Organix – Collect Implicit Organisational Knowledge Through a Gaming Platform

Valter Nepomuceno
Instituto Superior Técnico
Av. Rovisco Pais 1
1049-001 Lisboa – Portugal
valter.nepomuceno@ist.utl.pt

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

1. INTRODUCTION

Considering the organisational progress of nowadays' industries, it can be observed that innovation and process efficiency must be improved in order to attain global competitiveness. For that reason, business analysts and consultants are responsible for analysing the organisation and designing its business processes, so that strategic plans can be delineated in the interest of boosting the performance of the organisation's business processes, and therefore, better achieving previously delineated business goals.

When efforts are directed towards eliciting business processes, a manifold of approaches and techniques can be used to gather as much implicit business knowledge as possible. The sources from which knowledge can be collected range from available process documentation to real process participants. However, most of these approaches might be compromised by the subjectivity of professional capabilities of the practitioner or even by the lack of problem solving skills.

From the standpoint of increasing the efficiency and quality of business process elicitation, it would be of interest to explore the possibility of creating a solution that would tackle complications brought by present elicitation techniques and approaches. Sometimes, communication difficulties may undermine the results of process elicitation; therefore, the solution to be explored should not involve intermediaries who extract implicit knowledge from business participants. Thus, consultants can be allocated to perform activities that follow the business process modelling phase, reducing the financial costs significantly to both professional services firms and companies whose processes need redesign.

The leading challenge of this work subsists in designing a solution, as a serious game platform, that would engage business participants in contributing with implicit organisational knowledge, regarding the business processes they are involved. The solution must provide mechanisms that allow participants to cooperatively build business process representations of the processes to be elicited, and to collaboratively elect the business process models that best represent how processes are carried out in real life.

The conception of such solution requires a review of the state of the art work in business process modelling, serious games and game technology. Afterwards, follows a rationalisation behind which approaches should be used to incorporate the solution, and which alternatives should be explored in order to differentiate techniques used nowadays that lead to identified constraints and obstacles.

So that business participants are able to communicate their implicit knowledge about the processes they participate, in a structured and systematic way, it is essential that the solution includes a process model notation. The graphic notation should be as simple as possible and easy to use, in order to avoid constraints in the contribution of knowledge because of the lack of expertise in the area of business process modelling of participants.

Lastly, this dissertation plans to provide mechanisms that allow business analysts to export the explicit knowledge about

business processes, contributed by real world business participants, so that those process models can be used for posterior decision making and redesign, in order to improve their effectiveness and efficiency.

2. STATE OF THE ART

In this section it is addressed an assessment to related research work regarding the areas of business process modelling, serious games and game technology.

2.1 Business Process Modelling

A business process consists of a structured set of activities, which describes their logical order and dependencies, focused in business goals to produce a desired result [1].

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. Some properties that are of the utmost importance in a process language are [2]: completeness, every part of the business process should be represented and reached; clearness, a process model must be capable of communicating the process to stakeholders over different perspectives in a clear and intuitive way; and ease of use, the learning curve of the language should be as easy as possible.

A flowchart is a type of diagram that can be useful when it is required a high level of detail in the representation of processes. Flowcharts are structured graphical representations of work or manufacturing processes, whose symbols are used to represent operations, data, flow direction and equipment [1]. However, this technique arises a constraint related to the inability of assigning responsibilities to activities in the diagram.

The Unified Modelling Language, or most commonly known as UML, is used as a standard modelling language in the application area of software systems. Although the language was not designed for the representation of business processes, its notations have been used to model behaviour, originating activity diagrams.

Additionally, extended profiles have been created, augmenting the language with business concepts such as resources, goals, processes and rules [3]. Resources consist of objects that are part of the input/output of the business, namely people, materials, information and products. 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 [4].

The Business Process Modelling Notation, or BPMN as is more commonly known, was designed with the purpose of providing a mechanism that would allow the representation of business processes. The leading endeavour in the creation of this notation was for it to be easily communicated to a wide variety of business participants, without losing the ability to do it in a comprehensive and intuitive way [5].

The language provides four different categories of symbols objects: Flow and Connecting objects, Swimlanes and Artefacts. The Flow Objects category encompasses the symbols Event, Activity and Gateway. Events are characterised by a trigger and a result, affect and control the flow of the process, and can be differentiated as Start, Intermediate or End events. Activities are used to represent work performed by the organisation, and can either be atomic, names Tasks, or composed, names Sub-Processes. A Gateway is considered a business decision, representing the divergence or convergence of flow symbols.

Connecting Objects envelop Sequence Flows, Message Flows and Association, which are used to link together the other objects of the diagram. These objects establish the order of execution of activities, represent the flow of messages between business participants, and characterise the association between objects, namely the input and output of activities, 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 sub-partitions named Lanes.

