Making Games ALIVE: An Organisational Approach

Sergio Alvarez-Napagao[1], Fernando Koch[2],
Ignasi Gómez-Sebastià[1], and Javier Vázquez-Salceda[1]

[1] Universitat Politècnica de Catalunya
{salvarez,igomez,jvazquez}@lsi.upc.edu
[2] University of Melbourne
fkoch@unimelb.edu.au

Abstract. The AI techniques used in commercial games are usually predictable, inflexible and unadaptive, causing a lack of realism for the player. In this paper, we introduce a proposal of integrating the ALIVE framework, based on Organisational theory, into commercial games. The objective of our proposal is to provide game AI developers with a methodology and tools to model gaming scenarios using social structures.

1 Introduction

Artificial Intelligence (AI) in commercial games (Game AI) provides the means to enhance the two-way communication with the human player by delivering the illusion of "intelligence" in the non-player characters' (NPCs) behaviour. Usually, it encompasses a subset of academic AI techniques that implement *ad hoc* solutions in three groups[8]:

i Movement mechanisms, providing the decision process to control NPC's motion, e.g. optimised real-time versions of A^* algorithms.
ii Behaviour control used to control NPCs' actions.
iii Strategy techniques used to co-ordinate groups of NPCs.

Whilst algorithms in (i) have evolved to mature state-of-the-art, solutions commonly used in commercial games for (ii) and (iii) are far from aligned with academic AI and are based on simplistic, rule-, automata- or case-based methods optimised for performance. These domain-dependent approaches present the following limitations in most of the cases:

- Blind specifications: the NPCs are programmed on *how* to act in reaction to environmental and/or other players conditions, but not *why* to act in a given manner; hence, the actions are purposefuless and, in most cases, not "natural" from the human player's perception.
- Lack of flexibility and adaptiveness: the rule-based actions are limited and reactive to external conditions, not beeing able to evolve, and providing reduced pro-activeness.

F. Dignum (Ed.): Agents for Games and Simulations II, LNAI 6525, pp. 179–191, 2011.
© Springer-Verlag Berlin Heidelberg 2011

– Strange behaviour: the behaviour of the NPCs do no reflect the aspects of sociability and "participating in a whole", leading to unnatural actions from the human player's perception.
– Predictable behaviour: NPCs' tactics are easily discoverable by the human player and, after some time, predictable, leading to negative perception.
– Low reusability, as the solutions are commonly tailored to specific scenario domains and, therefore, not re-usable through different games even if they belong to the same genre.

We argue that it is possible to create elaborate solutions for the issues of (ii) behaviour control and (iii) strategy techniques by integrating models based on Organisation Theoretical methods to control NPCs' behaviour. This theory contributes to the systematic study of how actors behave within organisations. Hence, the actors in a game are described as an organisation which behaviour is based on specific roles, norms, dependencies, and capabilities (services). Our aim is to provide a methodology and tools for Game AI developers to model gaming scenarios using social structures. We demonstrate how this approach tackles the limitations of the current domain-dependent approaches aforementioned.

This research is part of the Project ALIVE[2], which combines ideas from organisational and coordination theories to create an integrated framework for the development of complex distributed systems. We describe this framework in the next section. Section 3 introduces our proposed architecture and how to integrate to existing games. Section 4 provides proof-of-concept case studies. Section 5 cross-relates our approach to other works. The paper concludes with Section 6, where we discuss our achievements and propose future work.

