



Reflective agents for personalisation in collaborative games

Damon Daylamani-Zad¹ · Harry Agius² · Marios C. Angelides²

© The Author(s) 2018

Abstract

The collaborative aspect of games has been shown to potentially increase player performance and engagement over time. However, collaborating players need to perform well for the team as a whole to benefit and thus teams often end up performing no better than a strong player would have performed individually. Personalisation offers a means for improving overall performance and engagement, but in collaborative games, personalisation is seldom implemented, and when it is, it is overwhelmingly passive such that the player is not guided to goal states and the effectiveness of the personalisation is not evaluated and adapted accordingly. In this paper, we propose and apply the use of reflective agents to personalisation ('reflective personalisation') in collaborative gaming for individual players within collaborative teams via a combination of individual player and team profiling in order to improve player and thus team performance and engagement. The reflective agents self-evaluate, dynamically adapting their personalisation techniques to most effectively guide players towards specific goal states, match players and form teams. We incorporate this agent-based approach within a microservices architecture, which itself is a set of collaborating services, to facilitate a scalable and portable approach that enables both player and team profiles to persist across multiple games. An experiment involving 90 players over a two-month period was used to comparatively assess three versions of a collaborative game that implemented reflective, guided, and passive personalisation for individual players within teams. Our results suggest that the proposed reflective personalisation approach improves team player performance and engagement within collaborative games over guided or passive personalisation approaches, but that it is especially effective for improving engagement.

Keywords Personalisation · Profiling · Collaborative games · Decision-making games · Reflective agents · Player engagement · Player performance

✉ Harry Agius
harry.agius@brunel.ac.uk

Damon Daylamani-Zad
d.d.zad@greenwich.ac.uk

Marios C. Angelides
marios.angelides@brunel.ac.uk

¹ Department of Computing and Information Systems, Faculty of Architecture, Computing and Humanities, Old Royal Naval College, University of Greenwich, Park Row, London SE10 9LS, UK

² Department of Electronic and Computer Engineering, Brunel University London, Kingston Lane, Uxbridge, Middlesex UB8 3PH, UK

1 Introduction

Multiplayer games fall into three broad categories (Zagal 2006): *competitive* games, where players form strategies in direct opposition to other players; *cooperative* games, where players' interests partially oppose and coincide with each other such that opportunities exist for cooperation but not necessarily for equal reward; and *collaborative* games, where all players share a common goal and work together as a team, sharing rewards and penalties. Collaborative games serve many purposes (e.g. training, role playing, entertainment) and offer numerous advantages (e.g. heightened game enjoyment, improved teamworking skills), but they are not without their problems, often because each team of players is required to work together and make various decisions. However, while the common expectation is that a group of players will regularly perform better than a single player, groups rarely perform as well as their best player would have performed individually and it is seldom feasible for such a player to be charged with all of the responsibility, information gathering and processing, and decision-making of the group (Linehan et al. 2009). Some players within the team will also *free ride* (Al-Dhanhani et al. 2014), giving less than their best effort, and where opportunity exists to change teams, *backstabbing* (Zagal 2006) becomes an issue, whereby players defect at opportune moments. Consequently, no matter how well matched a team is, not all players will be performing and engaging equally and optimally.

Personalisation offers a potential solution here, by customising aspects of the collaborative game with a view to improving player performance and engagement. For example, the most widespread and earliest use of personalisation was in providing explicit advice to players (Bakkes et al. 2012, 2013). Since then, personalisation has been found to produce benefits such as improved player satisfaction (Teng 2010), game enjoyment (Kim et al. 2015), and game engagement (Das et al. 2015). However, the application of personalisation to collaborative games specifically has received very limited attention within the research literature, and, overall, personalisation has been highlighted as an area of games where more empirical work and advances are needed (Caroux et al. 2015).

The aim of this paper is to apply personalisation in collaborative gaming to individual players within teams via a combination of individual player and team profiling with the intention of improving team player performance and engagement. Previously (Daylamani-Zad et al. 2016), we have experimented with personalisation via individual and team profiling with positive results in terms of player enjoyment and decision making. This paper extends that work by utilising reflective agents (Brazier and Treur 1999; Russell and Norvig 2009) for personalisation ('*reflective personalisation*'), such that players are actively guided to suitable goal states via a self-evaluative technique that constantly monitors the impacts of recommendations and adapts future recommendations accordingly, including the matching of players and the dynamic formation of teams.

The rest of this paper is structured as follows. Section 2 reviews related work and proposes three increasing levels of personalisation in collaborative games (passive, guided, reflective), considering existing personalised collaborative games within each level. Section 3 presents in detail our Lu-Lu agent-microservices architecture in pursuit of reflective personalisation, comprising adjustment, player and team profiling, personalisation agents, and a microservice gateway. Section 4 evaluates game personalisation within the architecture in terms of player performance and engagement using an experiment involving three different implemented versions of the game that reflect the passive, guided, and reflective personalisation levels presented in Sect. 2.3. Section 5 concludes and discusses future research and development directions.

2 Personalising collaborative games

This section reviews personalisation in gaming, the advantages and challenges presented by collaborative games, and then focuses on the state-of-the-art regarding personalisation in collaborative games, proposing three increasing levels of personalisation.

2.1 Personalisation in games

Karpinskyj et al. (2014) define personalisation as the automatic customisation of content and services based on a prediction of what the user wants, with game personalisation in particular involving the construction of a system capable of tailoring game rules and content to suit aspects of the player, such as their gameplay preferences, playing style or skill level. They survey work relating to game personalisation according to five ways that players are often said to differ from each other: by preferences (gameplay that players find appealing), by personality (distinctive character of players), by experience (how players emotionally and cognitively respond while playing), by performance (the degree and rate of player achievement/progression), and by in-game behaviour (player actions within the game). Their review focuses on the selection of meaningful player characteristics to drive personalisation, reliable and practical observation of the player characteristics, integration of the observation methods into a game adaptation system, and evaluation that verifies the personalisation has the intended effect.

Some research focuses on personalisation via broad *player types*. For example, Göbel et al. (2010) deploy a variation of Bartle's model (killer, achiever, socialiser, and explorer) which is updated as the player makes decisions, while Holmes et al. (2015) use game state changes and visual, auditory or haptic player feedback resulting from player interactions with the game mechanics to help determine players as disruptors, free spirits, achievers, players, socialisers, or philanthropists and promote certain behaviour changes. Ferro et al. (2013) derive relationships between player types, personality types and traits, and game elements and game mechanics. Hardy et al. (2015) describe an interdisciplinary framework for personalised, game-based training for the elderly and disabled which differentiates between three personalisation layers (sensors/actuator (constitutional), user experience, and training), which are used to adjust cognitive or physical challenges which may be dependent on the vital state of the player (e.g. heart rate) or the needs of target player types that share specific characteristics. However, the empirical validity of player types is questioned by some researchers, e.g. Busch et al. (2016) investigated the psychometric properties and predictive validity of player types within the popular BrainHex model and found that psychometric properties could be improved while predictive validity required significant further study.

Consequently, a broader body of research more specifically considers a personalised game to utilise *player profiles* (also known as player models) to tailor the game experience to the individual player rather than a type of player. For example, Bakkes et al. (2012, 2013) utilise one or more of the following personalisation components informed by difficulty-scaling techniques: space, mission/task, characters, game mechanics, narrative, or music/sound adaptation, and player matching in multiplayer games. Similarly, Machado et al. (2011) define player profiling as an abstract description of the current state of a player at a particular moment, which can be carried out according to satisfaction (player preferences), knowledge (what the player knows), position (player movement) and strategy (interpreting the player actions and relating them with game goals). These can be modelled in increasing levels of use, from online tracking (for predicting future actions), to online strategy recognition (iden-

tifying a set of actions as a higher level objective or strategy, often used for teams), to offline review (game log evaluations). Natkin and Yan (2006) distinguish three player profile levels: generic, localised, and personalised, the latter involving complex state variables such as a player's skill level, relationships and habits. These are mapped to global, context-oriented and character-based narration schemes in multiplayer mixed reality games, so that narration can be personalised to groups and individual players. Shaker et al. (2010) derive profiles from predicted player experience based on features of level design and playing styles, which are constructed using preference learning based on post-play questionnaires. Similarly, Bakkes et al. (2014) personalise levels taking into account challenge balancing within an enhanced version of *Infinite Mario Bros*, such that short new level segments are generated during gameplay by mapping gameplay observations to player experience estimates. Hocine et al. (2015) use player profiles in conjunction with a training module in serious games to model a stroke patient's motor abilities based on short-term prediction and their daily physical condition, which are then used to generate dynamically-customised game difficulty levels.

Given the above two approaches to personalisation, player types and player profiles, the latter have proven less controversial and profiles tend to be more comprehensive in their coverage as they are specific to a particular player, which facilitates broader means of personalisation within a game. However, the focus is overwhelmingly on the individual player and thus profiling groups of players has not been adequately considered, which is important to collaborative games involving teams.

2.2 Advantages and challenges of collaborative games

A range of benefits deriving from collaboration within a game environment has been evidenced in the research literature, including increased game enjoyment (Nardi and Harris 2006), increased sense of presence (von der Pütten et al. 2012), support and improvement of social interactions (Silva et al. 2014), improved communication, collaboration and teamwork skills (González-González et al. 2014), improved team training (O'Connor and Menaker 2008), and improved sympathy and empathy toward patients (Octavia and Coninx 2014). Nasir et al. (2015) found that playing a collaborative game resulted in increased interaction and participation in real-world collaborative tasks performed subsequently. Studies (van der Meij et al. 2013) have found that even a collaborative debriefing following a non-collaborative-gaming session can improve player score over time.

While games and gamification have long been seen as beneficial to education and training (Angelides and Paul 1993; Siemer and Angelides 1998; Malas and Hamtini 2016), collaborative games in particular have been shown to facilitate team training and thus have been designed as instructional games promoting team learning and transfer. Characteristics seen as promoting learning include: environment, reality, interactivity, role play, engagement, rules, persistence, efficiency, and fidelity (O'Connor and Menaker 2008). Consequently, many collaborative games have tended to be educational or serious in nature. For example, GAMEBRIDGE (Oksanen 2013; Oksanen and Hamalainen 2014) is a 3D serious collaborative game focused on teaching the requirements for future working life, while DREAD-ED (Linehan et al. 2009) places players in unique roles in an emergency management team and was found to demonstrate similar problems to those faced by real-world decision-making groups. Many collaborative games target the health domain with various objectives, e.g. helping hospitalised children to communicate and collaborate (González-González et al. 2014), collaborative rehabilitation of multiple sclerosis patients (Octavia and Coninx 2014), enabling patients to virtually exercise together and interact with each other under doctor

supervision (Lin et al. 2015), and supporting sharing, performance and interaction among autistic players (Silva et al. 2014).

As can be seen, the nature of collaboration can vary widely within a collaborative game, from informal encounters to highly-organised play within structured, collaborative groups (Nardi and Harris 2006). Collaborative gameplay is often supported through social networks, making it possible to establish relationships and encouraging collaboration as well as the transfer of strategies and knowledge linked with demands and executions specific to the game. In such games, Del-Moral Pérez et al. (2014) found four collaborative processes prioritised by players: collaborating and giving other players gifts, learning new strategies, sharing strategies, and helping other players expand. The level of awareness provided by a collaborative game varies but is necessary to fully undertake collaborative tasks effectively. Teruel et al. (2016) propose a comprehensive concept of ‘gamespace awareness’ that covers the present, the past, the future, and social and group dynamics. Other research has explored issues of player commitment and loyalty in collaborative games (Moon et al. 2013), by motivating players to embrace ownership of the game via enhancing their ability to control their game character and to develop an online social identity, and issues of participation in collaborative games (Kim et al. 2013) by establishing prices to encourage participation and retain players in order to maintain a healthy number of game participants.

Team formation is an important but much neglected aspect of collaborative games, yet it is a widely-explored topic in non-gaming domains, such as in collaborative filtering and recommendation (Retna Raj and Sasipraba 2015; Ghenname et al. 2015; Najafabadi and Mahrin 2016), collaborative design (Xu et al. 2010), collaborative crowdsourcing (Lykourentzou et al. 2016), and expert collaboration in social networks (Basiri et al. 2017). However, in multi-player games, the focus has overwhelmingly been on player matching and more so with a view to finding worthy opponents for players to play *against* rather than suitable players to play *with* in a team (Daylamani Zad et al. 2012). For example, Corem et al. (2013) propose a technique for quickly matching opponents with similar skill levels to facilitate fair tournaments, while Tsai (2016) propose a technique for matching competitors in educational multi-player online games. The exploitation of social media has enabled player matching to somewhat influence the personalisation function, as in the SoCom architecture (Konert et al. 2014), where currently-online friends of a player may contribute game content to specific calls from a game, or as in BLUE (Naudet et al. 2013), where museum visitors’ experiences are personalised through the use of gaming and social networks which exploit the player’s cognitive profile and personal interests as inferred from the choices they make during an associated Facebook game. Thus, in the vast majority of collaborative games, teams are usually self-formed by the players themselves and some methods have been proposed for facilitating that, e.g. in the PlayerRating interface add-on (Kaiser and Feng 2009) for World of Warcraft, prior experiences of a player’s peers are used to determine the reputation of all peers, allowing well-behaved players to safely congregate and avoid interaction with antisocial peers. Studies of multi-player games have consequently focused on understanding what motivates players’ choices of teammates and the relationship with performance, with one key finding being that large variations in competence within teams discourages repeated interactions (Alhazmi et al. 2017). This is in line with findings from collaborative learning environments, in which approaches that focus on automatically-forming optimally balanced student teams have been found to perform better than manual allocation to the student teams by experts (Yannibelli and Amandi 2012; Bergey and King 2014). Considering feedback from student collaborations has also been found to improve the group formation process (Srba and Bielikova 2015). Thus, an effective team in collaborative games is considered to be an automatically-formed, balanced team that may be reformed during the game according to feedback.

The wide-ranging impacts of collaboration within gaming environments coupled with increasingly complex social interactions possible through on-going technological advances, has resulted in rich, complex environments that offer great benefits but also pose significant challenges to personalisation. The additional challenges presented to personalisation by multiplayer and collaborative games typically mean that personalisation is often either avoided altogether, thereby limiting the games with regards to the level of heterogeneity of co-players they are able to support, or ‘handicaps’ against some players are utilised in an attempt to level the field, which may result in measurable psychological impacts (Streicher and Smeddink 2016). Thus, to date, personalisation within multiplayer games, and personalisation as it pertains to collaborative games specifically, has been typically neglected and not adequately addressed in the research literature to date.

The next section focuses more specifically on personalisation within collaborative games, proposing three increasing levels of personalisation which are used to review and position the small body of research within the area.

2.3 Levels of personalisation in collaborative games

Following Bakkes et al. (2012, 2013), we consider a personalised collaborative game to utilise profiles to tailor the game experience to the individual player within the collaborating team. As discussed in Sect. 2.1, although individual profiles have been used widely to personalise games, team profiles are not adequately explored within the literature. Therefore, we consider the importance of both and distinguish personalisation in collaborative games according to the incorporation of:

- *Player goal states* where personalisation seeks to steer the players towards specific goal states, such as optimal player types, gaming objectives, or similar.
- *Self-evaluation* where the effects of personalisation are regularly evaluated for the purposes of dynamically modifying the recommendations made and thus the personalisation itself.
- *Dynamic team formation* where the collaborative nature entails the matching of players and the formation and re-formation of teams as the game progresses, including appointing leaders and moving players between different teams. This enhances goal states and self-evaluation by providing a richer context for reflection and is enhanced by the use of team profiles.

The combination of the above results in three increasing levels of personalisation in collaborative games, as illustrated in Fig. 1. Considering the state-of-the-art in personalisation of collaborative games within these levels reveals that research has focused exclusively at the first two levels.

