Lu-Lu: A framework for collaborative decision making games

Damon Daylamani-Zad*, Marios C. Angelides†[1], Harry Agius†

*Department of Computing and Information Systems, Faculty of Architecture, Computing and Humanities, University of Greenwich, London SE10 9LS, UK

†Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, Uxbridge UB8 3PH, UK

d.d.zad@greenwich.ac.uk, marios.angelides@brunel.ac.uk[1], harryagius@acm.org

Abstract: This paper proposes Lu-Lu as an add-on architecture to open MMOGs and social network games, which has been developed to utilise a key set of ingredients that underline collaborative decision making games as reported within the research literature: personalisation, team matching, non-optimal decision making, leading, decisiveness index, scoring, levelling, and multiple stages. The implementation of Lu-Lu is demonstrated as an add on to the classic supply chain beer game, including customisation of Lu-Lu to facilitate information exchange through the Facebook games platform, e.g. Graph API and Scores API. Performance assessment of Lu-Lu using Behaviour Driven Development suggests a successful integration of all key ingredients within Lu-Lu's architecture, yielding autonomous behaviour that improves both player enjoyment and decision making.

Keywords: collaborative, decision making, game, profiling, personalisation, MPEG-7

1. Introduction

The ever increasing popularity of collaborative games, especially MMOGs and social networking games, has seen a rise in the deployment of collaborative gaming in areas such as learning, decision making and health [1–3] where gaming is traditionally regarded as a distraction. There are an

[1] Corresponding Author

estimated 400 million (and rising) MMOG players across the globe [4], while collaborative social games such as Zynga's FarmVille suite of games have gained immense popularity, commanding the engagement of 60 million monthly users on average, an estimated 1% of the world population, and more than 20% of Facebook users [5,6]. Gonzalès et al. [5] have proposed to deploy the multiplayer collaborative aspect of social games for improving scientific models of land-use. The use of collaborative systems has yielded major improvements in, for instance, reducing replications within systems [3,7]. Whilst games have been used widely as a paradigm for learning and education [8,9], recent studies reveal that whilst computer games in general have the potential to enhance learning interest and increase motivation [10–12], non-collaborative games may yield a negative impact on learning outcomes and self-alienation [10,13–16]. These studies highlight the importance of collaborative games and their strength in comparison to the more traditional non-collaborative games. Further studies reveal that collaborative games have the potential to improve player score over time on the one hand and increase game engagement on the other [17,18]. In a further study, Silva et al. [19] illustrate that collaborative games can even support and improve social interactions amongst players with autism.

The aim of this paper is to present the key ingredients of collaborative decision making games as reported by the research literature and deploy these in the development of an add-on to open MMOGs and Social Network games. The rest of the paper is organised as follows. Section 2 undertakes a literature review of collaborative decision making games with the goal of unravelling what they regard as their key ingredients. Section 3 presents the add-on framework to open MMOGs and Social Network games, designed to include all key ingredients with a view of breeding collaborative decision support into these games. Section 4 evaluates the add-on framework's performance using Behaviour-Driven Development (BDD), and Section 5 concludes.

2. Collaborative Decision Making Games in the Literature

In this section we have collated the most prevalent components reported in the research literature that underline collaborative decision making games, namely: personalisation, team matching, non-optimal decision making, leading, decisiveness index, scoring, levelling, and multiple stages. Whilst the benefits of these components have been extolled either individually or in smaller subsets by researchers, there has not been an attempt thus far to consider all of these together in the same game design or implementation. This is the underlining motivation behind this research.

2.1 Personalisation

Kim et al. [20] present a model that links game personalisation with game enjoyment. They have performed two separate studies using two different types of personalisation, i.e. functional versus aesthetic. Their results indicate a strong link between game personalisation and game enjoyment and explain this concept as feelings of autonomy and control. Naudet et al. [21] personalise an IT-driven museum visit based on an estimation of visitors' cognitive profiles and gaming behaviours on Facebook. They argue that Facebook provides significant inputs for user profiling and also enables users to share their museum visit experiences on their social network. Karadimitriou and Roussou [22] evaluate players' learning and fun informally in an interaction environment. The design of the game supports sustained motivation and strengthens the sense of accomplishment. This also contributes to dividing the goals into different levels, an essential characteristic of successful game design. Researchers argue that a comprehensive player model should include player skills, behaviour, and socio-personal information such as aggression [23–25]. Peer reputation is based on prior individual player experiences [26].

2.2 Team matching

Researchers have used a linear system for the dynamics of information types to formulate optimal information exchange over time as a control problem. The problem is then extended to represent social structure in teams as a regulator of the amount and type of information that members

exchange [2]. The team dynamics in a collaborative environment is considered a major contributing factor to the decision making process and therefore the enjoyment of the game [27].

2.3 Non-optimal decision making

The Stackelberg games [28–30] are typical of decision making games. Since Murphy's [31] Administration Decision Game developed to teach quantitative thinking and planning, researchers have been designing decision making tutoring games [32–34] whose context is usually real life. Brinkman [35] presents such a game aimed at corporate decision-making and whistle-blowing that allows students to experience managerial decision making.

2.4 Leading

Pita [36] presents several algorithms for security games deployed in airports and in air marshals' services. Stackelberg games [28,29] are considered the natural choice for oligopolistic markets and security domains and are the backbone of security systems such as ARMOR, IRIS, and GUARDS whose decision aim is resource allocation. Stackelberg games are also widely used in supply chain management [37,38]. In Stackelberg games a leader commits to a strategy and their followers make their own decisions having knowledge of the leader strategy. On the topic of the leader, Momo Kenfack et al. [39] also point out that a strategy set out by a leader might not be followed by a group of followers, as they may choose to ignore the strategy and make a decision based on their own logic and rationality. Various studies [40–43] in social choice indicate that a leadership role would affect social preferences. These studies show that the leader can go as far as to define social preference and thus steer a collaborative decision towards their vote [40]. Higgins [44] shows how a dominant employer choice can influence social preference and steer individual decisions to match their own choice.

2.5 Decisiveness index

Tao et al. [45] consider the weighting approach for collaborative decision making using fuzzy linguistic preference reactions. They build a group decision error matrix to reflect the deviations of all decision makers with given initial weighting vectors. They then design an iterative algorithm to lower the sum of the decision error so that a final convergence result may be obtained. Bouzarour-Amokrane et al. [46] suggest a similar approach, where they introduce a decisiveness index, which they refer to as confidence degree. Their decisiveness index can be positive or negative and it is formed by the confidence degree (opinion) that other members of the team have assigned to an actor based on their decisions and decision making behaviour. Binary voting systems [47,48] are among the most widely used collaborative decision making games with their decisiveness index [47][49]. Some researchers use games in performance evaluation and benchmarking. Wu et al. [50] deploy a Nash bargaining game [51] to improve cross-efficiency evaluation whereby each decision making unit is an independent player and the bargaining solution between CCR efficiency and cross-efficiency is achieved using Nash bargaining. Weighted majority, a social choice method, [52] combines weighted decisions into a group decision [53]. Rubin and Watson [54] use weighted majority for combining the decisions of multiple experts into an automated poker player. They use a series of expert imitators for limit and no-limit Texas Hold 'Em and investigate two separate approaches for combining their decisions in an attempt to improve performance. They demonstrate that combining decisions via weighted majority voting in the limit variation produces better results than any one expert alone or simple majority voting.

2.6 Scoring

PackPlay [55] leverages a community of distributed web users in order to build semantically-rich annotated corpora generated from two collaborative games: "Entity Discovery" and "Name That Entity". Whilst the Entity Discovery game has a set score for every answer, Name That Entity has a sliding scale. Whitaker et al. [56] present the game Heuristica that uses a set of problem diagnosis

and repair or observation scenarios on a space station in order to teach how to recognize and mitigate cognitive biases. They score players on three different aspects of the game; this allows them to monitor the players' progress through different aspects of the game and encourage players to collaborate in order to improve their scores on aspects where they are weaker. Bonnet et al. [57] present BrainArena a multiuser videogame in which two users play a simple football game by means of two brain-computer interfaces (BCI) either collaboratively to score goals or against each other.

2.7 Levelling

Some researchers have developed simulation games aimed at different levels of enterprise decision making [58]. These researchers consider that breaking the decision making process into multiple discrete levels allows for better decision making as well as making the game more enjoyable [22,59]. Ben-Zvi [60] promotes business simulation games for Information Systems teaching whilst [61] promote operations management decision making. Douma et al. [62] argue that despite the benefits of multi-agent systems very few have been deployed in practice. They then present a real-time multi-player simulation game to address the barge handling problem in the Port of Rotterdam. Katsaliaki et al. [63] present a game approach in facilitating decision making for perishable products and argue that playing the various game levels would improve understanding and performance.

2.8 Multiple stages

Pato and Delgado-Mata [64] experiment with dynamic adjustment of a set of game variables in order to induce new stages and to regulate difficulty. Their algorithm operates during game play and makes the changes with a predetermined rate of frequency in order to increase game enjoyment. Chen and Wang [65] propose an algorithm for multi-facility work order formulated as a stochastic non-autonomous Lotka–Volterra difference game with a closed loop control scheme.