2 The ALIVE Framework

The ALIVE framework is being developed as part of the Project ALIVE[2]. It aims to combine existing work in coordination and organisational structures with the state-of-the-art in service-oriented computing. It will allow system architects to build service-oriented systems based on the definition of organisational structures and how they interact. This framework defines three structural levels, as depicted in Figure 1:

– The *Service Level* augments and extends existing service models in order to make components aware of their social context and of the rules of engagement with other services via semantic demarcation technologies.
– The *Coordination Level* specifies the high-level patterns of interaction (known as *workflows*) among services by using powerful coordination techniques from recent agent research. These *workflows* can be adapted at runtime, which is useful when the system has to react to unexpected events (such as failures and exceptions).
– The *Organisational Level* provides the *Service* and *Coordination* levels with a social context, specifying the organisational rules that govern interaction. This level makes services *organisational aware*, that is, services are aware

of system's high-level objectives, structure and normative restrictions. This reflects in task allocation, *workflow* generation and agreement at the coordination level. For instance, the system will prevent *workflows* that violate normative restrictions from being generated and tasks are to be allocated to appropriate actors as defined on the organisational structure. This level also benefits from recent research in organisational dynamics to allow the structural adaptation of the system when changes on rules or restrictions happen.

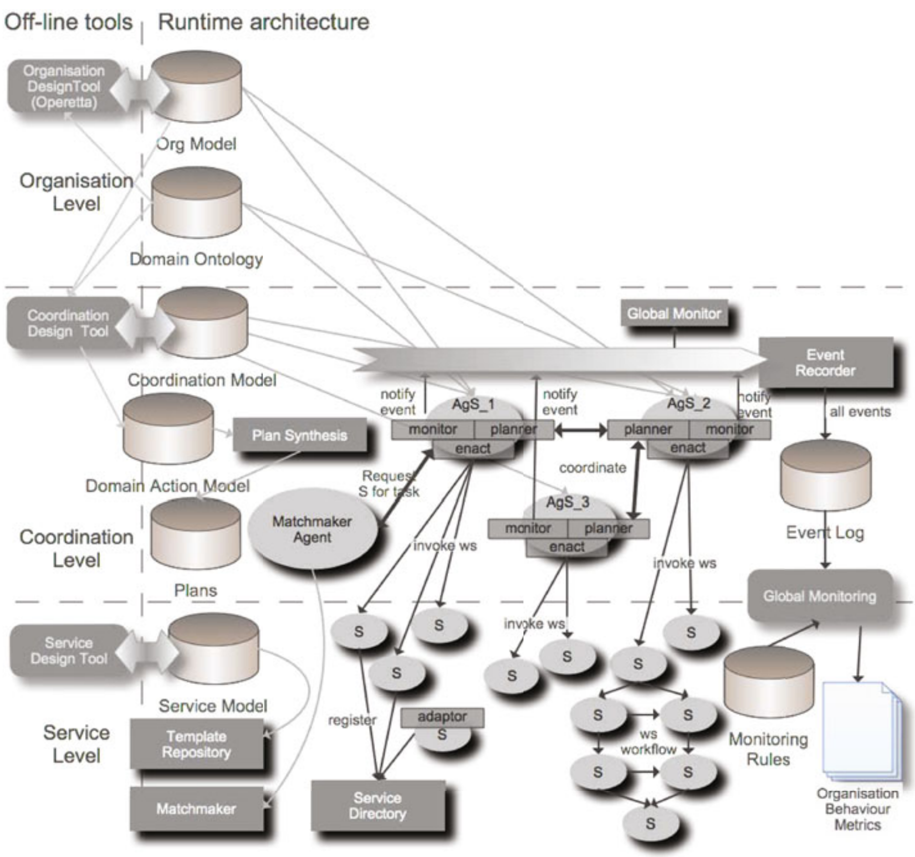

Fig. 1. ALIVE architecture (*S* stands for *Service*)

The ALIVE Framework allows Game AI developers to think in terms of why-what-how when defining the decision-making actions for NPCs. That is, at the Organisational level, the developer defines "why to do something" by describing the elements of the organisational structure in terms of organisation objectives, roles, norms, and restrictions. At the Coordination level, the developer defines "what to do" based on possible solutions and tasks to realise in specific situations; finally, at Service level, the developer defines "how to do it" in terms of which actual, low-level actions to perform in order to realise those tasks.

