td>
<td></td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Alignment between strategic objectives and game rules.

5.4 Scoring System

This section endeavours to explain the reasoning behind the conception of the scoring system chosen for the gaming platform. Furthermore, a statement is presented about players behaviour that are to be compensated and penalised according to the strategic objectives introduced in the previous chapter. Moreover, a definition of the formula that leads to each player’s final score is made. Lastly, test cases are delineated, to illustrate and demonstrate how scoring aspects work regarding rewards and penalties.

5.4.1 Bonifications and Penalisations

Regarding process creation and publication, the process author is rewarded for each non-empty created activity (activities with no name assigned accommodate no business knowledge; therefore, no bonuses should be involved). To avoid processes with a great number of unnecessary activities and players who create processes with an excessive number of activities exclusively to achieve more score points,
A mechanism was implemented that compensates players who attain processes with the ideal number of activities and penalises players who create processes whose number of activities diverges from the expected number of activities.

The expected (or ideal) number of activities depends on the process itself, and for this reason, this value can be defined by administrators at the beginning of each game session. The implementation of this mechanism follows a modified Poisson Distribution $\text{Poi} \sim (k, E) = \frac{e^k e^{-k}}{k!}$, $e = 2.71828$, whose value $E$ is set in the game configuration, and $k$ the number of activities of the processes created by each player.

The modified distribution can be obtained as $\text{Poi}' \sim (k, E) = 2\\text{Poi} \sim (k, E) - \text{Poi} \sim (E, E)$, in order to assign a positive percentage of the process score to the author if the number of activities converges to the expected value, or a negative percentage of the process score, otherwise.

### Bonuses
- Publishing new process/version;
- Convergent number of activities per process/version created;
- Creating process/version with positive vote rate;
- Creating process/version marked as duplicate but considered not a duplicate;
- Creating process/version with best vote rate in process tree;
- Voting for process/version;
- Marking a process/version as a duplicate, convergent from players’ voting;
- Earning achievements;
- Earning medals;
- Creating process/version that is ultimately included in the Final Consensus.

### Penalties
- Divergent number of activities per process/version created;
- Creating process/version with negative vote rate;
- Creating process/version marked as duplicate and considered a duplicate by other players;
- Creating process/version with inferior vote rate in process tree;
- Abstention from voting;
- Marking a process/version as a duplicate, divergent from players’ voting.

<table>
<thead>
<tr>
<th>Bonuses</th>
<th>Penalties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publishing new process/version;</td>
<td>Divergent number of activities per process/version created;</td>
</tr>
<tr>
<td>Convergent number of activities per process/version created;</td>
<td>Creating process/version with negative vote rate;</td>
</tr>
<tr>
<td>Creating process/version with positive vote rate;</td>
<td>Creating process/version marked as duplicate and considered a duplicate by other players;</td>
</tr>
<tr>
<td>Creating process/version marked as duplicate but considered not a duplicate;</td>
<td>Creating process/version with inferior vote rate in process tree;</td>
</tr>
<tr>
<td>Creating process/version with best vote rate in process tree;</td>
<td>Abstention from voting;</td>
</tr>
<tr>
<td>Voting for process/version;</td>
<td>Marking a process/version as a duplicate, divergent from players’ voting.</td>
</tr>
<tr>
<td>Marking a process/version as a duplicate, convergent from players’ voting;</td>
<td></td>
</tr>
<tr>
<td>Earning achievements;</td>
<td></td>
</tr>
<tr>
<td>Earning medals;</td>
<td></td>
</tr>
<tr>
<td>Creating process/version that is ultimately included in the Final Consensus.</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Summary of player’s actions that lead to bonuses or penalties in the game.

Respecting the vote rate of quality voting, the author of processes with positive vote rates is rewarded, as opposed to processes with negative vote rates, leading to penalties. The process with the best vote rate in the process tree is rewarded, additionally. This mechanism not only motivates players to build process models with the best quality possible (so process versions created posteriorly earn no more positive votes than the original one), but also to motivate players to create better versions of original processes, therefore obtaining a higher quality vote rate, being compensated for it.
With respect to process duplication, whenever a process is marked as a duplicate, if players deter-
mine consensually that the process is indeed a duplicate, then the process author is penalised and the
player who placed the mark is compensated. Otherwise, the process author is compensated, and the
player who placed the mark is penalised.

Compensations related to quality voting and duplication voting were already stated in the previous
sections, and are further specified later in this section, as well as achievements, medals and creation of
processes that belong to the final consensus.

Table 5.5 summarises the described bonuses and penalties for the actions performed in the game.

5.4.2 Bonus-Malus System

Given the previous identification of compensations and penalisations directed at target situations in the
game and the elaborated game rules that balance the valorisation of game objects created by players,
a mapping between the scoring factors that leverage the bonus-malus system and the result of these
actions is outlined in this section.

Rewards and penalties can be segregated into different scoring aspects. Each scoring aspect may
encompass multiple scoring factors, which are more specific bonus-malus rules that compensate or
penalise players (e.g. the expected number of activities is a scoring factor encompassed in the process
authorship scoring aspect). The defined scoring aspects are described subsequently:

- **Process and Version Authorship**: Takes into consideration everything related to publishing pro-
cesses and process versions, namely: score of created activities, expected number of activities,
quality votes received, duplicate marks and votes received, and best vote rate in process tree.

- **Voting for Quality and Duplication**: Examines votes placed by players and the conformity or
nonconformity of these votes with the convergent decision. Additionally, players who abstain from
voting are pinpointed.

- **Duplicate Detection**: Players who marked processes as duplicates are scrutinised according to
what other players voted in the duplication voting.

- **Achievements and Medals**: Regarding the defined achievements and medals defined in sections
5.2.6 Achievements and 5.2.7 Medals, these game objects are attributed to players who performed
accordingly in the game.

- **Final Consensus**: Authors of processes elected to be included in the final consensus set are
identified so they can be compensated.