2.9 Conclusions

In order to accommodate **personalisation**, we need to monitor and record players' every action, all their preferences and behaviour and all the events that occur in the game. We also need to profile the players both at player and team levels to enable evaluation and personalisation at both levels. To help the players improve and make better decisions, they need to be divided into dynamically created teams which match the player's level of proficiency. **Team matching** helps the players improve and progress and also avoids situations where some of the team members are significantly better than others as this would discourage less proficient players from engaging. Further, the proficiency of players in a team should vary so that they do not lose encouragement to improve their performance. As part of encouraging the players to improve and to reduce **non-optimal decision making**, each team needs a **leader**, preferably one that evolves naturally during game play. The leader of a team is the strongest and most proficient player in the team and, therefore, a player would occupy or vacate that position based on their performance. A team should be notified of leader decisions as these should serve as a guideline for either making good decisions or non-viable strategies, the latter through demonstration of the consequences of making a bad decision. Using a weighted majority function allows the combination of decisions as well as encouraging players to raise the weight of their decision in the team. Using the "**decisiveness index**" as a decision weight of the competence level and performance of each player, it can be a determining factor for each player's decision in the process of creating the final collective decision. A **scoring** system should encourage players not only to continue playing but to guide each other into making better decisions. Players would score positively or negatively based on the outcome of the collective decision and each player will be given a decisiveness index based on the final collective decision. Individual decisions should also determine a player score. In order to monitor the player progress, a **levelling** system needs to be implemented that increases the game difficulty as players progress to higher levels. Levelling provides a sense of accomplishment and reward. A game should include **multiple**

stages to progress through, e.g. through characterisation, which they may change, upgrade and update as they progress.

3. An Architecture for Collaborative Decision Making Games

In this section, we turn the ingredients outlined in the previous section into functional modules which we include in an architecture that aims to support collaboration in decision making games.

3.1 The Lu-Lu Add-on Architecture

The architecture presented in Figure 1 is designed to support collaborative decision making using the functional ingredients presented in the previous section. It builds upon our previous work in collaboration and personalisation in digital games [66,67]. The architecture amalgamates individual player decisions to a team decision. In order to allow users to improve, each player's influence on the team decision is based on their ability. The player's decisiveness index is calculated according to their standing within the team, which, in turn, is based on score, level and loyalty. The architecture caters for team profiling in order to ensure the teams are well balanced to encourage players to progress. The Lu-Lu architecture consists of two zones: the Lu-Lu *profiling zone* and the Lu-Lu *adjustment zone*. The game connects to the adjustment zone which interacts directly both with the game and the profiling zone. All the information about the players and teams are stored in the profiling zone and retrieved by the adjustment zone when needed. The framework is designed so that the profiling zone is independent of the game, therefore, it will not change for different games. Alternatively, the adjustment zone needs to be re-set for each game. Such design allows for portability of player models, enabling players to use their profile across multiple games. As proof of concept we have implemented the Lu-Lu architecture for the Supply Chain or Beer Game [68] that follows the process depicted in Figure 2. The player goals are to minimize total inventory cost when ordering from their immediate supplier and to maximise profit. How much to order is based on own prediction of future demand by customers [69,70]. Each player will start at one of the stages.

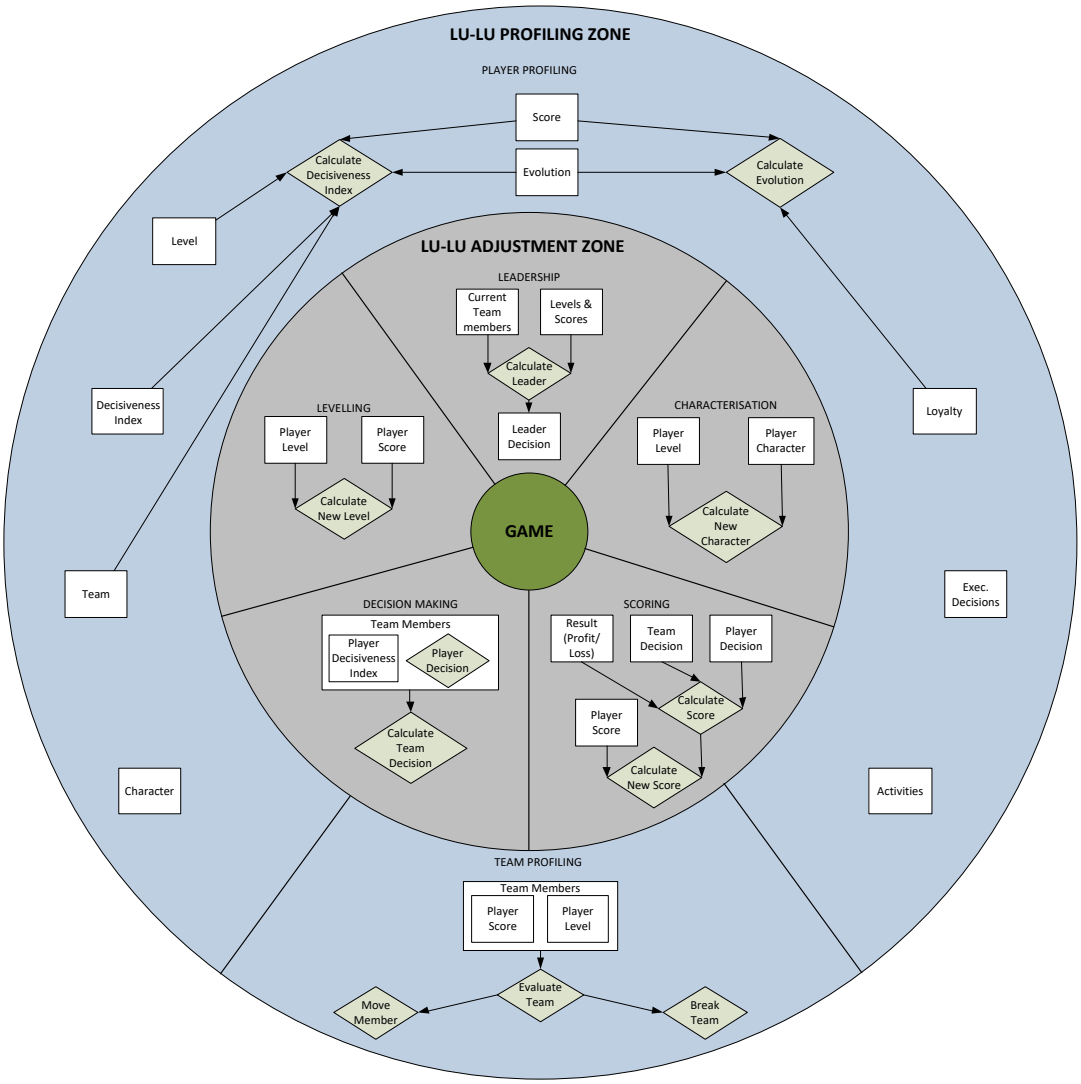

Figure 1: Lu-Lu's Architecture

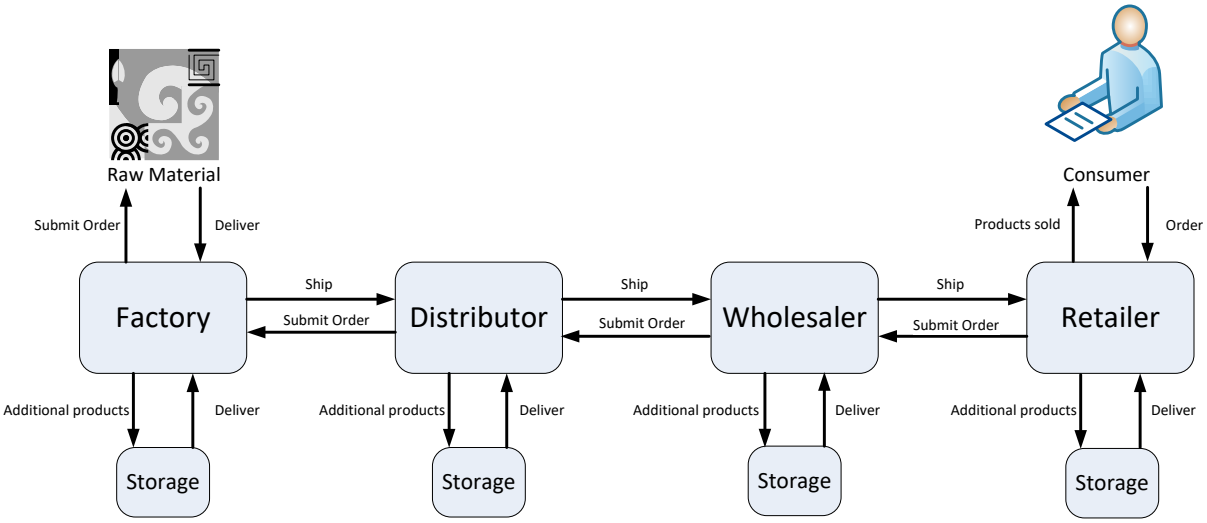

Figure 2: The Original Supply Chain Game

The Beer Game is an asymmetric-stochastic type of game whose asymmetry influences the manufacturer team equilibrium and uncertainty influences the retailer team equilibrium. Whilst the final game outcome will always be either profit or loss, with this type of game we cannot predetermine possible options for a team or its players. Furthermore, any form of advice with regards to team or player actions other than assessing individual and team performance and balancing of teams will introduce an exogenous bias in the bullwhip effect.

3.2 Lu-Lu Profiling Zone

The profiling zone stores both the player and team models, which hold all information about a player, their team and all their game actions. The profiling zone also stores an evaluation of a player's decisiveness index in the game, state of evolution, and team structure in order to ensure a balanced distribution of players. Player profiling stores the player model, which records a player's in-play stats: score, level, character, team, loyalty, activities, (their) decisiveness index, and evolution. Decisiveness index and evolution are calculated for each player before being recorded in their model while the rest of the player information is sourced from the adjustment zone and team profiling. Player profiling contributes towards personalisation. The player model is modelled in MPEG-7 through various Description Schemes (DSs). Figure 3 shows how the player profile maps to the player model. Figure 4 shows how personal information such as character, team, and level are modelled.