Artefacts are useful when building context for process models, and the category contain Data Objects, to communicate readers the required or produced data of an activity; Groups may be used with the intention of organising diagram objects; and Annotations can be used to add textual comments within the process scope.

2.2 Serious Games

The definition of serious game ranges considering different authors, who describe the concept under many perspectives. In order to determine which concept is more adequate to achieve the objectives of this research work, it is important to determine a concept definition that will serve as a basis for the development of this project. Under the most relevant definitions, 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 [6];
- Considered a mental challenge, played under specific game rules, using entertainment for purposes of training, communicating and learning, among others [7];
- Consists of a voluntary activity within the context of a simulated, conceptual or imaginary world created purposely for the game, to capture the players' attention, being the game conducted under defined rules [8].

The concept definition that will be considered in the fulfilment of this work is a combination of the previous definitions and states that a serious game is a game whose main purpose is beyond pure entertainment of the player, being used in a variety of tasks, such as learning, training, simulating, informing or advertising.

2.3 Game Technology

The video games industry is presently valued as a $70+ billion business, and due to the fast market growth and worldwide demand, the evolution of improved development software and tools have been growing accordingly [9]. In order to magnify game complexity without overloading complexity in development, cutting edge game engines have been developed to assist developers in computer science areas such as graphics, artificial intelligence, networking, algorithms, parallelism, etc. [10].

Concerning the decision on which game engine should be chosen to develop a certain game, the team of developers must identify the priority features that the framework ought to deliver, so that technological limitations imposed by the engine impact the development phase as little as possible.

A brief search on the existing and available game engines reveals that the number of results is considerably generous. Further analysis was performed in 7 game engines that best matched the objectives of this work: Unreal Engine (UE), IdTech (IdT), CryEngine (CE), Unity 3D (U3D), IW Engine (IW), Source Engine (SE) and RAGE (RG). A comparison of the most important features of the aforementioned game engines is outlined in Table 1 (Appendix).

3. PROBLEM ANALYSIS

This section aims at providing an assessment on how business process elicitation is performed nowadays. Then follows an identification of the major complications brought by approaches used in the collection and representation of business knowledge about processes.

3.1 Stakeholders

Concerning process elicitation, the types of actors that can be identified are business participants, administrators and consultants.

- **Business participants** are directly involved in the business processes performed by an organisation, and therefore possess implicit knowledge that facilitate the construction of business process models.
- **Administrators** decide the scope of process elicitation and regulate how it is going to be performed. Commonly, administrators are integrated in the top management of a company, and therefore are driven by business goals; so, it is of their best interest that elicitation projects follow certain guidelines.
- **Consultants** perform decision making on the built process models, in order to perfect efficiency of business activities performed by the organisation.

3.2 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**: Business analysts conduct interviews to business participants, in order to extract business knowledge about processes. Storytelling activities and role-playing games may also be performed.
- **Analysis of Documentation**: The available documentation on business processes is studied by business analysts, whose information such as schedules, stakeholders' identification and procedure descriptions may be attainable.
- **Direct Observation**: Business participants are observed while performing their daily activities, enclosed in business processes to be elicited.
- **Brainstorming Sessions**: Modellers meet to collaboratively interpret the gathered business knowledge and build business process models, using graphic languages that represent information in a structured way.

3.3 Substancial Complications

Although the aforementioned approaches have been used in enterprise environments for quite a while, they implicate certain flaws that may invalidate the results of the process elicitation. Some problems that may arise from the practice of these techniques are detailed below:

1. **Difficulties in communication**
   The majority of elicitation techniques involve communication between two parts, and therefore communication gaps may occur. In complex business domains (e.g. insurers, law firms, etc.), consultants may struggle to fully grasp the tacit knowledge being transmitted, making it difficult to formulate adequate questions to business participants because of the lack of common vocabulary. Due to the subjectivity of languages, it may also possible that respondents provide ambiguous information without noticing, making it difficult for consultants to detect these incongruities.
2. **Lack of explicit information**
   In case of lack or non-existence of documentation regarding business processes, consultants are forced to rely on other techniques that might not contribute with the same adequate result.
3. **Restriction imposed by skills**
   While modeling business processes, consultants may settle for a certain approach based on their expertise, despite these techniques might not be appropriate for the problem in question. On the other hand, consultants may choose the correct approach for the situation, but lack experience in practicing it.
4. **Time-consuming process modelling**
   Commonly, process elicitation is carried out using a combination of techniques, performed by a team of analysts and modellers. Accordingly, the execution of these approaches result in time-consuming projects, allocating a large amount of human resources, and therefore bringing substantial costs to the organisation.