The table below aims at defining the scoring aspects that compose the final score of a player. Each
scoring aspect encompasses a group of scoring factors, described formerly, that can be assigned to
tasks or macro tasks that can be performed in the game. To each scoring factor is assigned a condition
that determines which formula or mathematical model is to be applied to calculate the new score of the
player(s) in question.
<table>
<thead>
<tr>
<th>Scoring Aspect</th>
<th>Condition</th>
<th>Bonus-Malus (points)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Process Authorship</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Score (1)</td>
<td>Added Activity</td>
<td>+ 25</td>
</tr>
<tr>
<td></td>
<td>Added Composed Activity</td>
<td>+ 50</td>
</tr>
<tr>
<td></td>
<td>Added Ad-Hoc Activity</td>
<td>+ 50</td>
</tr>
<tr>
<td>Expected Num. Act. (2)</td>
<td>True</td>
<td>+ PS x (2 x Poi – (k, E) - Poi – (E, E))</td>
</tr>
<tr>
<td>Positive Vote Rate (3)</td>
<td>PPV &gt; 0.5</td>
<td>+ PS x (PPV/4)</td>
</tr>
<tr>
<td></td>
<td>PPV &lt; 0.5</td>
<td>- PS x (PNV/4)</td>
</tr>
<tr>
<td>Marked as Duplicate (4)</td>
<td>Dup.PPV &gt; 0.5</td>
<td>- PS x 25% (out of FC)</td>
</tr>
<tr>
<td></td>
<td>Dup.PPV &lt; 0.5</td>
<td>+ PS x 10%</td>
</tr>
<tr>
<td>Best Vote Rate in</td>
<td>True</td>
<td>+ PS x 10%</td>
</tr>
<tr>
<td>Process Tree (5)</td>
<td>False</td>
<td>- PS x 5%</td>
</tr>
<tr>
<td><strong>Version Authorship</strong></td>
<td>Version Score</td>
<td>Same as in (1)</td>
</tr>
<tr>
<td></td>
<td>Expected Num. Activ.</td>
<td>Same as in (2)</td>
</tr>
<tr>
<td></td>
<td>Positive Vote Rate</td>
<td>Same as in (3)</td>
</tr>
<tr>
<td></td>
<td>Marked as Duplicate</td>
<td>Same as in (4)</td>
</tr>
<tr>
<td></td>
<td>Best Vote Rate in Tree</td>
<td>Same as in (5)</td>
</tr>
<tr>
<td><strong>Voting (Quality/Duplication)</strong></td>
<td>Participation</td>
<td>Convergent Vote</td>
</tr>
<tr>
<td></td>
<td>Non-Convergent Vote</td>
<td>- 5</td>
</tr>
<tr>
<td></td>
<td>Tie</td>
<td>+ 10</td>
</tr>
<tr>
<td>Abstention</td>
<td>Quality Voting</td>
<td>- 10</td>
</tr>
<tr>
<td></td>
<td>Duplication Voting</td>
<td>- 25</td>
</tr>
<tr>
<td><strong>Duplicate Detection</strong></td>
<td>(Marking as Duplicate)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dup.PPV &gt; 0.5</td>
<td>+ PS x 10%</td>
</tr>
<tr>
<td></td>
<td>Dup.PPV &lt; 0.5</td>
<td>- PS x 10%</td>
</tr>
<tr>
<td></td>
<td>Dup.PPV = 0.5</td>
<td>+ PS x 5%</td>
</tr>
<tr>
<td><strong>Achievements</strong></td>
<td>First Process Published</td>
<td>+ 75</td>
</tr>
<tr>
<td></td>
<td>Vote Processes All Players</td>
<td>+ 50</td>
</tr>
<tr>
<td></td>
<td>Modify Other Process</td>
<td>+ 25</td>
</tr>
<tr>
<td></td>
<td>Correct Own Process</td>
<td>+ 10</td>
</tr>
<tr>
<td><strong>Medals</strong></td>
<td>Best Convergence Rate</td>
<td>+ 150</td>
</tr>
<tr>
<td></td>
<td>First Process Published</td>
<td>+ 100</td>
</tr>
<tr>
<td></td>
<td>Best Quality Content</td>
<td>+ 75</td>
</tr>
<tr>
<td></td>
<td>Duplication Detection</td>
<td>+ 50</td>
</tr>
<tr>
<td><strong>Final Consensus</strong></td>
<td>Belongs to the set</td>
<td>+ PS x 25%</td>
</tr>
</tbody>
</table>

Table 5.6: Blueprint of the scoring system.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS</td>
<td>Process Score.</td>
</tr>
<tr>
<td>PPV</td>
<td>Percentage of Positive Votes (quality voting).</td>
</tr>
<tr>
<td>PNV</td>
<td>Percentage of Negative Votes (quality voting).</td>
</tr>
<tr>
<td>FC</td>
<td>Final Consensus.</td>
</tr>
<tr>
<td>Dup.PPV</td>
<td>Percentage of Positive Votes (duplication voting).</td>
</tr>
<tr>
<td>E</td>
<td>Expected number of activities per process.</td>
</tr>
<tr>
<td>Poi (E)</td>
<td>Probability of Poisson Distribution with expected value $E$.</td>
</tr>
</tbody>
</table>

Table 5.7: Description of abbreviations used in the scoring system table.

### 5.4.3 Final Consensus

As described previously, the final consensus consists of a set of processes, created in the game, that are elected to be the final representation of the elicited process. The algorithm behind the selection of processes is based on the game rules defined in earlier sections.

The algorithm starts with a process set, which gathers all processes and process versions published in the game. The first step consists of removing from this set the duplicates, by excluding processes marked as duplicate that have more positive duplication votes than negative ones. The output of this step is a set of non duplicate processes.

The second step orders the previous process set by descending order of process score. This means that non duplicate processes with higher scores occupy the first places.

The third and last step retrieves from the ordered set the top-1 processes with higher score. The result can either be only one process or more than one process, in case there is a tie between process scores of two or more processes. The processes contained in this subset are selected to be the final consensus of the game.

![Figure 5.2: Algorithm that retrieves the final processes contained in the Final Consensus.](image)

The second step orders the previous process set by descending order of process score. This means that non duplicate processes with higher scores occupy the first places.

The third and last step retrieves from the ordered set the top-1 processes with higher score. The result can either be only one process or more than one process, in case there is a tie between process scores of two or more processes. The processes contained in this subset are selected to be the final consensus of the game.

### 5.4.4 Player's Final Score

The final score of a player can be calculated as the combination of the applicable scoring factors, and can be obtained by applying the game data that concerns the target player into the following equation:
\[
\text{FinalScore}(\text{player}) = \sum_{i=1}^{I} \text{ProcessAuthorship}(\text{createdProcess}_i)
\]

\[
+ \sum_{j=1}^{J} \text{VersionAuthorship}(\text{createdVersion}_j)
\]

\[
+ \sum_{k=1}^{K} \text{VotingReward}(\text{VotePlaced}_k) - \text{VoteAbstention}()
\]

\[
+ \sum_{l=1}^{L} \text{DuplicateDetection}(\text{mark}_l)
\]

\[
+ \sum_{m=1}^{M} \text{FinalConsensus}(\text{createdProcess}_m) + \text{Achievements}() + \text{Medals}()
\]

Once the game ends and the final score of each player is calculated and assigned, the server orders players by descending order of player score, so it can be known which players performed best in the game.

### 5.5 User Interface

This section is intended to demonstrate how players are able to interact with the game, and how they can navigate the graphic interface to perform possible and allowable actions. Furthermore, it is explained the way information is organised, and how it can be manipulated. At last, a specification of keyboard shortcuts is delineated, which can be used to accomplish tasks with less effort.

#### 5.5.1 Graphic Interface

In the screen pictured in Figure 5.3 players can carry out process creation. The menu at the top of the screen is in charge of some of the operations that can be performed regarding the manipulation of processes (process removal, process publication and insertion of primitives). At the time of publication, the platform performs a validation of the process language grammar, and in case incongruities are found, warning messages are displayed as notifications in the bottom left corner of the screen, to indicate the author that he must correct them in order to proceed with the publication of the process.