Moreover, the ALIVE framework applies *substantive norms* that define commitments agreed upon actors and are expected to be enforced by authoritative agents, imposing repair actions and sanctions if invalid states are reached. Substantive norms allow the system to be flexible, by giving actors –human or computer-controlled– the choice to cause a violation if this decision is beneficial from an individual or collective perspective.

Finally, the ALIVE Framework provides useful tools to define these elements, such as *OperettA*, a visual tool implemented as an Eclipse IDE plugin, which allows to specify the organisational concepts of roles, interactions and norms. These structures are implemented as coordination agents, used to build coordination plans for groups of agents enacting roles withing the organisation. Agents interact for enacting their roles either via direct communication or via service invocation. Monitors observe agent interactions. When these interactions are put together with the normative and organisational states – e.g. obligations, permissions, roles – they allow the agents to reason about the normative effects of their actions. The detection of normative states is a passive procedure that consists in monitoring past events and checking them against a set of active norms[1].

This set of tools and methods provides inherent support to the development of complex, re-usable Game AI solutions.

3 Proposal

Our argument is to create elaborate solutions for the issues of (ii) behaviour control and (iii) strategy techniques by integrating the ALIVE framework to academic and commercial games. This approach will provide extended flexibility to the elements that imply intelligent behaviour, e.g. actors and characters, teams of individuals, and narrative storylines. In addition, it will provide metrics that can be applied to evaluate the organisational behaviour using the games' environments as simulation scenarios. Hence, it would be possible to compare, learn, and improve NPC's behaviour with an approach based on organisation theoretical solutions for Game AI. This would contribute to overall flexibility and adaptiveness.

Figure 2 depicts the proposed architecture. We are providing in our solution:

1. A practical solution to couple agents to the Game Engine, by defining the Game Enactor programming interface.
2. A tool to describe the Organisation Ontology, which contains a representation of agent structures.
3. The elements to describe game actors' behaviour via social structures based on norms, roles and their enactment, promoting the balance between autonomy and story direction.

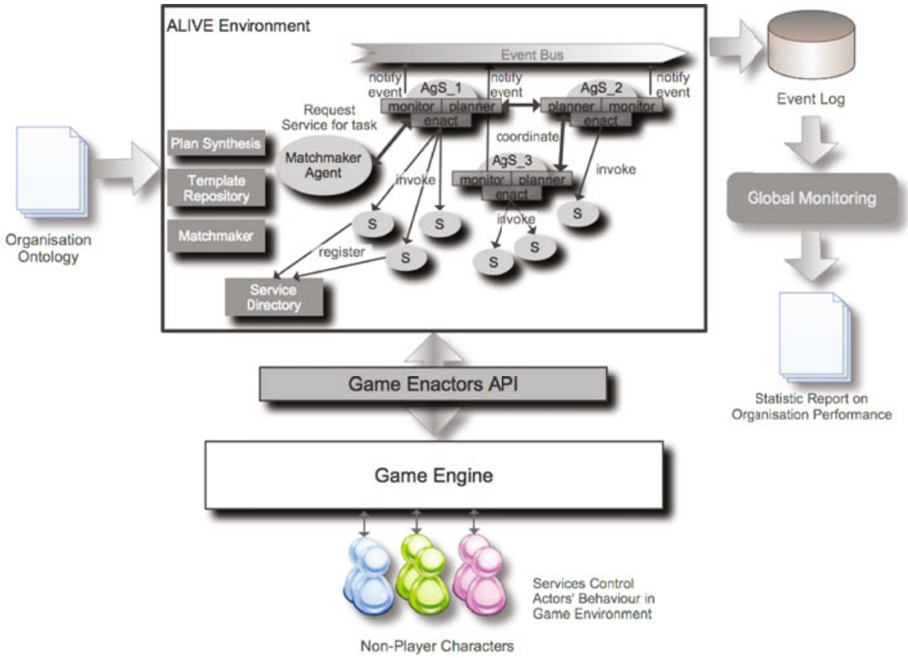

Fig. 2. ALIVE-Gaming coupling infrastructure

We propose that this solution is applicable to both *fun games* and *serious games*. For the former, we foresee our solution helping to improve the games' actors' functioning with more flexibility and promoting natural behaviour. For the latter, the model reflects the socio-environmental behaviour of human societies, providing the basis for games and simulations that can be used in the emerging field of Computational Social Sciences. Next, we present a number of case studies in the applicability field.