Passive personalisation is the most common personalisation approach in collaborative games, whereby individuals and/or teams are passively served with personalised content. While player and team profiling may be apparent to achieve this, this approach does not guide players to particular goal states, and while the effects of personalisation may be monitored, personalisation techniques are not dynamically adapted. For example, in the 3D collaborative game, OSgame (Terzidou and Tsiatsos 2014), an intelligent agent interacts with individual players and the team, providing support through messages that provide assistance and hints during gameplay. The agent’s dialogue is triggered based upon players’ actions. While a progress map is included in the game, it serves only as a visual reference for players, and while the intelligent agent operates according to predefined goals, it does not seek to progress the players or teams to specific states. Similarly, in our earlier Lu-Lu architecture (Daylamani-Zad et al. 2016), messages are sent to players in order to encourage engagement

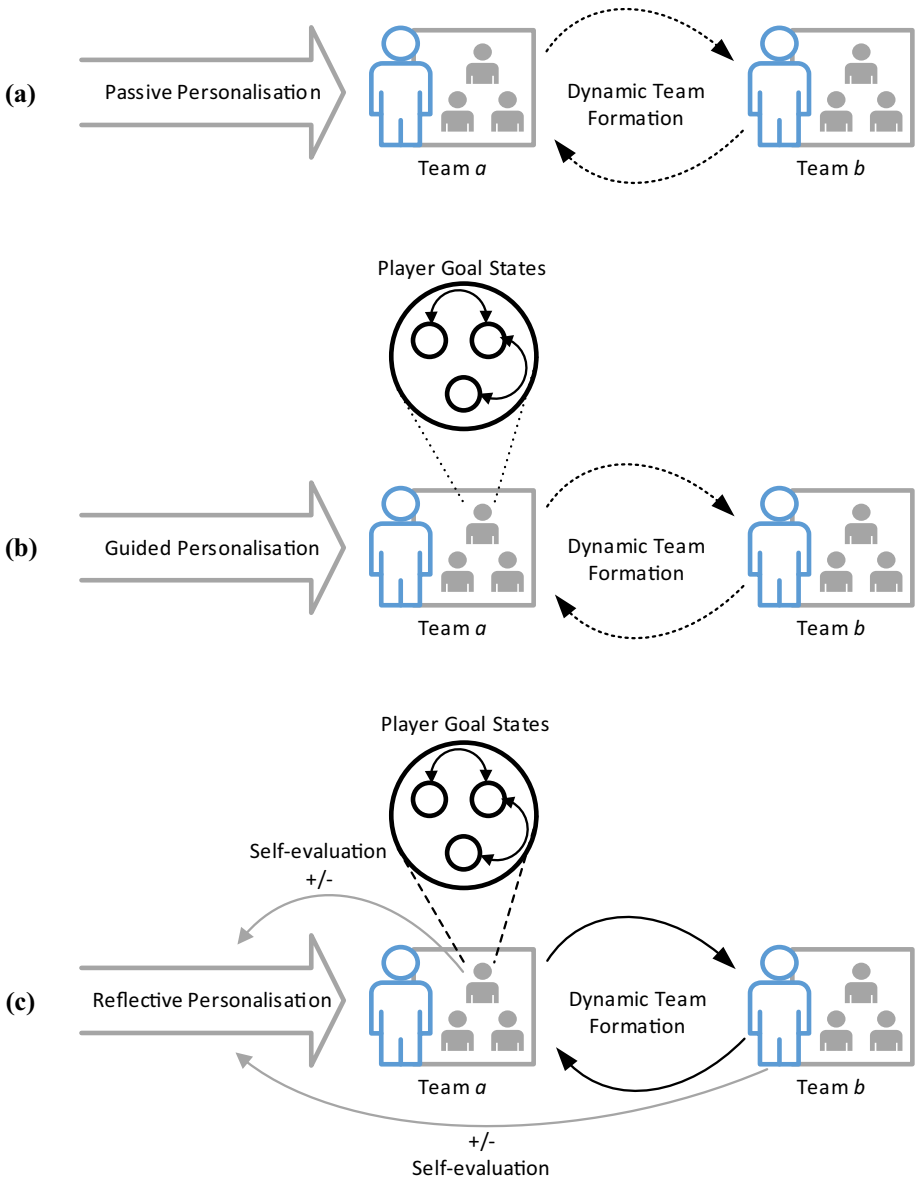


Fig. 1 Increasing levels of personalisation in collaborative games: **a** passive, **b** guided, and **c** reflective

and performance. These are triggered by decision trees which do not take player’s historical activities into account. Due to the lack of guidance and self-evaluation, passive personalisation is often at the exclusion or expense of in-game team formation considerations, and thus teams are typically self-formed, although dynamic team formation may be incorporated optionally (indicated by dashed arrows in Fig. 1). For example, in the Social Maze Game (Octavia and Coninx 2014) for collaborative rehabilitation of multiple sclerosis patients, the patients receive social support from family, friends, therapists, caregivers, and fellow patients by actively participating in the therapy sessions (sympathetically, by performing the collab-

orative training exercise together with the patient, or empathetically, by two patients training together). Although it is collaborative, involving two players, the team is formed at the start of the game only, and thus personalisation is focused on only one player: the patient requiring rehabilitation.

Guided personalisation represents a more enhanced, and less common, personalisation approach for collaborative games, in that the objective of personalisation is to guide players towards particular goal states, such as player types and/or gaming objectives. For example, in NUCLEO (Sancho et al. 2009, 2011), a collaborative game set in a 3D virtual world for teaching computer programming, small teams of students solve missions and are provided with hints, guidelines, cues and feedback to steer them towards completion of the mission which serves as a goal state. Similarly, the PLAGER-VG architecture (Padilla-Zea et al. 2017) supports game-based collaborative learning, consisting of personalisation, design, groups, monitoring and game sub-systems, in which players are loosely steered towards educational goal states through learning, profile and game instance personalisation. In guided personalisation, team formation and player matching may be dynamic but it is not required (again, indicated by dashed arrows in Fig. 1) as guidance may operate independently of team considerations. This is the case in both NUCLEO and PLAGER-VG where, despite the incorporation of dynamic team formation, it serves only to balance roles or learning styles and in the case of NUCLEO occurs only prior to the commencement of each new mission.

The third level, *reflective personalisation*, is differentiated by the integration of self-evaluation. Self-evaluation necessitates the use of intelligent agents embodying reflective behaviour for personalisation. Various researchers have worked on multi-agent architectures to improve learning and facilitate training without human supervision (Yee et al. 2007; Goldberg and Cannon-Bowers 2015; Shute et al. 2015). The use of agents has become more prominent in recent years due to advances in profiling and available computational power, leading to an increase in the use of agents in educational games, especially decision-making games (Visschedijk et al. 2013; Veletsianos and Russell 2014). Moreover, various types of agents have been used in games generally for a wide range of purposes, e.g. emotional game agent architectures (Spraragen and Madni 2014), agent-centric game design methodologies (Dignum et al. 2009), memetic computing within games (Miche et al. 2015), and adaptive agents in social games (Asher et al. 2012). Galway et al. (2008) extensively survey machine learning within digital games (neural networks, evolutionary computation, and reinforcement learning for game agent control), which provides a means to improve the behavioural dynamics of game agents by facilitating the automated generation and selection of behaviours according to the behaviour or playing style of the player. Reflective agents are intelligent agents that are capable of reasoning about their own behaviour and that of other agents' as well as reasoning about the behaviour of the external world (Brazier and Treur 1999). They have been used in a wide variety of domains with promising results, such as Web interfaces (Li 2004), natural language processing and learning (van Trijp 2012), game theory (Fallenstein et al. 2015), and mobile robot navigation (Kluge and Prassler 2004).

Reflective personalisation represents the most advanced personalisation approach for collaborative games, where personalisation is both guided and self-evaluative due to the incorporation of reflective agents to improve the personalisation function. The effects of the personalisation for the individual and team are monitored and reasoned about, with recommendations modified as and when necessary to guide players to particular goal states and to match players while dynamically forming and re-forming teams as necessary during gameplay. Dynamic team formation is required (indicated by solid arrows in Fig. 1) to fully support the self-evaluation carried out by the reflective agent so that the full range of necessary cor-

Table 1 Comparing personalisation in existing collaborative games with reflective personalisation

Personalisation approach	Player goal states	Self-evaluation	Dynamic team formation
OSgame (Terzidou and Tsiatsos 2014)	X	X	X
Social Maze Game (Octavia and Coninx 2014)	X	X	X
Original Lu-Lu (Daylamani-Zad et al. 2016)	X	X	✓
NUCLEO (Sancho et al. 2009, 2011)	✓	X	✓
PLAGER-VG (Padilla-Zea et al. 2017)	✓	X	✓
Reflective Lu-Lu	✓	✓	✓

reflective actions can be fulfilled, which includes reforming teams according to feedback to improve team balance and effectiveness.

It is important to emphasise that this is in an automated fashion. Individual players are not the only stakeholders in the game and some research seeks to cater for these additional (non-player) stakeholders within the (non-reflective) personalisation function (Streicher and Smeddinck 2016). For example, in the cloud-based serious game for obese patients presented by Alamri et al. (2014), therapists/caregivers can access player's health data remotely to change the game complexity level and accordingly provide recommendations during the game session. Likewise, the Personalised Educational Game Architecture (Peeters et al. 2016), uses a flexible multi-agent organisation and ontology for personalisation whereby a human instructor can decide to control selective parts of the training, while leaving the rest to intelligent agents (i.e. adaptive automation), or can fully instruct agents in advance of the game on how to personalise the environment. However, these approaches involve human intervention in the personalisation process, whereas the majority of definitions of personalisation (Karpinskyj et al. 2014) stipulate automatic customisation.

Table 1 compares the reflective Lu-Lu personalisation approach proposed in this paper with the existing collaborative game personalisation approaches described above. To the best of our knowledge, the reflective level of personalisation has not been previously incorporated into collaborative games. Consequently, the remainder of this paper seeks to propose an architecture that implements a reflective-agent-based personalisation and to evaluate that approach within the architecture. In this way, we aim to take some steps towards meeting the personalisation challenges caused by collaboration discussed above.

3 Lu-Lu reflective personalisation architecture for collaborative games

In this section, we propose an architecture for reflective personalisation for collaborative games. To do so, we build upon our previous Lu-Lu architecture (Daylamani-Zad et al. 2016), which fostered collaboration through the implementation of collaborative features such as team matching, leadership, non-optimality, identity awareness, and passive personalisation. The architecture incorporates multiple reflective agents to facilitate reflective personalisation and aims to work with games that have existing collaborative capabilities. Through this approach, we aim to increase player *engagement* and *performance* within such games by aspiring to guide individual players within teams to a goal state that matches the definition of optimal player for each individual game. The architecture incorporates profiling both at player

and team levels in order to be able to reflectively communicate through recommendations and facilitate team formation/re-formation. Recommendations are a commonly-used and effectual personalisation technique which may be realised in various forms, ranging from direct recommendations to the player during gameplay (Alamri et al. 2014) to in-game assistance and hints (Sancho et al. 2009, 2011; Terzidou and Tsiatsos 2014) or more elaborate improvement reports (Padilla-Zea et al. 2017). As the architecture aims to be portable and scalable, it is designed to take advantage of existing game APIs provided by game developers and communicates through such APIs with the games, while profiles are portable across games. This approach enables the architecture to engage and improve players with minimal interference with the game itself. Game content is not changed as facilities to do so outside of the game are often not provided, and when they are, this would result in the architecture having to be vastly customised in a bespoke way for each individual game, thereby contradicting its purpose. Our research is therefore beyond the scope of strictly personalised games, which seek to tailor all content to the individual player (Bakkes et al. 2012), instead being focused on personalisation for individual players within teams from Lu-Lu's side of the game API not the game's side (Lu-Lu environment and external environment respectively in Fig. 2).

3.1 Architecture overview

Multi-agent systems work well within a webservice environment as the distributed and independent nature of the agents calls for a decentralised architecture (Florio 2015). We adopt a microservices architecture pattern (Lewis 2012; Fowler and Lewis 2014) which consists of a collection of collaborating services that implement a set of narrow, related functions that may be deployed independently and thus are easier to update and customise (Richardson 2017). Figure 2 presents the Lu-Lu reflective personalisation architecture, which distinguishes between the Lu-Lu environment, consisting of a set of zones (Profiling, Adjustment, and Personalisation) and a Microservice Gateway, and the external environment, consisting of a game interfaced by its associated API. While the Profiling and Adjustment Zones are modified from the previous architecture, the Personalisation Zone is an entirely new addition designed to facilitate the use of reflective agents. The various microservices allow for decentralisation as well as creating communication layers (the Microservice Gateway and the Adjustment and Personalisation Zones) between external games (via Game API) and the user modelling services (Profiling Zone).

All information about players and teams are stored in the Collective Memory which spans across the Profiling and Personalisation Zones and is retrieved by the Adjustment and Personalisation Zones when needed. The Profiling Zone is designed to be independent of the game and therefore does not need to be modified as the player participates in new games. Instead, as a result of using a microservices architecture, new services may be deployed within the Adjustment and Personalisation Zones for each individual game. Combined with the use of cloud-based services, this facilitates rapid scalability, which has recently been recognised as being essential for sustained gameplay and deployment support (Scacchi 2017). Such design also allows for portability and independency of player profiles, enabling them to persist across multiple games.

The Microservice Gateway component acts as an interface between the services in the cloud and the external games, through the provided game API, thus Lu-Lu is only able to work with games that provide such APIs for external communication. This is a necessary requirement for portability and scalability which allows for third-party integration of the architecture. The Game Monitor service receives all messages from the games and passes

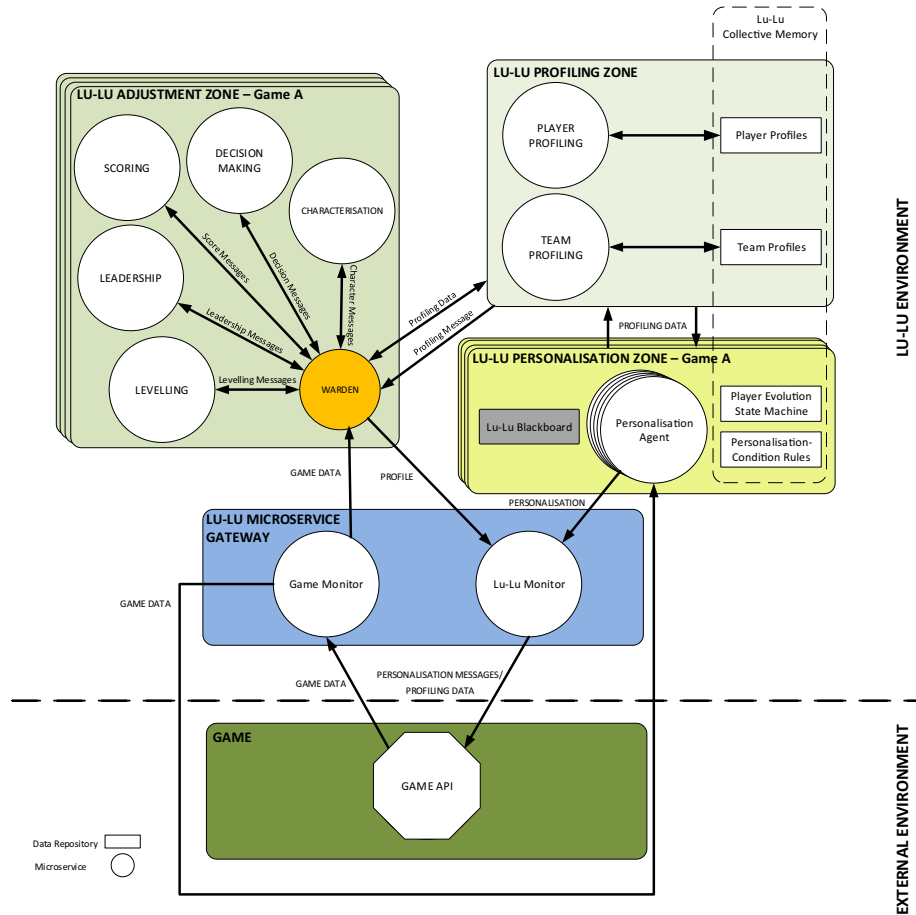


Fig. 2 Lu-Lu reflective personalisation architecture for collaborative games

them to the corresponding component in the Adjustment Zone. Each component in the Adjustment Zone is a unique implementation of the Adjustment Zone’s services tailored to, and corresponding with, a supported game. Characterisation, Decision Making, Scoring, Leadership and Levelling were previously proposed in Daylamani-Zad et al. (2016) although here they are realised as microservices. The Warden microservice is an addition to the previous architecture and acts as the internal gateway for each implementation of the Adjustment Zone. When messages are received from the Game Monitor, the Warden requests the profiling data of the player, analyses the message, and informs the five corresponding microservices. The output of the process from each microservice in the Adjustment Zone is then sent back to the Warden who passes these to the Profiling Zone. The Profiling Zone applies these new changes to the profiles and based on these changes sends profiling messages back to the corresponding game’s Adjustment Zone through its Warden to inform the game of any changes to the profile of the player. Once all messages have been received, the Warden then packages all changes into one message that is relayed back to the corresponding game via the Lu-Lu Monitor microservice. The Game Monitor service also sends the received information to the corresponding Personalisation Agent in the Personalisation Zone. As with the Adjustment

Fig. 3 Sample GameData sent in JSON

```
{
  "Identifier": "55744ddb-a08e-4d9d-a007-fa1f8b75323c",
  "GameName": "Beer Game v3.3.1-AM",
  "Player": "JohnG20",
  "rScore": 532,
  "PlayerAction": [...],
  "TimeStamp": "2016-09-03T17:19:06.0552948+01:00"
}
```

Table 2 Player actions

Field name	Description	Type
Action name	For example, issuingOrder, playerlogin	String
Action type	Decisions Actions that directly affect gameplay, depending on genre, e.g. issuing an order, making a profit, move, attack, steer Loyalty General information not directly affecting gameplay such as login, logoff, update details Activities General information not directly affecting gameplay, interactions and game events: clicks, taps, swipes, button press, character updates	String
Start time	Action start time	Datetime
Duration	Either action duration or action end time (to calculate duration)	Datetime
Affected object(s)	Reference to any object(s) related to the action, e.g. if action relates to making a decision, a reference to that decision	Reference

Zone, there are multiple components within the Personalisation Zone, each uniquely implemented for a specific game. The Personalisation Agent uses the message and, based on the information from its Blackboard and Collective Memory, arrives at a decision that is sent back to the game through the Lu-Lu Monitor service.

All microservices in the architecture have been implemented as RESTful webservice using the .NET framework Web API in C# and messages are serialised in JSON. Figure 3 presents a typical GameData object as received in JSON format by the Game Monitor. The PlayerAction field is a list of player activities which are returned as a time-stamped list in DateTime format. The action data structure is broken down in Table 2. Player Actions are divided into three types: Decisions, Loyalty, and Activities. Decisions are defined in a broad way, as actions that have direct impact on the gameplay. This could include decisions and orders in a decision-making game or could be actions such as movements which in essence could be considered decisions by the player on how to proceed in the game. Thus, any action that has a direct effect on the gameplay is considered a decision type action. This might vary depending on the game genre and the mechanics implemented within the game. Loyalty actions are those that do not have direct impact on the gameplay and relate to account actions and engagement of the player, including login and logoff actions. Activity actions include all other actions that do not directly impact on the gameplay such as interaction activities, character updates, and so on. The start-time and duration of activities are used by Lu-Lu to calculate player engagement as described in the following section.