Figure 5.4 represents the home screen of the application, in which can be seen the same operations menu, available with the option of logging out of the game and examining the game data and statistics regarding the logged player. Processes are segregated in two groups: draft and published processes. Notifications regarding earned achievements and medals can be seen in the bottom left corner.

When it comes to creating versions of a currently existing process, once the original process is published, the version creation panel appears (and can be additionally hidden) at the bottom of the screen. This panel displays information and access to proposed versions of a given process (Figure 5.5).
Figure 5.3: Screen responsible for the creation and manipulation of processes.

The screen in Figure 5.6 illustrates the operation of marking a process as a duplicate. Once the “Mark as Duplicate” button is hit, a panel appears, in which it is listed all the processes and versions added to the game (except the process being marked). The player is to select the process he considers is the original process of the one he is currently marking as a duplicate. In case the selected process was created after the marked process, a warning message is presented.

Figure 5.4: Main screen where all processes are displayed.
As aforementioned, the screen in Figure 5.6 illustrates the operation of marking a process as a duplicate. Once the “Mark as Duplicate” button is hit, a panel appears, in which it is listed all the processes and versions added to the game (except the process being marked). The player is to select the process he considers is the original process of the one he is currently marking as a duplicate. In case the selected process was created after the marked process, a warning message is presented.
Figure 5.7: Screen responsible for the display of final scores.

At the end of the game, players can analyse how well they performed in the game. In Figure 5.7 it is represented the final scores’ screen, in which it is displayed the performance of players through different gaming aspects, such as process authorship, process voting, duplicates detection and earned achievements and medals.

Figure 5.8: Screen responsible for the definition of game context.

The screen depicted in Figure 5.8 is reserved for administrators, considering it belongs to the server of the application. Displayed is the panel used to define the game context for each game session, in
which administrators can set the game length, maximum number of players, name of the process to be elicited and the expected number of activities for each process.

5.5.2 Keyboard Shortcuts

Select primitive

A primitive can be selected to either reveal further informations about the primitive, or perform other operations, such as opening composite primitives or deleting primitives. One primitive can be selected by pressing the left mouse button over the primitive’s graphic representation.

Reposition primitive

The position of each language primitive added to the process pool can be changed using the computer mouse. To reposition an element, the player has to drag and drop the element using the right mouse button. If the element’s position is set to outside the process lanes, the position is considered invalid and the element is not repositioned to the new location.

Edit activity name

The name of an activity can be edited by keeping pressed the CONTROL button and clicking with the left mouse button on the primitive. A text field appears, allowing the player to change the name and save it, by hitting the ENTER button.

Open composite activity

Ad-hoc activities and composed activities can be opened by first selecting the activity in question and pressing the SPACE button.

Figure 5.9: Summary of keyboard shortcuts.

Delete primitive
Primitives can be removed by pressing the **DELETE** button. In order to do that, first, players need to select the primitive they want to eliminate. Some primitives cannot be removed, such as start or end events, or lanes that contain the aforementioned events.

**Options menu**

The options menu can be triggered by pressing the **ESCAPE** button. This menu can be accessed to operate on the application’s state and control features. Players can save the instance of the currently being played game, load a previously saved instance of a game, export the processes created in the game into an XML document, or quit the game.

**Help info**

The help info panel can be accessed by pressing the **F1** button. This panel lists all the keyboard shortcuts that can be used within the game to facilitate the operations carried out by players (see figure 5.9).
Chapter 6

Game Implementation

This chapter endeavours to go over the most influential decisions regarding the implementation of the gaming platform. Firstly it is presented the train of thought that lead to the choice of development environment. Then, a blueprint of how the application is grouped into modules of functionality is traced. Posteriorly, the communication protocols between application instances is discussed. Lastly, restrictions imposed by the choice of technology are exposed and it is understood their impact in the objectives of this dissertation.

6.1 Development Environment

From the exposition of desired behaviour for the proposed gaming platform, laid out in Section 4.2, Game Requirements, it can be understood that some of the game engine’s features compared in Section 2.4 Game Technology matter most for the development of a solution than others.

The most relevant features are: 1) Networking, since it is involved cooperation between business participants playing simultaneously; 2) Fast Learning Focus, so the development of the project is not delayed due to a learning stage; 3) Scripting and IDE, to fasten the development process and support the debugging phase; 4) Multi Platform, so technological limitations do not lessen the potential of the application; and 5) GUI System, being a powerful tool for the implementation of user interfaces, due to the fact that the game is two-dimensional.

By doing an examination over the comparison of features between different game engines, established in Table 2.1, it can be quickly acknowledged that the game engine that best accommodates the desired features elicited above is Unity 3D, being the game engine chosen for the development of the application. Unity guarantees integration with an IDE named MonoDevelop, using Unity scripting languages C#, JavaScript or Boo.

Given the multiplicity of data types that are used to accommodate the game domain, the most appropriate programming language for development is strongly typed. Strong typing guarantees that variables are bound to a specific data type. Therefore, type checking happens at compile time, thus reducing errors earlier in the development cycle.
Since JavaScript is a weakly typed language, the choice of programming languages narrows down to C# and Boo. Due to a wider online support of C# and previous experience with C-based object-oriented languages, it was deliberated that the platform would be developed in the programming language C#, using the MonoDevelop environment, which incorporates a debugger that is directly linked to Unity running instances.

6.2 Game Architecture

This section endeavours to describe detailedly the architecture of the gaming platform. The modular architecture diagram displayed below depicts the structure of the gaming platform, segregating the application architecture into different modules according to the type of functionality.

![Modular architecture diagram of the gaming platform.](image)

The following sections aim at characterising each of the architecture’s modules, specifying and giving a brief description of the implemented functionality. First, the analysis focus on modules common to both client and server sides. Then, a description of the modules delegated specifically to the client side is performed, followed by a description of the server modules.

6.2.1 Common Modules

Both User Interface modules provide a visual interface for player-game interaction. This high level module is directly linked to the computer’s input interface, the keyboard and mouse, used to manipulate game elements and enter information regarding processes.

Networking modules embody all the communication made between player instances and server. Since network functionality is natively implemented in Unity, there is no need to code components that create and manage connections, or even deal with network faults, since these features are similarly supported by the chosen game engine. Although technically, network functionality is inserted in the
Game Engine module, these two modules are distinguished to acquaint the nature of communication between different application instances (transversal to most of other modules).

**Data Models** are all grouped in one single module, which is used by both client and server sides, indiscriminately. It is important that both sides use the same data models so there is no constraints when sending data instances across networks (via RPC). This decision also allows for similar entities to be handled equally in the client and server sides, preserving the synchrony of game state across application instances while the game is being played.

The In-Memory Database modules are responsible for storing all the information regarding the game instance being played, whether that be business knowledge contributed by players about processes, information regarding process entities, defined game settings, player’s scores and accomplishments, game managing attributes, among others.

Despite the Game Engine module is not part of the solution proposed in Chapter 4 Proposed Solution, and therefore is obviously not implemented in the scope of this dissertation project, it constitutes a crucial layer in the platform implementation, influencing the rest of the application architecture.

