Towards a Design Framework for Controlled Hybrid Social Games

Harko Verhagen [1] and **Pablo Noriega** [2] and **Mark d'Inverno** [3]

We propose a framework for designing and deploying games where social behaviour is kept under control. This framework may also be used for designing other dynamic coordinated social spaces.

1 INTRODUCTION

In digital game research it has been noted that even though game developers use AI as a selling argument, the intelligence is rather shallow. A close analysis of current game agents (or *non-player characters*, NPCs) reveals many deviations of intelligent behaviour.

According to Bartle [4], NPCs may have several functions in a game:

- buy, sell, and make stuff
- provide services
- guard places
- get killed for loot
- dispense quests (or clues for other NPCs quests)
- supply background information (history, lore, cultural attitudes)
- do stuff for players
- make the place look busy.

Many times, NPCs are unaware of their environment, causing them to miss essential information and die as a result. They are also usually unable to dynamically build temporary coalitions, reason about norms, or use any other social coordination mechanism that will control their decision making and behaviour. In fact, most NPCs are very simple script machines that have very poor social skills. Dynamic NPCs would be one solution to the boredom of having to deal with these simple messengers. However, game designers fear that the loss of control over the agents may cause the game to get out of hand and destroy the gaming experience by breaking the storyline.

Our proposal is that in addition to using different NPC architectures based on the function of NPCs (thus making them more challenging or engaging), one may rely on *social coordination mechanisms* that apply within such hybrid social games (hybrid here implies a mix of human and NPC participants interacting) and thus achieve an acceptable level of control over characters. We understand that both elements are non-trivial and carry different concerns, hence we propose to separate those concerns by proposing a clear separation between the game itself and the characters that participate in it. In this paper we focus on the first direction.

Although in this paper we limit our discussion to hybrid social games, the design approach that we propose applies to other sorts of dynamic coordinated sociotechnical systems like participatory simulation environments and open regulated MAS.

[1] Stockholm University. Sweden; email:verhagen@dsv.su.se
[2] IIIA-CIC, Spain; email:pablo@iiia.csic.es
[3] Goldsmiths, University of London, UK; email:dinverno@gold.ac.uk

2 RELATED WORK

Some related works exists in both game research (concerning the design framework as such) and participatory simulation research (dealing with hybrid social spaces). In game research, the MDA framework presented in [11] is of interest. The framework is meant to make iterative design processes involving developers, researchers, and designers more easy by distinguishing between the game development from a designer perspective and from a player perspective. In the view of [11], the developer would focus on the game mechanics (M), while the player is more focussed on (or expressed in ideas of) the game aesthetics (E), expressed in the emotions the game produces in the player while playing. The runtime behaviour of the game is called dynamics (D). In contrast to our framework presented below, the MDA framework is including the basic game idea (aesthetic) while remaining under-specified with respect to the mechanics and elements involved and also lacks a separation between the control of the game elements from the design and play perspective. In participatory simulation (such as [1]), the mix of human and nonhuman agents is needed to build a better understanding (or increase knowledge of) a real world situation or system by the human agents. Thus, there is a target system that is known and is characterised by (verified) empirical data and hypotheses. In computer games, the target system does not exist, it is a designed world. Thus the relationships between the game elements cannot be empirically verified nor theoretically grounded in a decisive way.

3 DESIGN FRAMEWORK

3.1 Games and Coordinated Social Spaces

We can see games—and several other coordinated social spaces—as having three complementary and interrelated views.

1. The first view, (circle "I" in Fig. 1) is the *ideal* game where some (ideal) characters interact according to the rules of the game. The"'I" view contains an idealised description of the landscape where the game takes place, refers to knights and aliens or football players that will be involved in the game and exchange messages or kill each other according to the scripts of "Assassin's Creed" or whatever.
2. Another view (square "T"), consist of the *technological artefacts* that implement and support that ideal game. This support is of two sorts: On one hand, there are the technological artefacts that implement and run the ideal game, namely, it includes the code of avatars that will be used by humans to play, as well as the code of NPCs,it makes operational the actions of characters on the game "landscape", and, in general, makes sure that the game is executable and follows the conventions that govern the ideal game.

On the other hand, there is the software and hardware needed to support the running game: communications, data-bases, interfaces, etc.

3. Finally, the third view (triangle "W"), is the physical world where the game takes place (the room, the screen, the console and the humans that play the game).

In this paper we will look mainly into what the contents of "I" and "T" should be, and how these two views of a game are related.