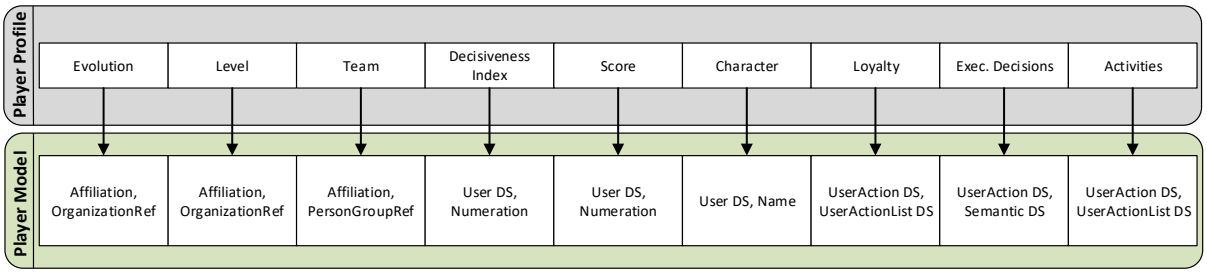

Figure 3: Player profile mapping to MPEG-7 player model

Character holds information about character roles that a player has assumed in the game as well as their personal and login information. These are stored under Name. Score records the player's

respective score in the game which is stored in Numeration under the score title. Player's decisiveness index is stored in Numeration under decisiveness index title. Player's team affiliation is stored through PersonGroupRef with an XPath reference to the teams' model. The player's level and evolution are modelled using OrganizationRef with an XPath reference to the respective models for more detail on the level and evolution. Loyalty records the information about the player's gaming habits; it includes every login and logoff action, time spent in the game, the progress pace and regularity of game play. Executed (exec.) decisions stores outcomes of decisions made by a player with a reference to the decision model. Activities stores all the actions a player makes during game play excluding decisions and login-logoff actions. This includes movements, achievements and preferences. Figure 5 provides samples of various actions being stored. The figure also illustrates a sample of action definition for the "UserLogin" action as well as a sample of a decision model.

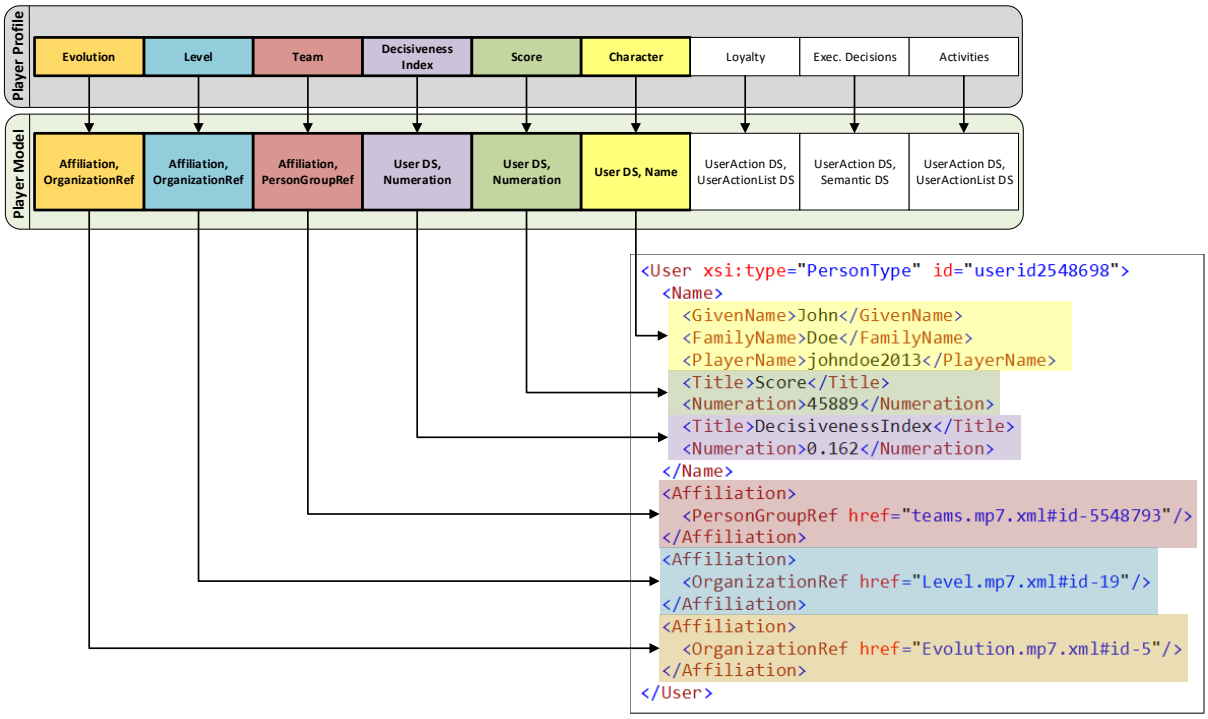

Figure 4: Player profile mapping to MPEG-7 player model: Player character, level and team

Evolution allows tracking and classification of player progress based on their performance using their score, loyalty and current evolution state. Decisiveness index shows how effective a player's decision

would be informing the team decision. This is also an indication of the player's competence in the game. Decisiveness index also affects players' share of the loss/profit as a result of the team decision. Decisiveness index is calculated using the player's level, score, evolution and current decisiveness index. Figure 6 shows that the player's score compared to that of the team members is the first source for calculating the decisiveness index followed by level. Players within a team are matched based on loyalty and rank.

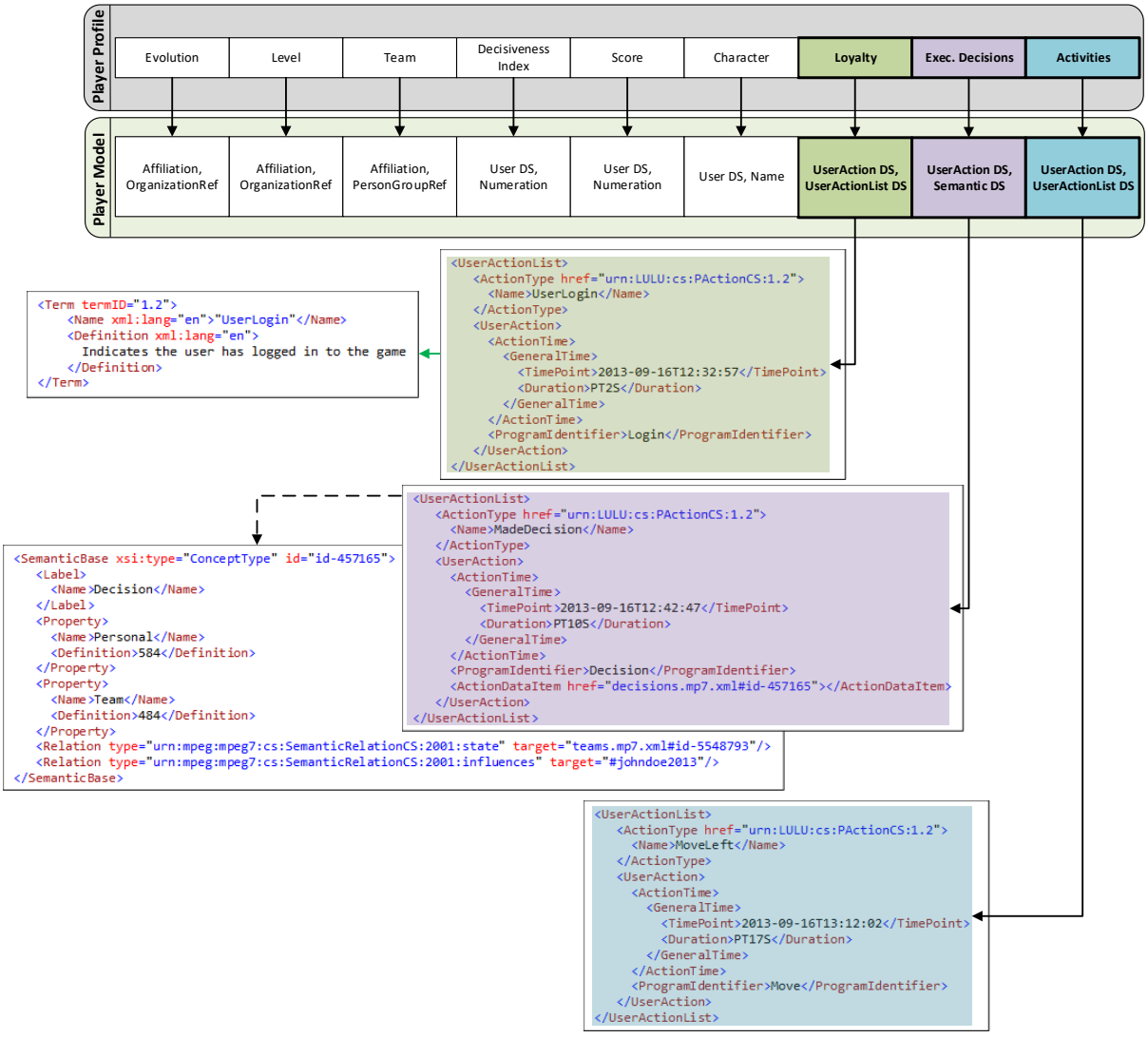

Figure 5: Player profile mapping to MPEG-7 player model: Player Loyalty, Executed Decisions and Activities

```
function CalculateDecisivenessIndex(myPlayer)
{    teamMembers = GetAllPlayersTeam(t); baseScoreLevel = 0;
     foreach (player in teamMembers)
         {baseScoreLevel = baseScoreLevel + player.Score * player.Level;}
     loyaltyRank = RankTeamPlayerLoyalty();
     decisivenessIndex = loyaltyRank x (myPlayer.Score x myPlayer.Level) / baseScoreLevel;
 }
```