3.4 Conclusions

Because of the nature of complications brought by described techniques, an hypothetical solution must guarantee that process elicitation is made in a systematic and automated way, 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: business participants.

4. PROPOSED SOLUTION

In this section, it is taken into consideration the major problems in process elicitation techniques used, in order to design a solution that overcomes these obstacles.

This dissertation proposes a solution based on the creation of a serious game that engages players in eliciting the business processes they are involved. The platform should provide mechanisms that enable the creation of business process models, by motivating players in building the most accurate and realistic representation according to their local view of the processes.

Whilst the game is being played, a consensus must be reached between players, regarding which models are elected to be portraying the elicited processes more correctly, consisting this process model set in the output of the game.

4.1 Game Requirements

Functional requirements specify the desired functionality for the application, allowing players to accomplish their tasks, therefore satisfying the business requirements. An alignment between use cases and requirements of the system is delineated at the end of each description.

REQ.1 - Contribute with business knowledge (UC01)
The player shall be able to define organisational scenarios, by creating business process models that communicate the local view of the processes the business actor (player) participates.

REQ.2 - Graphic representation of business processes (UC01, UC02 and UC04)
The platform must provide a graphic notation that assists players in the representation of business processes. The language should be simple and easy to learn and use.

REQ.3 - Propose modifications to other players' process models (UC02)
A player should be able to propose modifications to process models created by other players, stating that the alterations will better correspond to how processes are performed in real life.

REQ.4 - Rate other players' contributions (UC03)
The platform shall allow players to rate process models created by other players. Voting for a process may either assert that the voter thinks the representation is accurate, or does not represent well how the process is carried out in real life.

REQ.5 - Identify a process model as duplicate of other (UC03)
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.5 - Game settings (UC05)
The platform must allow for administrators to set configurations that will be applied to the following game sessions, such as game length, number of players and scoring parameterisations.

REQ.6 - Export business knowledge (UC07)
Administrators shall be able to export business process models created in the game, described in a structured and well-known markup language (*e.g.* XML).

REQ.7 - Save and load game instances (UC06)
The platform must allow administrators to save information about game sessions being played at the moment, in order to load them later, so that players can resume the process elicitation that was in course.

REQ.8 - Visualisation of game results (UC04)
The platform must allow players to visualise the final results of the game instance they played.

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.9 - Multiplayer application
The platform must allow for different players to play the same game session simultaneously.

REQ.10 - Cross-platform application
Players must be able to play the game using computers with different configurations and/or operating systems.

4.2 Use Cases

The following use case diagram depicts the multiple interactions that users of the proposed gaming platform are allowed to perform.

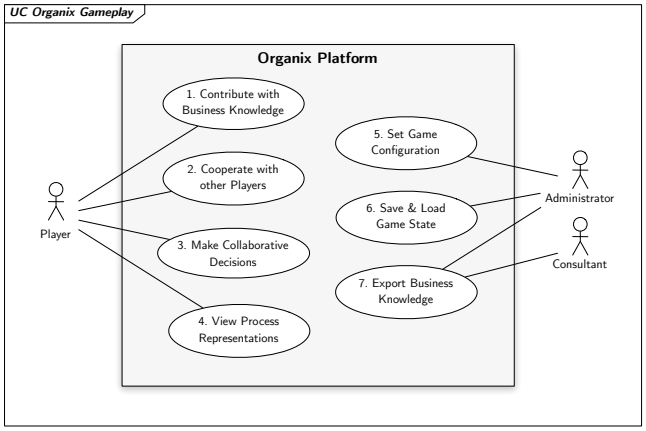


Figure 1: Use case diagram for the proposed solution.

UC1: Contribute with Business Knowledge: Players build representations of business processes they are involved professionally in their organisation.

UC2: Cooperate with other Players: Players play the game simultaneously, in a way that allows them to cooperate with each other in the contribution of business knowledge.

UC3: Make Collaborative Decisions: Players collaboratively evaluate process models created in the game, by rating them regarding quality and accuracy.

UC4: View Business Process Representations

Players visualise the resultant models of the elicited business processes, as well as his individual contributions.

UC5: Set Game Configuration
Administrators set game configurations that will parameterise the game sessions to be posteriorly played.

UC6: Save & Load Game State
Administrators save information concerning the game session being currently played, or restore that information, so the game session can continue to be played.

UC7: Export Business Knowledge
Administrators export business knowledge elicited in the platform by its players.