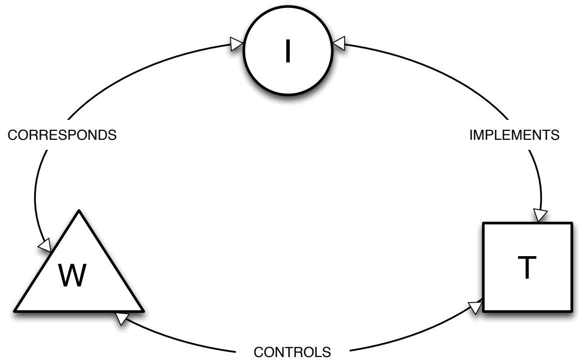

Figure 1. The three views of a coordinated social space: The ideal game, the technological artefacts that implement it and the actual world where the game is played

We propose that the design of a game should take into consideration a number of aspects that become instantiated and assembled to become what we call a *game space*, which is part of the "I" view. The outcome of that assembly will be a precise description of an ideal game that may then be specified, implemented and run within its *game environment* as a technological artefact. The instantiation and assembly of the game components, however, may take different forms if the repertoire of aspects is rich enough. Thus, we claim it is worth separating those elements that may help in choosing the better components from those needed for assembling those components. For that purpose we propose to have a *design space* (again in"I") and couple it with a *design environment* that includes the game environment and the repositories and services that support and complement it (in "T").

3.2 The Game Space

Building loosely on a theatrical metaphor, we may visualise the game space as a play. The author of the play creates a plot and organises it into scenes where some characters exchange dialogue and gestures and move around the staged rooms according to some directions. Likewise, the game space is an abstract entity, designed by someone, where humans and NPCs interact, by means of some interaction mechanisms, within collective activities that happen in particular "rooms" subject to some procedural and behavioural conventions.

More precisely, a game space defines an interaction framework where agents—that may be humans or software and may be created by the game designer with a specific purpose in mind or may be bonafide players—are able to perform certain (admissible) actions subject to some ways of imbuing "acceptable" social order. We propose that in order to define that interaction framework, the designer needs to address ten aspects for which a collection of conceptual constructs are available. In fact, the instantiation of particular constructs and their assembly will constitute the actual game space.

These are the aspects that we believe are necessary and sufficient to describe a game space and we exemplify the type of constructs each should include:

Ontology. It is worth distinguishing between game-generic and game-specific ontologies. In both cases we mean ontology as "entities", or a collection of "terms" in"I" that are eventually mapped into "W".

- *game-generic constructs* are needed to define contexts of collective interaction and their interrelations. For example: action, agent, role, and notably *collective contexts* (ideal locations or activities where several agents interact simultaneously, sharing the *same state*) like game level, scene, transition, challenge, ...
- *game-speciic*. This will list the elements the are used to define the content of collective contexts and "interactions" (actions). For example, swords, ditch, wall; dig, climb, exit, acquire role, improve prestige,...; raise hand, ...

Agent types These include the two main types of "embodied" participants: PCs and NPCs, and perhaps some server agents, which are not visible to players, that deal with some game management functions (for instance performing police-like and time-keeping functions).
Notice that although we want agents to participate in the game, we do not include them as part of the game space. However, we specifically want to distinguish between playing characters and NPC. The former are assumed to be independent of the designer while in the second type, the designer has control over their definition within the *design space* as we shall see below.

Social constructs. Describe the way individuals are related among themselves and also serve as means to refer to individuals and collectives by the role they play rather than by who they actually are. These may include: roles; relations among roles (n-ary relationships between individuals as well as higher-order relationships. i.e, groups, hierarchies of roles, power relationships and so on); organisations (groups plus coordination conventions)

Actions . It is worth distinguishing at least three types: individual actions (pick up a sword, climb a wall, move towards an object); interactions (actions involving two or more agents like attack an enemy, ask for directions, proclaim an outcome) and actions towards game-generic constructs (change a level, embark in a challenge)

Languages. These are needed to define the behaviour of the system and the way it is regulated. These may be organised as a hierarchy of languages that starts with a *domain* language (to refer to the basic game objects: mountains, walls, sword, attire, coins,...) that includes terms of higher *action* languages (description of an action); followed by *constraint* languages (preconditions and post-conditions of actions); then *normative* languages (procedural, functional or operational directions; behavioural rules,...) and so on, depending on the complexity of the definition of the gae and the particular choice of aspects.