### 6.2.2 Client Side Modules

The Game Manager module coordinates the components that control the way the game unfolds. The module keeps track of the value of game elements that are crucial to guide players visually and informationally through different phases of the game (*e.g.* game screen management, notification system, primitives under inspection, etc).

In the Game Logic module it is encompassed functionality related to the possible interactions between players and the game. In case players make requests that need database update, this module forwards these requests to the network. Furthermore, the responsibilities of this module extend to the governance of access privileges of game information (*e.g.* determine which players have access to which segments of game data) and moderation of possible actions (*e.g.* forbid edition of read-only information), among others.

### 6.2.3 Server Side Modules

The Server module is in charge of receiving the RPC requests from players, pipeline the requests that need further verification to other modules of functionality, and dispatch success or unsuccess acknowledgements to the player, informing the result of the request.

Language Constructor is the module responsible for the construction of business processes, encompassing every possible action players can execute using the process language described in Section 4.4 Business Process Graphic Language. This includes the creation, edition and removal of process primitives (lanes, activities, events or flows), to assure that every attempted move is valid under the grammar and rules of the process language. The verification of players’ moves authenticates not only the validity of action (*e.g.* it is never possible to create loop cycles in events) but also taking into consideration the context in which it occurs (*e.g.* it is possible to create composed activities, but not within
The **Game Mechanics** module incorporates functionality related to the implementation of mechanics that regulate players’ behaviour, specified minutely in Section 5.3 *Game Mechanics*. Examples of actions regulated by this module include voting for quality, marking process as duplicated, proposing modifications to other players’ processes, etc.

Every time server instances are shut down, the **Physical Database** module assures that all information produced since the server was started is saved. Due to security reasons over the internet, Unity Web Player does not allow to work with physical files. As a result, the data persistency of the server state is supported by Unity’s *PlayerPrefs*, a class that allows the storage and access of information between game sessions. The restoration of game information from a previous session is implemented similarly, except when an administrator imports XML content.

### 6.3 Data Persistency

To ensure that players can play the same game through multiple sessions, and that all business information created in the game is saved to be posteriorly used, data persistency mechanisms had to be implemented to accomplish these requirements. Due to technological limitations further explained in section 6.5 *Technological Limitations*, it is not possible to save the game state into files; so, alternatives were explored.

In order to avoid data loss from application bugs or crashes, whenever the application quits, the current game state is saved into Unity’s *PlayerPrefs*, which are player preferences and information that are maintained throughout game sessions, using object serialisation. Additionally, when the server application starts, the previous game state stored in *PlayerPrefs* is loaded and restored. These operations are hidden from the player side, since the game state is only sent through the network as soon as the players logs in successfully.

![Technological implementation of data persistency](image)

**Figure 6.2:** Technological implementation of data persistency.

So that business information elicited in the game is used for posterior decision making, it is needed that the exportation of this information is made following a structured format. For that reason, the exportation of the game state is done using the XML format and this operation can be performed anytime in the game. Additionally, state load can also be performed by pasting in the application the XML text containing the game state to be loaded.
6.4 Multiplayer Game

The communication between clients and server is performed using RPC calls, following a client-server model. Most cases consist of the client sending a request to the server and receiving an acknowledge about the success or failure of that request. The used approach implements an authoritative server, meaning that the network node that executes as a server will process and store the game data, apply game rules and determine the responses to RPC requests from players. Since cooperation and communication between players involve the action of an intermediate node in the network (the server that regulates and validates actions within the game), players are not able to communicate between each other directly. As a consequence, end users can experience privacy regarding information entered into the game, as other players do not possess knowledge about the ownership of the game information entered.

In Figure 6.3 it is depicted the most common client-server interaction, implemented in the addition or edition of processes or primitives in the game. This type of interaction starts with the request from a player to change the state of the game, containing information regarding the action to be performed, such as primitive’s identifiers and attributes. Once the server receives the call, the request is evaluated, and a response to be posteriorly send to the player is built. Lastly, when the client receives the response, the action requested is performed in case the response indicates success, or else the game state remains unchanged in case the response signals failure.

In some cases, the requests made by clients involve the change in game information that other players do not have access at the moment (e.g. when a player is building a process that is not yet finished). So, the response will be only sent to the player that made the request (Figure 6.3 (a)). In other cases, such as the change in a process that can be accessed by everyone in the game, the response is sent to all players, containing an update to the game state (Figure 6.3 (b)).

![Figure 6.3: Client-server interactions upon client request.](image)

As an example, a certain player sends a request to the server to add a new activity to a certain lane belonging to a process’ pool. The server will receive this request, analyse its game state, decide on whether the activity can be added in the particular context or not, and send back an RPC request acknowledge to the client, informing if the activity was successfully added, or failed to be performed and...
In figure 6.4, it is represented client-server interactions that do not start with a request from the client, such as asynchronous events.

(a) Single player update
(b) Global update

Figure 6.4: Client-server interactions without previous client request.

As an example, once the server detects that one player made a certain achievement in the game that deserves an award, it sends an RPC call to that player to inform him about the bonus. The RPC call will contain information about the nature of the award and the increase in score that the player earned, among others.

It is widely known that a game with networked multiplayer features may arise some problems and very particular scenarios, due mostly to: a) RPC calls with exchanged order of arrival to the receiver; and b) lost RPC calls that the receiver never gets to acknowledge. The existing online documentation of the game engine Unity about RPC guarantees that RPC calls are always to be executed in the same order as they are sent. Hence, system requirements that prevent these scenarios are not part of the proposed solution, since such implementations already exist and rely on the networking system of the chosen game engine.

6.5 Technological Limitations

Despite the extensiveness of the GUI elements supplied by the GUI system of the chosen game engine, the implementation of lines and arrows is not native. For that reason, in order to implement the traditional arrows, the collision of GUI elements must be calculated for every frame in every flow between two elements, which implies heavy processing that could be avoided by seeking a different approach. Therefore, the choice of putting the arrow in the middle of the flow instead of at the head of the line was preferred to avoid unnecessary processing work at execution, without compromising the nature of the representation.

Since the main focus of this dissertation is not to develop a game using highly refined graphic techniques, but to develop a gaming platform to aid in the elicitation of business processes inside an organ-
isation, the aforementioned graphic choices for the process language can be seen as the relocation of challenges that are out of the scope of this project.

Regarding data persistency, a few alternatives to writing the game information in the filesystem needed to be explored. Unity's webplayer security sandbox imposes some restrictions concerning the access of data on a domain other than the one defined in the .unity3d file. Thus, applications running in the webplayer are not allowed to write to physical files. Alternatively, it is possible to write and read from a Unity preferences file, whose content is maintained along multiple game sessions. Consequently, the technological decision made was to serialise the game state of a game session and write it as a string in PlayerPrefs when the application quits, and read and deserialise the game state posteriorly, once the application starts again. Since this decision has no negative impact in the functionality of the game, this limitation can be considered overcome.
Chapter 7

Evaluation

With the purpose of assessing the overall usefulness and acceptance of a gaming platform intended to collaboratively elicit business processes, an evaluation process has been delineated. First, the defined requirements for the gaming platform of this work are revisited, and for each, an acknowledgement of the degree of fulfilment is made. Thereafter, the designed test cases directed at testing the balance and coherence of the scoring system are explained. Posteriorly, a summary of the proceedings adopted in the playtesting sessions is attained, along with a description of the testing environment and involved testers. Afterwards, a discussion of the results obtained from the aforementioned methods is accomplished. Lastly, remarks are drawn from the gathered information while evaluating the application through various lenses.