The rest of this section will discuss the use of individual player profiling to support reflective personalisation, the facilities provided for encouraging collaboration and the mechanism for team profiling, and the role of reflective agents for reflective personalisation within the architecture.

3.2 Individual player profiling

To effectively implement personalisation in collaborative games, all player actions, preferences, and behaviour need to be profiled. For reflective personalisation, profiles at both player and team level are required. In order to help players improve, they need to be divided into dynamically created teams which match the players' levels of proficiency (as discussed in Sect. 2.3). The player and team profiles, which form part of the collective memory, are stored in the Profiling Zone and hold all information regarding the player, their team and their actions within the game.

Player Profiles store the player models which contain a player's in-play status as illustrated in Table 3. The profile is modelled using the XML-based standard MPEG-7 (ISO/IEC 2002, 2003, 2004, 2005) as it facilitates a detailed, structured representation within multimedia systems such as digital games (Agius and Angelides 2004; Angelides and Agius 2006) and also provides facilities that support distributed player profiling. Score is calculated through the Scoring service which, using the recorded player gameplay, determines and feeds the player's score back into the game. The result of each team's decision affects the score of each player based on their decisiveness index (described in Table 3 and later formally defined in Eq. 2) on the team decision as well as their personal decision. As mentioned earlier, we define decisions in a broader sense that includes all actions that have a direct impact on gameplay. Players share the team score based on the influence they have had in achieving it. The player score is calculated using Eq. 1. This is Lu-Lu's definition of score which might not be the same as the in-game score. A Scoring microservice in the Adjustment Zone is implemented and launched for each individual game such that rewards and progress, even in games that do not have a conventional score for players, can be mapped to a unified Lu-Lu representation of score (hereafter referred to as just 'score'). For example, in a time-based game the time finished could be translated to a score whilst in a certificate-based game, the number of certificates and pass/fail in acquiring a certificate could be mapped as a score.

$$\begin{aligned}
 \text{WeightedScore}_{\text{player}_i} &= \text{DecisivenessIndex}_{\text{player}_i} \\
 &\quad \times \frac{\text{Result}_{\text{Decision}_{\text{Team}}} + \text{Result}_{\text{Decision}_{\text{player}_i}}}{2} \\
 \text{Score}_{\text{player}_i} &= \text{CurrentScore}_{\text{player}_i} + \text{WeightedScore}_{\text{player}_i}
 \end{aligned} \tag{1}$$

In any game, there is a maximum achievable score for each round/decision. Considering this as a basis for calculating the score in the proposed architecture allows for a gain/loss score system which can result in positive and negative scores. Therefore, each round of gameplay results in a real number and a player's score can go down as well as up which can be a great indication of the player's progress or decline during gameplay.

Levelling and characterising are interlinked. Levelling uses a player's score to determine their Lu-Lu level (hereafter referred to as simply 'level'). Even when a player's score is dropping they still retain their level achievement. Player character will change throughout the game to reflect progress. Players can upgrade their characters when they reach a level where a new character is available to them. Not all games include levels and characters in

Table 3 Player profile and player model example in MPEG-7

Player Profile	Description	MPEG-7 DS/types	Player Model
Character	Assumed character role in the game. Player personal information. Player login information.	UserDescription, Person DS, PersonName	<pre><User xmlns:type="PersonType" id="userids776928"> <Name> <GivenName>John</GivenName> <FamilyName>G1111ian</FamilyName> <PlayerName>JohnG28</PlayerName> </User></pre>
Score	Player's score as defined in Lu-Lu.	PersonName	<pre><Title>Score</Title> <Numeration>124875</Numeration></pre>
Decisiveness Index	Reflects how effective a player is within the team and an indication of the player's competence in the game.	PersonName	<pre><Title>DecisivenessIndex</Title> <Numeration>0.427</Numeration> </Name></pre>
Team	Reference to the player's current team affiliation.	Reference	<pre><Affiliation> <PersonGroupRef href="teams.mp7.xml#id-14578256"/> </Affiliation></pre>
Level	Player's level as defined in Lu-Lu which is derived from the score.	Reference	<pre><Affiliation> <OrganizationRef href="Level.mp7.xml#id-12"/> </Affiliation></pre>
Evolution	Allows tracking and classification of player progress which is derived from their score, loyalty and current evolution state.	Reference	<pre><Affiliation> <OrganizationRef href="EvolutionState.mp7.xml#id-5"/> </Affiliation> <Affiliation> <OrganizationRef href="EvolutionAction.mp7.xml#id-13"/> </Affiliation></pre>
Decisions	Outcomes of decisions made by a player with a reference to the decision model.	UsageHistory DS, UserActionHistory DS, UserActionList DS	<pre><UserActionList> <ActionType href="urn:LULU:cs:PActionCS:1.2"> <Name>MadeDecision</Name> </ActionType> <UserAction> <ActionTime> <GeneralTime> <TimePoint>2016-06-11T16:03:48</TimePoint> <Duration>PT08S</Duration> </GeneralTime> </ActionTime> <ProgramIdentifier>Decision</ProgramIdentifier> <ActionDataItem href="decisions.mp7.xml#id-87124593"></ActionDataItem> </UserAction> </UserActionList></pre>
Loyalty	Player's gaming habits. Includes every login and logoff action, time spent in the game, progress pace, and regularity of gameplay.	UserActionList DS	<pre><UserActionList> <ActionType href="urn:LULU:cs:PActionCS:1.2"> <Name>UserLogin</Name> </ActionType> <UserAction> <ActionTime> <GeneralTime> <TimePoint>2016-06-11T15:27:32</TimePoint> <Duration>PT1S</Duration> </GeneralTime> </ActionTime> <ProgramIdentifier>Login</ProgramIdentifier> </UserAction> </UserActionList></pre>
Activities	All actions a player makes during gameplay excluding decisions and loyalty actions. Includes: movements, achievements, preferences, and records of personalised recommendations assigned to player, including the results they produced once actuated to the player. Acts as a history of effectiveness of recommendations for each player which is used for reflective personalisation.	UserActionList DS	<pre><UserActionList> <ActionType href="urn:LULU:cs:PActionCS:1.2"> <Name>Personalisations</Name> </ActionType> <UserAction> <ActionTime> <GeneralTime> <TimePoint>2016-06-11T18:54:21</TimePoint> <Duration>PT4H0M5</Duration> </GeneralTime> </ActionTime> <ProgramIdentifier>Utility Value</ProgramIdentifier> <ActionDataItem href="PersonalisRules.mp7.xml#id-71"></ActionDataItem> </UserAction> </UserActionList></pre>

their original form, but utilising this architecture enables these facilities to be included in the game and to potentially increase engagement.

Lu-Lu aims to progress each player into a goal state where they are active and high scoring to support the guided element of reflective personalisation. In order to achieve this, players are classified into one of eight states (Fig. 4) adopted from those suggested by Angelides and Agius (2000). While being somewhat similar to player types that were discussed in Sect. 2.1, they are not personality or trait based, but rather performance and engagement based, which is applicable generally to collaborative games. They are fuzzy dynamic states as our approach aims to progress players into Dominance or the closest state possible, with players then being balanced within teams. This is described in more detail in Sect. 3.4. Players are classified into an evolution state and are moved between states based on their score, their loyalty and current state. As a player's score (and thus performance) and activity (and thus engagement)

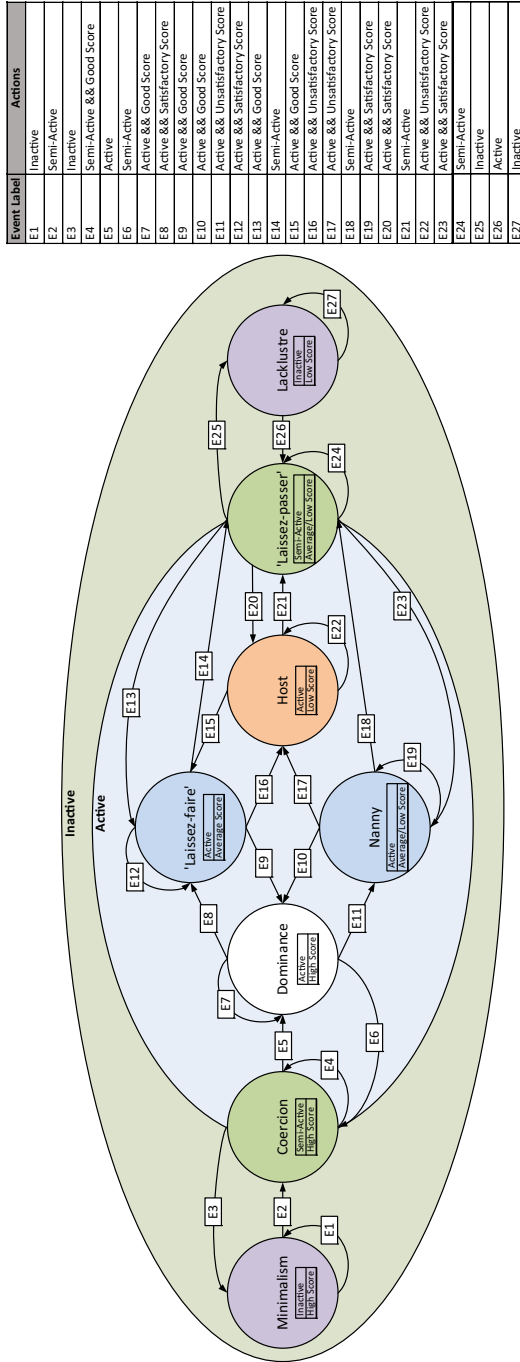


Fig. 4 The player evolution state machine

changes, their state changes accordingly, with the player's current state recorded in the profile under Evolution.

While some states seem to bear similar characteristics, the source or destination states and the events that progress the players to them are different. As these are fuzzy states, player movements are not rigid and can have multiple outcomes. The fuzzy implementation of the evolution state machine allows players to be in 'multiple' states; passing out of one while emerging into another. This fuzzy approach is suitable for individualising players as the extent of emergence or pass-out along with the source and destination are valuable for distinguishing players beyond basic classification. For example, while players in 'Laissez-faire' or Nanny states are all active with an average score, they may be differentiated by the event that progressed them from Dominance to each of these states and the extent to which they have emerged into or are passing out of it. A player in Dominance who scores satisfactorily would move to 'Laissez-faire' but a player in Dominance who scores unsatisfactorily would move to Nanny. Such players would now record average scores. There are no paths between Nanny and 'Laissez-faire', as if players in either of these states perform the same, they will remain in their current state. Any improvement or decline in engagement or performance would result in them moving to respective states for that behaviour.

In order to classify players, there is an Upper Limit and Lower Limit defined for each game in the Adjustment Zone, set as fuzzy values and customised for each game in order to allow a dynamic movement suitable for each individual game. A player scoring consistently above the Upper Limit is classified as Good scoring; between the Upper Limit and the Lower Limit is considered Satisfactory; scoring consistently below the Lower Limit is classified as an Unsatisfactory score.

Players are classified into Active, Semi-Active and Inactive based on their engagement with the game. The engagement conditions are adjusted based on individual games and player activity, which includes frequency of playing, duration of gameplay and frequency of activities during play. For example, a player who has been playing 75% or more of the days since he joined the game, has not had an absent day or the number of days since he last played is still within the range of Active Limit, may be considered Active; a Semi-Active player may be one who has played between 50% and 75% of days since joining the game, and also the number of days since he last played the game is between the Active and Inactive limits; and an Inactive player may have played less than 50% of the days since they joined the game and has not been playing for more than the Inactive Limit number of days.

3.3 Team profiling and collaboration

In order to support collaboration through profiling, team profiling as proposed here includes distinct profiles for each team as well as the individual player profiles as presented earlier. This combination aims to create effective teams in order to increase player engagement and promote performance. The Team Profile, presented in Table 4, is used to support collaboration and for player matching and dynamic team formation and re-formation, including leadership, based on a team status of *balanced*, *imbalanced*, or *ineffective*, as described in Daylamani-Zad et al. (2016). As discussed in Sect. 2.3, an effective team is considered to be one which is balanced and which may need to be re-formed during the game. Our approach considers a balanced team to be one which is performing well and in which the members are well balanced as defined by the player levels. An imbalanced team has one or more members that need to be moved out of the team because they are either much stronger (much higher

Table 4 Team profile and team model example in MPEG-7

Team Profile	Description	MPEG-7 DS/types	Team Model
Team	Team identifier	PersonGroup DS	<code><PersonGroup xsi:type="PersonGroupType" id="id-5548793"></code>
Status	Used for team reformation: <ul style="list-style-type: none"> • balanced • imbalanced • ineffective 	TermUse	<code><Kind> <Name>balanced</Name> </Kind></code>
Members	Members of team including their character information, their decisiveness index in the team, a reference to the profile and their role within the team: <ul style="list-style-type: none"> • Member • Leader 	Person DS	<code><Member xsi:type="PersonType" id="userid5776928"> <Name> <GivenName>John</GivenName> <FamilyName>Gillian</FamilyName> <Title>Leader</Title> <PlayerName>JohnG20</PlayerName> <Title>DecisivenessIndex</Title> <Numeration>0.427</Numeration> </Name> <NameTerm href="johnG20.mp7.xml"> </NameTerm> </Member> <Member xsi:type="PersonType" id="userid1784265"> <Name> <GivenName>Jane</GivenName> <FamilyName>Walker</FamilyName> <Title>Member</Title> <PlayerName>epictree</PlayerName> <Title>DecisivenessIndex</Title> <Numeration>0.120</Numeration> </Name> <NameTerm href="epictree.mp7.xml"> </NameTerm> </Member> <Member xsi:type="PersonType" id="userid5542718"> ... </Member> ... </PersonGroup></code>

level) or much weaker (much lower level) than the other team members, yet the team is still effective enough that it does not need to be re-formed. An ineffective team is a team that has such diversity that it needs to be re-formed; meaning that either the divergence between the players is higher than the upper limit or the divergence is less than the lower limit which means that the players are too similar in their levels. The team formation and re-formation technique is presented in Fig. 5. This approach avoids situations where some team members are significantly better than others as this discourages less proficient players from engaging (as discussed earlier). The proficiency of players in the team should also vary so that they do not lose the encouragement to improve their performance. However, on commencement, with no gameplay history for any players, team formation results in a random team allocation as all levels are zero, and through player matching the teams are later re-formed to create balanced teams.

As mentioned previously, the decisiveness index represents how effective a player’s decision is in forming the team decision and is an indication of the player’s competence in the game. It is calculated using the player’s level, score and rank within a team based on their loyalty. Equation 2 exemplifies how it is calculated in a decision-making game, for a player ($Player_m$) within a team, where $SortedTeam$ is the players of the team sorted in descending loyalty. Once a formed team has begun to play, the decisiveness index of each player is calculated using their level, score and loyalty (C3). In cases where all the players within the team are new to the game, and thus have a score of zero, then the decisiveness index for all is assigned equally (C1). The decisiveness index of a new player (player with a score of zero) who joins an existing team (whose members have already been playing the game and have score values) is calculated by assuming the new player has the same score and level as the weakest player ($Player_w$) in the team (C2).

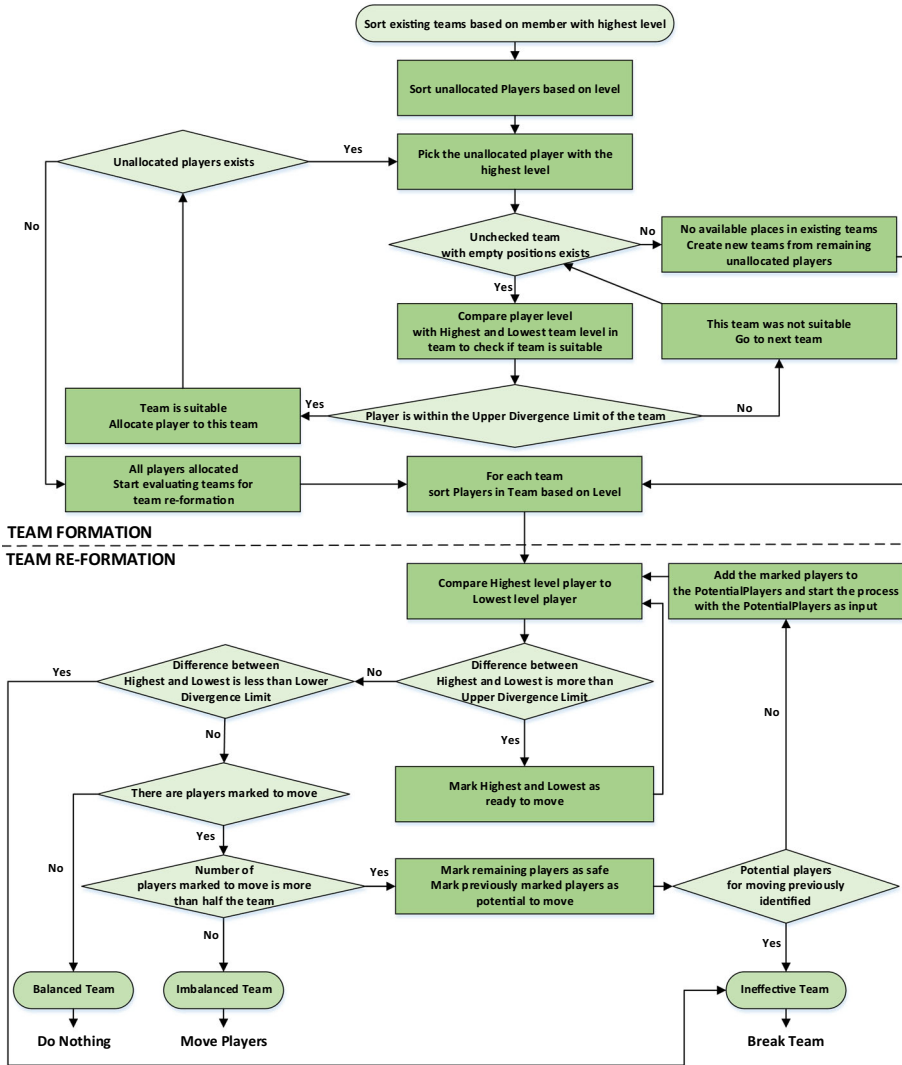


Fig. 5 Player matching for team formation and re-formation

$$DecisivenessIndex_{Player_m} = \begin{cases} \frac{1}{Team\ Size} & C1 \\ \frac{(Level_{Player_w} \times Score_{Player_w}) \times Rank_{Loyalty}(Player_m)}{\sum_{i=1}^{Team\ Size} (Level_{Player_i} \times Score_{Player_i}) + Level_{Player_w} \times Score_{Player_w}} & C2 \\ \frac{(Level_{Player_w} \times Score_{Player_w}) \times Rank_{Loyalty}(Player_m)}{\sum_{i=1}^{Team\ Size} (Level_{Player_i} \times Score_{Player_i})} & C3 \end{cases}$$

$C1$: All players in Team are new, $C2$: Only $Player_m$ is a new player, $C3$: No new players in Team

$Player_m, Player_w \in Team$

$Player_w = P \in Team : Level_{Player_w} = \text{Min}_{Level}(Team) \wedge Score_{Player_w} = \text{Min}_{Score}(Team)$

$Rank_{Loyalty}(Player_m) = \text{Index of } (Player_m) \text{ in Sorted Team : TeamSortedonLoyalty}_{Player}$ (2)

The team leader among players is selected as the player with the highest decisiveness index within the team by considering the score and level of all players in the team as suggested by Yang et al. (2015) and Daylamani-Zad et al. (2016). The player with the highest combination of level and score is announced as the leader (the most engaged and best performing player in the team). This is a dynamic selection process that may change after every interaction. Once selected, a leader can send messages to team members to advise them on game strategy and players are made aware of their leader’s decisions.