Social order constructs. To allow top-down or bottom-up articulation of interactions, the usual device is to use different types of norms: procedural, constitutional, rules of behaviour,...

Social order mechanisms. To allow top-down or bottom-up governance. Among these: regimentation (rendering some actions impossible, strict application of sanctions,...); social devices (trust, reputation, prestige, status, gossip); policing devices (law enforcement),...

Evolution. The game may evolve over time as a result of emerging social conventions, adaptation to different populations of players or to some performance criteria like the quality of engagement or the success rate in some challenges. The definition of the game should include the devices through which that change happens: performance indicators, normative transition functions and such.

Inference. In case the description of situations is somewhat normative, the designer may want to postulate different ways of inferring intended or observed behaviour. For example, *classical logical inference* to allow norm-aware agents to decide whether or not to comply with a norm at some point, to allow police-like NPC to infer a potential misconduct, and so on; reasoning under *uncertainty*; *coherence* as alternative to classical forms of inference when validating game conventions off-line or monitor on-line evolution of a game.

Information structures. that are associated with the main entities of the game, agent profiles and the profiles of active game-specific constructs. In particular, every game needs to keep that information that may change as the game is played: the (shared) state of the system (the value of each and every variable that may change through the action of some agent or the passing of time),

3.3 The Game Environment

While the game space is an ideal description of the game, there should be some device to turn that description into code that allows humans and software agents to be part of an enactment of a hybrid social game. The game environment is made by those technological artefacts that allow that to happen.

As Fig.2 suggests, the game environment contains all the data structures and operations that allow the implementation of the instantiations of all those constructs needed to make a precise description of a game space and the op erational semantics that determine when an input to the system is admitted and its effect.. Thus one needs to have data structures and algorithms to refer to swords and agent types, to denote the attempt to climb a wall or attack and alien ship, to represent challenges and other collective contexts, to establish regimented regulations, allow adequate transparency for social order constructs to apply, and in essence to represent, control, and update the state of the game at every instant the game is being played.

Ideally, there should be a formal definition of these data structures, operations and semantics so that (i) they may be implemented in one or various architectures (centralised, distributed or mixed), (ii) an appropriate specification language may be built and (iii) the corresponding middleware produce a run-time version of the game (in "T") that allow players (in"W") to engage in the game.

3.4 Design Space and Design Environment

While the game space is the ideal game, the designer has to make choices as to what are going to be the actual components of the game in function of some design criteria and assumptions regarding the decision-making capabilities of participants (PCs as well as NPCs), the intended level of complexity, the expected or desired evolution of the game, the purported level of engagement , etc. While we presume that most challenges are dealt with thanks to the repertoire of constructs available for the game space, the design space will contain other constructs to complement the activation of the game space.

All those components, in turn, are then supported by technological artefacts that will eventually constitute the environment where the game is designed, assembled , enacted and maintained. Schematically, as depicted in Fig. 3, the design environment will include:

- The design and activation of NPC and support agents:

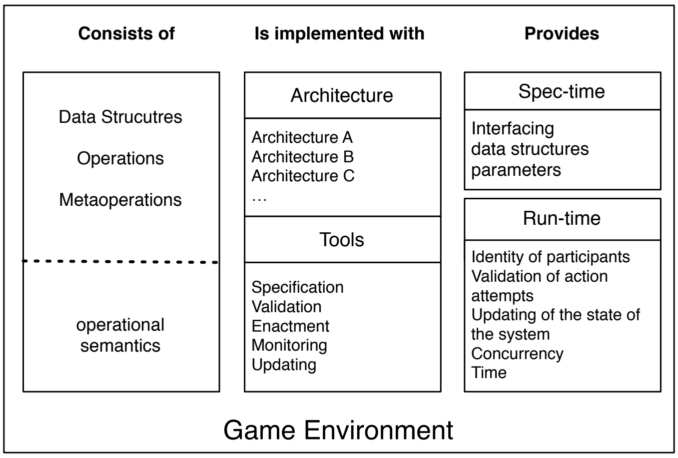

Figure 2. Game Environment

- Tools for monitoring and updating the game.
- In general, services and repositories of data and of knowledge. Services like a model to calibrate parts of the game, trust and quality assessment, and in social simulation systems, econometric or demographic models to supplement agent-based models, scenario specification, performance indicators, parameter changing functions,... . Repositories like heuristics for experimental design or calibration of performance indicators, environmental or economic data and so on.
- Finally, one may want to consider as part of the design space other support and management technologies that, being associated with the game, are arguably not part of it. For instance, a forum-like facility to exchange messages among players; a polling device to get opinions about possible game extensions or other games; and of course all the back-office support.