7.1 Fulfillment of Requirements

From the set of requirements defined initially in the game design, it is performed a qualitative analysis of their fulfilment or unfulfilment in Table 7.1.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Fulfilment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>R1. Contribute with business knowledge</strong></td>
<td>It is possible for players to build representations of the business processes they are involved, contributing with business knowledge within the local view they have of the processes as business participants.</td>
</tr>
<tr>
<td><strong>R2. Graphic representation of processes</strong></td>
<td>As the platform provides a business process graphic language, players are able to represent business processes graphically. Hence, players are challenged to transform and structure their tacit knowledge of processes into explicit representations.</td>
</tr>
<tr>
<td><strong>R3. Propose modifications to other player’s processes</strong></td>
<td>Once processes are published, it is possible for a player, other than the author, to propose an alternative modified version of the process in question, posteriorly existing the original process and the proposed version.</td>
</tr>
<tr>
<td><strong>R4. Rate other player’s contributions</strong></td>
<td>It is possible for players to vote for other players’ process models in two ways. Voting for quality states the opinion of the voter regarding the accurateness of the model. Voting for duplication states if a player thinks a process model is a duplicate of another or not.</td>
</tr>
</tbody>
</table>
R5. Identify a process model as a duplicate of other

It is possible for players to mark a process as a duplicate of another already existing one. Once a process is marked as a duplicate, other players are able to vote for duplication, stating that either they agree that the process is a duplicate or not.

R6. Game settings

Administrators are able to define parameterisations that impact the game sessions to be posteriorly played, and the way players are going to be scored.

R7. Export business knowledge

Whenever the platform is running, administrators are able to export the business knowledge contributed by players in the game, in XML format.

R8. Save and load game sessions

The platform provides data persistency mechanisms that allow for game data to be saved, and loaded posteriorly. Consequently, game sessions can be paused, to be resumed and played again later.

R9. Visualisation of game results

At the end of the game, both players and administrators are able to visualise elicited business process models, the models that belong to the final consensus set, and the scores of players.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5 Pos. Votes 8 Neg. Votes</td>
<td>Yes 7 Pos. Votes 3 Neg. Votes</td>
<td>No 7</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>4 Pos. Votes 1 Neg. Votes</td>
<td>Yes 5 Pos. Votes 11 Neg. Votes</td>
<td>Yes 11</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>3 Pos. Votes 3 Neg. Votes</td>
<td>No N/A N/A</td>
<td>No 16</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1 Pos. Votes 7 Neg. Votes</td>
<td>No N/A N/A</td>
<td>Yes 11</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.2: Process data for Test Case 1.

As it can be observed in the table, all the functional requirements defined in Section 4.2 Game Requirements are fully implemented. The non-functional requirements established in the same section are implemented likewise. As a result, it can be stated that the implemented solution is valid in this context, and is a possible solution for the challenge presented by this dissertation.

7.2 Test Cases

Given that the designed scoring system is somewhat compound and quite numerous regarding the aspects focused when scoring players, test cases were delineated to evaluate the validity and balance of the system.

The following test cases are responsible for a practical demonstration of bonuses and penalties which result from players’ actions performed in the game. For reasons of simplicity, each test case focus in one individual scoring aspect, using different initial process data, to reflect the influence and impact of each factor in the scoring aspect.

Since final scores consist of summations of the value of each scoring aspect regarding the player being scored, it is redundant to build a test case regarding the final score of a player.

7.2.1 Test Case 1: Process Authorship

The player creates and publishes a process with 6 activities, 1 ad-hoc activity with 4 sub-activities, posteriorly achieving the following process data:
The rewards and penalisations regarding process authorship, with process data depicted in Table 7.2, are segregated by scoring factors in the following table:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>+ 300</td>
<td>- 17.6</td>
<td>- 46.2</td>
<td>- 75</td>
<td>- 15</td>
<td>+ 146.2 pts</td>
</tr>
<tr>
<td>B</td>
<td>+ 300</td>
<td>+ 35.8</td>
<td>+ 60</td>
<td>+ 30</td>
<td>+ 30</td>
<td>+ 455.8 pts</td>
</tr>
<tr>
<td>C</td>
<td>+ 300</td>
<td>+ 0</td>
<td>+ 0</td>
<td>N/A</td>
<td>- 15</td>
<td>+ 285 pts</td>
</tr>
<tr>
<td>D</td>
<td>+ 300</td>
<td>+ 35.8</td>
<td>- 65.6</td>
<td>N/A</td>
<td>+ 30</td>
<td>+ 300.2 pts</td>
</tr>
</tbody>
</table>

Table 7.3: Results of Test Case 1.

### 7.2.2 Test Case 2: Process Voting

The player places votes for quality and duplication in a certain process with the following process data:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Voting Type</th>
<th>Vote Placed</th>
<th>Quality Voting</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Any</td>
<td>Positive</td>
<td>Pos. Votes: 1 Neg. Votes: 5</td>
</tr>
<tr>
<td>B</td>
<td>Any</td>
<td>Negative</td>
<td>Pos. Votes: 3 Neg. Votes: 6</td>
</tr>
<tr>
<td>C</td>
<td>Quality</td>
<td>Abstention</td>
<td>Pos. Votes: 4 Neg. Votes: 3</td>
</tr>
<tr>
<td>D</td>
<td>Duplication</td>
<td>Abstention</td>
<td>Pos. Votes: 3 Neg. Votes: 5</td>
</tr>
</tbody>
</table>

Table 7.4: Process data for Test Case 2.

The rewards and penalisations regarding process voting, with process data depicted in Table 7.4, are segregated by scoring factors in the following table:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Participation</th>
<th>Abstention</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>+ 5</td>
<td>N/A</td>
<td>+ 5 pts</td>
</tr>
<tr>
<td>B</td>
<td>+ 20</td>
<td>N/A</td>
<td>+ 20 pts</td>
</tr>
<tr>
<td>C</td>
<td>N/A</td>
<td>- 10</td>
<td>- 10 pts</td>
</tr>
<tr>
<td>D</td>
<td>N/A</td>
<td>- 25</td>
<td>+ 300.2 pts</td>
</tr>
</tbody>
</table>

Table 7.5: Results of Test Case 2.

### 7.2.3 Test Case 3: Marking Process as Duplicate

The player marks a process, with process score of 300 points, as a duplicate with the following process data:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Duplication Voting</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Positive Votes: 6  Negative Votes: 2</td>
</tr>
<tr>
<td>B</td>
<td>Positive Votes: 3  Negative Votes: 5</td>
</tr>
<tr>
<td>C</td>
<td>Positive Votes: 2  Negative Votes: 3</td>
</tr>
</tbody>
</table>

Table 7.6: Process data for Test Case 3.