To exemplify the decisiveness index, we have chosen to present the scenario for a decision-making game. While players are scored based on their individual decisions, the decisions of the players in a team are combined into a team decision using an adaptation of a weighted majority social choice function as described by Kolter and Maloof (2003) and Yager (1988). This provides the players with the opportunity to learn from each other, collaborating at a level higher than just the game. Equation 3 illustrates the high-level approach to deriving team decisions (Daylamani-Zad et al. 2016). *Decisions* is the vector of all decisions (d_j) submitted by the players. *UniqueDecisions* is a normalised vector representing all the unique alternatives by removing the duplicate decisions from *Decisions*. *DW* (Decision Weights) is the vector representing the weight of each alternative by amalgamating the Decisiveness Index of players who have voted for each decision. With these vectors, a 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.

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

$$UniqueDecisions \subseteq Decisions$$

$$w_i = \sum DecisivenessIndex_{player_j} | Decision_{player_i} = d_i \wedge d_i \in UniqueDecisions$$

$$Decision_{Team} = d_i \rightarrow \begin{cases} w_i = Max(DW) & \exists Max(DW) \\ w_i = Max(DW) \wedge d_i = Decision_{Leader} & \exists! Max(DW) \end{cases} \quad (3)$$

3.4 Reflective personalisation through utility-based agents

Reflective personalisation requires that all players be guided towards an ideal state. In order to move players effectively through different states, a utility-based agent architecture (Russell and Norvig 2009) has been designed as such agents choose among multiple actions that are capable of satisfying a particular goal. The utility-based agent uses utility to self-evaluate its own performance, therefore enabling reflective personalisation. Our proposed architecture is inspired by the proposal-critique-driven agent architecture from Orkin and Roy (2009), Singh and Minsky (2003) and Wang and Zhao (2014) where the agents interact with humans in virtual environments. However, while we work within their overall framework, we modify their approach so that the Collective Memory accommodates player and team profiles, player evolution, and personalisation-condition rules, the action selector is replaced with an Arbitrator that proposes and critiques personalisation solutions, and the Blackboard caters specifically to player activities, states and actions as well as personalisation statuses, utilities and a chronicle. In this way, human gameplay is used to learn and generate personalised recommendations. The proposed architecture therefore learns and generates social behaviour from human gameplay in the form of actions.

The architecture is presented in Fig. 6 and indicates how it sits within the overall reflective Lu-Lu architecture from Fig. 2. It is a multi-agent architecture where each player has an agent that is responsible for its personalisation. When a player starts the game a new instance of an agent is created for the player. This agent is tasked with sending personalised communications

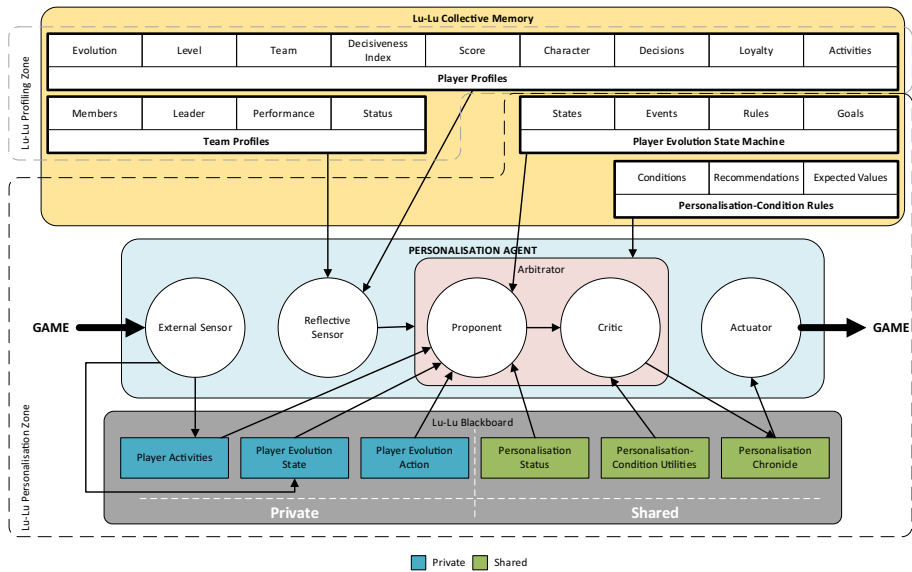


Fig. 6 Utility-based agent architecture for reflective collaborative game personalisation

to this specific player only. The Collective Memory is not affected directly by the agents and therefore acts as an external component in the architecture while the Blackboard is affected only by the agents and therefore is designed as an internal component of the architecture. All the information in the Collective Memory is accessible to all agents at all times. Yet, the Blackboard is divided into two areas, such that each agent has a private area in the Blackboard that is only accessible to them which they use to post information relating to their specific player, while the shared area is accessible (retrieval only) to all the agents.

Player Profiles and Team Profiles are implemented to make personalised recommendations that help the player improve in gameplay and increase player engagement. All agents access the Lu-Lu Collective Memory component of the architecture which includes all Player Profiles, Team Profiles, a Player Evolution State Machine and Personalisation-Condition Rules. The latter define the personalised recommendations available based on the conditions of the player such as their current state and the event they entered their current state with. The agents monitor their recommendations and the results of their recommendations to exhibit reflective behaviour.

The External Sensor in the agent receives the information from the game needed to analyse the player. This information arrives from the Warden and, similar to elements in the Adjustment Zone, is customised to cater for the specific game that the agent has been implemented for, e.g. for the supply chain game (presented later, in Sect. 4.1), the external sensor is activated at the end of each round and is waiting for results from the round directly from the game (these include player's current state once a round has been consolidated as well as the event that assigned them to this state). This information is passed to the Blackboard along with the player and team profiling information retrieved by the Reflective Sensor. Based on this information, the Proponent recognises the goal state that the agent would want the player to enter. Then the Proponent chooses the available Personalisation-Condition Rules, checks for the recommendations made to players with similar conditions, and evaluates if these recommendations have been reaching their expected values and at which level. All this

information is then passed on to the Critic, who chooses the rule with the best utility and passes it to the Personalisation Chronicle, who tailors the rule to the current player and passes it to the Actuator, who executes the personalised recommendation to the game.

3.4.1 Steering players towards goal states

The Player Evolution State Machine (described earlier in Sect. 3.2) is key to providing guidance within the reflective personalisation as it stores information concerning all the states, the events that transition between the states, and the actions that trigger these events. It also stores the information about goal states for each state, which are the states the player should be promoted to from their current state. The goal states for each state are set by non-player stakeholders (such as those discussed in Sect. 2.3) and can be modified at any time. All information is stored as rules within the architecture. Equation 4 presents examples of the rules storing the transitions and Eq. 5 illustrates the rules suggesting the next goal states. Rules in Eq. 4 are presented as *if* Conditions *then* Event ($\Rightarrow E1$), where \therefore indicates the meaning of the event. In Eq. 5, the rules are presented as *if* State *then* State(s) represented by \Rightarrow .

$$\begin{aligned} \text{RULE} : (\text{State} \equiv \text{Minimalism}) \& (\text{Player Activity} \equiv \text{Inactive}) \\ \Rightarrow E1 \therefore \text{Move to Minimalism} \end{aligned}$$

$$\begin{aligned} \text{RULE} : (\text{State} \equiv \text{Dominance}) \& (\text{Player Activity} \equiv \text{Active}) \\ \& (\text{Round Score} \equiv \text{Satisfactory Score}) \Rightarrow E8 \therefore \text{Move to 'Laissez - faire'} \quad (4) \\ \text{Host} \Rightarrow \text{'Laissez - faire'} \parallel \text{Nanny} \\ \text{Nanny} \Rightarrow \text{Dominance} \\ \text{Lacklustre} \Rightarrow \text{'Laissez - passer'} \quad (5) \end{aligned}$$

The rules that abstract the recommendations for players based on their current and previous performance, which are stored in the Personalisation-Condition Rules component, can also be dynamically changed by non-player stakeholders. The rules are based on the player's current evolution state, the event that they entered this state by and the previous recommendation rule that had been applied to them. Equation 6 illustrates some example rules. Each rule is formed of two parts, the conditions and the recommendation, where the conditions lead into the recommendation and each recommendation is represented as R_n and $Event$ represents the event that causes a state change.

$$\begin{aligned} R07 : (\text{State} \equiv \text{'Laissez - faire'}) \& (\text{Event} \equiv E15) \& (\text{Previous Recomm} \equiv R19) \\ \Rightarrow \text{Double points if "Player" scores above "Max(PreviousDailyScore)" today} \\ R08 : (\text{State} \equiv \text{'Laissez - faire'}) \& (\text{Event} \equiv E15) \& (\text{Previous Recomm} \equiv R19) \\ \Rightarrow \text{"Player" can double guess today} \\ R23 : (\text{State} \equiv \text{'Laissez - faire'}) \& (\text{Event} \equiv E13) \& (\text{Previous Recomm} \equiv R52) \\ \Rightarrow \text{"Player" can get double points if he plays 4 rounds everyday for 3 days in a row.} \quad (6) \end{aligned}$$

These rules guide the agents to send a suitable message to the player and take into account the player's current state, the event that has brought the player into this state and any previous recommendations in order to make a decision. The player's current state is an important factor in making any recommendations; as mentioned earlier, the current state is what drives any guided recommendation. The event that has brought the player to this state is key; for example, the player in 'Laissez-faire' who has come from 'Laissez-passer' is very different to

the player in ‘Laissez-faire’ who has come from Host. Finally, the previous recommendation is also an important factor to consider. If the player has been offered a similar type of recommendation but is not responding well to it, continuing to make similar recommendations is probably not going to improve their performance and therefore there is a need for a change in the recommendation strategy. Yet, if a player is responding well to a certain type of recommendation then they should be targeted with the same strategy. The final section of each rule is the body of the recommendation shown to the player and requiring substitution with values that relate to the specific player.

The expected values have different scores that allow the agent to evaluate the effectiveness of each rule and help it self-evaluate by increasing the accuracy of the utility of each rule. Each rule can have four possible expected values scored $\{-v, 0, +v, 2v\}$ based on the level of success. The expected value function for each rule is stored in the Collective Memory. These values are chosen to show the effectiveness of each rule for a player at an exact point in time. The lowest value represents that the rule was not only unable to improve the player but the player is actually performing worse than before. The 0 value represents when the rule has not had any measurable impact on the player while the $+v$ value is a representation of the rule having had a positive impact on the player. The final high value of $2v$ is assigned to the rule when not only the recommendation has led to improved performance but it has had such a considerable impact that the player has moved to the goal evolution state as a result of this rule, hence registering double the expected value of just improved performance. In our implementation we have assigned $v=5$ to allow for more precision and granularity when combining the expected utilities of rules to score the overall performance of the rule and to avoid long fractional parts which due to rounding can cause false positives. Equation 7 illustrates an example of the expected value function of Rule 07 from Eq. 6 for a player.

$$E_{R07}(Player_i) = \begin{cases} Lowscore || Semi-Active || Inactive & -5 \\ Averagescore \wedge Active \wedge Samescore & 0 \\ Averagescore \wedge Active \wedge Betterscore & 5 \\ Newstate \equiv Dominance & 10 \end{cases} \quad (7)$$

Each time a recommendation rule is applied, the agent monitors its impact to assign a value based on expected values such as those in Eq. 7. These scores are summed to make what is known as the utility of each recommendation as discussed in the next section and presented in Eq. 9. The agent uses this self-evaluation process to increase the effectiveness of its recommendations.

3.4.2 Self-evaluation of personalisation effects

The Personalisation Agent is the processing core of the proposed architecture. As shown in Fig. 7, the External Sensor takes information from the game consisting of the current decision, current round results and the current game state which is transferred to the Blackboard. The information that is directly changed by the agents are stored in the Blackboard. Not all of this data is shared amongst all agents, rather it acts as a memory for each agent where some of the data is shared. Player Activities, Evolution State and Evolution Action are only accessible to the specific agent assigned to a player but Personalisation Chronicle, Personalisation Status and the Utilities are shared amongst all agents as any change in them could affect other agents. Player Activities, Evolution State and Evolution Action which are retrieved from the Player Profile, are private to each agent. This information is only useful for tracking the progress of a specific player and therefore this level of granularity is only useful to the agent in charge of personalisation for this player.

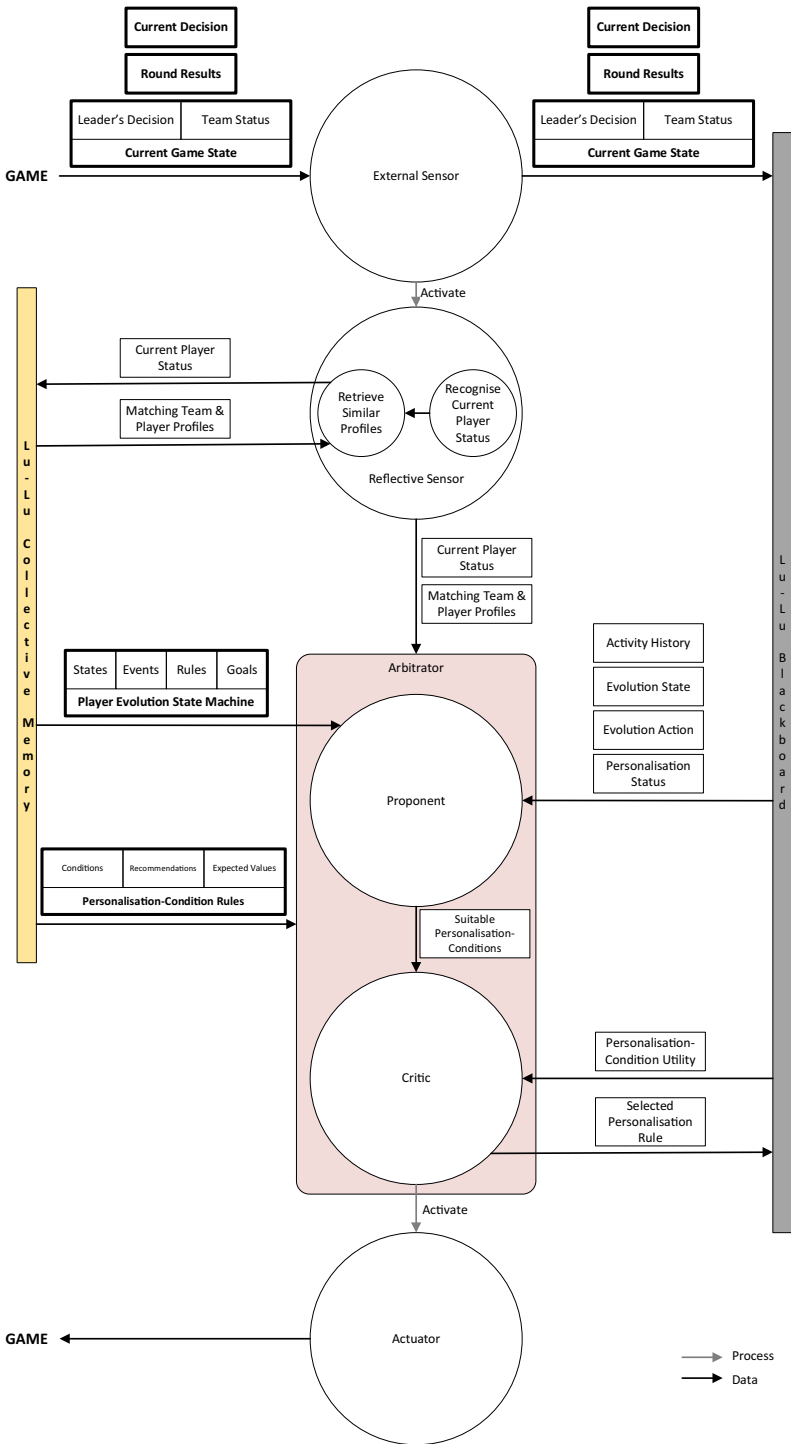


Fig. 7 Agent process and data flow

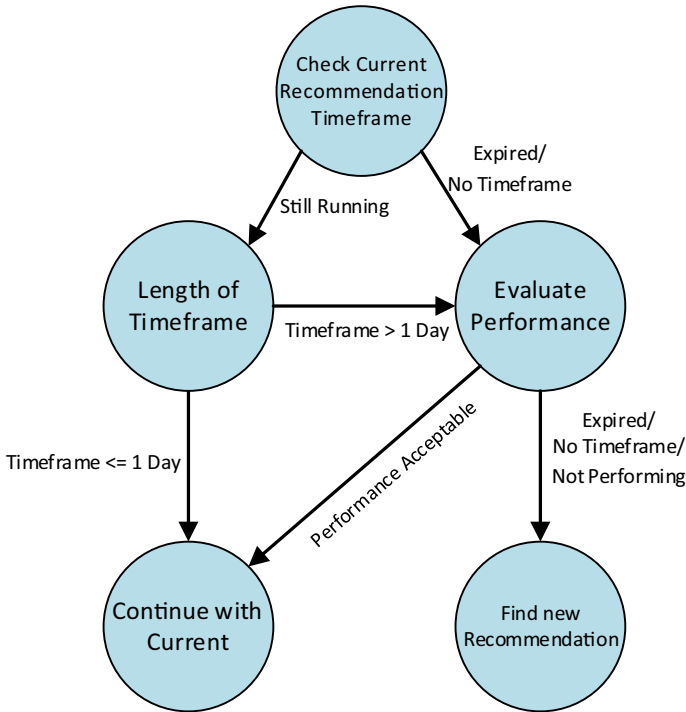


Fig. 8 Proponent process

The Reflective Sensor retrieves the information of all Player Profiles and Team Profiles which match the current player’s status. The reflective information, the current external information along with the history of external player information are passed to the Arbitrator.