4.3 *Business Process Graphic Language*

In order to construct a graphic language capable of representing business processes, various desired features can be identified so that objectives for building the notation converge with the goals of this research work:

1. **Easiness to use**: Business participants may vary in knowledge of process modelling, education, level of skills when dealing with computers and software or even in ability to solve problems. Therefore, the notation must be as simple as possible and easy to use.
2. **Completeness of domain**: The language domain must encompass a minimum set of graphic symbols, even though it is capable of representing comprehensively, accurately and detailedly the business processes of any organisation.

To design a business process notation that has the capacity to represent business processes following the goals outlined above, a combination and adaptation of business process modelling languages studied in the chapter of related work, in section 2.1 *Business Process Modelling*, is attained. The constructed graphic notation grasps the concepts and simplicity of representation of UML and Flowchart diagrams, while simultaneously adopting a minimum set of symbols of the thorough domain of BPMN.

4.3.1 Pool and Lane

Pools group language symbols 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.

Figure 2: Domain model for the proposed solution.

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 lane.

4.3.2 Event

Start, Merge and End events may be grouped into a set responsible for the control of 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.

(a) Start Activity (b) Merge Activity (c) End Activity

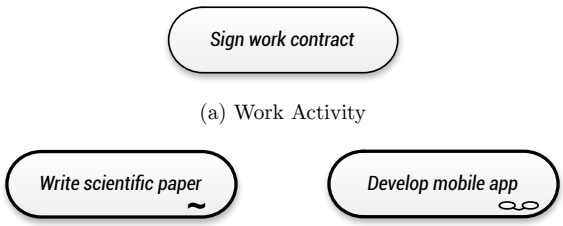
Figure 3: Graphic representation of events in charge of controlling the process flow.

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.

4.3.3 Activity

There are three different kinds of activities: Work Activities, Composed Activities and Ad-Hoc 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.



(a) Work Activity

(b) Composed Activity (c) Ad-Hoc Activity

Figure 4: Graphic representation of activities.

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.

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.

4.3.4 Flow

Sequence flows communicate the order of execution of primitives, and may be accompanied by a condition.

Information flows acquaint the exchange of data, information, business artefacts or even materials between participants/lanes.

(a) Sequence Flow (b) Information Flow

Figure 5: Graphic representation of flows.

Rules:

1. Each primitive must be reached through flows and must connect to a subsequent primitive, except for start and end events.
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.
3. Only work, composed and ad-hoc activities are allowed to have loop cycles. The execution of the loop will be performed for as long as the condition of the flow remains true.

4.4 Domain Model

A list of the existing classes in the domain model of the platform is displayed posteriorly:

- **Process**: Set of sequenced and structured activities that compose and represent the business process being elicited. Main entity of the domain.
- **Process Version**: 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.)
- **Vote**: Vote of one player in polls regarding quality or duplication of a Process.
- **Lane**: 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.
- **Primitive**: Abstraction of Activities, Events and Flows.
- **Flow**: Connects two Primitives, which can either be Activities or Events.
- **Event**: Controls sequence and information Flows between Primitives (except Flows them- selves), beginning and terminating in start and end Events, respectively.

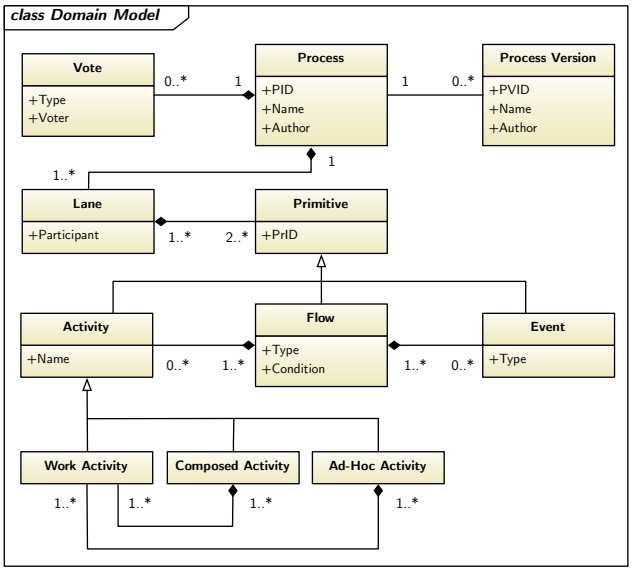


Figure 6: Domain model for the proposed solution.

- **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.

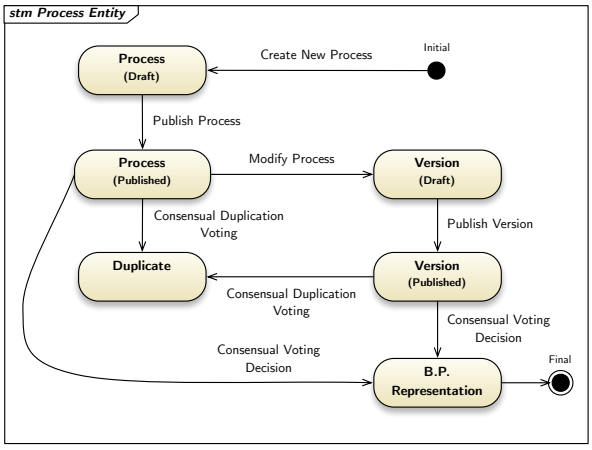


Figure 7: State diagram for the entity `Process`.