The rewards and penalisations regarding process duplication, with process data depicted in Table 7.6, are represented in the following table:
7.3 Playtesting

In order to conceptually and technologically assess the architecture of the implemented solution, playtesting sessions were performed with more than one tester playing the game simultaneously. For comparison purposes, all participants took part in two different case studies, so that collected data from the sessions were not compromised by contextual limitations. Descriptions and rationalisations of the case studies are delineated below, to understand process domains explored with testers. After, a summary of the proceedings undertook in the playtesting sessions is laid out.

7.3.1 Case Studies

- **Case Study 1: Triage of Patients**

To exemplify the applicability of the process notation, a case study that is publicly well-known to the general society is examined. The business process being modelled consists in the triage of patients in a hospital, in which it is described the course of a patient from the moment of arrival to the hospital until the moment the patient leaves the hospital. A textual description of the process is laid out below, derived from the simplification of an already existing description [35].

"Every Patient arriving at the ED (Emergency Department) must be registered by a Secretary, being requested to give some personal information. The Secretary writes the data in a registration form and provides an identifier to the Patient. After that, the Patient goes to a triage room where a Nurse triages them, assigns a priority level, and defines Patients routing, giving them a printed wristband. Then, Patients must wait for their turn, according to their urgency level.

Once called, Patients go to a treatment room where a Doctor handles their problem. To handle Patients’ problems, the Doctor may have to prescribe treatment. Finally, Patients are discharged with their problems solved, and may need to pay some fees, depending on the healthcare system or the owned insurance."

The most reasonable approach to adopt as a first step to model the business process is to identify the activities of the process. Next, an assignment of each activity to the corresponding business participant is made. Finally, it is analysed the relationships between primitives, so sequence and information flows are added to the process to sequence the order of execution of activities. The identification of activities is laid out in Table 7.8 and the representation depicted in Figure 7.1.
<table>
<thead>
<tr>
<th>Bus. Participant</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Patient</strong></td>
<td>Give personal information; Go to triage room; Wait for his/her turn; Go to treatment room; Pay fees.</td>
</tr>
<tr>
<td><strong>Secretary</strong></td>
<td>Write registration form.</td>
</tr>
<tr>
<td><strong>Nurse</strong></td>
<td>Triage patient; Assign priority level; Define patients routing.</td>
</tr>
<tr>
<td><strong>Doctor</strong></td>
<td>Handle patient’s problem; Prescribe treatment.</td>
</tr>
</tbody>
</table>

Table 7.8: Alignment between business participants and activities of process “Triage of Patients”.

![Diagram](image)

Figure 7.1: Representation of process “Triage of Patients”.

- **Case Study 2: Finalise Dissertation (UTL-IST)**

In order to obtain a specification of this process, an analysis over a document issued by the Governing Board of IST [36] was performed, along with a simplification of the corresponding representation built with BPMN [37].

“To begin, Students deliver their dissertations. The Coordinator is responsible for proposing a Jury, and waiting for the approval of the proposal. Once the proposal is approved, the Jury propose a date for discussion. After the Student defends the dissertation work, the Jury evaluate the Student’s work. In case it is decided that the Student is not to be approved, the course of action ends here. Otherwise, upon improvements suggested in the dissertation defense, the Student is to submit the final version of the dissertation. Then, the Orientator is to check all documents, so that finally the Coordinator is able to insert the classification in the system.”

The same approach is used as in case study 1; therefore, activities are identified using bold font. The assignment of activities to business participants is established in Table 7.9, and the representation
of the process using the designed process language is depicted in Figure 7.2.

<table>
<thead>
<tr>
<th>Bus. Participant</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Student</strong></td>
<td>Deliver dissertation; Defend dissertation work; Improve dissertation work; Submit final version of dissertation.</td>
</tr>
<tr>
<td><strong>Coordinator</strong></td>
<td>Propose jury; Insert classification.</td>
</tr>
<tr>
<td><strong>Jury</strong></td>
<td>Propose date for discussion; Evaluate student’s work.</td>
</tr>
<tr>
<td><strong>Orientator</strong></td>
<td>Check documents.</td>
</tr>
</tbody>
</table>

Table 7.9: Alignment between business participants and activities of process "Finalise Dissertation".

![Process Diagram](image)

Figure 7.2: Representation of process "Finalise Dissertation (UTL-IST)".

### 7.3.2 Testing Environment

When coordinating the playtesting sessions, the main intention was to assess the application under a relatively controlled environment. So, a total of 5 participants were gathered to test the gaming platform simultaneously. The testing sessions had the following agenda:

1. **Game Explanation** (10min): A brief explanation of the game is provided, along with the communication of the main objectives and the expected result at the end of the game.

2. **Free Game Session** (15min): Players are given the opportunity of playing a purposeless session, with the goal of getting to know the game and how to use tools provided to build process models.

3. **Case Study 1** (30min): A textual description of case study 1 is handed to players, for them to interpret the definition and collaboratively build the corresponding process model, applying their local view of the process.
4. **Case Study 2** (30min): The assignment practiced for case study 1 is similarly given to players, only that the process to be elicited is the one depicted in case study 2, explained previously.

5. **Questionnaire** (5min): A questionnaire is handed to participants, so that it is possible to gather individual information on each tester regarding personal traits that matter in the context of the sessions, and the perception of gameplay of the application.

**Estimated Time: 1 hour and 30 minutes**

To guarantee anonymity of information in the game, avoiding bias of behaviour in players when reviewing processes of other players, each player was confidentially given a username, in order to anonymously identify each player in process creation and associated game mechanics.

In order to better understand the data gathered from the testing sessions, and to more easily infer conclusions based on a rationalisation of the facts, the questionnaire demands information regarding demographics of participants, previous knowledge of business process modelling and serious games, and answers concerning the fulfilment of crucial functional requirements of the application. Hence, it can be understood if there is a correlation between that information and the good or bad performance in the game. In order to maximise the spectrum of answers on the aforementioned subjects, the questionnaire was designed using the **Likert-type scale**. The complete questionnaire is provided in Figure A.1 (Appendix), and in Table 7.10 and 7.11 is depicted gathered information concerning participants' gender and age:

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Percent (%)</th>
<th>Cumulative Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>Male</td>
<td>60.0</td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>40.0</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 7.10: Frequency table for the participant's gender.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Percent (%)</th>
<th>Cumulative Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>23</td>
<td>20.0</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>60.0</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>20.0</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 7.11: Frequency table for participants' age.

The following table characterises the distribution of participants concerning their level of education and professional area.

These data combined with gathered game data are going to be posteriorly analysed in order to determine if exists correlation between the experience of players and their performance in the game.
Table 7.12: Individual information collected through questionnaires regarding players.