The Proponent, illustrated in Fig. 8, uses this information to evaluate if the current recommendation is still valid or has expired. Personalisation Status within the Blackboard stores the current status of the personalisation rule that has been applied to the player. If the timeframe of the rule has passed, it holds the result of the rule based on the expected values. Otherwise, this holds the current status of the rule such as time left and the current progress of the player according to the expected values.

The examples in Eq. 6 show that some recommendations might have a timeframe such as one day. In cases when the timeframe is no more than one day, the agent allows the time frame to expire before re-evaluating the personalisation. Yet, if the timeframe is longer than one day, if there is no timeframe, or it has expired, then the Proponent evaluates the performance of the recommendation and stores it. The Proponent then finds the most suitable rule or rules within the Personalisation-Condition Rules, as illustrated in Eq. 8, where, for example, $State_R$ represents the State of rule R and $State_{Player_i}$ represents the State of $Player_i$.

$$ProposedRules_{Player_i} = \left\{ R \mid \begin{array}{l} State_R = State_{Player_i} \wedge \\ Event_R = Event_{Player_i} \wedge \\ PreviousRecomm_R = PreviousRecomm_{Player_i} \wedge \\ NewState_R = GoalState_{Player_i} \end{array} \right\} \quad (8)$$

The identified most suitable rule or rules are passed to the Critic process which evaluates these against their utilities. The Critic is the evaluative unit of the reflection/self-evaluation process, taking in the rules and evaluating their previous performance, finally choosing the most effective rule for a player in each instance. The Critic’s operation is represented in Eq. 9 and Eq. 10, calculating the utility of each rule for a player and evaluating them based on the calculated utilities. The utility of each rule reflects its performance as it has been applied to players. It is calculated as the sum of the results of the expected value function (as presented in Eq. 7) of the same personalisation rule having been applied to players of a similar status as presented in Eq. 9, where $E_{R_n}(Player_k)$ represents the expected value of rule R_n when applied to $Player_k$. Similarity of player status is determined based on team and individual profiles, which would include their evolution state and evolution history. Players who have recently been in similar evolution states are more likely to be assessed as being similar. The score and level are also used for the similarity search which is performed on a need basis and follows a fuzzy approach to find the minimum required players for the agent to be able to make a conclusive choice. Players are filtered based on their evolution history, and only players who have been in the same state as the current player are selected. Then, a fuzzy value is assigned to these players based on evolution history, level, score and their profiles such as leadership status, activity and loyalty. For example, players who went through the same event as the current player to enter the current player’s state are considered more similar to this player (the farther away their source state is, the lower their similarity).

$$Utility(R_n) = \sum_{k=0}^{|Players|} E_{R_n}(Player_k)$$

$$Player_k \in Players = \left\{ P \mid (P \in All_Players) \wedge (\exists S \in \{ \forall Status_p \} \mid S \approx Status_{Current\ Player}) \wedge (R_n \in \{ \forall Recommendations_P \}) \right\} \tag{9}$$

$$CritiquedRule = R \Rightarrow Utility(R) = \max Utility(R_j) : R_j \in ProposedRules_{Player_i} \tag{10}$$

The Personalisation Chronicle within the Blackboard holds a history of all the personalisation rules applied to the player with the most current flagged. It is modelled in UsageHistory DS using UserAction DS. Each action references to its corresponding rule using Action-DataItem and XPath. Figure 9 presents examples of how the Personalisation Chronicle is modelled. The rule with the highest utility is selected by the Critic (as defined in Eq. 10) and sent to the Actuator which in turn applies the rule by the in-game representor of the agent.

4 Evaluation of the proposed reflective personalisation approach

To assess the performance and engagement effects of comprehensive profiling and a utility-based proposal-critique-driven agent in support of reflective personalisation in collaborative games, as proposed in Sect. 3, we undertook an experiment which is described in this section. We define performance as the player’s score and rate of scoring. In order to counter the effects of recommendation messages that activate reward multipliers as incentives, such as double scoring, we normalise the scoring data captured during the experiment against a score log that stores the round score of each player without any reward multipliers. This ensures that they are consistently comparable across players based on their performance. For engagement, we use the definition applied in the evolutionary state machine in categorising the players into

```

<UserActionList>
  <ActionType href="urn:LULU:cs:PActionCS:1.2">
    <Name>Personalisations</Name>
  </ActionType>
  <UserAction>
    <ActionTime>
      <GeneralTime>
        <TimePoint>2016-06-11T18:54:21</TimePoint>
        <Duration>PT24H0M0S</Duration>
      </GeneralTime>
    </ActionTime>
    <ProgramIdentifier>-5</ProgramIdentifier>
    <ActionDataItem href="PersonalisRules.mp7.xml#id-71"></ActionDataItem>
  </UserAction>
  <UserAction>
    <ActionTime>
      <GeneralTime>
        <TimePoint>2016-06-13T15:02:14</TimePoint>
        <Duration>PT12H0M0S</Duration>
      </GeneralTime>
    </ActionTime>
    <ProgramIdentifier>5</ProgramIdentifier>
    <ActionDataItem href="PersonalisRules.mp7.xml#id-74"></ActionDataItem>
  </UserAction>
  <UserAction>
    <ActionTime>
      <GeneralTime>
        <TimePoint>2016-06-18T06:00:43</TimePoint>
        <Duration>PT24H0M0S</Duration>
      </GeneralTime>
    </ActionTime>
    <ProgramIdentifier>OnGoing</ProgramIdentifier>
    <ActionDataItem href="PersonalisRules.mp7.xml#id-8"></ActionDataItem>
  </UserAction>
</UserActionList>

```

Fig. 9 Personalisation history in MPEG-7

Active, Semi-Active and Inactive categories (Sect. 3.2). The engagement of a player is based on how often they play the game and how long their gameplay sessions last.

4.1 Method

A variant of the decision-making Supply Chain or Beer Game (Sterman 1989) was implemented using Unity and C#. There are four sequential stages in this chain related to producing and delivering units of beer: Factory, Distributor, Wholesaler, and Retailer. Each player starts at one stage and tries to achieve the goal of maximising profit while minimising long-term system-wide total inventory cost in ordering from their immediate supplier. Players collaborate in their stage to deal with orders arriving. Deciding how much to order may depend on a player's own prediction of future demand by their customer based on their own observations. Other rules may also be used by players in making decisions on how much to order from a supplier (Kimbrough et al. 2002; Yuh-Wen et al. 2010).

Figure 10 shows player "JohnSmith2014" at the Wholesaler stage in a team of six players. The game interface comprises the game area, where the main game is played, and the player

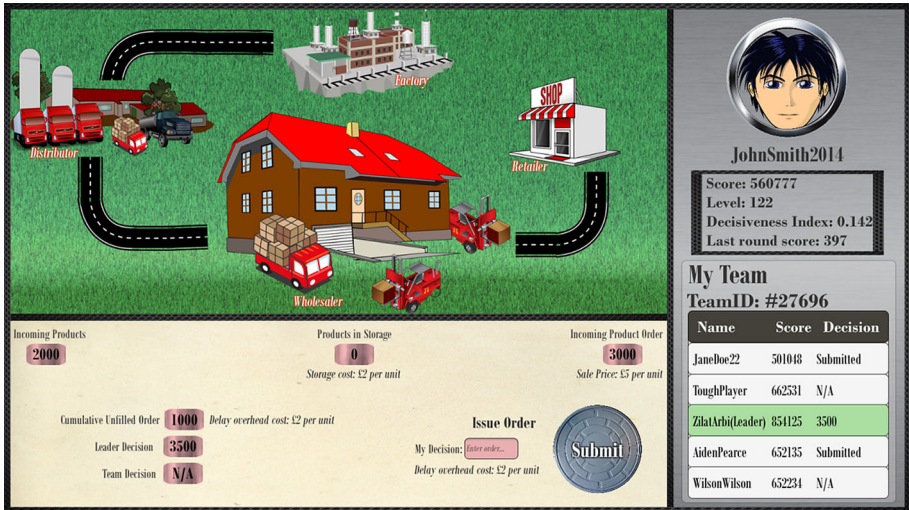


Fig. 10 A player is prompted for a decision at Wholesaler stage

area, where the player’s stats and team information are displayed. A Leader tag on the team list identifies the leader and their decisions are shown in the team members’ table and game area. In addition, the leader’s decision is included both in the “My Team” area and in the left-hand side of the game interface so that players are aware of the decision made by the strongest player in the team. A player can view their team’s scores and decisions even when they have not made any.

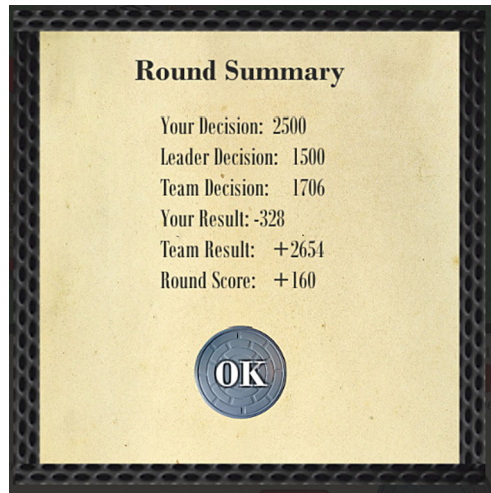
A player’s decisiveness index, along with their level and score from the last round are also included in the player area. Once all players in a team have made their decisions, the team decision is derived and executed. Once the result from a decision has been applied, a new round begins. The result of a round normally changes the scores and may result in changing the current recommendation, the level and decisiveness index for players as well as resulting in changes in team structure and formation, i.e. appointment of a new leader, change of team for a player or break up of a team.

In the case where the result of a team decision is considerably different to that of an individual player, the impact on the individual score is significant. Figure 11 demonstrates the end of a round during which a player decision has led to a significantly different result to the team. Damage caused by bad decisions results in reduction of a player’s share of the team score. When a player reaches a certain score limit, they will level up.

Players receive various recommendation messages from the games as a result of the events that occur during gameplay. Some of these messages relate to progress they have made in a round (Fig. 11). Others relate to unlocking a new character in a game (Fig. 12a), or relate to team matching (Fig. 12b), or are personalised messages to encourage or help players (Fig. 12c).

To maintain consistent results and isolate the specific effects of reflective personalisation in a collaborative context, we deployed three versions of the game, each sharing the user interface shown above, but with an increasing level of personalisation that reflected the three levels presented in Sect. 2.3, and incorporating dynamic team formation in all three versions:

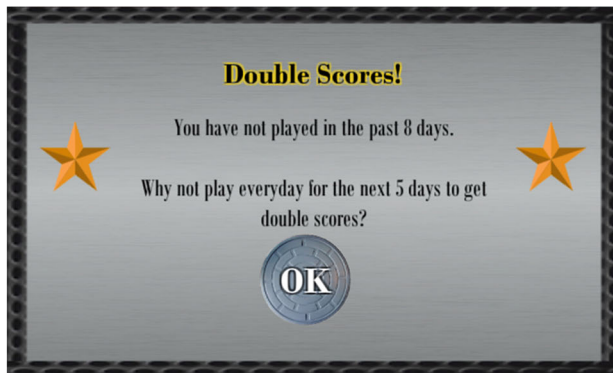
Fig. 11 Different decisions with results



(a)



(b)



(c)

Fig. 12 Progress, team matching and personalised messages. **a** Unlocking a new character in the game. **b** Player matched to a new team. **c** Personalised encouraging message to the player, containing incentives for the player

1. *Beer Game with Reflective Personalisation (REFLECTIVE)* This version implements the reflective personalisation architecture proposed in Sect. 3, and is hypothesised to be the most effective in terms of improving player performance and engagement. In this version, the personalisation guides players to goal states and is self-evaluative. The agent

personalises for individual players within the team while monitoring the effects of the personalisation and taking actions to modify its recommendations as and when necessary in order to guide the players to particular states. For example, the personalisation agent would realise that a player is high-scoring but semi-active, therefore it would need to target this player’s activity and not their scoring. The agent would also take into account that this player was inactive but high-scoring previously. Based on this information, it would shortlist several recommendations. Then the agent would look at the performance of these recommendations, choose the one that has performed the best previously, and apply that to the player.

2. *Beer Game with Guided Personalisation (GUIDED)* The personalisation in this version of the game guides players to goal states but is not self-evaluative. It implements all of the architecture in REFLECTIVE except that the performance of the rules as applied to the players is not taken into account when choosing a recommendation rule. Therefore, while players are guided to particular states, evaluative decision-making is replaced with random choices. This means that in this version of the game, the Critic in the Personalisation Agent has been replaced by a random chooser. The effect of this change on the game is that the utilities and expected value of the rules as presented in Eq. 7, Eq. 9 and Eq. 10 are not used. Instead the Critic only chooses a rule from the proposed rules at random, as shown in Eq. 11, where R represents a rule and \sim represents a random choice. For example, for the player mentioned in the previous example, the agent would shortlist a set of suitable recommendations for them based on their current state and previous history but would choose at random which recommendation to send to them.
3. *Beer Game with Passive Personalisation (PASSIVE)* This version of the game uses the passive version of the Lu-Lu architecture which was developed previously, as presented in Daylamani-Zad et al. (2016). It does not include the Player Evolution State Machine (as in GUIDED and REFLECTIVE) nor the Personalisation Agent (as in REFLECTIVE). Consequently, it does not guide players to particular states and does not self-evaluate its personalisation decisions. Messages are sent to players to encourage improvements in engagement and performance, but unlike REFLECTIVE, these are not critiqued and are triggered by decision trees which do not take player’s historical movements into account. For example, in comparison to the previous examples, the game would merely send recommendations indiscriminately that would try to encourage the player to score better and to be more active, with no specific focus beyond that.

$$CritiquedRule = R \Rightarrow R \sim U(ProposedRules_{Player_i}) \tag{11}$$

These three versions of the game enable the experiment to reveal whether the engagement and performance of the players is improved by the reflective personalisation architecture proposed in this paper. For this purpose, two sets of hypotheses are devised, focused around player engagement (Eq. 12) and player performance (Eq. 13).

$$H_{1E} : \mu_{E_{REFLECTIVE}} > \mu_{E_{GUIDED}} > \mu_{E_{PASSIVE}} \tag{12}$$

$$H_{0E} : \mu_{E_{REFLECTIVE}} \leq \mu_{E_{GUIDED}} \leq \mu_{E_{PASSIVE}}$$

$$H_{1P} : \mu_{P_{REFLECTIVE}} > \mu_{P_{GUIDED}} > \mu_{P_{PASSIVE}} \tag{13}$$

$$H_{0P} : \mu_{P_{REFLECTIVE}} \leq \mu_{P_{GUIDED}} \leq \mu_{P_{PASSIVE}}$$

90 participants volunteered for the experiment, 49% male, 51% female, aged 20-39. Participants were of various backgrounds and were randomly divided into three groups, with each group tasked with playing one of the three versions of the game (REFLECTIVE, GUIDED or PASSIVE). This helped to limit any potential bias within the groups, and to ensure that,

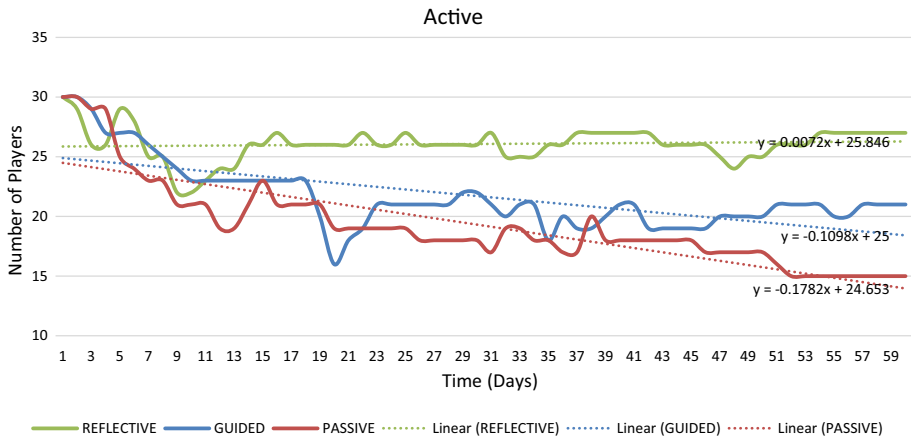


Fig. 13 Active players during the experiment period for the three game versions

for the REFLECTIVE and GUIDED groups, the player evolution state machine would not be skewed towards particular states. Moreover, we initialised the experiment by considering all participants to be active and average scoring, so that no assumptions were made and all commenced the game at the ‘Laissez-faire’ state, subsequently moving through the states depending on their performance compared to their peers. Participants were asked to play their assigned game for a period of two months. Their performance was recorded and monitored but they were not communicated with outside of the automated communications of the game itself. Results were analysed according to player engagement and performance. For engagement, results considered the frequency in which the participants played the game and the length of each gameplay session. For performance, average total player scores were compared across the three different groups in order to evaluate the effect of the key elements of the architecture. A two-tailed paired t test and a Mann–Whitney test were applied to the data to explore the significance of the experiment results based on the null hypotheses (H_{0E} and H_{0P}).