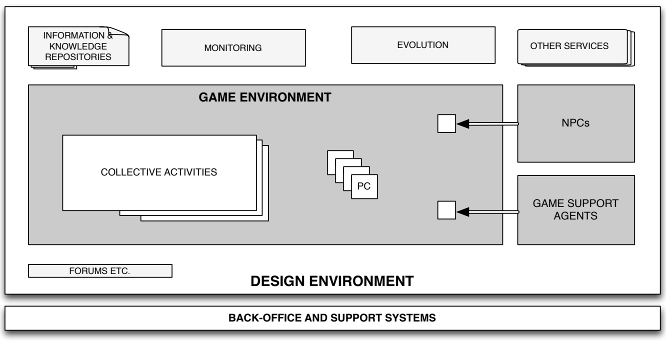

Figure 3. The Design Environment includes the game environment plus those services that make the game playable

4 CLOSING REMARKS

4.1 Separation of concerns

While we motivated this paper as a way of making the use of artificial intelligence more valuable for games, we postulated to address this problem by dividing it in two. We proposed an obvious separation of concerns between design and implementation of those NPC that that the designer wants to have in the game, on one side, and on the other, those of the game milieu and its components. The paper dealt only with the second aspect by making three proposals:

- We propose a three-fold understanding of games (depicted in Fig. 1) and from that picture we proposed a second and perhaps not so obvious, separation of concerns between *game design* and *game implementation*.
- We propose a list of *aspects* that need to be taken into account in the design of a game, propose that for each of these aspects, there may be different "conceptual constructs" that may be assembled to make those aspects concrete for a given game, and for each of these conceptual constructs and their assembly there should be a *computational counterpart* that implements them.
- Finally we propose that the design and enactment of the actual game requires another layer composed by a design space that includes conceptual means to choose the constructs of the design space and an environment where the implementation of the game is complemented with other artifacts that allow its enactment

We are confident that this approach is one reasonable way of addressing the clumsiness of NPCs without loosing control over the game that we mentioned at the top of the paper. Moreover, it is our impression that game developers take a shortcut from the game or simulation idea to the tools and architectures, then iterate back to the game space. This limits of course the game space contents and constructs used. Computer scientists may use all elements but lack the input from the social science on how to fill the game or deign space with values based on sound theories and empirical work from the social sciences. Thus the contribution potential of both communities to each other is clear. Analysing existing games using these constructs as well as rebuilding them using the design proposal is a natural next step.

4.2 Backing

A substantial influence is the work on electronic institutions. The EI framework (see [8]) is a particular, restricted, version of the framework we propose here. Figure 4 suggest how the constructs we propose for the game space extend the EI ones. Moreover, some of the ideas of the design space are already present in an extension of the EI framework with services for simulation [3]. Likewise, the experience of an implementation architecture and the corresponding EIDE development tools [10], their light-weight variants suggested in [9, 12] and their 3–D extensions [13] give an inkling of what is involved in the tasks ahead.

4.3 Games as a special case of coordinated social spaces

Our game space is but a particular collection of aspects, most of which are also relevant for several types of controlled spaces where many individuals interact in some endeavour that they cannot achieve in isolation. Similarly, the game environment would also have a counterpart in these coordinated social spaces. What may not be so obvious, though is that an analogue of the design space and the design environment are also applicable. An immediate example is the case of participatory agent-based simulation, where a straightforward specialisation of the game space would be the actual simulation model and the environment, the implemented model.

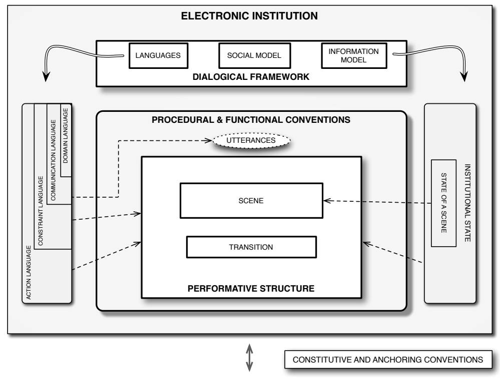

Figure 4. The conceptual space of electronic institutions

One layer up, the design space and environment would include the conceptual tools—statistical techniques, scenario definition, macro-view models—and their technological counterparts—services and repositories—that allow for the design, enactment and analysis of experiments. It would be appropriate to contrast the proposed game space components with the conceptual needs of sociological theories and enrich the game space accordingly.