<table>
<thead>
<tr>
<th>Player</th>
<th>Gender</th>
<th>Education Degree</th>
<th>Professional Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>F</td>
<td>Bachelor</td>
<td>Computer Science</td>
</tr>
<tr>
<td>California</td>
<td>M</td>
<td>Bachelor</td>
<td>Computer Science</td>
</tr>
<tr>
<td>Ohio</td>
<td>M</td>
<td>Bachelor</td>
<td>Computer Science</td>
</tr>
<tr>
<td>Florida</td>
<td>F</td>
<td>Bachelor</td>
<td>Finance</td>
</tr>
<tr>
<td>Washington</td>
<td>F</td>
<td>Bachelor</td>
<td>Banking</td>
</tr>
</tbody>
</table>

7.4 Results

This section endeavours to scrutinise data gathered from playtesting sessions, structure and transform it into information, and reason on what conclusions can be attained regarding the main objectives of the gaming platform. Firstly, it is analysed data that is related to implemented features that meet the requirements designed in Chapter 4 Proposed Solution. Then, a parallelism between these data and the participant’s background is established. Lastly, answers to the questionnaires are exposed, so that more rationalisations can be reached.

Box plots depicted in Figure 7.3 graphically represent the distribution of numerical data from the number of activities and number of votes placed for each process in both case studies. The Expected Number of Activities parameter was defined in the server side with the number of activities of case studies 1 and 2 depicted in Figures 7.1 and 7.2, respectively.

Analysing the left plot, it can be acknowledged that the median number of activities of each process created by players is not far from the expected number of activities in each case study (in case study 1 the value diverges by 1 activity, and in case study 2 the median and expected number of activities coincide). However, the range between minimum and maximum number of activities for each process (9 to 15 in case study 1 and 9 to 13 in case study 2), is somewhat wide, indicating that the platform allows for players to freely represent processes with arbitrary level of detail, one of the initial premises of this dissertation work.

![Box plot for the number of activities and votes for each process.](image)
The plot on the right depicts the distribution of votes placed for each process in each case study. In case study 1, it can be observed that positive quality votes placed varied from 2 to 3 in each process, as in negative votes varied from 0 to 1. In case study 2, positive votes varied from 0 to 2 and negative votes from 2 to 3 for each process. Accordingly, in the triage of patient’s process, all process models had a positive vote rate; in the finalise dissertation’s process, processes created either earned negative vote rates, or got caught between a tie. This information do not uncover any particular pattern; so, that may reveal that disparity found in the number of positive and negative votes placed might relate to the perception of players about the quality itself of process models created, which was the crucial objective when designing the mechanics for this feature. Additionally, this disparity may also be associated to the adjustment of strategies adopted by players in the game, although further iterations of playtesting sessions with different populations need to be performed to more precisely understand the true relationship.

In Figure 7.4 it is represented the total number of achievements and medals earned by players in both sessions. By reviewing game mechanics regarding this matter, it is evident why in both sessions the total number of the earned achievement and medal First Process Published is 5 (total number of players) and 1, respectively. Regarding the Correct Own Process achievement, 2 players earned it, being 1 the average number of players earning this achievement in each session. Since there is no record of modifications to processes created by other players, participants were casually asked at the end of the sessions the reason for that, which was answered that either they felt it was not necessary to correct other processes, or while getting carried away performing other tasks, their full attention was not directed at these aspects. The same answers were given regarding detection of duplicates.

![Figure 7.4: Bar chart for achievements and medals earned by players for each process.](image-url)

The Vote Processes All Players achievement has a significant increase from the first session to the second, being that 2 players reached this achievement in case study 1, and twice the players reached it for case study 2. Excluding randomness, it could be understood that the increase in votes

71
might be induced by players learning what tasks are encouraged in the game in order to gain more score points. The occurrence of 3 Best Quality Content medals in each session can be justified by ties in quality voting (in each session there were 3 process models with best quality vote rate). Lastly, it can be observed a threefold increase of earned Best Convergence Rate, which indicates that more consensual voting occurs in the second session, when it comes to deciding the quality of process models.

In tables 7.13 and 7.14 is exposed the distribution of earned achievements and medals by players in each case study.

<table>
<thead>
<tr>
<th>Player</th>
<th>Case Study #</th>
<th>Achievements</th>
<th>Medals</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>1</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>California</td>
<td>1</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ohio</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Florida</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Washington</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.13: Game data relating earned achievements with players.

From individual information provided in the previous section, it can be noticed that the professional background of players is not of the utmost importance when it comes to earn achievements or medals.

<table>
<thead>
<tr>
<th>Player</th>
<th>Case Study #</th>
<th>Medals</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>California</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Ohio</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Florida</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Washington</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.14: Game data relating earned medals with players.

Box plots depicted in Figure 7.5 disclose answers given by participants in the questionnaires handed at the end of sessions. Regarding previous knowledge of business process modelling, players were averagely familiar with the subject, whether only knowing the basic concepts, or even being familiar with few traditional languages. When it comes to previous knowledge of serious games, the majority of players was poorly familiar with the matter.

In regard to the easiness of understanding the process language and creating processes with it, the majority of players answered these questions as easy to understand and create. Concerning the
easiness to create processes in the game, fairness of game rules and effectiveness of the application in eliciting processes, the median given answer corresponds to easy, fair and effective, respectively.

Figure 7.5: Box plot for the answers to the questionnaires.

In order to draw conclusions and recognise patterns between players’ backgrounds and performances in the game, in Table 7.15 is mapped the individual answers given by players in the questionnaires.

Table 7.15: Game data relating players and answers given by them in the questionnaires.

Table 7.16 shows the process models determined as belonging to the final consensus set.

Table 7.16: Game data relating to process models belonging to the final consensus set.

In case study 1, the author of the winning process is the participant that answered the questionnaire as being not familiar with both business process modelling and serious games. In case study 2, the author answered as being poorly familiar with both knowledge areas. This further indicates that it is not required to be familiar with both subjects to be successful in the game.
7.5 Concluding Remarks

From the results gathered and disclosed throughout this chapter, it may be remarked that the application was generally well accepted by players in collaboratively building business process models. It can be additionally deduced that the professional background of players or previous knowledge of business process modelling is not essential to guarantee good performances in the game.

Concerning the architecture of the implemented solution, results hint that game mechanics were fairly combined. Answers given in the questionnaires further corroborate this belief. When it comes to issues during playtesting sessions and suggestions made by players during that time, it was proposed improvements to the user interface.

Considering the positive results, it is likely that the application may be considered effective as a collaborative platform where business participants can build business process models and reach a consensus on which model portrays the best representation of how the process is carried out.
Chapter 8

Conclusions

Recalling the challenge proposed at the beginning of this dissertation, the main objective of this research work is to evaluate the possibility of creating a gaming platform that encourages business participants to cooperatively build business process models and collaboratively elect the best representation. Initial assumptions dictate that the application must collect implicit organisational knowledge about business processes from direct business participants, in a systematic and automated way, in order to diminish obstacles brought by current process elicitation approaches.

8.1 Accomplished Work

In order to attain such solution, an assessment over state of the art work in areas necessary to intervene was performed, serving as a solid base for this research work. A study was addressed regarding conceptual frameworks, assessing organised structures commonly used in business process modelling, so that decisions could be made regarding the structure to be used in the application. Then, an examination over diversified aspects of business process modelling was performed, along with a summary of relevant graphic notations used to represent business processes. Furthermore, an analysis over current work on serious games and desired guidance when building such systems is outlined. Lastly, game technology is explored, with the purpose of going over game engines that are resourceful in the development of this project.