4.2 Results: player engagement

Categorising the engagement of players as active, semi-active, and inactive as defined in Sect. 3.2 helps to monitor the effect of personalisation using the three different games utilised in the experiment. We assume all players are active at the start and therefore aim to retain a higher number, and reduce the fall in the number, of active players. Figure 13, 14, and 15 illustrate the number of players in the three categories in each version of the game during the two-month period of the experiment. Figure 13 indicates that the full implementation of the reflective personalisation architecture (REFLECTIVE) results in a much lower fall in the number of active players, and while there are fluctuations in the active number, overall this is much lower than the other two game versions (REFLECTIVE variance in active players of 1.929 compared to 8.028 for GUIDED and 13.003 for PASSIVE). The charts indicate that REFLECTIVE retains more players in the active category than the other two, while GUIDED outperforms PASSIVE. A similar pattern can be observed in semi-active players (Fig. 14), where GUIDED retains more semi-active players compared to PASSIVE, suggesting that profiling manages to considerably reduce the number of inactive players.

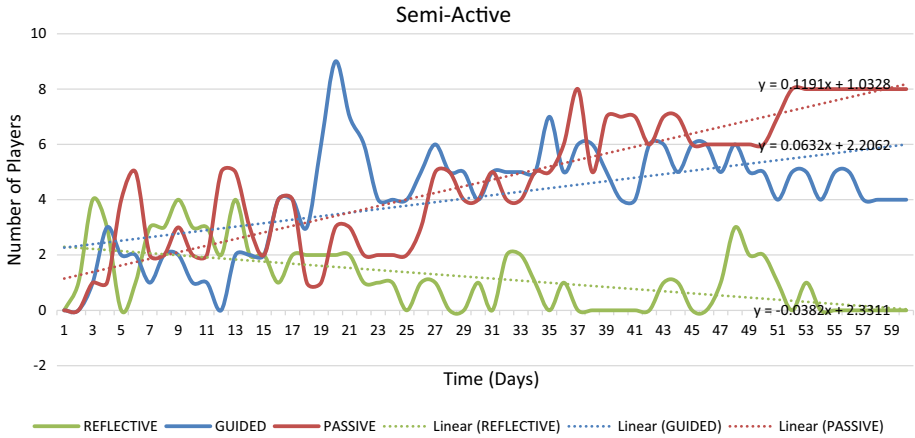


Fig. 14 Semi-active players during the experiment period for the three game versions

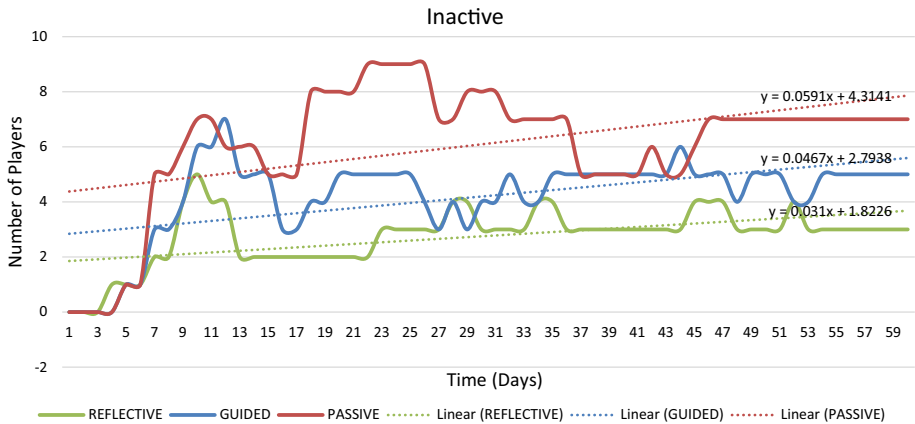


Fig. 15 Inactive players during the experiment period for the three game versions

For semi-active (Fig. 14) and inactive players (Fig. 15), we can observe notable trends in the player engagements. For inactive players, REFLECTIVE indicates a more gradual rise compared to the other two (0.031) and the slope is much steeper for GUIDED (0.0591) compared to PASSIVE (0.0467). For semi-active players, the trend is rising for GUIDED (0.0632) and PASSIVE (0.1191), but for REFLECTIVE there is a falling trend (-0.0382), suggesting that, overall, the trend in the number of semi-active players in the REFLECTIVE game is diminishing. This correlates with the trends in the active players (Fig. 13), where in the REFLECTIVE game there is a rise in the number of active players by a value of 0.0072. Consequently, it is possible to conclude that REFLECTIVE is more convincingly converting semi-active players back to being active players. The trend in the active players in both GUIDED and PASSIVE is falling with respective values of -0.1098 and -0.1782, suggesting that these two games do not perform as well as REFLECTIVE in converting semi-active players back to being active. There is a spike in Fig. 14 between days 19 and 25; this was discovered to be related to a cluster of players in the GUIDED group whose

Table 5 Mean of engagement for the three game groups

	Mean		
	Active	Semi-active	Inactive
REFLECTIVE	26.07	1.17	2.77
GUIDED	21.65	4.13	4.22
PASSIVE	19.22	4.67	6.12

other commitments prevented them from engaging over this period, resulting in them being profiled as semi-active.

Four participants dropped out of the experiment by day 12; two from the REFLECTIVE group and two from the PASSIVE and GUIDED groups. By disregarding the players who dropped out, it is possible to see that the REFLECTIVE game has managed to retain all players but one in the non-inactive categories of active and semi-active (93%) while GUIDED manages 86% of players and PASSIVE reaches 79%. Table 5 illustrates the means for each game; based on these values, REFLECTIVE has performed 35% better than PASSIVE in retaining the active players while GUIDED has performed 13% better. Similarly, REFLECTIVE has performed 54.8% better than PASSIVE in preventing the players becoming inactive. Finally, GUIDED outperforms PASSIVE by 31%.

Based on these preliminary results, it is possible to conclude that REFLECTIVE outperforms both other games and GUIDED also performs much better than PASSIVE in terms of player engagement and therefore H_{IE} may be initially established. Since the participants playing the three different games were different, the results were tested against the null hypothesis H_{0E} . A two-tailed paired t-test was performed on the data. The results are presented in Table 6. Apart from one instance, $p=0.000$ and thus the null hypothesis may be rejected. In the case of semi-active players for GUIDED-PASSIVE, $p=0.020$, which is also significant (with $p=0.02$ even a one-tailed comparison would have given a result of 0.040) and it is safe to reject the null hypothesis in this case as well. As we can rule out the null hypothesis in both cases, it is possible to conclude that the results are sufficiently different. Therefore, to a greater extent REFLECTIVE, and to a lesser extent GUIDED, can be considered to improve player engagement compared to PASSIVE.

Figures 16, 17, and 18 further visualise the results as box plots and indicate a clear distinction between the means and distributions amongst the three personalisation approaches. REFLECTIVE has the highest mean and lowest deviation for active players, followed by GUIDED. Results for semi-active and inactive players also seem to confirm the hypothesis since the REFLECTIVE game has the least number of players in these categories followed by GUIDED, and PASSIVE has the most number of players in these categories. There are some consistent outliers (at values 29 and 30) across all three approaches, but given that we start the experiment with the assumption that every player is active and as time goes by this number will drop, this is to be expected (we would only have near-100% active players at the start of the experiment when the trends have not yet been established).

Similarly, the outliers in the inactive plot are from the start of the experiment where no player was yet considered to be inactive (and gradually the number of inactive players increases until a pattern establishes). The fluctuation in the number of active players for REFLECTIVE also reveals three low outliers, including the lowest) which occurred at discreet instances relating to the point in time when the aforementioned two players completely dropped out of the experiment. This coincided with a long weekend which explains the low number of active players, but this occurrence has been an isolated incident and not part of

Table 6 Paired *t* test results for engagement

	Paired differences				t	df	Sig. (2-tailed)	
	Mean	SD	SE mean	95% CI of the difference				
				Lower				Upper
<i>Active</i>								
REFLECTIVE-GUIDED	4.417	2.936	.379	3.658	5.175	11.652	59	.000
REFLECTIVE-PASSIVE	6.850	3.663	.473	5.904	7.796	14.485	59	.000
GUIDED-PASSIVE	2.433	2.094	.270	1.892	2.974	9.001	59	.000
<i>Semi-active</i>								
REFLECTIVE-GUIDED	-2.967	2.604	.336	-3.639	-2.294	-8.826	59	.000
REFLECTIVE-PASSIVE	-3.700	3.321	.429	-4.558	-2.842	-8.631	59	.000
GUIDED-PASSIVE	-.733	2.378	.307	-1.348	-.119	-2.388	59	.020
<i>Inactive</i>								
REFLECTIVE-GUIDED	-1.450	1.032	.133	-1.717	-1.183	-10.882	59	.000
REFLECTIVE-PASSIVE	-3.350	1.745	.225	-3.801	-2.899	-14.871	59	.000
GUIDED-PASSIVE	-1.900	1.526	.197	-2.294	-1.506	-9.644	59	.000

the pattern hence they represent themselves as outliers. Together with Fig. 13, the box plot confirms the increase in the number of active players overall.

The boxplots also highlight that the data is not evenly distributed and is skewed. A Shapiro–Wilk normality test (Table 7) confirmed that the data significantly deviated from a normal distribution. Therefore, we performed a further analysis using the Mann–Whitney test for nonparametric data. Table 8 presents the results along with the 2-tailed asymptomatic significance of the results. Aside from one case, significance is 0.000 which allows rejection of the null hypothesis. We may therefore conclude that REFLECTIVE better engages the players than GUIDED, while GUIDED better engages the players than PASSIVE. In the case of GUIDED-PASSIVE, for semi-active players the significance is 0.073 which is > 0.05 and thus we cannot decisively reject the null hypothesis. However, the t-test showed a much lower significance value. The nature of the data as presented in Fig. 14 shows that, while the data for REFLECTIVE and PASSIVE is evenly distributed and is not skewed, the data for GUIDED is skewed. Furthermore, GUIDED and PASSIVE means are close, causing the significance of the result to be lower than the threshold. Consequently, the difference in the number of semi-active players between the GUIDED and PASSIVE approaches is not statistically significant and these two approaches are not as successful as REFLECTIVE in engaging semi-active players to become active. The GUIDED personalisation was, in fact, almost as successful as PASSIVE is engaging these players.

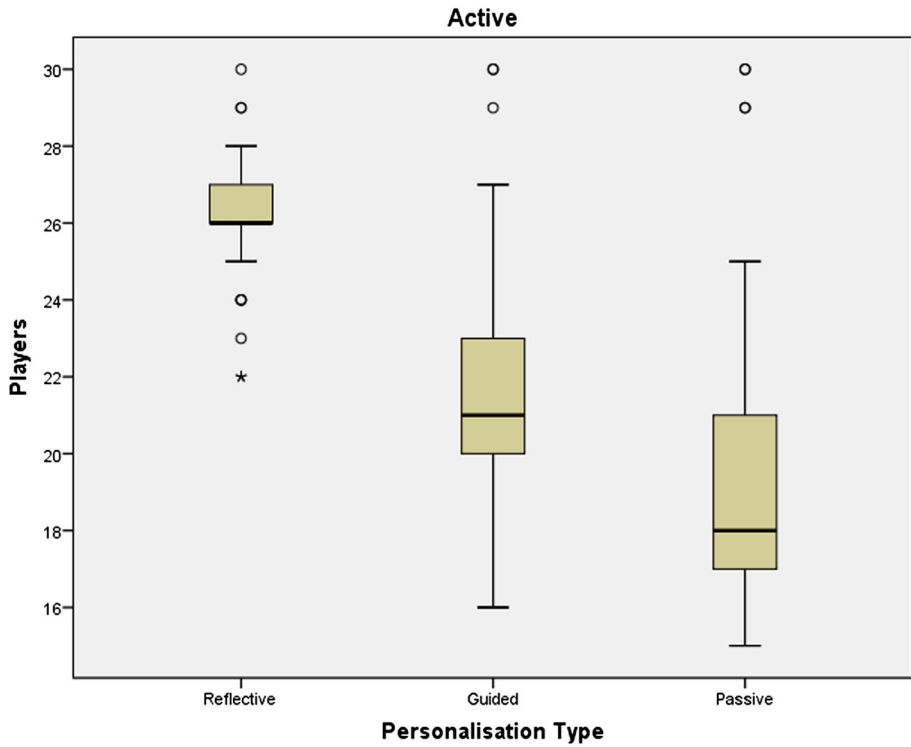


Fig. 16 Active players plotted for the three games

Considering these results, we can confirm H_{IE} and demonstrate that reflective personalisation incorporating guidance to player goal states and self-evaluation has had a positive impact on player engagement. The added player evolution state machine in the GUIDED approach has shown a clear improvement compared to PASSIVE. Aiming to guide players into specific evolution states based on their performance and engagement and sending personalised messages that are based upon their current state has shown considerable improvement in player engagement with the game, retaining more active players and converting players back from less active positions compared to PASSIVE.

More notably, the use of self-evaluative agents through the Critic has had a higher impact on player engagement. As observed in Fig. 16, 17, and 18, REFLECTIVE has engaged players considerably more than GUIDED, highlighting the impact of the Critic and the self-evaluation of reflective agents in engaging the players with the game. Evaluating their own performance based on the players' evolution state, the agents have been able to send more effective personalised messages to the players resulting in a much better retention of active players.

4.3 Results: player performance

The performance of the players can be evaluated by their scores in the game as they increase during gameplay. It is a good indicator of the performance of the players during gameplay and also the effectiveness of the game in improving player decision making. As the score (Lu-Lu

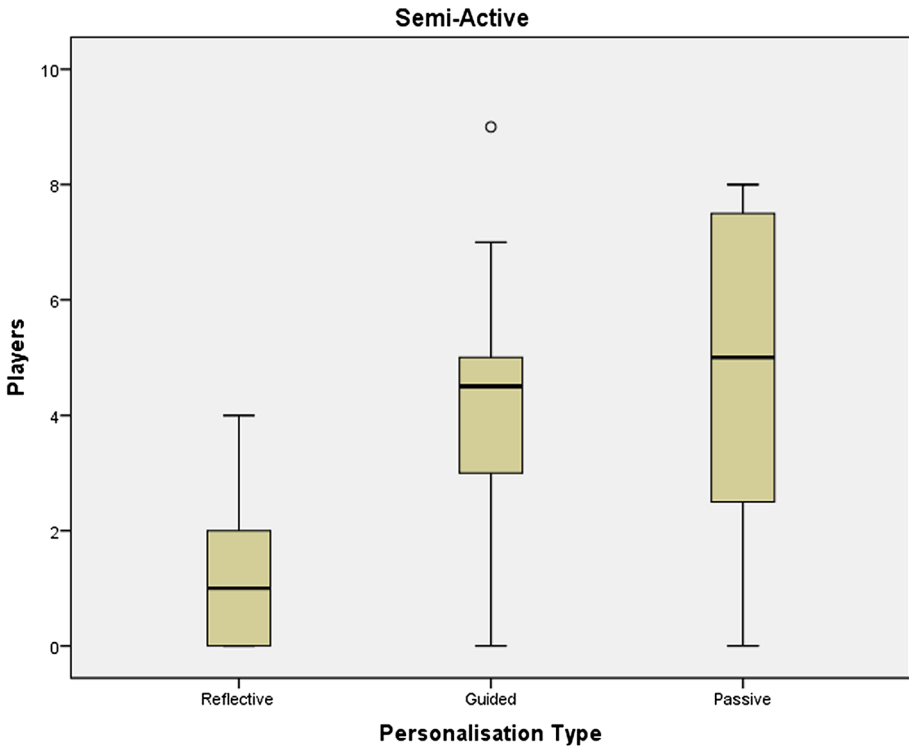


Fig. 17 Semi-active players plotted for the three games

score) can be both positive and negative, the rise and fall of the score would correspond closely with the performance of players. Figure 19 illustrates the average total score of the player within each group (the player dropouts mentioned earlier have been excluded). The line charts indicate that all three game versions have had a positive impact on the average performance of the participants. However, the REFLECTIVE group with a trend of 1548.1 has had a much steeper improvement compared to the other two groups. The GUIDED group (1383.6) is also much closer to REFLECTIVE’s performance compared to the PASSIVE group’s 912.14, highlighting that the profiling and Player Evolution State Machine have had a stronger impact on performance compared to REFLECTIVE. Comparing the means across the three groups (Table 9), it is evident that REFLECTIVE and GUIDED have had a stronger impact on improving the players’ decision making and thus performance (REFLECTIVE has performed 52% better than PASSIVE and GUIDED has performed 31.6% better than PASSIVE).