4 Case Studies

We are testing the solution in different games in order to validate our proposal, as depicted in Figure 3. We intend to analyse what is the advantage in terms of realism, flexibility and adaptability. Moreover, our application to simulation environments will provide results useful for organisational research. In the case of commercial games, it requires access to the internal game control structures. We selected three representative examples for our case studies, considering the complexity and validity of achieved results:

i Grand Theft Auto IV, from Rockstar Games;
ii Warcraft III, from Blizzard, and;
iii Lincity, an open-source, SimCity like city building simulation.

(a) GTA IV (b) Warcraft III (c) Lincity

Fig. 3. Games used as case studies

4.1 Sandbox Game: GTA IV

First we will test our environment on sandbox games, also known as free-roaming games. In these kind of games, players are given a large amount of freedom, with non-linear storylines and different paths to completion. For example, the Grand Theft Auto (GTA) series allows the player to wander around a whole city and interact with hundreds of NPCs and objects.

In *free roaming* games such as GTA, most of the interactions with characters are scripted, giving the player a feeling of repetitiveness after a few hours of play. On the other hand, the higher-than-normal freedom given to the player also provides less realism.

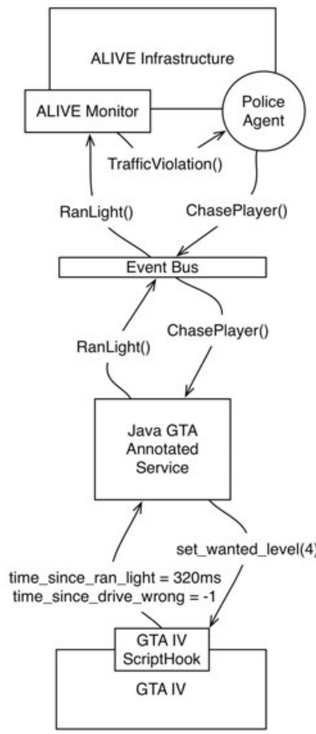

Fig. 4. ALIVE-Gaming coupling infrastructure for GTA

Our objective is to define a high level social structure, simulated by the ALIVE coordination layer, with dynamic adaptation of interaction patterns, using GTA as the graphic interface of such a social environment. For example, in GTA the player is almost free to behave in a violent way while driving a car. Passing red lights, driving in the wrong direction, and running over people are actions that have no consequences in the vast majority of cases. We have already implemented a prototype by designing, at the organisational level, traffic norms and roles defining authority figures, i.e. police (see Figure 5). Police agents plan and reason about the sanctions to apply when detecting a traffic norm violation, which can consist on imposing a fine or initiating a car chase, depending on the gravity of such violation.