Figure 6: The decisiveness index function

Equation 1 calculates the level of player activity. An active player is a member of the active set, which means that the player has played 75% or more since first joining the game and his recent inactivity has been less than the active limit. An inactive player is a player that has not played more than 50% since joining and his recent absence has been more than the inactivity limit. A semi-active player's activity lies between the active and inactive limits. The player's activity is divided into three categories: a player with a good score has a score higher than the upper limit, an unsatisfactory score is lower than the lower limit, and a satisfactory score lies somewhere between the two limits. (This requires as input the player score which is calculated in Equation 4.) Team profiling uses player scores to resolve situations of significantly uneven player competences. This may result in moving a player to another team or breaking up a team. Figure 7 presents the team profiling process.

$$Active = Player \in \{P | P \in ActiveSet\}$$

$$ActiveSet = \{P | D_{NotPlayed} = 0\} \cup \{P | D_{LastPlayed} < ActiveLimit \ \& \ (D_{Played}/D_{Total}) > 0.75\}$$

$$Inactive = Player \in \{P | P \in InactiveSet\}$$

$$InactiveSet = \{P | D_{LastPlayed} \geq InactiveLimit \ \& \ (D_{Played}/D_{Total}) < 0.5\}$$

$$SemiActive = Player \in \{P | P \in SemiActiveSet\}$$

$$SemiActiveSet = \{P | ActiveLimit \leq D_{LastPlayed} < InactiveLimit \ \& \ 0.5 < (D_{Played}/D_{Total}) < 0.75\}$$

$$Activity\ (Score_{Player}) = \begin{cases} Good\ Score & Score_{Player} \geq UpperLimit \\ Satisfactory\ Score & LowerLimit < Score_{Player} < UpperLimit \\ Unsatisfactory\ Score & Score_{Player} \leq LowerLimit \end{cases}$$

Equation 1: Level of player activity

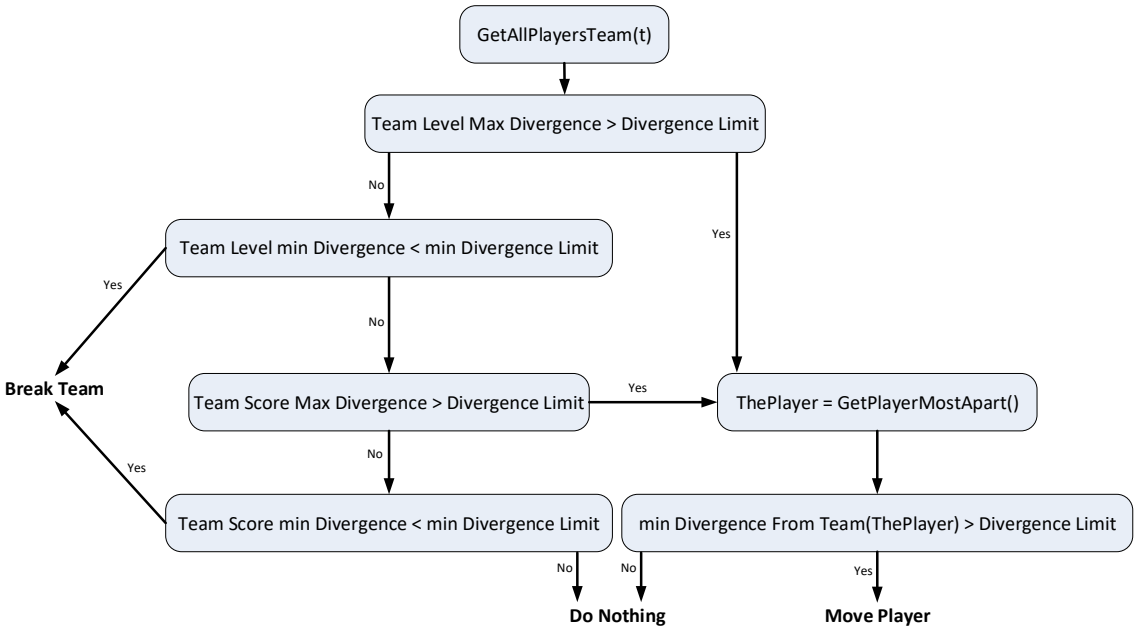

Figure 7: Team profiling

3.3 Lu-Lu Adjustment Zone

The adjustment zone communicates directly with the game. It consists of five areas: leadership, decision making, scoring, levelling and characterisation. Leadership selects a team leader among players with the highest decisiveness index within the team by considering the score and level of all players in the team. The player with the highest combination of level and score is announced as the leader. This is a dynamic selection process that may change after every decision. Once selected, a leader can send messages to team members to advise them on game strategy. Players are made aware of their leader's decisions. In the Supply Chain game, at the beginning of each round, when an order is forwarded to a team, the leader of each team submits their order before the rest of the players within the team. All other players are then notified of their leader's decision, hence, they will issue their order based on how much stock the leader has ordered. Figure 8(a) illustrates how the Leader title is added to the team profile for a team member whilst Figure 8(b) shows how decisions are stored in MPEG-7. Whilst players will be scored based on their individual decisions, the decisions of the players in a team are amalgamated into a team decision using an adaptation of weighted majority social choice function [52,71] as described by Kolter and Maloof [72] and Yager [73]. This provides the players with the opportunity to learn from each other, collaborating in decision making

at a level higher than just the game. The team decision is derived using

$$Decisions = \langle d_1 \quad \dots \quad d_n \rangle, UniqueDecisions = \langle d_1 \quad \dots \quad d_m \rangle, DW = \langle w_1 \quad \dots \quad w_m \rangle$$

$$w_i = \sum DecisivenessIndex_{Player\ j} \mid Decision_{Player\ i} = d_i \wedge d_i \in UniqueDecisions$$

$$Decision_{Team} = d_i \longrightarrow \begin{cases} w_i = Max(DW) & \exists\, Max(DW) \\ w_i = Max(DW) \wedge d_i = Decision_{Leader} & \exists!\, Max(DW) \end{cases}$$

Equation 2.

```
<PersonGroup xsi:type="PersonGroupType" id="id-5548793">
   <Member xsi:type="PersonType" id="userid2548698">
      <Name>
         <GivenName>John</GivenName>
         <FamilyName>Doe</FamilyName>
         <Title>Leader</Title>
         <PlayerName>johndoe2013</PlayerName>
      </Name>
      <NameTerm href="johndoe2013.mp7.xml">
      </NameTerm>
   </Member>
</PersonGroup>
```

(a)

```
<SemanticBase xsi:type="ConceptType" id="id-457165">
   <Label>
      <Name>Decision</Name>
   </Label>
   <Property>
      <Name>Personal</Name>
      <Definition>584</Definition>
   </Property>
   <Property>
      <Name>Team</Name>
      <Definition>484</Definition>
   </Property>
   <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:state"
      target="teams.mp7.xml#id-5548793"/>
   <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:decisivenessIndex"
      target="#johndoe2013"/>
</SemanticBase>
```

(b)

Figure 8: Modelling of (a) team leadership and (b) decisions

$$Decisions = \langle d_1 \quad \dots \quad d_n \rangle, UniqueDecisions = \langle d_1 \quad \dots \quad d_m \rangle, DW = \langle w_1 \quad \dots \quad w_m \rangle$$

$$w_i = \sum DecisivenessIndex_{Player\ j} \mid Decision_{Player\ i} = d_i \wedge d_i \in UniqueDecisions$$

$$Decision_{Team} = d_i \longrightarrow \begin{cases} w_i = Max(DW) & \exists\, Max(DW) \\ w_i = Max(DW) \wedge d_i = Decision_{Leader} & \exists!\, Max(DW) \end{cases}$$

Equation 2: Deriving a team decision

Decisions represents all decisions submitted by players and is normalised by removing duplicate decisions in *UniqueDecisions*. *DW* (Decision Weights) represents the weight of each decision which is sourced by combining the *DecisivenessIndex* of players who have voted for each decision. With these vectors, team decision is formed by choosing the decision with the most associated weight; if maximum is not unique then the option backed by the leader is chosen. In the supply chain game the usual decision involves placing an order. The players within a team will receive an order and they need to place an order based on the order they have received. They need to strike a balance

between the cost of storage and a penalty for under ordering when making the decision on the amount of products they will be ordering. Figure 9(a) illustrates the receipt of an order; Figure 9(b) illustrates placing an order; Figure 9(c) defines an IssuingOrder action. Once all the team members submit their orders, the team's collective order is calculated using

$$Decisions = \langle d_1 \quad \dots \quad d_n \rangle, UniqueDecisions = \langle d_1 \quad \dots \quad d_m \rangle, DW = \langle w_1 \quad \dots \quad w_m \rangle$$

$$w_i = \sum DecisivenessIndex_{Player\ j} \mid Decision_{Player\ i} = d_i \wedge d_i \in UniqueDecisions$$

$$Decision_{Team} = d_i \longrightarrow \begin{cases} w_i = Max(DW) & \exists\, Max(DW) \\ w_i = Max(DW) \wedge d_i = Decision_{Leader} & \exists!\, Max(DW) \end{cases}$$

Equation 2.

```
<UserActionList>
   <ActionType href="urn:LULU:cs:PActionCS:1.2">
     <Name>ReceivingOrder</Name>
   </ActionType>
   <UserAction>
     <ActionTime>
       <GeneralTime>
         <TimePoint>2014-05-16T12:32:47</TimePoint>
         <Duration>PT10S</Duration>
       </GeneralTime>
     </ActionTime>
     <ProgramIdentifier>Order</ProgramIdentifier>
     <ActionDataItem href="scg.mp7.xml#Wholesaler"></ActionDataItem>
   </UserAction>
</UserActionList>
```

(a)

```
<UserActionList>
   <ActionType href="urn:LULU:cs:PActionCS:1.2">
     <Name>IssuingOrder</Name>
   </ActionType>
   <UserAction>
     <ActionTime>
       <GeneralTime>
         <TimePoint>2014-05-16T12:32:57</TimePoint>
         <Duration>PT10S</Duration>
       </GeneralTime>
     </ActionTime>
     <ProgramIdentifier>Order</ProgramIdentifier>
     <ActionDataItem href="scg.mp7.xml#Factory"></ActionDataItem>
     <ActionDataItem href="decisions.mp7.xml#id-457165"></ActionDataItem>
   </UserAction>
</UserActionList>
```

(b)

```
<Term termID="1.2">
    <Name xml:lang="en">"IssuingOrder"</Name>
    <Definition xml:lang="en">
      Indicates the player has issued a new order
    </Definition>
</Term>
```

(c)

Figure 9: Modelling of the receipt (a) and issuing (b, c) of orders.