We may therefore initially conclude that REFLECTIVE performs much better than GUIDED or PASSIVE with respect to increasing player performance, and GUIDED also performs much better than PASSIVE, therefore H_{1P} appears to be supported by the results. Again, a null hypothesis H_{0P} was considered, then a two-tailed paired t-test was performed on the data. Results are presented in Table 10. For all tests, $p=0.000$ and thus the null hypothesis is rejected. It is therefore possible to conclude that the results are significantly different enough and thus, to a greater extent REFLECTIVE and to a lesser extent GUIDED, do indeed improve player performance compared to PASSIVE.

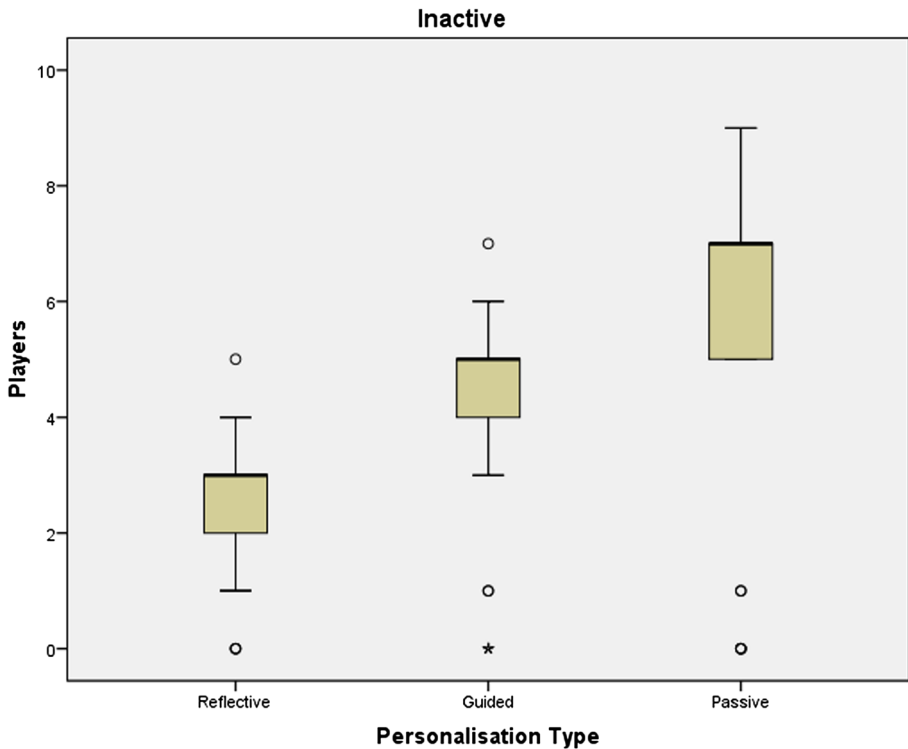


Fig. 18 Inactive players plotted for the three games

Table 7 Shapiro–Wilk normality test for engagement

	Statistic	<i>df</i>	Sig.
<i>Active</i>			
REFLECTIVE	.886	60	.000
GUIDED	.868	60	.000
PASSIVE	.840	60	.000
<i>Semi-active</i>			
REFLECTIVE	.844	60	.000
GUIDED	.929	60	.002
PASSIVE	.909	60	.000
<i>Inactive</i>			
REFLECTIVE	.863	60	.000
GUIDED	.756	60	.000
PASSIVE	.795	60	.000

Figure 20 provides the results in box plot format. The figure indicates that there is an improvement in player performance amongst the three approaches. REFLECTIVE has a slightly higher mean than GUIDED and PASSIVE, while GUIDED shows a clear increase in player performance compared to PASSIVE which reconfirms the observation on Fig. 19. The distribution of REFLECTIVE and GUIDED is wider than PASSIVE due to the total aver-

Table 8 Mann–Whitney test for engagement

	Mann–Whitney <i>U</i>	Wilcoxon <i>W</i>	<i>Z</i>	Asymp. Sig. (2-tailed)
<i>Active</i>				
REFLECTIVE-GUIDED	385.000	2215.000	− 7.505	.000
REFLECTIVE-PASSIVE	257.000	2087.000	− 8.162	.000
GUIDED-PASSIVE	843.000	2673.000	− 5.073	.000
<i>Semi-active</i>				
REFLECTIVE-GUIDED	399.500	2229.500	− 7.445	.000
REFLECTIVE-PASSIVE	384.000	2214.000	− 7.518	.000
GUIDED-PASSIVE	1463.000	3293.000	− 1.790	.073
<i>Inactive</i>				
REFLECTIVE-GUIDED	601.000	2431.000	− 6.477	.000
REFLECTIVE-PASSIVE	351.000	2181.000	− 7.711	.000
GUIDED-PASSIVE	609.000	2439.000	− 6.449	.000

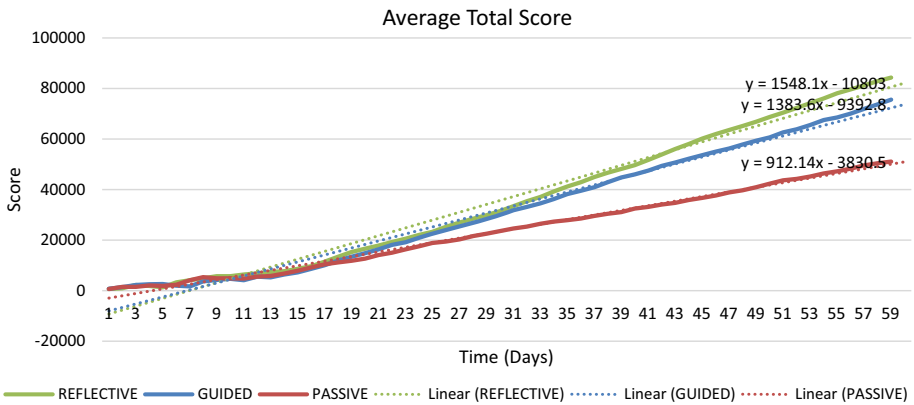


Fig. 19 Average total scores for the three games

age scores being used. While total scores are increasing to higher values over time, scores all commence the experiment at zero; therefore, the higher distribution does not reveal any outliers. This demonstrates that there is a steady increase in the scores and the distribution is well-formed. Based on these observations, we conclude that REFLECTIVE offers improvements in player and team performance over GUIDED and PASSIVE, and in turn GUIDED offers improvements over the PASSIVE approach.

Table 9 Mean performance for the three games used

Mean		
Reflective	Guided	Passive
35,046.633	31,579.75	23,141.62

Both REFLECTIVE and GUIDED have an evident improvement over PASSIVE, illustrating that profiling and the player evolution state machine have had a positive impact on the performance in the game. Players have improved in their decision making when guided through the states which has allowed the players and teams to perform better. The REFLECTIVE mean has had a positive increase of nearly 4000 and average total score has reached a much higher value compared to GUIDED, indicating that reflective personalisation and the implementation of self-evaluation in the agents has had a positive impact player performance. In the Beer Game, a round score can be positive or negative as it is based on loss/profit. Considering that players can actually make a loss and therefore have a lower score than the previous round, the increase in the means becomes more interesting. This indicates that there were fewer negative scores in REFLECTIVE compared to GUIDED and PASSIVE. The data shows that there far fewer negative scores in REFLECTIVE after the eleventh day of the experiment and no negatives were scored in the last four days, while there were still negative scores until the last day in the GUIDED and PASSIVE groups.

Yet, the difference between REFLECTIVE and GUIDED is not as much as anticipated. It is possible to conclude that the self-evaluation in the reflective agents has not had as much of an impact as was foreseen on the performance. This observation warrants further research into implementation of reflective agents with respect to player performance. It is possible that the personalisation messages sent to the players based on their performance were not rich enough to have a considerable impact on their performance. The inclusion of more learning-focused messages, incorporating an expert system into the agent architecture to help the Critic and revising the rules for performance could potentially improve the impact of self-evaluation and reflective agents on performance.

4.4 Limitations

A key assumption of the experiment was that personalisation improves engagement and performance, as reported by repeated studies and was discussed in Sects. 1 and 2. Consequently, the experiment did not utilise a control group but aimed to comparatively evaluate the effectiveness of the three personalisation approaches with respect to engagement and performance.

The main limitation of the experiment is that it was focused on a single, specific collaborative game (the Beer Game). Moreover, the experiment did not test for the specific influence of each individual aspect of the architecture, such as the player and team profiles, the player evolution state machine, and the team formation approach. Where present, they were kept consistent across the three versions of the game. Consequently, while the results are promising, some caution should be exercised regarding transferability. Further research involving correlational and/or regression analysis of these aspects together with further study across several different collaborative games would help to reveal if there are any collaborative gaming elements or collaborative game types that especially benefit from or hinder the reflective personalisation approach. This would help to establish specific criteria

Table 10 Paired *t* test results for performance

	Paired differences				<i>t</i>	<i>df</i>	Sig. (2-tailed)	
	Mean	SD	SE mean	95% CI of the difference				
				Lower				Upper
REFLECTIVE-GUIDED	3466.883	3093.813	399.410	2667.667	4266.100	8.680	59	.000
REFLECTIVE-PASSIVE	11,472.867	10,906.303	1407.998	8655.470	14,290.263	8.148	59	.000
GUIDED-PASSIVE	8005.983	7950.087	1026.352	5952.258	10,059.708	7.800	59	.000

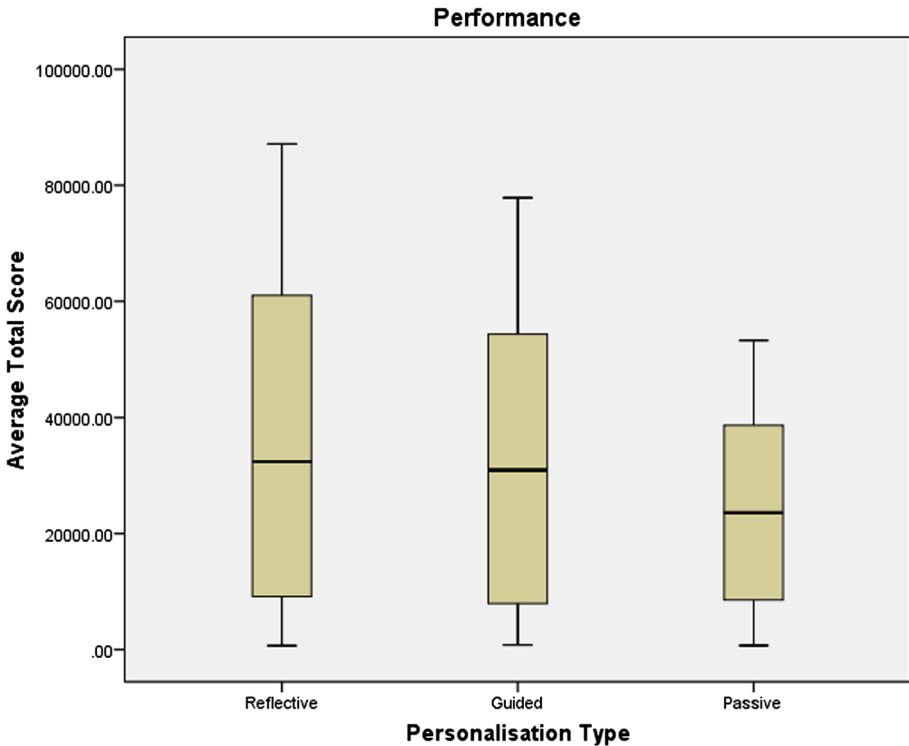


Fig. 20 Average total scores for the three games

for transferability. Nevertheless, while the Profiling Zone requires characterisation, levelling and scoring mechanics which are not supported by all games, the implementation of characterisation, levelling and scoring within the Adjustment Zone enables provision of these mechanics to games which do not support them. However, the results are not transferable to non-collaborative games which do not involve cooperation, and the architecture requires that games have exposed API endpoints so that they can interface with the gateway. Likewise, while the architecture fosters collaboration, recommendations are not currently made to players about how to collaborate well. The further study described above could also serve to form the basis of incorporating additional recommendations of this nature.

Another limitation is that both the architecture and experiment use only primary measures of performance, reflected by scores, and engagement, reflected by frequency and length of play. Incorporating multidimensional measures may provide different or more specific results, such as improvements in particular aspects of performance or engagement.

Regarding the participants taking part in the experiment, although they were diverse and were assigned randomly to the three versions of the game, they were volunteers. Consequently, it is possible that some participants may have already known each other and therefore communicated privately outside of the game. This may have affected the results, e.g. sharing tips or convincing a player to return to the game. However, in an everyday gaming situation, this would also occur. Furthermore, despite the varied backgrounds, random group assignment, and initialisation of all players into the same default state, it is possible that some bias still remained. Use of self-reporting at the start of the experiment may have acted as a further check for this.

Finally, it should be noted that increasing the number of participants and the duration of study may have resulted in different patterns and thus greater differentiation in results. Additionally, use of self-reporting at the end of the experiment may have helped to reveal further conclusions and complement the existing analysis, despite the usual validity issues.

5 Concluding discussion

This paper has proposed an architecture for the use of reflective agents in collaborative games for personalisation of individual players within teams via a combination of individual player and team profiling. Reflective personalisation involves player guidance towards goal states, self-evaluation of the effects of personalisation, and dynamic team formation. This is contrasted with guided personalisation, where players are steered towards goal states but without any self-evaluation, and passive personalisation, involving neither goal states nor self-evaluation, both of which do not necessarily incorporate dynamic team formation (our evaluation considered it across all three cases for consistency). Experimentation revealed that reflective personalisation has a positive impact on player and thus team performance and engagement within collaborative games, over passive or guided personalisation approaches. Therefore, using autonomous utility-based agents to make informed interventions based on profiles is a valuable technique to deploy for personalisation in collaborative games, and the agents should be self-evaluative to produce the best results. However, in our experiment, the effects on engagement appeared to be more pronounced. Further research is needed to understand whether this is a consequence of the types of personalised recommendations used, the measure of performance used, or the type of collaborative game used, and subsequently to explore methods for raising performance more substantially.

Further research related to limitations was discussed in Sect. 4.4. In addition, for more refined player profiling and personalisation, it would be sensible to also explore the inclusion of an individual state machine for each profile. In the existing approach, we know the current state of the player and the state the player should be guided to, but it is possible that in-between states could be based on each individual player's performance. It would also be useful to investigate whether the incorporation of team state machines in combination with individual player state machines would yield any significant benefits. Furthermore, as profiling continues to gain traction, we would seek to increase the level of automation of the profiling and to create more comprehensive player and team profiles, observing the effects on performance and engagement. This may also facilitate the incorporation of a Critic Presenter (Oh and Oh 2017) that provides more human-like gameplay critiquing as part of the reflective personalisation process. Finally, future research and development could also seek to enhance the dynamic team formation approach and explore how different team formation techniques affect performance and engagement.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed consent Informed consent was obtained from all individual participants included in the study.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Agius H, Angelides MC (2004) Modelling and filtering of MPEG-7-compliant meta-data for digital video. In: Proceedings of the 2004 ACM symposium on applied computing—SAC'04. ACM Press, New York, pp 1248–1252
- Alamri A, Hassan MM, Hossain MA et al (2014) Evaluating the impact of a cloud-based serious game on obese people. *Comput Hum Behav* 30:468–475. <https://doi.org/10.1016/j.chb.2013.06.021>
- Al-Dhanhani A, Mizouni R, Otrok H, Al-Rubaie A (2014) A game theoretical model for collaborative groups in social applications. *Expert Syst Appl* 41:5056–5065. <https://doi.org/10.1016/j.eswa.2014.02.050>
- Alhazmi E, Horawalavithana S, Iamnitchi A et al (2017) An Empirical Study on Team Formation in Online Games. In: 2017 IEEE/ACM international conference on advances in social networks analysis and mining, pp 431–438
- Angelides MC, Agius HW (2000) Eight scenarios of national information superhighway development. *J Inf Technol* 15:53–67. <https://doi.org/10.1080/026839600344401>
- Angelides MC, Agius H (2006) An MPEG-7 scheme for semantic content modelling and filtering of digital video. *Multimed Syst* 11:320–339. <https://doi.org/10.1007/s00530-006-0012-6>
- Angelides MC, Paul RJ (1993) Towards a framework for integrating intelligent tutoring systems and gaming simulation. In: Proceedings of 1993 winter simulation conference—(WSC'93). IEEE, pp 1281–1289
- Asher DE, Zaldivar A, Barton B et al (2012) Reciprocity and retaliation in social games with adaptive agents. *IEEE Trans Auton Ment Dev* 4:226–238. <https://doi.org/10.1109/TAMD.2012.2202658>
- Bakkes S, Tan CT, Pisan Y (2012) Personalised gaming: a motivation and overview of literature. In: Proceedings of the 8th Australasian conference on interactive entertainment playing the system—IE'12. ACM Press, New York, pp 1–10
- Bakkes S, Tan CT, Pisan Y (2013) Personalised gaming. *J Creat Technol* 3. <https://ojs.aut.ac.nz/journal-of-creative-technologies/index.php/JCT/article/view/13>
- Bakkes S, Whiteson S, Li G et al (2014) Challenge balancing for personalised game spaces. In: 2014 IEEE games media entertainment. IEEE, pp 1–8
- Basiri J, Taghiyareh F, Ghorbani A (2017) Collaborative team formation using brain drain optimization: a practical and effective solution. *World Wide Web* 20:1385–1407. <https://doi.org/10.1007/s11280-017-0440-6>
- Bergey P, King M (2014) Team machine: a decision support system for team formation. *Decis Sci J Innov Educ* 12:109–130. <https://doi.org/10.1111/dsji.12027>
- Brazier FMT, Treur J (1999) Compositional modelling of reflective agents. *Int J Hum Comput Stud* 50:407–431. <https://doi.org/10.1006/ijhc.1999.0249>
- Busch M, Mattheiss E, Orji R et al (2016) Player type models—towards empirical validation. In: CHI extended abstracts, pp 1835–1841. <https://doi.org/10.1145/2851581.2892399>
- Caroux L, Isbister K, Le Bigot L, Vibert N (2015) Player–video game interaction: a systematic review of current concepts. *Comput Hum Behav* 48:366–381. <https://doi.org/10.1016/j.chb.2015.01.066>
- Corem Y, Brown N, Petralia J (2013) Got skillz? Player matching, mastery, and engagement in skill-based games. In: 1st international conference on gameful design, research, and applications, gamification 2013. ACM Press, New York, pp 115–118
- Das S, Zook A, Riedl MO (2015) Examining game world topology personalization. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems—CHI'15. ACM Press, New York, pp 3731–3734
- Daylamani Zad D, Angelides MC, Agius H (2012) Personalise your massively multiplayer online game (MMOG) with Artemis. *Multimed Syst* 18:69–94. <https://doi.org/10.1007/s00530-011-0237-x>
- Daylamani-Zad D, Angelides MC, Agius H (2016) Lu-Lu: a framework for collaborative decision making games. *Decis Support Syst* 85:49–61. <https://doi.org/10.1016/j.dss.2016.02.011>
- Del-Moral Pérez M-E, Guzmán-Duque A-P, Guzmán-Duque A-P (2014) CityVille: collaborative game play, communication and skill development in social networks. *J New Approaches Educ Res* 3:11–19. <https://doi.org/10.7821/naer.3.1.11-19>
- Dignum F, Westra J, van Doesburg WA et al (2009) Games and agents: designing intelligent gameplay. *Int J Comput Games Technol* 2009:1–18. <https://doi.org/10.1155/2009/837095>