Figure 6 unveils the existing entities that belong to the domain model of the desired solution, focusing on the relationships and multiplicities between entities.

The `Process` entity consolidates the implicit business knowledge, contributed by players. It also has assigned game data that classifies the process scores, and is related to `Process`

`Versions`, which are resemblant to processes. Figure 7 depicts how the `Process` entity evolves throughout the game.

5. GAME DESIGN

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 cycle is divided into the following phases:

1. **Pre-Gaming Phase**: A consensus reach amongst administrators is attained, regarding which process is to be elicited by players in the game.
2. **Process Publication Phase**: Subsists in players building and publishing business process models.
3. **Collaboration Phase**: Players can execute multiple actions directed at enforcing multiplayer collaboration in process reviewing, rating and duplication detection.

At the end of the game, according to process ratings, 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.

5.1 Game Concepts

The following list aims at disclosing the most important concepts used throughout the game:

- **Process Creation**: The platform provides mechanisms that allow players to create process models under the process language rules specified in section 4.3 *Business Process Graphic Language*.
- **Final Consensus**: Minimum set of published process models with best rate of positive quality votes, which are not consensually marked as duplicates. An elected representation indicates that players consensually elected that process model as the one that best represents how the process is performed in real life.
- **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.
- **Process Duplication**: Players are able to mark one process model as duplicate of another, being permanently considered a duplicate if the majority of players agree that the model in question is indeed a duplicate.
- **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.
- **Achievements & Medals**: Achievements consist of milestones that acknowledge the progress made by a player in the game:
  - **First Process Published**: The player publishes his first process in the game, whether or not other players have already published processes.
  - **Correct Own Process**:The player proposes a modification to an original process published by himself.
  - **Modify Other Process**: The player proposes a modification to an original process published by another player.
  - **Vote Processes All Players**: The player places a vote in at least one of the published processes of all other players.

  Medals mark exceptional behaviour attained by players:
  - **First Process Published**: The player publishes the first ever published process in the game.
  - **Duplication Detection**: The player marks a process as a duplicate and the vote rate indicates that other players agree.
  - **Best Quality Content**: The player publishes the process with best quality vote rate in the game.
  - **Best Convergence Rate**: The votes placed by the player meet the most the voting decisions made by other players.

5.2 Game Mechanics

The main goal of players is to build representations of the business process of the organisation they belong using the process language provided by the game.

5.2.1 Player's Actions

The following actions can be performed in the game by the role *Player* discussed in section 4 *Proposed Solution*:

- **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. Addition- ally, 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.

5.2.2 Game Rules

The conception of game rules directs its focus at valorising and devaluing process scores, according to actions carried

out by players, whose outcomes are considered to enforce or renounce the objectives of the application.

Process/Version Creation

- **GR1**: Players can only create processes under the process language rules defined in Section 4.3.
- **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.

5.3 Scoring System

This section endeavours to explain the reasoning behind the conception of the scoring system chosen for the gaming platform.

5.3.1 Bonifications and Penalisations

Regarding process creation and publication, the process author is rewarded for each non-empty created activity. To avoid situations in which players create processes with an excessive number of activities exclusively to achieve more score points, a mechanism was implemented that compensates/penalises players who attain processes with convergent/divergent number of activities compared to the ideal number, value set by administrators.

The implementation of this mechanism follows a modified Poisson Distribution. The modified distribution can be obtained as $Poi' \sim (k, E) = 2Poi \sim (k, E) - Poi \sim (E, E)$, in order to assign a positive/negative percentage of the process score to the author if the number of activities converges/diverges from the expected value. The value E is set in the game configuration, and k the number of activities of the processes created by each player.

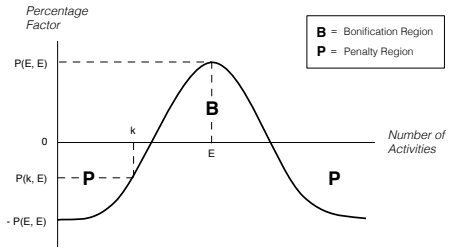


Figure 8: Modified poisson distribution of the scoring system.

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.

With respect to process duplication, whenever a process is marked as a duplicate, if players determine 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.