Scoring records the players' gameplay and determines and feeds the player's score back into the game. The result (profit/loss) of each team's decision affects the score of each player based on their decisiveness index on the team decision as well as their personal decision. A player who has made a decision that resulted in being more profitable than the team decision scores more points than a player whose decision was not more profitable than the team decision. Similarly, a player whose decision did not make a great impact on the team decision does not score higher than a player who

had made significant impact on the team decision. The player score is calculated using

$$RoundScore_{Player_i} = DecisivenessIndex_{Player_i} \times (Result_{Decision_{Team}} + Result_{Decision_{Player_i}})/2$$

$$Score_{Player_i} = CurrentScore_{Player_i} + RoundScore_{Player_i}$$

Equation 3.

$$RoundScore_{Player_i} = DecisivenessIndex_{Player_i} \times (Result_{Decision_{Team}} + Result_{Decision_{Player_i}})/2$$

$$Score_{Player_i} = CurrentScore_{Player_i} + RoundScore_{Player_i}$$

Equation 3: Calculating a player score

Using this equation for calculating the round score ensures that the expected utility of each decision is directly affected by the decision of both the team and a player. Whilst in each round a player can either make a profit or loss or neither, loss or lack of profit is packaged as the latter. Considering that a player is aware of their leader's decision but not the outcome, both player and leader have a choice to seek making profit or loss. In such a case, the team decision would be similar to that of the leader, the player with the highest decisiveness index. This is illustrated in Figure 10.

		Leader	
		Loss	**Profit**
Player	**Loss**	X_1, Y_1	X_2, Y_2
	Profit	X_3, Y_3	X_4, Y_4

$X_1 < X_2$	$Y_1 < Y_2$	$X_1 < X_2$
$X_3 < X_4$	$Y_3 < Y_4$	$X_3 < X_4$
$X_1 < X_3$	$Y_1 < Y_3$	$Y_1 < Y_2$
$X_2 < X_4$	$Y_2 < Y_4$	$Y_3 < Y_4$

Figure 10: Profit/Loss game model

The figure shows that regardless of a leader's decision, profit strictly dominates loss for a player and equally profit strictly dominates loss for the leader regardless of players' choices. It is not rational for any player to try to make a loss intentionally. Yet it might be a strategically bad decision. Intentional bad decisions would result in loss of score in the short term and a negative score value, but in the long term this strategy would only damage the player's progress, standing and level. Lack of any gains would help discourage players from making non-optimal decisions.

In the supply chain game, the result of each round of decision making is either profit or loss. Therefore, a player's score reflects how much money they have made or lost while playing the game. As the result of each round of decision making would result in a real number, which can be positive or negative, a player's score can go down as well as up. This fluctuation in a player score can be a great indication of the player's progress or decline during game play. In the supply chain game the *Result* is calculated using Equation 4. In the current round i the products shipped would be $P_{Shipped(i)}$ which has value *V*. The products shipped in each round yield the income and profit of the round. Any product left in storage from the previous round $P_{Storage(i-1)}$ would have accumulated a cost $C_{Storage}$. If the team had under ordered products for the previous round's order by an amount $P_{order(i-1)}$ then there is a backlog cost of $C_{Backlog}$ per product. The cost of ordering more products is set as C_{Order} and the number of products on order in round i is set as $P_{Ordered(i)}$.

$$Result = V \times P_{Shipped(i)} - (C_{Storage} \times P_{Storage(i-1)} + C_{Backlog} \times P_{Order(i-1)} + C_{Order} \times P_{Ordered(i)})$$

Equation 4: Profit and Loss during round i

Levelling and characterising are interlinked; the processes are illustrated in Figure 11. Levelling uses a player's score to determine their level, which is shown in Figure 11(a). The original supply chain game does not feature levels, however, this add on introduces levelling. Even when a player's score is dropping they still retain their level achievement. Player character changes throughout the game to reflect the player progress. Players can upgrade their character when they reach a level where a new character is available to them. Figure 11(b) illustrates the characterising process. Once players have completed all four stages, they can freely swap between the four stages when they reach an upgrade level.

```
function Levelling(myPlayer)
{     levelMax = GetMaxScore(myPlayer.Level);
      if (myPlayer.Score >= levelMax)
          {return GetLevel(myPlayer.Score);}
      else
          {return myPlayer.Level;}
}
```

(a)

```
function Characterising(myPlayer)
{     levelMax = GetMaxScore(myPlayer.Level);
      if (myPlayer.Level = GetUpgradeLevel(myPlayer.Character))
          {return UpgradeCharacter(myPlayer);}
      else
          {return myPlayer.Character;}
}
```

(b)

Figure 11: Levelling (a) and characterising (b) processes.

3.4 Selection of an optimal decision in Lu-Lu

Nash's equilibrium assumes that each player's decision is an optimal response to the other players' decisions because it would maximise his or her expected outcomes. This assumes knowledge or accurate prediction of the decisions of the other players. However, knowledge of the other players' decisions might be the result of prior agreement or communication and accurate prediction of their decisions may result from common knowledge of strategies and outcomes and from optimisation behaviour. As many games may have multiple equilibriums and predictions may be incomplete or based on imperfect information, equilibrium is not a determinant criterion, and therefore we need to 'refine' this in order to obtain accurate predictions. The beer game is one of many multi-optimisation games with multiple equilibriums. Therefore, selecting an optimal D_i becomes a multi-optimisation problem (MOP) which seeks to optimise objectives $\bigcup_{d=1}^{k} O_d(D) \leq 0$ subject to limit constraints $\bigcup_{d=1}^{m} L_d(D) \leq 0$: $F(\vec{x}) = (f_1(\vec{x}).....f_k(\vec{x}))$ where $\vec{x}$ is an n-dimensional decision variable vector $(\vec{x}=x_1,....,x_n)$.

Sofokeous and Angelides [74] suggest the use of Pareto Optimality (PO) as the Nash equilibrium refinement solution and Genetic Algorithms (GAs) as the best effort implementation tool. GA returns multiple best fit solutions evolved over many generations and PO evaluates and ranks each GA generation's solutions against the set of optimisation objectives (O_i), e.g. profit and number of orders, without the need to assign weights to each one. During each generation the GA creates a population of chromosomes with each chromosome encoding a solution to the problem, e.g., the decision vector above where each chromosome is a D_i. During a chromosome verification, the GA verifies that each chromosome in a population is valid. Non-valid chromosomes are discarded and valid chromosomes are evaluated against the limit constraints. Valid chromosomes satisfy all limit constraints, e.g. cost, and the Nash equilibrium.

Valid chromosomes are passed on to the PO whose first objective is to distinguish between dominated and non-dominated chromosomes. The PO evaluates each D_i chromosome against the optimisation constraints in order to identity those that are dominated and those that are not. The PO then calculates fitness values for both the dominated and non-dominated chromosomes using a fitness function. With the PO fitness function, in contrast to that used by a GA, each non-dominated chromosome is assigned a strength value that is proportional to the number of the dominated chromosomes it weakly dominates because it is considered as good as the number it weakly dominates. If the chromosome population (P) is expressed as $P = P_E \bigcup P_I$, where P_E contains the non-dominated and P_I contains the dominated chromosomes, the strength value of a non-dominated C_i is expressed as $S(\mathrm{C_i}) = \frac{|\{\mathrm{C_j} \mid \mathrm{C_j} \in P_I \ \& \ \mathrm{C_i} \preceq \mathrm{C_j}\}|}{|P_I| + 1}$, where $\mathrm{C_i} \in P_E$. The strength value is then used for calculating the fitness value of dominated chromosomes. Each dominated chromosome C_j is assigned a value that is equal to the sum of strength values of the non-dominated chromosomes that weakly dominate C_j, i.e. $F(C_j) = 1 + \sum_{i=1}^{|P_E|} \begin{cases} S(C_i), if \ C_i \preceq C_j \\ 0 \quad , otherwise \end{cases}$, where $C_i \in P_E$.

A dominates B, denoted as $A \prec B$, iff $O_i(A) \le O_i(B), \forall i = 1, \ldots, D$ and $O_i(A) < O_i(B)$ for some *i* and A weakly dominates B, denoted $A \preceq B$, iff $O_i(A) \le O_i(B), \ \forall i = 1, \ldots, D$, where *D* is the number of optimisation constraints.

When the termination criterion is satisfied, i.e. the number of evolutions, the GA stops and a non-dominated chromosome is selected, usually randomly, to depict the optimal D_i. If the termination criterion is not satisfied, it proceeds to select chromosomes for reproduction using a fitness-based selection operator, such as roulette, that chooses fitter chromosomes from an existing population. Crossover and mutation operations are then performed on the selected chromosomes to produce new offspring chromosomes. Crossover and mutation are controlled by the corresponding rates that define the probability that two chromosomes will swap their parts, and the probability that a gene

will be altered within a chromosome. During successive crossovers and mutations, non-valid offspring chromosomes may be produced which will be discarded during verification. The outcome is a Pareto Front of optimal D_i (C_j) all of which satisfy: $f : P'_E = \{C_1, C_2, \dots, C_n\} \longrightarrow C_j \in P'_E$ where *n* is the number of chromosomes.