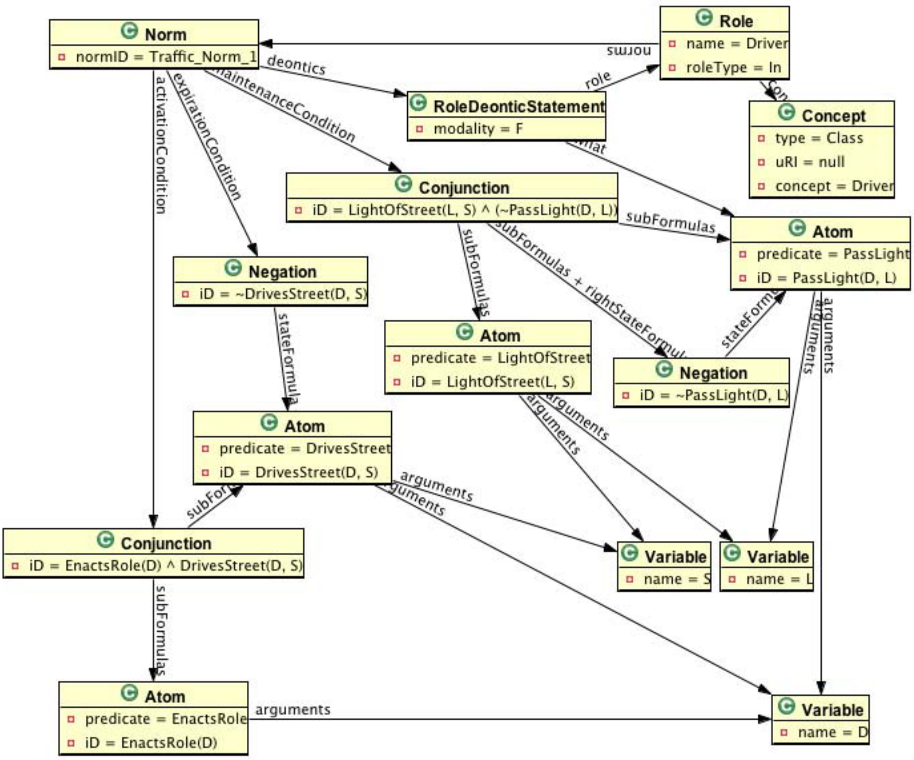

Fig. 5. Graphical representation of the norm *it is forbidden to pass under a red light* (*OperettA* Tool)

For the GTA connection with ALIVE we have used *GTA ScriptHook*, an open-source tool which allows us to capture all possible events and execute all possible actions in the game running environment, including the control of NPCs' behaviour. The following is a typical flow of information on our system (see Figure 4):

1. An event, i.e. running past a red traffic light, happens on the game.
2. *GTA ScriptHook* captures the event and provides it to the *Java GTA Annotated Service*.
3. The service interprets the game event as a low-level event and puts it on the *Event Bus*.
4. The *ALIVE Monitor* captures the event from the *Event Bus* and infers its high-level interpretation.

5. This interpretation triggers the generation of a new event *ChasePlayer* that is registered in the *Event Bus*.
6. The *Java GTA Annotated Service* captures the *ChasePlayer* event from the *Event Bus* and, via *ScriptHook*, modifies the game.
7. This will effectively make something happen on the game, i.e. player being chased by police forces, as a response of having run a traffic light.

As we have already seen in Section 2, norms modelled using the ALIVE tools are not regimented but substantive, which means that the player –as well as any NPC– can decide not to fulfill them. Thus, a player can decide to break traffic rules if the police is not around or at line of sight, or if the player has no concerns about the possible sanctions enforced by the NPCs. This is a simple example, but more complex examples can be designed to create obstacles or motivations for the player, by reasoning at runtime about the social-environmental context in the game at a certain point of time.

Our intention is to design a full set of organisational constraints, i.e. norms, individual objectives and roles, in order to define high-level social structures in the game, therefore improving realism through sensible and adaptive interactions with NPCs.

4.2 Real-Time Strategy Game: Warcraft III

For many years, computer wargames have been designed as turn-based games. Real-time Strategy (RTS) games are an evolution of turn-based wargames, in which the player has to command a team of virtual individuals with diverse capabilities to achieve a common objective, commonly to defeat the teams of the human- or computer-controlled rivals. Other (sub)objectives include the capture and micro-management of resources, technological evolution, and so on. RTS games are interesting for our purpose in the sense that the concepts they deal with can be directly mapped to the ALIVE domain, i.e. organisational structure, roles, role hierarchy, objectives, and coordination.

At the moment, we have modelled the organisational specification of an abstract RTS game (see Figures 6 and 7), including roles, e.g. worker, defender, attacker, objectives – e.g. gather resources, defeat the enemy armies –, and norms – e.g. it is forbidden to create military units until there are enough workers to support them. This specification can be reused through different RTS implementations. For our tests we are using Warcraft III.