5.3.2 Bonus-Malus System

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. The defined scoring aspects are described subsequently:

- **Process and Version Authorship**: Takes into consideration everything related to publishing processes, 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 section 5.1 *Game Concepts*, 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 depicted in Figure 14 (Appendix) 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.

5.4 Final Consensus

The algorithm that retrieves the process representation(s) that belong to the final consensus set is represented in Figure 9. At the beginning, all processes and process versions are gathered, and representations that have been consensually determined as duplicates are filtered out.

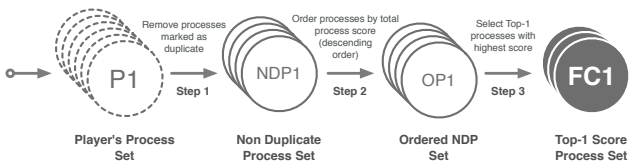


Figure 9: Algorithm that retrieves the final processes contained in the Final Consensus.

Then, the list of remaining processes is sorted by process score in descending order. Ultimately, it is retrieved the top-1 processes with higher process score. The output may consist of a list of more than one process, in case there is a tie between process scores of two or more processes.

5.5 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:

$$FinalScore(player) = \sum_{i=1}^{I} ProcAuth(Proc_i) + \sum_{j=1}^{J} VersAuth(Vers_j) + \sum_{k=1}^{K} VotReward(Vote_k) - VotAbs() + \sum_{l=1}^{L} DupDet(mark_l) + \sum_{m=1}^{M} FinalCons(Proc_m) + Achiev() + Medals()$$

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.

6. GAME IMPLEMENTATION

Analysing the objectives of this research work in section **??** *Objectives*, the following key features preferably provided by a game engine can be identified:

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;
5. **GUI System**, being a powerful tool for the implementation of user interfaces, due to the fact that the game is two-dimensional.

Examining the comparison of game engines defined in Table 1 (Appendix), it can be acknowledge that the game engine that best accommodates the desired features is **Unity 3D**. Unity guarantees integration with an IDE named MonoDevelop, using the scripting languages C#, JavaScript or Boo. Due to the multiplicity of used data types, and the preferability over popular languages, the chosen programming language was **C#**.

6.1 Game Architecture

This section endeavours to describe detailedly the architecture of the application. Since the platform provides multiplayer functionality, the chosen distributed computing architecture is the client-server model, so that players can manipulate information through client instances, being the information orchestrated and regulated by the server instance.

Client and server instances are individually designed as modular architectures, emphasising the separation of functionality into independent modules. There are common modules to both client and server sides, namely the modules `User Interface`, `Data Models`, `Networking`, `In-Memory Database` and `Game Engine`, which are self-explanatory.

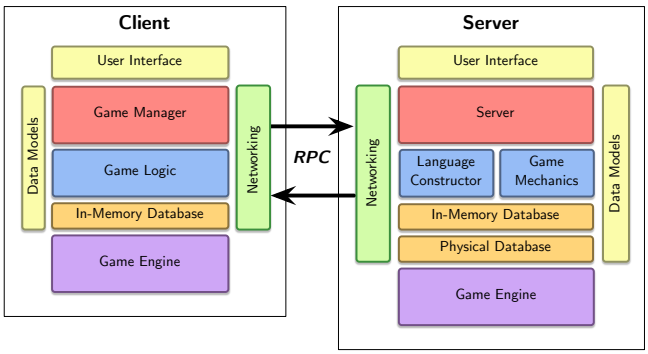


Figure 10: Client-server modular architecture for the gaming platform.

Modules `Game Manager` and `Game Logic` are incorporated exclusively in the client side. The former coordinates the game components that control the way the game unfolds, and guides players visually and informationally through different game phases. The latter implements interactions between players and the game, and extends to the governance of access privileges of game information.

Regarding the server side, the `Server` module is in charge of receiving RPC requests, pipelining requests that need further verification to other modules, and dispatching acknowledgement messages back to players. The `Language Constructor` is responsible for the construction of business processes and the validation of those process models, according to the conformity with the process language rules. `Game`

`Mechanics` incorporates the mechanics that regulate the behaviour of players, specified minutely in section 5.2 *Game Mechanics*. The `Physical Database` assures that all game information produced since the server was started is saved.

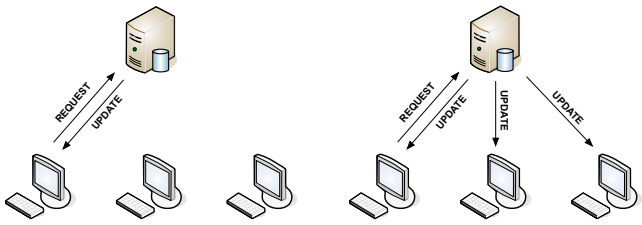
6.2 Data Persistency

In order to avoid data loss from application bugs or crashes, whenever the application quits, the current game state is saved physically, using object serialisation. Additionally, when the server application starts, the previous game state 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 players log in successfully.