Game theorists have long considered that the distribution of optimal solutions on a Pareto Front may suggest the existence of a "best-fit" optimal D_i. Sofokleous and Angelides [75] refine further the selection process of an optimal D_i by putting the Pareto distribution though a Self-Organizing Neural Network (SONN) in order to rank the non-dominated chromosomes. The SONN calculates Euclidean Distances (ED) between each chromosome, a measure that calculates the similarity between two chromosomes: $ED(C_j, C_i) = \sqrt{(O_1(C_j) - O_1(C_i))^2 + \dots + (O_m(C_j) - O_m(C_i))^2}$, where $O_k(C_z)$ is the evaluation result of O_k using the chromosome C_z values. The closer C_j is to C_i, the more likely is that C_j will result in the same satisfaction as C_i. The fitness value of each chromosome is then re-calculated and the one with the highest $f(C_i)$ value is chosen as the best-fit optimal D_i:

$$f(C_i) = \left(10 + \sum_{C_j \in P'_E} \frac{UserRating}{1 + ED(C_j, C_i)}\right) / (1 + |P'_E|)$$

3.5 Lu-Lu Add-On to a game

Lu-Lu's adjustment zone can be customised in order to interface with a game's API for information exchange with the game. How Lu-Lu's adjustment zone can be customised in order to enable information exchange with a game's API is demonstrated through the Facebook[2] games platform. The popularity of social networking games has risen due to their social nature, which in turn yields a great potential for collaborative decision making games. There are numerous social networks, such as Kakao[3], that encourage social gaming; however, Facebook is one of the most popular with possibly the most number of games hosted. Another advantage with using Facebook games is the

[2] http://www.facebook.com/
[3] http://www.kakao.com/

variety of APIs in use, e.g. Graph API and Scores API, which enables exchange of information of fine granularity. Typical information that Lu-Lu needs to extract from the game is: name of the player, user ID, score, and activities. Activities include a player's logbook, decisions, movement, mouse clicking, and behavioural traits. Not all games would present the information in the necessary format; therefore, once access to the game is established through its API, Lu-Lu's adjustment zone would retrieve the necessary information and then convert it to the necessary format. The retrieval and conversion process is illustrated using Facebook APIs. Facebook's Graph API enables access to user information with consent. Once a Facebook user has agreed to allow access to their profile an access token is generated and the user-player profile can be accessed using the API. The player profile is encapsulated in a node called /{user-id} that includes a number of fields Lu-Lu needs access to. The id, first_name and last_name fields map directly to that of Lu-Lu's user ID, first name and surname.

Figure 12 shows the use of the Graph API. Figure 12(a) illustrates access to the /{user-id} node, while Figure 12(b) shows retrieval of a player's first name. Facebook includes Scores API as part of Graph API. Access to a player's score is through one of the edges of the /{user-id} node, i.e. /{user-id}/scores. When accessing this edge, the API would return an array of objects that make up the player's score. The /{user-id} node in Graph API includes an edge named /{user-id}/activities which provides access to user activities. Lu-Lu needs access to all user activities that take place, including the logbook. Graph API may return a list of user activities, time-stamped in date/time format. The retrieval of user activities by the Graph API is shown in Figure 12(c).

```
FB.api(
    "/{user-id}",
    function (response) {
      if (response && !response.error) {
        /* We have access to player profile */
        Lu-Lu.Convert(response); } });
```

(a)

```
FB.api(
    "/{user-id}?fields=first_name",
    function (response) {
        var PlayerFirstName = response.first_name; });
```

(b)

```
FB.api(
    "/{user-id}/activities",
    function (response) {
        if (response && !response.error) {
            /* We have access to player activities */
            Lu-Lu.Convert(response); } } );
```

(c)

Figure 12: Using the Facebook Graph API: (a) accessing /{user-id} node, (b) retrieving a player's first name, and (c) retrieving player activities

4. Assessing performance with Behaviour Driven Development

Behaviour-Driven Development (BDD) [76] is a software development process with a focus on defining specifications for the behaviour of the targeting system, thus concentrating all development efforts on delivering of such behaviour, i.e. what a system should do, specified so that both the developers and the domain experts would understand. BDD enables users to describe the required system behaviour as executable user stories and the acceptance criteria as executable scenarios attached to these [77,78]. The user stories should have a clear and explicit title, a short introductory narrative section that specifies Who (**As a**), Which (**I want**) and What (**So that**) and acceptance criteria or scenarios which would describe each specific case of the narrative [76]. Once the scenarios are created, the steps for testing the scenarios are defined. These steps translate to methods that use existing code to test if the scenarios have been successful.

Narrative: The system architecture nurtures player decision making in a game that is enjoyable.

Formal representation in BDD of the narrative can be presented as:

So that the game is enjoyable, **As a** researcher, **I want** the players to improve in decision making.

Four scenarios have been devised for the above narrative. Figure 13 illustrates the four scenarios. The BDD tests for this research have been developed using SpecFlow Cucumber for .NET and NUnit testing framework [79] in Microsoft Visual Studio. The system was made accessible to a group of 45 players, 20 females and 25 males aged 21-46 with different professional backgrounds. They were asked to play the game for 30 days and all behaviour in player models were used as data for BDD scenarios. During testing of scenario 1, players would play the game less frequently but when close to a new level or stage limit, the frequency would increase and reach a peak.

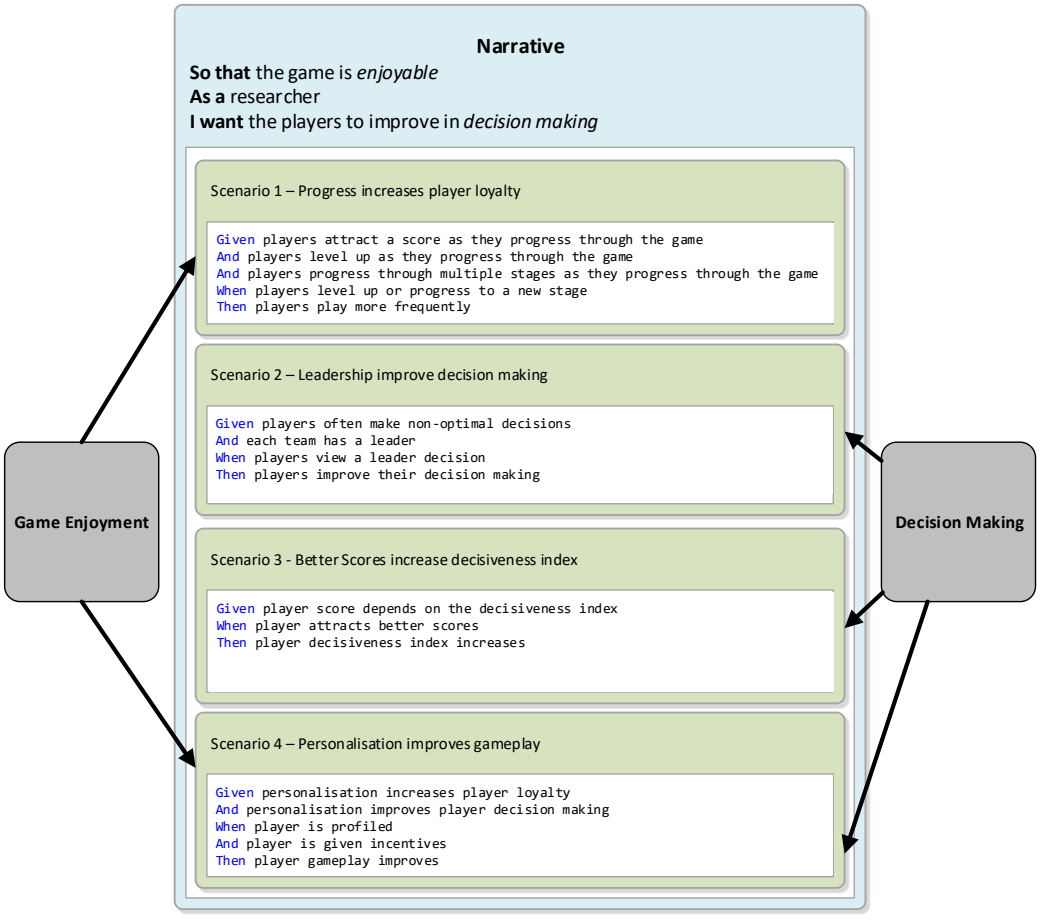

Figure 13: The BDD Scenarios

Figure 14 shows a typical player. With scenario 2, players were monitored over 50 decisions they made with their decision compared to the team leader's decision. The team leader was changed based on the team performance. Some teams replaced members or new teams were created through team matching. The results revealed that over time player decisions converged towards the team leader's decision and vice versa.