We have coded a Java service that is connected to Warcraft 3 game, allowing for bidirectional communication via sockets. This Java service is used by the ALIVE framework. With this implementation, agents are able to:

– Perceive the “state of the world”, reacting to events happening in the game at runtime, e.g. a unit being created, or a soldier spotting an enemy.
– Reason about which actions should be taken in the game taking into account the current state of the world.

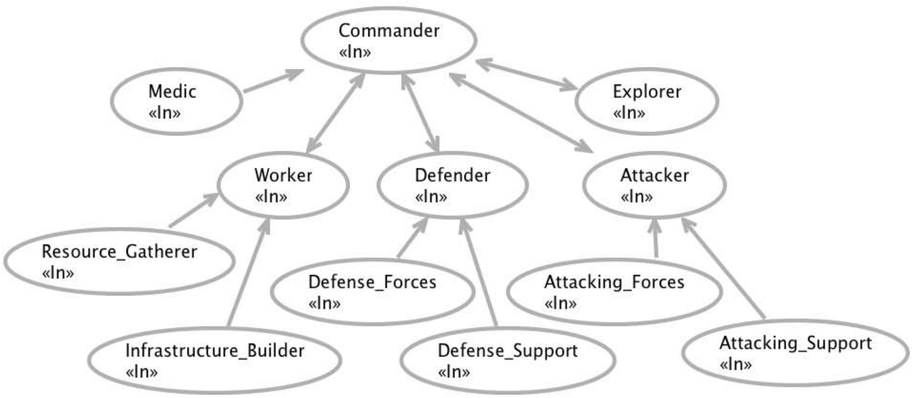

Fig. 6. Social structure for generic RTS games (*OperettA* Tool)

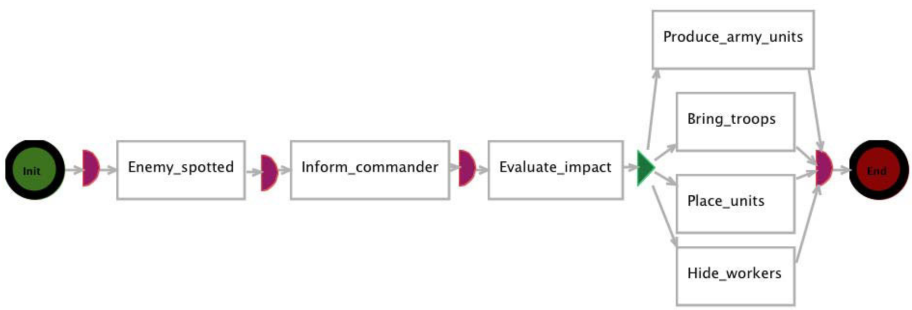

Fig. 7. Interaction structure for *Defend city* (*OperettA* Tool)

- Include the ALIVE specification in their reasoning. Agents will take into account the organisational structure: roles, plans and norms, defined in the ALIVE model, and proactively decide at each moment which actions to enact in order to accomplish the organisational objectives. The agents may also decide to discard some actions to be enacted if an organisational norm is forbidding to enact them given the current state of the world.
- Enact actions in the game (see Figure 8). Once the reasoning process has decided which are the next actions to be performed, agents are able to communicate with the game, making the unit responsible of each action to enact it according to the role and plan structures defined in the organisational specification.

This is the scenario that could best benefit from the adaptability offered by the ALIVE infrastructure. A common issue of RTS games is that after some amount of time spent on it, computer opponents are predictable and easily defeatable by using simple yet optimal strategies. We aim to produce computer opponents capable of adapting to unpredictable scenarios by dynamically improving at the

Fig. 8. Warcraft III units enacting actions sent from the ALIVE platform

organisation and coordination layers. Moreover, this type of game would provide us a clear visual interface to execute simulations of organisations in real-time.