- Fallenstein B, Taylor J, Christiano PF (2015) Reflective oracles: A foundation for game theory in artificial intelligence. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Springer, Berlin, pp 411–415
- Ferro LS, Walz SP, Greuter S (2013) Towards personalised, gamified systems: an investigation into game design, personality and player typologies. In: Proceedings of the 9th Australasian conference on interactive entertainment matters of life and death—IE'13. ACM Press, New York, pp 1–6
- Florio L (2015) Decentralized self-adaptation in large-scale distributed systems. In: Proceedings of the 2015 10th joint meeting on foundations of software engineering. ACM, New York, pp 1022–1025
- Fowler M, Lewis J (2014) Microservices. <http://www.martinfowler.com/articles/microservices.html>. Accessed 01 Mar 2018
- Galway L, Charles D, Black M (2008) Machine learning in digital games: a survey. *Artif Intell Rev* 29:123–161. <https://doi.org/10.1007/s10462-009-9112-y>
- Ghenname M, Abik M, Subercaze J et al (2015) Hashtag-based learning profile enrichment for personalized recommendation in e-learning environments. *Int Rev Comput Softw (IRECOS)* 10:891–899. <https://doi.org/10.15866/irecos.v10i9.6261>
- Göbel S, Wendel V, Ritter C, Steinmetz R (2010) Personalized, adaptive digital educational games using narrative game-based learning objects. In: *Edutainment 2010*. Springer, Berlin, pp 438–445
- Goldberg B, Cannon-Bowers J (2015) Feedback source modality effects on training outcomes in a serious game: pedagogical agents make a difference. *Comput Hum Behav* 52:1–11. <https://doi.org/10.1016/j.chb.2015.05.008>
- González-González C, Toledo-Delgado P, Collazos-Ordoñez C, González-Sánchez JL (2014) Design and analysis of collaborative interactions in social educational videogames. *Comput Hum Behav* 31:602–611. <https://doi.org/10.1016/j.chb.2013.06.039>
- Hardy S, Dutz T, Wiemeyer J et al (2015) Framework for personalized and adaptive game-based training programs in health sport. *Multimed Tools Appl* 74:5289–5311. <https://doi.org/10.1007/s11042-014-2009-z>
- Hocine N, Gouaïch A, Cerri SA et al (2015) Adaptation in serious games for upper-limb rehabilitation: an approach to improve training outcomes. *User Model User Adap Interact* 25:65–98. <https://doi.org/10.1007/s11257-015-9154-6>
- Holmes D, Charles D, Morrow P, et al (2015) Rehabilitation game model for personalised exercise. In: Proceedings—2015 international conference on interactive technologies and games, ITAG 2015. IEEE, pp 41–48
- ISO, IEC (2002) Information technology—multimedia content description interface—part 2: description definition language. ISO/IEC, Geneva
- ISO, IEC (2003) Information technology—multimedia content description interface—part 5: multimedia description schemes. ISO/IEC, Geneva
- ISO, IEC (2004) Information technology—multimedia content description interface—part 5: multimedia description schemes: amendment 2: multimedia description schemes extensions. ISO/IEC, Geneva
- ISO, IEC (2005) Information technology—multimedia content description interface—part 9: profiles and levels. ISO/IEC, Geneva
- Kaiser E, Feng W (2009) PlayerRating: a reputation system for multiplayer online games. In: 2009 8th annual workshop on network and systems support for games (NetGames). IEEE, Paris, pp 1–6
- Karpinskyj S, Zambetta F, Cavedon L (2014) Video game personalisation techniques: a comprehensive survey. *Entertain Comput* 5:211–218. <https://doi.org/10.1016/j.entcom.2014.09.002>
- Kim K, Yoo B, Kauffman RJ (2013) Valuation of participation in social gaming. *Int J Electron Commer* 18:11–49. <https://doi.org/10.2753/JEC1086-4415180201>
- Kim K, Schmierbach MG, Bellur S et al (2015) Is it a sense of autonomy, control, or attachment? Exploring the effects of in-game customization on game enjoyment. *Comput Hum Behav* 48:695–705. <https://doi.org/10.1016/j.chb.2015.02.011>
- Kimbrough SO, Wu DJ, Zhong F (2002) Computers play the beer game: can artificial agents manage supply chains? *Decis Support Syst* 33:323–333. [https://doi.org/10.1016/s0167-9236\(02\)00019-2](https://doi.org/10.1016/s0167-9236(02)00019-2)
- Kluge B, Prassler E (2004) Reflective navigation: Individual behaviors and group behaviors. In: IEEE international conference on robotics and automation, 2004. Proceedings. ICRA '04, pp 4172–4177. <https://doi.org/10.1109/robot.2004.1308926>
- Kolter JZ, Maloof MA (2003) Dynamic weighted majority: a new ensemble method for tracking concept drift. In: Third IEEE international conference on data mining. IEEE Computer Society, pp 123–130
- Konert J, Göbel S, Steinmetz R (2014) Video game personalization via social media participation. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), pp 35–46
- Lewis J (2012) Micro services—Java, the Unix Way. In: 33rd degree conference, Kraków, Poland

- Li X (2004) Reflective web interface agent. In: *Advanced web technologies and applications*. Springer, Berlin, pp 146–151
- Lin C-H, Sun P-Y, Yu F (2015) Space connection: a new 3D tele-immersion platform for web-based gesture-collaborative games and services. In: *2015 IEEE/ACM 4th international workshop on games and software engineering*. IEEE, pp 22–28
- Linehan C, Lawson S, Doughty M, Kirman B (2009) Developing a serious game to evaluate and train group decision making skills. In: *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era on—MindTrek'09*. ACM Press, New York, New York, USA, p 106
- Lykourantzou I, Wang S, Kraut RE, Dow SP (2016) Team dating. In: *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems—CHI EA'16*. ACM Press, New York, pp 1243–1249
- Machado MC, Fantini EPC, Chaimowicz L (2011) Player modeling: towards a common taxonomy. In: *Proceedings of CGAMES'2011 USA—16th international conference on computer games: AI, animation, mobile, interactive multimedia, educational and serious games*. IEEE, pp 50–57
- Malas RI, Hamtini T (2016) A gamified e-learning design model to promote and improve learning. *Int Rev Comput Softw (IRECOS)* 11:8–19. <https://doi.org/10.15866/irecos.v11i1.7913>
- Miche Y, Lim M-H, Lendasse A, Ong Y-S (2015) Meme representations for game agents. *World Wide Web* 18:215–234. <https://doi.org/10.1007/s11280-013-0219-3>
- Moon J, Hossain MD, Sanders GL et al (2013) Player commitment to massively multiplayer online role-playing games (MMORPGs): an integrated model. *Int J Electron Commer* 17:7–38. <https://doi.org/10.753/JEC1086-4415170401>
- Najafabadi MK, Mahrin MN (2016) A systematic literature review on the state of research and practice of collaborative filtering technique and implicit feedback. *Artif Intell Rev* 45:167–201. <https://doi.org/10.1007/s10462-015-9443-9>
- Nardi B, Harris J (2006) Strangers and friends: collaborative play in World of Warcraft. In: *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work—CSCW'06*. ACM Press, New York, pp 149–158
- Nasir M, Lyons K, Leung R, et al (2015) The effect of a collaborative game on group work. In: *Proceedings of the 25th annual international conference on computer science and software engineering*, pp 130–139
- Natkin S, Yan C (2006) User model in multiplayer mixed reality entertainment applications. In: *Proceedings of the 2006 ACM SIGCHI international conference on advances in computer entertainment technology—ACE'06*. ACM Press, New York, Article No. 74
- Naudet Y, Lykourantzou I, Tobias E et al (2013) Gaming and cognitive profiles for recommendations in museums. In: *8th international workshop on semantic and social media adaptation and personalization*. IEEE, Bayonne, pp 67–72
- O'Connor DL, Menaker ES (2008) Can massively multiplayer online gaming environments support team training? *Perform Improv Q* 21:23–41. <https://doi.org/10.1002/piq.20029>
- Octavia JR, Coninx K (2014) Adaptive personalized training games for individual and collaborative rehabilitation of people with multiple sclerosis. *Biomed Res Int* 2014:1–22. <https://doi.org/10.1155/2014/345728>
- Oh Y, Oh YK (2017) A computational model of design critiquing. *Artif Intell Rev* 48:529–555. <https://doi.org/10.1007/s10462-016-9509-3>
- Oksanen K (2013) Subjective experience and sociability in a collaborative serious game. *Simul Gaming* 44:767–793. <https://doi.org/10.1177/1046878113513079>
- Oksanen K, Hamalainen R (2014) Game mechanics in the design of a collaborative 3D serious game. *Simul Gaming* 54:255–278. <https://doi.org/10.1177/1046878114530799>
- Orkin J, Roy D (2009) Automatic learning and generation of social behavior from collective human gameplay. In: *Proceedings of the 8th international conference on autonomous agents and multiagent systems*, vol 1, pp 385–392
- Padilla-Zea N, Medina NM, Gutiérrez Vela FL et al (2017) PLAGER-VG: platform for managing educational multiplayer video games. *Multimed Tools Appl* 77:2115–2152. <https://doi.org/10.1007/s11042-017-4376-8>
- Peeters MMM, van den Bosch K, Meyer J-JC, Neerinx MA (2016) Agent-based personalisation and user modeling for personalised educational games. In: *Proceedings of the 2016 conference on user modeling adaptation and personalization—UMAP'16*. ACM Press, New York, pp 303–304
- Retna Raj J, Sasipraba T (2015) Quality web services recommendation system based on enhanced personalized hybrid collaborative filtering approach. *Int Rev Comput Softw (IRECOS)* 10:249–255. <https://doi.org/10.15866/irecos.v10i3.4890>
- Richardson C (2017) Pattern: microservice architecture. <http://microservices.io/patterns/microservices.html>. Accessed 01 Mar 2018

- Russell S, Norvig P (2009) *Artificial intelligence: a modern approach*, 3rd edn. Prentice Hall, Upper Saddle River
- Sancho P, Fuentes-Fernández R, Gómez-Martín PP, Fernández-Manjón B (2009) Applying multiplayer role-based learning in engineering education: three case studies to analyze the impact on students' performance. *Int J Eng Educ* 25:665–679
- Sancho P, Torrente J, Marchiori EJ, Fernandez-Manjon B (2011) Enhancing moodle to support problem based learning. The Nucleo experience. In: 2011 IEEE global engineering education conference (EDUCON). IEEE, pp 1177–1182
- Scacchi W (2017) Practices and technologies in computer game software engineering. *IEEE Softw* 34:110–116. <https://doi.org/10.1109/MS.2017.20>
- Shaker N, Yannakakis GN, Togelius J (2010) Towards automatic personalized content generation for platform games. In: *AIIDE*, pp 63–68
- Shute VJ, D'Mello S, Baker R et al (2015) Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Comput Educ* 86:224–235. <https://doi.org/10.1016/j.compedu.2015.08.001>
- Siemer J, Angelides MC (1998) Towards an intelligent tutoring system architecture that supports remedial tutoring. *Artif Intell Rev* 12:469–511. <https://doi.org/10.1023/A:1006588626632>
- Silva GFM, Raposo A, Suplino M (2014) PAR: a collaborative game for multitouch tabletop to support social interaction of users with autism. *Procedia Comput Sci* 27:84–93. <https://doi.org/10.1016/j.procs.2014.02.011>
- Singh P, Minsky M (2003) An architecture for combining ways to think. In: *International conference on integration of knowledge intensive multi-agent systems*, 2003. IEEE, Cambridge, 30 Sept–4 Oct, pp 669–674
- Spraragen M, Madni AM (2014) Modeling of emotional effects on decision-making by game agents. *Procedia Comput Sci* 28:736–743. <https://doi.org/10.1016/j.procs.2014.03.088>
- Srba I, Bielikova M (2015) Dynamic group formation as an approach to collaborative learning support. *IEEE Trans Learn Technol* 8:173–186. <https://doi.org/10.1109/TLT.2014.2373374>
- Sterman JD (1989) Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment. *Manag Sci* 35:321–339. <https://doi.org/10.1287/mnsc.35.3.321>
- Streicher A, Smeddinck JD (2016) Personalized and adaptive serious games. In: Dörner R, Göbel S, Kickmeier-Rust M, Masuch M, Zweig KA (eds) *Entertainment computing and serious games*. Springer, Berlin, pp 332–377
- Teng C-I (2010) Customization, immersion satisfaction, and online gamer loyalty. *Comput Hum Behav* 26:1547–1554. <https://doi.org/10.1016/j.chb.2010.05.029>
- Teruel MA, Navarro E, González P et al (2016) Applying thematic analysis to define an awareness interpretation for collaborative computer games. *Inf Softw Technol* 74:17–44. <https://doi.org/10.1016/j.infsof.2016.01.009>
- Terzidou T, Tsiatsos T (2014) The impact of pedagogical agents in 3D collaborative serious games. In: 2014 IEEE global engineering education conference (EDUCON). IEEE, pp 1175–1182
- Tsai F-H (2016) The effectiveness evaluation among different player-matching mechanisms in a multi-player quiz game. *J Educ Technol Soc* 19:213–224
- van der Meij H, Leemkuil H, Li J-L (2013) Does individual or collaborative self-debriefing better enhance learning from games? *Comput Hum Behav* 29:2471–2479. <https://doi.org/10.1016/j.chb.2013.06.001>
- van Trijp R (2012) A reflective architecture for language processing and learning. In: Steels L (ed) *Computational issues in fluid construction grammar*. Springer, Berlin, pp 51–74
- Veletsianos G, Russell GS (2014) Pedagogical agents. In: Spector JM, Merrill MD, Elen J, Bishop MJ (eds) *Handbook of research on educational communications and technology*. Springer, New York, pp 759–769
- Visschedijk GC, Lazonder AW, van der Hulst A et al (2013) Modelling human emotions for tactical decision-making games. *Br J Educ Technol* 44:197–207. <https://doi.org/10.1111/j.1467-8535.2012.01286.x>
- von der Pütten AM, Klatt J, Ten Broeke S et al (2012) Subjective and behavioral presence measurement and interactivity in the collaborative augmented reality game TimeWarp. *Interact Comput* 24:317–325. <https://doi.org/10.1016/j.intcom.2012.03.004>
- Wang B, Zhao D (2014) A Kalman filter-based actor-critic learning approach. In: 2014 international joint conference on neural networks (IJCNN). IEEE, Beijing, 6–11 July, pp 3657–3662
- Xu Y, Wang Y, Gao X, Wei L (2010) Collaborative design team formation method based on three-phase comprehensive evaluation. In: *The 2nd international conference on information science and engineering*. IEEE, pp 1907–1910
- Yager RR (1988) On ordered weighted averaging aggregation operators in multi criteria decision making. *IEEE Trans Syst Man Cybern* 18:183–190. <https://doi.org/10.1109/21.87068>

- Yang D, Jiao J, Ji Y et al (2015) Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game. *Eur J Oper Res* 246:263–280. <https://doi.org/10.1016/j.ejor.2015.04.022>
- Yannibelli V, Amandi A (2012) A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Syst Appl* 39:8584–8592. <https://doi.org/10.1016/j.eswa.2012.01.195>
- Yee N, Bailenson JN, Rickertsen K (2007) A meta-analysis of the impact of the inclusion and realism of human-like faces on user experiences in interfaces. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, New York, pp 1–10
- Yuh-Wen C, Larbani M, Chen-Hao L (2010) Simulation of a supply chain game with multiple fuzzy goals. *Fuzzy Sets Syst* 161:1489–1510. <https://doi.org/10.1016/j.fss.2009.10.015>
- Zagal JP (2006) Collaborative games: lessons learned from board games. *Simul Gaming* 37:24–40. <https://doi.org/10.1177/1046878105282279>