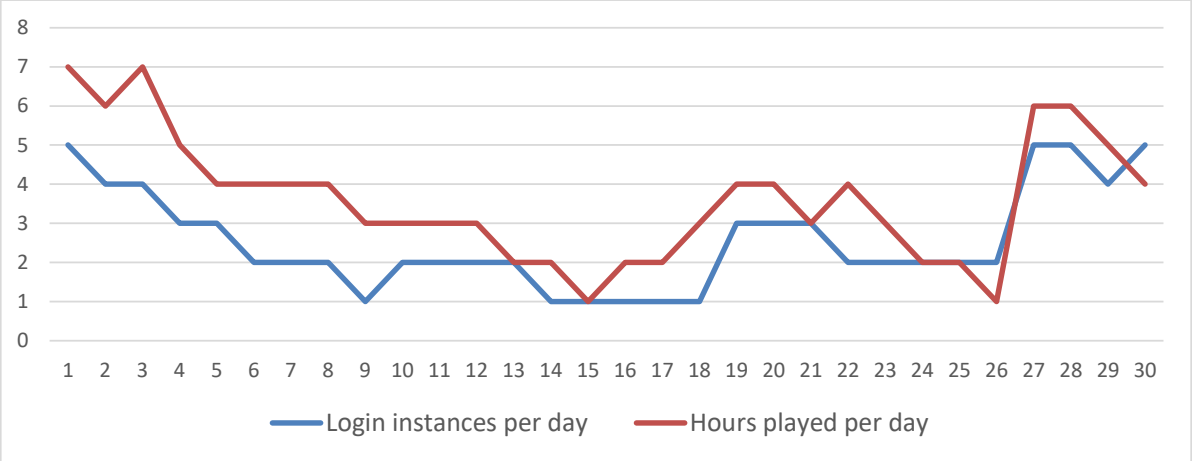

Figure 14: Player loyalty increases when levelling up and assuming new characters

Figure 15 shows that player decisions grow increasingly similar to that of team leaders. With scenario 3, as player scores increase during a game, the player decisiveness index increases and that affords a bigger share in the total team profit. This scenario was tested by monitoring players over 30 decisions.

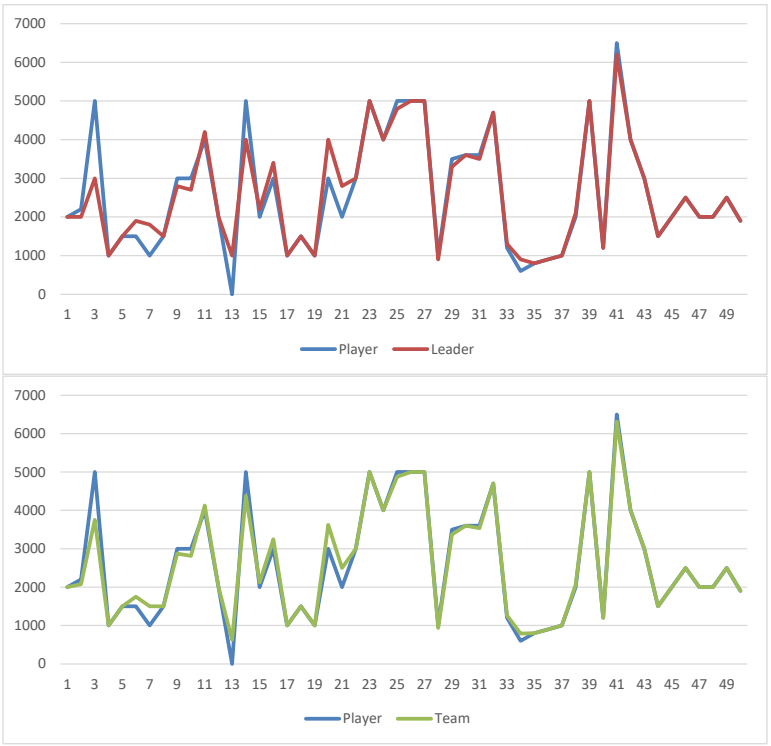

Figure 15: Player decisions matching team Leader and team decisions

Figure 16 shows the increase in decisiveness index as player scores grow. There is a sharp increase in the player's decisiveness index on decision 16 with a move to a new team. With scenario 4, players who were either less active or not scoring well were chosen to participate. These players were incentivised such as with an opportunity to attract double scores or prompted to adopt the leader decision making pattern in order to improve their scores. Once incentivised, less active players began to engage regularly, seizing the opportunity to attain a higher standing within their teams or follow in their leader's footsteps.

Figure 16: A player Score and Decisiveness Index

5. Concluding Discussion

This paper collates the key ingredients reported in research literature on collaborative decision making games: personalisation, team matching, non-optimal decision making, leading, decisiveness index, scoring, levelling and multiple stages. The set is incorporated in the development of Lu-Lu's architecture as an add-on to open MMOGs and Social Network games and Lu-Lu's deployment is demonstrated in the beer game [68] using Facebook's games platform. A BDD performance analysis reveals successful integration of all key ingredients in Lu-Lu and an autonomous behaviour that improves player enjoyment and decision making.

References

[1] A. Al-Dhanhani, R. Mizouni, H. Otrok, A. Al-Rubaie, A game theoretical model for collaborative groups in social applications, Expert Systems with Applications. 41 (2014) 5056–5065.

[2] S.D. Silver, Designing technology for managing the information exchange of decision making teams, Decision Support Systems. 61 (2014) 136–146.

[3] M.S. Hossain, M. Masud, G. Muhammad, M. Rawashdeh, M. Mehedi Hassan, Automated and user involved data synchronization in collaborative e-health environments, Computers in Human Behavior. 30 (2014) 485–490.

[4] L. Lu, C. Shen, D. Williams, Friending your way up the ladder: Connecting massive multiplayer

online game behaviors with offline leadership, Computers in Human Behavior. 35 (2014) 54–60.

[5] R. Gonzalès, J. a. Cardille, L. Parrott, Agent-based land-use models and farming games on the social web—Fertile ground for a collaborative future?, Ecological Informatics. 15 (2013) 14–21.

[6] M.T. Wyman, Making Great Games: An Insider's Guide to Designing and Developing the World's Greatest Games, Focal Press, Boston, 2013.

[7] M. Masud, I. Kiringa, H. Ural, Update processing in instance-mapped P2p data sharing systems, International Journal of Cooperative Information Systems. 18 (2009) 339–379.

[8] N.E. Cagiltay, Teaching software engineering by means of computer-game development: challenges and opportunities, British Journal of Educational Technology. 38 (2007) 405–415.

[9] H. Tüzün, M. Yılmaz-Soylu, T. Karakus, Y. Inal, G. Kızılkaya, The effects of computer games on primary school students' achievement and motivation in geography learning, Computers & Education. 52 (2009) 68–77.

[10] G.J. Hwang, H.Y. Sung, C.M. Hung, L.H. Yang, I.W. Huang, A knowledge engineering approach to developing educational computer games for improving students' differentiating knowledge, British Journal of Educational Technology. 44 (2013) 183–196.

[11] T.Y. Liu, Y.L. Chu, Using ubiquitous games in an English listening and speaking course: impact on learning outcomes and motivation, Computers & Education. 55 (2010) 630–643.

[12] M.D. Dickey, Murder on Grimm Isle: the impact of game narrative design in an educational game-based learning environment, British Journal of Educational Technology. 42 (2011) 456–469.

[13] H.-Y. Sung, G.-J. Hwang, Y.-F. Yen, Development of a contextual decision-making game for improving students' learning performance in a health education course, Computers & Education. 82 (2015) 179–190.

[14] G.-J. Hwang, H.-Y. Sung, C.-M. Hung, I. Huang, C.-C. Tsai, Development of a personalized educational computer game based on students' learning styles, Educational Technology Research and Development. 60 (2012) 623–638.

[15] M.D. Kickmeier-Rust, D. Albert, Micro-adaptivity: protecting immersion in didactically adaptive digital educational games, Journal of Computer Assisted Learning. 26 (2010) 95–105.

[16] H.-Y. Sung, G.-J. Hwang, A collaborative game-based learning approach to improving students' learning performance in science courses, Computers & Education. 63 (2013) 43–51.

[17] H. van der Meij, H. Leemkuil, J.-L. Li, Does individual or collaborative self-debriefing better enhance learning from games?, Computers in Human Behavior. 29 (2013) 2471–2479.

[18] A. Gegenfurtner, K. Veermans, M. Vauras, Effects of computer support, collaboration, and time lag on performance self-efficacy and transfer of training: A longitudinal meta-analysis, Educational Research Review. 8 (2013) 75–89.

[19] G.F.M. Silva, A. Raposo, M. Suplino, PAR: A Collaborative Game for Multitouch Tabletop to Support Social Interaction of Users with Autism, Procedia Computer Science. 27 (2014) 84–93.

[20] K. Kim, M.G. Schmierbach, S. (Saras) Bellur, M.-Y. Chung, J.D. Fraustino, F. Dardis, et al., Is it a sense of autonomy, control, or attachment? Exploring the effects of in-game customization on game enjoyment, Computers in Human Behavior. 48 (2015) 695–705.

[21] Y. Naudet, I. Lykourentzou, E. Tobias, A. Antoniou, J. Rompa, G. Lepouras, Gaming and Cognitive Profiles for Recommendations in Museums, in: 8th International Workshop on

Semantic and Social Media Adaptation and Personalization, Ieee, Bayonne, 2013: pp. 67–72.

[22] K. Karadimitriou, M. Roussou, Studying Player Experience in a Collaborative Embodied Interaction Game, in: The 2011 Third International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), 2011: pp. 199–206.

[23] M. Suznjevic, O. Dobrijevic, M. Matijasevic, Hack, Slash, and Chat: A study of players' behavior and communication in MMORPGs, in: Proceedings of 8th Annual Workshop on Network and Systems Support for Games (NetGames), Paris, France, 2009: pp. 1–6.

[24] J.L. Miller, J. Crowcroft, Avatar movement in World of Warcraft battlegrounds, in: The 8th Annual Workshop on Network and Systems Support for Games (NetGames), Paris, France, 2009: pp. 1–6.

[25] S. Natkin, C. Yan, User model in multiplayer mixed reality entertainment applications, in: Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, ACM, Hollywood, California, 2006: p. 85.

[26] E. Kaiser, F. Wu-chang, PlayerRating: A reputation system for multiplayer online games, in: Proceedings of 8th Annual Workshop on Network and Systems Support for Games (NetGames), Paris, France, 2009: pp. 1–6.