4.3 Macro-economic Simulation: Lincity

One of our main objectives, as noted earlier, is to use games as an interface for visualising the results of simulations of organisational specifications. Real-time simulation (RTS) games can be used for this, but they are only useful for scenarios where the range of events types are limited. For discrete-event and off-line simulations we propose the use of games that deal with macro-economic variables.

In our case, we have chosen *Lincity*, which is an open-source city-building simulator[1] developed in C++. The idea behind this is to simulate the actions taken by large amounts of virtual individuals and present the results at the macro-economic level, i.e. the evolution of the city. Reorganisation techniques will be intensively used in this scenario.

For instance, we intend to simulate crisis management scenarios during natural disasters[11]. City-simulations are perfect environments to describe intuitive

[1] More particularly, it is a clone of the SimCity series.

disaster situations, such as fire and floods. In this situation, the diverse stakeholders have to coordinate their activities to handle such large scale crises. There are various dynamic aspects that must be dealt with, as the crisis may escalate, which require elaborate coordination mechanisms among the various stakeholders involved: (i) Emergency Call Centre; (ii) Fire Station; (iii) Fire fighter Team; (iv) First-Aid Station; (v) Police Station, and; (vi) Fire Fighter Truck.

We intend to introduce "unexpected events", such as resources becoming unavailable, escalation of the problem, road block, and so on, that would require the handling of exceptions in ongoing coordinations. In this case, modelling of organizational knowledge has a crucial role, where role and objective concepts define *why* the stakeholders must operate and letting the system define *what to do* and *how*. Changes in the environment and in stakeholders needs are inherently reflected in a top-down approach. When the service and coordination levels fail, the system turns to the organizational level for continuing the operations.

Therefore, this elaborated simulation environment could also be used as a tool to model real world situations, allowing for testing and evaluating different response approaches to natural disaster situations.

5 Related Work

The current issues of commercial games AI introduced in Section 1 are related to high-level concepts of gaming such as realistic virtual actors, automatic content and storyline generation, dynamic learning, or social behaviour. Tackling these issues could represent a qualitative improvement on gaming experience from the player perspective and academic research on AI has good opportunities to provide solutions to these challenges[3,9].

Adaptiveness in games has been already explored in academic AI research. However, existing approaches are either focused on individual reasoning[5,6], or do not take into account high-level definitions that would allow for reasoning *why* to make a particular decision on a specific context[12]. These approaches can get advantage of ALIVE by extending individual agents' reasoning cycle with *organisational awareness*.

Organisational frameworks such as OperA[4] are already being explored for their use in *serious games*. In [13], organisational specifications are used to create a distributed intelligent task selection system that adapts to the player skill level and to model the storyline. With our work we intend to advance on this line of work by generalising the use of organisational models for *fun games*, more focused on the realism of gaming experience, rather than on user modelling and learning.

There are already examples showing that higher levels of abstraction can be successfully used in commercial games' AI. Actually, some recent important commercial games such as *F.E.A.R*[10], *Fallout 3*, or *Empire: Total War*, have started to apply more complex cognitive patterns by using *GOAP* (Goal-Oriented Action Planning), a simplified and optimised version of *STRIPS* that allows for real-time planning of actions with pre- and post-conditions, even outperforming *Finite State Machine*-based algorithms in some scenarios[7]. Thus, these games execute complex symbolic reasoning not only about *how* to execute certain actions, but also about *what* to execute at each moment. We believe that, by using an even higher level of abstraction in order to reason also about *why* actions have to be performed, methods such as GOAP can be complemented and improved.

6 Conclusions

Our research intends to address a common problem of commercial Game AI solutions by providing an approach based on the integration of an organisation theoretical control system for NPC. We suggest that this combination contributes to Game AI solutions by providing an adaptive, extensible and flexible solution to game development industry.

The main advantage of this approach is that now developers can specify NPCs' behaviour in terms of "why" they should do something, not only "what' and "how" to do it. That is, the actors in a game are described as an organisation which behaviour is based on specific roles, norms, dependencies, and capabilities (services). Our aim is to provide a methodology and tools for Game AI developers to model gaming scenarios using social structures.