For the benefit of representing information in a structured format, 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.

6.3 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.



(a) Single player update (b) Global update

Figure 11: Client-server interactions upon client request.

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.

7. EVALUATION

In order to evaluate the implemented gaming platform as broadly as possible, an evaluation process has been delineated. An assessment on the degree of fulfilment of requirements is performed, along with the development of test cases to validate the coherence of the scoring system. Also, playtesting sessions with real world business participants are performed.

7.1 Fulfilment of Requirements

From the set of requirements defined initially in the game design, it is now performed a qualitative analysis of their fulfilment or unfulfilment. The full text contains a detailed table with all the requirement fulfilment analysis.

The developed solution implements every functional and non-functional requirement specified in section 4 *Proposed Solution*. Therefore, it can be concluded that this project was successful in implementing the solution comprehensively, as defined in specifications.

7.2 Test Cases

To validate the balance of the scoring system and its conformity with the specified behaviour, test cases were developed and implemented so that collected data could be analysed and corroborated.

A set of 3 test cases was designed, and its main focus was bonifications and pensalisations associated to players' actions. This validation not only confirms that the implemented game rules are behaving as determined in game design, but also exposes the direct results of those game rules into players' scores, exposing the balance of the bonus-malus system.

7.3 Playtesting

Playtesting sessions were performed to conceptually and technologically assess the architecture of the implemented solution. A total of 5 participants were gathered to test the gaming platform simultaneously.

At the beginning of the testing sessions, a 10 minute explanation of game concepts, objectives and rules was provided, along with a description of the process language used. Then, participants played two different sessions of 30 minutes each, with different case studies. For each session, players were handed a textual description of well known processes and were asked to collaboratively elicit a business process model. At the end of these sessions, a questionnaire was delivered to participants, to examine previous knowledge players had of business process modelling and serious games, and also to study their perception of the application and the fulfilment of its objectives. Answers of participants are depicted in figure 12.

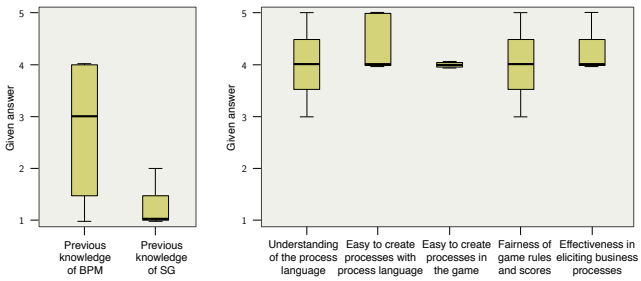


Figure 12: Box plot for the answers to the questionnaires.

Additionally, all gathered game data was analysed, processed and interpreted, so that information regarding players' performances could be disclosed, and also to evaluate decisions made by the platform. Gathered game data was also

mapped to corresponding players, in order to recognise patterns between players' backgrounds and their performance in the game.

In figure 13 it is depicted the evolution of achievements and medals earned through the two sessions. It can be perceived that an increase in achievement `Vote Processes All Players` is quite wide, suggesting that, disregarding randomness, players learned by playing that it would be beneficial to collaborate to elect the process belonging to the final consensus.

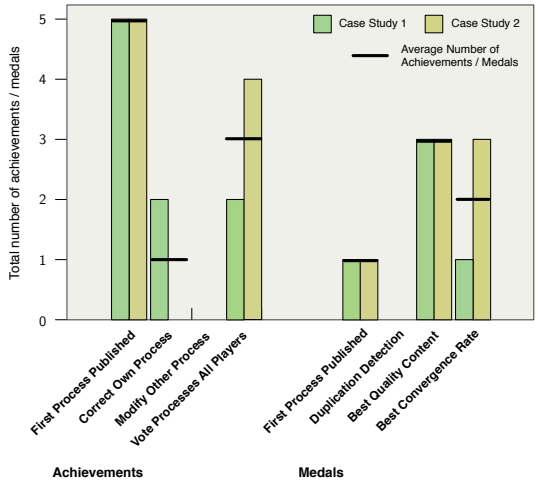


Figure 13: Bar chart for achievements and medals earned by players for each process.

Furthermore, earned medals `Best Convergence Rate` also increased in number largely, which indicates that more consensual voting occurs in the second session.

From the results gathered, 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. 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.

8. CONCLUSIONS AND FUTURE WORK

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. Therefore, a posterior rationalisation by business analysts to these process models is essential, to identify incongruities or propose improvements to representations.