We have drawn inspiration from the work in normative multiagent systems ([7, 6, 5]. In particular, the reader will notice a closer affinity of this proposal with the proceedings of the Dagstuhl Normas2012 workshop [2] and a version of Fig. 1 and something akin to the design space are included in the forthcoming follow-up volume of the workshop. We claim that our framework would fit nicely with a large number of open regulated multiagent systems.

Finally, we would like to note that the intuitions behind Fig.1 apply not only to hybrid social games or normative MAS but we believe the three-fold correspondence applies to a large variety of sociotechnical systems and our framework could be tuned to the peculiarities of several of them. We propose to abstract from the game space to a space where *agreements* take place and reify the environment as an open environment where computation by agreement is feasible. Perhaps this is one way to move towards the understanding of social intelligence.

ACKNOWLEDGEMENTS

We would like to thank the referees for their comments which helped improve this paper. This research was partly funded by SINTELNET FET Open Coordinated Action (FP7-ICT-2009-C Project No. 286370) and Consolider AT project CSD2007-0022 INGENIO 2010 of the Spanish Ministry of Science and Innovation.

REFERENCES

[1] Diana F. Adamatti, Jaime Simão Sichman, and Helder Coelho, 'An analysis of the insertion of virtual players in gmabs methodology using the vip-jogoman prototype', *Journal of Artificial Societies and Social Simulation*, **12**, (2009).

[2] Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leon van der Torre, 'Normative Multi-Agent Systems (Dagstuhl Seminar 12111)', *Dagstuhl Reports*, **2**(3), 23–49, (2012).

[3] Josep Lluis Arcos, Pablo Noriega, Juan A. Rodriguez-Aguilar, and Carles Sierra, 'E4mas through electronic institutions.', in *Environments for Multi-Agent Systems III.*, eds., D. Weyns, H.V.D. Parunak, and F. Michel, number 4389 in Lecture Notes in Computer Science, 184–202, Springer, Berlin / Heidelberg, (08/05/2006 2007).

[4] Richard A. Bartle, *Designing virtual worlds*, New Riders., 2003.

[5] Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, eds. *Normative Multi-Agent Systems*, volume 09121 of *Dagstuhl Seminar Proceedings*, Dagstuhl, Germany, 15/03/2009 2009. Schloss Dagstuhl, Leibniz-Zentrum fuer Informatik, Germany.

[6] Guido Boella, Gabriella Pigozzi, and Leender Van der Torre, 'Five guidelines for normative multiagent systems', in *Legal Knowledge and Information Systems. JURIX 2009*, ed., Guido Governatore, pp. 21–30, Amsterdam, (October 22-24 2009). IOS Press.

[7] Guido Boella, Leendert van der Torre, and Harko Verhagen, 'Introduction to the special issue on normative multiagent systems', *Autonomous Agents and Multi-Agent Systems*, **17**, 1–10, (2008).

[8] Mark d'Inverno, Michael Luck, Pablo Noriega, Juan A. Rodriguez-Aguilar, and Carles Sierra, 'Communicating open systems', *Artificial Intelligence*, **186**(0), 38 – 94, (2012).

[9] Marc Esteva, Juan A. Rodriguez-Aguilar, Josep Lluis Arcos, and Carles Sierra, 'Socially-aware lightweight coordination infrastructures', in *AAMAS'11 12th International Workshop on Agent-Oriented Software Engineering*, pp. 117–128, (2011).

[10] Marc Esteva, Juan A. Rodrguez-Aguilar, Josep Lluis Arcos, Carles Sierra, Pablo Noriega, and Bruno Rosell, 'Electronic Institutions Development Environment', in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '08)*, pp. 1657–1658, Estoril, Portugal, (12/05/2008 2008). International Foundation for Autonomous Agents and Multiagent Systems, ACM Press.

[11] R. Zubek R. Hunicke, M. LeBlanc, 'Mda: A formal approach to game design and game research', in *Proceedings of the Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence.*, (2004).

[12] David Robertson, 'A lightweight coordination calculus for agent systems', in *Declarative Agent Languages and Technologies. DALT 2004*, volume 3476, pp. 183–197. Springer, (2005).

[13] Tomas Trescak, Inmaculada Rodriguez, Maite López-Sánchez, and Pablo Almajano, 'Execution infrastructure for normative virtual environments', *Engineering applications of artificial intelligence*, **26**, 51–62, (01/2013 2013).