We proposed an architecture for the integration of Project ALIVE's organisation specification and coordination framework to existing commercial games. We propose the introduction of a middleware 'Game Engine Interface' that proxies information in two-ways: from the game environment to the ALIVE-based Game AI component, allowing developers to plug the proposed solution to existing games, as long as the basic interface methods to control NPCs actions are available. We demonstrate the architecture, steps and expected improvements of promoting this solution in three representative commercial games.

We conclude that this approach contributes to the issues of improved behaviour control and advanced strategy techniques, tackling some of the main issues in Game AI solutions, by providing:

- open specifications where NPCs are programmed in terms of *why* they must act in a certain way;
- enhanced flexibility and adaptiveness by describing NPC's behaviour based on specific roles, norms, dependencies, and capabilities;
- more "natural behaviour" as NPC will act autonomously, respecting environmental conditions and organisational objectives that will be perceived as "natural" by human player's perception;
- broader behaviour range, as NPCs' behaviour is more autonomous and driven by overall organisation objectives, and;
- improved reusability, as the proposed solution is generic and can be attached to a variety of existing commercial games through a common interface ad customised organisation models.

As future work, we intend to complete our implementations and extend the models with realistic social descriptions. We are also analysing the integrability and extension of our approach by exploiting the integration to other commercial games, such as World of Warcraft, and The Sims 3.

Acknowledgments. This research has been funded by the FP7-215890 ALIVE project, funded by the European Commission. Javier Vázquez-Salceda's work has been also partially funded by the "Ramón y Cajal" program of the Spanish Ministry of Education and Science.

References

1. Aldewereld, H., Alvarez-Napagao, S., Dignum, F., Vázquez-Salceda, J.: Making Norms Concrete. In: Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (2010)
2. Alvarez-Napagao, S., Cliffe, O., Vázquez-Salceda, J., Padget, J.: Norms, Organisations and Semantic Web Services: The ALIVE approach. In: Workshop on Coordination, Organization, Institutions and Norms at MALLOW 2009 (2009)
3. Charles, D.: Enhancing gameplay: Challenges for artificial intelligence in digital games. In: Proceedings of the 1st World Conference on Digital Games (2003)
4. Dignum, V.: A model for organizational interaction: based on agents, founded in logic. PhD Thesis, Utrecht University (2004)
5. Laird, J., Newell, A., Rosenbloom, P.: SOAR: An architecture for general intelligence. Carnegie-Mellon University Technical Report (1987)
6. Leite, J., Soares, L.: Evolving characters in role playing games. Cybernetics and Systems (2006)
7. Long, E.: Enhanced NPC behaviour using goal oriented action planning. PhD Thesis, University of Abertay-Dundee (2007)
8. Millington, I., Funge, J.: Artificial Intelligence for Games. Morgan Kaufmann, San Francisco (2009)
9. Nareyek, A.: Game AI Is Dead. Long Live Game AI! IEEE Intelligent Systems (2007)
10. Orkin, J.: Three states and a plan: the AI of FEAR. In: Proceedings of the 2006 Game Developers Conference (2006)
11. Quillinan, T.B., Aldewereld, H., Wijngaards, F.N., Brazier, F., Dignum, F., Dignum, V., Penserini, L.: Developing Agent-based Organizational Models for Crisis Management. In: Decker, K.S., Sierra, C., Castelfranchi, C. (eds.) Proceedings of the 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), pp. 45–52 (2009)
12. Spronck, P., Ponsen, M., Sprinkhuizen-Kuyper, I.: Adaptive game AI with dynamic scripting. Machine Learning (2006)
13. Westra, J., van Hasselt, H., Dignum, V., Dignum, F.: On-line Adapting Games using Agent Organizations. In: IEEE Symposium on Computational Intelligence and Games (CIG 2008) (2008)