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:

- Implement a validator that informs players about invalid actions as soon as they try to perform them, instead of only validating when the process is being published.
- Testers suggested that improvements to the user interface might improve their performance in the game.
- It would be interesting to configure in the server side the weights of bonifications and penalisations.
- Perform further playtesting iterations, with the purpose of tuning the scoring system; and to reach more conclusions regarding the capability of the application.

References

[1] R. S. Aguilar-Saven, "Business Process Modelling: Review and framework," *International Journal of Production Economics*, vol. 90, pp. 129–149, July 2004.

[2] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business Process Management: A survey," *Proceedings of the 1st International Conference on Business Process Management*, vol. 2678, 2003.

[3] H. Eriksson and M. Penker, *Business Modeling with UML: Business Patterns at Work.* New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 2000.

[4] N. Russell, W. M. P. van der Aalst, A. H. M. ter Hofstede, and P. Wohed, "On the suitability of UML 2.0 activity diagrams for Business Process Modelling," *3rd Asia-Pacific Conference on Conceptual Modelling (APCCM)*, vol. 53, pp. 95–104, 2006.

[5] S. A. White, "Introduction to BPMN," July 2004.

[6] K. Corti, "Games based learning: A serious business application," February 2006.

[7] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38, no. 9, pp. 25–32, 2005.

[8] D. R. Michael and S. L. Chen, *Serious Games: Games That Educate, Train, And Inform.* Thomson Course Technology, 2006.

[9] A. Herwig and P. Paar, "Game Engines: Tools for landscape visualization and planning?," *Trends in GIS and Visualization in Environmental Planning and Design*, pp. 161–172, 2002.

[10] B. Kot, B. Wuensche, J. Grundy, and J. Hosking, "Information visualisation utilising 3D computer game engines case study: a source code comprehension tool," *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on computer-human interaction: making CHI natural, New York, NY, USA*, pp. 53–60, 2005.

	UE	idT	CE	U3D	IW	SE	RG
High Performance Rendering	●	●	●	●	●	●	●
High Fidelity Rendering	●	◗	●	●	●	◗	◗
Multi Platform	◗	◗	◗	●	◗	◗	◗
Networking	●	●	●	●	●	●	●
AI System	●	◗	●	❍	●	●	●
Animation Engine	●	●	●	●	●	●	●
Physics Engine	●	●	●	●	●	●	◗
Audio Engine	●	●	●	●	●	●	❍
GUI System	●	❍	❍	●	❍	●	❍
Scripting	●	●	●	●	❍	❍	❍
Integrated Editor	◗	◗	●	●	❍	❍	❍
Asset Store	❍	❍	❍	●	❍	❍	❍
Fast Learning Focus	●	●	❍	●	❍	❍	❍
Free (non-commercial/academic)	●	●	●	●	❍	●	●

Table 1: Comparison of features among researched game engines.

Scoring Aspect		Condition	*Bonus-Malus* (points)
Process Authorship	Process Score (1)	Added Activity	+ 25
		Added Composed Activity	+ 50
		Added Ad-Hoc Activity	+ 50
	Expected Num. Act. (2)	True	+ PS x (2 x Poi ~ (k, E) - Poi ~ (E, E))
	Positive Vote Rate (3)	PPV > 0.5	+ PS x (PPV/4)
		PPV < 0.5	- PS x (PNV/4)
	Marked as Duplicate (4)	Dup.PPV > 0.5	- PS x 25% **(out of FC)**
		Dup.PPV < 0.5	+ PS x 10%
	Best Vote Rate in Process Tree (5)	True	+ PS x 10%
		False	- PS x 5%
Version Authorship	Version Score	Same as in (1)	
	Expected Num. Activ.	Same as in (2)	
	Positive Vote Rate	Same as in (3)	
	Marked as Duplicate	Same as in (4)	
	Best Vote Rate in Tree	Same as in (5)	
Voting (Quality/ Duplication)	Participation	Convergent Vote	+ 20
		Non-Convergent Vote	+ 5
		Tie	+ 10
	Abstention	Quality Voting	- 10
		Duplication Voting	- 25
Duplicate Detection (Marking as Duplicate)		Dup.PPV > 0.5	+ PS x 10%
		Dup.PPV < 0.5	- PS x 10%
		Dup.PPV = 0.5	+ PS x 5%
Achievements		First Process Published	+ 75
		Vote Processes All Players	+ 50
		Modify Other Process	+ 25
		Correct Own Process	+ 10
Medals		Best Convergence Rate	+ 150
		First Process Published	+ 100
		Best Quality Content	+ 75
		Duplication Detection	+ 50
Final Consensus		Belongs to the set	+ PS x 25%

Figure 14: Blueprint of the scoring system.