In pursuance of identifying major complications brought by approaches used nowadays by business analysts to elicit business processes of an organisation, an overview on such techniques was conducted. The identification of issues related to this domain is attained, and conclusions are drawn concerning desired improvements or characteristics for a possible solution, which serves as input to the next chapters, when discussing the proposed solution.

Considering the aforementioned identified issues, a conceptual definition of the game is reached, along with a specification of desired behaviour and interactions between players. Additionally, a blueprint of the chosen graphic notation to represent processes in the game is laid out.

The chapter reserved to game design intends to further establish how players are to behave in
the game, explaining basic concepts used throughout this dissertation, defining game mechanics that regulate players' actions, and culminating with a specification of what bonifications were developed to encourage players to perform certain tasks, and penalisations applied in behaviour that is to be avoided in the game, based on conclusions reached throughout the previous chapters.

Posteriorly, clarifications regarding the architecture of the implemented solution were stated. Not only this part provides a detailed explanation of blocks of functionality that put together culminate in the architecture of the game, but also delineates nonlinear implementation aspects that leverage the way the application reaches its objectives.

The evaluation of the accomplished work is attained then, focusing on assessing the implemented solution in a way that verifies the fulfillment of previously defined requirements, validates the correct conduct of the application, and justifies reached deductions with results from real world scenarios where business participants interact with the platform.

Performing an assessment to the accomplished work throughout this dissertation, it can be perceptible that it is possible to overcome the challenge of building a solution as a gaming platform, which encourages players to collaboratively reach a consensus regarding a unanimous representation of a business process of an organisation. However, it must be taken into consideration that process models elicited in the game and determined by players as the best representation may not depict the process with complete accuracy, because that depends on the individual skills of players regarding problem solving, general interpretation of information and capacity to transform their implicit knowledge into explicit representations. Therefore, a posterior rationalisation by business analysts to these process models is essential, to identify incongruities or propose improvements to representations.

### 8.2 Limitations

The decisions settled when designing the chosen approach arise some limitations which are important to be analysed and discussed. A list with the identified limitations is featured below:

- The main premise taken into consideration when designing the business process graphic language was that the lack of knowledge in business process modelling should not compromise the ability of players to build models effectively and accurately. Consequently, the minimalism of the language domain leads to limitations regarding the choice of available symbols by players, who may have difficulties when trying to represent certain kinds of information, especially players familiar with business process modelling.

- Data gathered from playtesting sessions is insufficient to accurately determine correlations and dependencies between variables in the game. Given the complexity of the application regarding game rules and scoring system, two playtesting sessions is only enough to recognise patterns and provide a qualitative analysis on collected game data.

- Technological decisions established when choosing a method of implementation deteriorates players' game experience when building models of the business process to be elicited (e.g. when
building a model with 4+ lanes, the rendering of flows is damaged due to graphical constraints of the chosen game engine).

8.3 Future Work

Despite positive results reached when evaluating the gaming platform, there is always opportunity for improvement. The following list provides suggested improvements where future work should focus on:

- **Upgrade process language validator.** Instead of validating the compliance of models with process language rules when players are publishing the process, it would be interesting to implement a validator which informs players about invalid actions as soon as they try to perform them.

- **Improve user interface.** Although HCI is out of the scope of this dissertation, participants of playtesting sessions suggested that improvements to the user interface might generally improve their performance in building business process models.

- **Parameterise scoring system.** Organisations interested in using the platform to elicit processes may find useful to configure in the server side the weights of bonuses and penalties, or even append further game rules that encourage certain behaviour in the game.

- **Perform further playtesting iterations,** with the purpose of tuning the scoring system to more precisely motivate and engage players in their tasks; and to reach more conclusions regarding the capability of the application and the fulfilment of its main objectives.
Appendix A

Appendix

A.1 XML Export of Triage Process

<GameState xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Processes>
    <Process Name="Triage of Patients" Author="PlayerA" Published="true" PID="1">
      <Pool>
        <Lane Participant="Patient">
          <LaneElements>
            <Primitive xsi:type="Event" Category="Start" />
            <Primitive xsi:type="Event" Category="End" />
            <Primitive xsi:type="Activity" Name="Give personal information" />
            <Primitive xsi:type="Activity" Name="Go to triage room" />
            <Primitive xsi:type="Activity" Name="Wait for his/her turn" />
            <Primitive xsi:type="Activity" Name="Go to treatment room" />
            <Primitive xsi:type="Activity" Name="Pay fees" />
          </LaneElements>
        </Lane>
        <Lane Participant="Secretary">
          <LaneElements>
            <Primitive xsi:type="Activity" Name="Write registration form" />
          </LaneElements>
        </Lane>
        <Lane Participant="Nurse">
          <LaneElements>
            <Primitive xsi:type="Activity" Name="Triage patient" />
            <Primitive xsi:type="Activity" Name="Assign priority level" />
          </LaneElements>
        </Lane>
      </Pool>
    </Process>
  </Processes>
</GameState>
Note: This XML example is simplified in the sense that irrelevant attributes were removed, such as empty flow conditions or flow positions, which do not leverage in any way the processing of game information by the gaming platform. Attributes devoid of business knowledge are omitted, since it is only considered when dealing with game mechanics.
A.2 Questionnaires of Playtesting Session

ORGANIX
Collect Implicit Organisational Knowledge Through a Gaming Platform

Questionnaire:
(English)

Age: ________  Sex:  M ☐  F ☐

Degree of Education: __________________________

Professional Area: ___________________________  Date: ___ / ___ / ______

Previous knowledge of Business Process Modelling:


Previous knowledge of Serious Games:


1. Is it easy to understand the elements of the process language of the game?


2. Is it easy to build process models with the process language of the game?


3. Is it easy to build process models with the tools provided by the game?


4. Are the game rules and scores assigned to players fair and just?


5. Would you consider the game effective in collaboratively eliciting business processes?


Suggestions: ____________________________________________

_____________________________________________________

Figure A.1: English version of the questionnaire given to participants of playtesting session.
ORGANIX
Collect Implicit Organisational Knowledge Through a Gaming Platform

Questionário:
(Português)

<table>
<thead>
<tr>
<th>Idade:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sexo:</strong></td>
<td>M</td>
</tr>
<tr>
<td><strong>Grau de Escolaridade:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Área Profissional:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Data:</strong></td>
<td></td>
</tr>
</tbody>
</table>

| --- | --- | --- | --- | --- | --- |

1. É fácil de entender os elementos da linguagem de processos do jogo?  

2. É fácil de construir modelos de processos de negócio com a linguagem de processos do jogo?  

3. É fácil de construir modelos de processos de negócio com as ferramentas providenciadas pelo jogo?  

4. As regras de jogo e as pontuações atribuídas aos jogadores são justas?  

5. Consideraria o jogo eficaz na elicitação colaborativa de processos de negócio?  

Sugestões:  

Figure A.2: Portuguese version of the questionnaire given to participants of playtesting session.
Bibliography