[27] N. Selvaraj, B. Fields, Rethinking collaborative decision making across distributed work communities in complex work settings, in: Proceedings of the 30th European Conference on Cognitive Ergonomics, 28 – 31 August, Edinburgh, Scotland, 2012: pp. 8–14.

[28] H. Von Stackelberg, The Theory of the Market Economy, Oxford University Press, Oxford, 1952.

[29] M. Simaan, J.B. Cruz, On the Stackelberg strategy in nonzero-sum games, Journal of Optimization Theory and Applications. 11 (1973) 533–555.

[30] R. Kicsiny, Z. Varga, a. Scarelli, Backward induction algorithm for a class of closed-loop Stackelberg games, European Journal of Operational Research. 237 (2014) 1021–1036.

[31] M.J. Murphy, The Administration Decision Game: A simulation for developing planning skills in educational managers, in: The 10th Conference on Winter Simulation - Volume 2, IEEE Computer Society Press, Miami Beach, FL, 1978: pp. 767–778.

[32] M.C. Angelides, R.J. Paul, Developing an intelligent tutoring system for a business simulation game, Simulation Practice and Theory. 1 (1993) 109–135.

[33] M.C. Angelides, A.K.Y. Tong, Implementing multiple tutoring strategies in an intelligent tutoring system for music learning , Journal of Information Technology. 10 (1995) 52–62.

[34] J. Siemer, M.C. Angelides, Evaluating intelligent tutoring with gaming-simulations, in: Winter Simulation Conference, Arlington, VA, 1995: pp. 1376–1383.

[35] B. Brinkman, The heart of a whistle-blower: a corporate decision-making game for computer ethics classes, ACM SIGCSE Bulletin. 41 (2009) 316–320.

[36] J. Pita, Real-world security games: toward addressing human decision-making uncertainty, in: The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3, International Foundation for Autonomous Agents and Multiagent Systems, Taipei, Taiwan, 2011: pp. 1359–1360.

[37] X. Liu, S. Gao, Modeling collaborative decision-making for supplier and customer in cooperative knowledge creation, in: 4th IEEE International Conference on Software Engineering and Service Science (ICSESS), 23-25 May, Beijing, China, 2013: pp. 521–525.

[38] D. Yang, J. (Roger) Jiao, Y. Ji, G. Du, P. Helo, A. Valente, Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game,

European Journal of Operational Research. 246 (2015) 263–280.

[39] J.A. Momo Kenfack, R. Pongou, B. Tchantcho, The stability of decision making in committees: The one-core, Economics Letters. 122 (2014) 390–395.

[40] A.M. Feldman, R. Serrano, Welfare Economics and Social Choice Theory, 2nd ed., Springer Science & Business Media, New York, NY, 2006.

[41] D.W. Rohde, K.A. Shepsle, Democratic Committee Assignments in the House of Representatives: Strategic Aspects of a Social Choice Process, American Political Science Review. 67 (1973) 889–905.

[42] A. Sen, Environmental Evaluation and Social Choice: Contingent Valuation and the Market Analogy, Japanese Economic Review. 46 (1995) 23–37.

[43] C. González-González, P. Toledo-Delgado, C. Collazos-Ordoñez, J.L. González-Sánchez, Design and analysis of collaborative interactions in social educational videogames, Computers in Human Behavior. 31 (2014) 602–611.

[44] M.C. Higgins, Follow the Leader?: The Effects of Social Influence on Employer Choice , Group & Organization Management . 26 (2001) 255–282.

[45] Z. Tao, X. Liu, H. Chen, Z. Chen, Group decision making with fuzzy linguistic preference relations via cooperative games method, Computers & Industrial Engineering. 83 (2015) 184–192.

[46] Y. Bouzarour-Amokrane, A. Tchangani, F. Peres, A bipolar consensus approach for group decision making problems, Expert Systems with Applications. 42 (2015) 1759–1772.

[47] F. Carreras, J. Freixas, M.A. Puente, A note on decisive symmetric games, Decision Support Systems. 51 (2011) 424–433.

[48] K. Kwiat, A. Taylor, W. Zwicker, D. Hill, S. Wetzonis, S. Ren, Analysis of binary voting algorithms for use in fault-tolerant and secure computing, in: The 2010 International Conference on Computer Engineering & Systems, IEEE, 30 November - 2 December, Cairo, Egypt, 2010: pp. 269–273.

[49] T. Hertzog, J.-C. Poussin, B. Tangara, I. Kouriba, J.-Y. Jamin, A role playing game to address future water management issues in a large irrigated system: Experience from Mali, Agricultural Water Management. 137 (2014) 1–14.

[50] J. Wu, L. Liang, F. Yang, H. Yan, Bargaining game model in the evaluation of decision making units, Expert Systems with Applications. 36 (2009) 4357–4362.

[51] J.F. Nash Jr, The bargaining problem, Econometrica. 18 (1950) 155–162.

[52] P.C. Fishburn, The Theory of Social Choice, Princeton University Press, 2015.

[53] N. Littlestone, Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm, Machine Learning. 2 (1988) 285–318.

[54] J. Rubin, I. Watson, On combining decisions from multiple expert imitators for performance, in: Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16-22 July, 2011: pp. 344–349.

[55] N. Green, P. Breimyer, V. Kumar, N.F. Samatova, PackPlay: mining semantic data in collaborative games, in: The Fourth Linguistic Annotation Workshop, Association for Computational Linguistics, Uppsala, Sweden, 2010: pp. 227–234.

[56] E. Whitaker, E. Trewhitt, M. Holtsinger, C. Hale, E. Veinott, C. Argenta, et al., The effectiveness of intelligent tutoring on training in a video game, in: 2013 IEEE International Games Innovation Conference (IGIC), IEEE, Vancouver, BC, 2013: pp. 267–274.

[57] L. Bonnet, F. Lotte, A. Lécuyer, Two Brains, One Game: Design and Evaluation of a Multiuser BCI Video Game Based on Motor Imagery, IEEE Transactions on Computational Intelligence and AI in Games. 5 (2013) 185–198.

[58] M.C. Angelides, R.J. Paul, A methodology for specific, total enterprise, role-playing, intelligent gaming-simulation environment development, Decision Support Systems. 25 (1999) 89–108.

[59] K.-C. Huang, Y.-H. Kuo, Learning collaborative decision-making parameters for multimodal emotion recognition, in: 2013 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 15-19 July, San Jose, CA, 2013: pp. 1–6.

[60] T. Ben-Zvi, The efficacy of business simulation games in creating Decision Support Systems: An experimental investigation, Decision Support Systems. 49 (2010) 61–69.

[61] D.-J. van der Zee, B. Holkenborg, S. Robinson, Conceptual modeling for simulation-based serious gaming, Decision Support Systems. 54 (2012) 33–45.

[62] A.M. Douma, J. van Hillegersberg, P.C. Schuur, Design and evaluation of a simulation game to introduce a Multi-Agent system for barge handling in a seaport, Decision Support Systems. 53 (2012) 465–472.

[63] K. Katsaliaki, N. Mustafee, S. Kumar, A game-based approach towards facilitating decision making for perishable products: An example of blood supply chain, Expert Systems with Applications. 41 (2014) 4043–4059.

[64] V.M.Á. Pato, C. Delgado-Mata, Dynamic Difficulty Adjusting Strategy for a Two-player Video Game, Procedia Technology. 7 (2013) 315–321.

[65] Y.J. Chen, W.-L. Wang, Orders dispatching game for a multi-facility manufacturing system, Expert Systems with Applications. 36 (2009) 1885–1892.

[66] D. Daylamani Zad, M.C. Angelides, H. Agius, Personalise your massively multiplayer online game (MMOG) with Artemis, Multimedia Systems. 18 (2012) 69–94.

[67] D.D. Zad, M.C. Angelides, H. Agius, Collaboration through Gaming, in: Handbook of Digital Games, 2014: p. 784.

[68] J.D. Sterman, Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment, Management Science. 35 (1989) 321–339.

[69] S.O. Kimbrough, D.J. Wu, F. Zhong, Computers play the beer game: can artificial agents manage supply chains?, Decision Support Systems. 33 (2002) 323–333.

[70] C. Yuh-Wen, M. Larbani, L. Chen-Hao, Simulation of a supply chain game with multiple fuzzy goals, Fuzzy Sets and Systems. 161 (2010) 1489–1510.

[71] H. Raiffa, J. Richardson, D. Metcalfe, Negotiation Analysis: The Science and Art of Collaborative Decision Making, Harvard University Press, 2002.

[72] J.Z. Kolter, M.A. Maloof, Dynamic weighted majority: a new ensemble method for tracking concept drift, in: Third IEEE International Conference on Data Mining, IEEE Comput. Soc, 2003: pp. 123–130.

[73] R.R. Yager, On ordered weighted averaging aggregation operators in multi criteria decision making, IEEE Transactions on Systems, Man and Cybernetics. 18 (1988) 183–190.

[74] A.A. Sofokleous, M.C. Angelides, DCAF: An MPEG-21 Dynamic Content Adaptation Framework, Multimedia Tools and Applications. 40 (2008) 151–182.

[75] A.A. Sofokleous, M.C. Angelides, Dynamic selection of a video content adaptation strategy from a pareto front, Computer Journal. 52 (2009) 413–428.

[76] D. North, Introducing BDD, Better Software Magazine. (2006).

[77] C. Solis, X. Wang, A Study of the Characteristics of Behaviour Driven Development, in: 37th EUROMICRO Conference on Software Engineering and Advanced Applications, Ieee, Oulu, 2011: pp. 383–387.

[78] I. Lazăr, S. Motogna, B. Pârv, Behaviour-Driven Development of Foundational UML Components, Electronic Notes in Theoretical Computer Science. 264 (2010) 91–105.

[79] P. Hamill, Unit Test Frameworks: Tools for High-Quality Software Development, O'Reilly Media, Sebastopol, CA, 2004.