Higher-Order Knowledge in Computer Games

Andreas Witzel and **Jonathan A. Zvesper**[1]

- *Hello, I'm looking for the room for making love.*
- *Oh, right. You must mean the Honeymoon Suite. Well, that's straight that way, can't miss it.*
- *I know where it is. I just wanted you to know that I know where I'm going, so you needn't bother with me.*

"The Missing Ingredient"
First part of the movie "Four Rooms" (1995)

Abstract. Our main aim is to raise awareness of higher-order knowledge, i.e. knowledge about knowledge, as an issue in simulating realistic non-player characters in computer games. We motivate the importance of higher-order knowledge with arguments, as well as a few examples. We survey existing games and literature to show that this issue is currently neglected. We also refer to our earlier work to illustrate one approach to simulating higher-order reasoning, which we call "explicit knowledge programming". Finally we describe a number of issues which arise when carrying out such an implementation, some of which go beyond the scope of the present motivation of computer gaming, and are of more general interest.

1 Introduction

If you ask people why they do things, they often give you reasons in terms of knowledge and beliefs.[2] The answer to the question, "Why did the chicken cross the road?", i.e., "To get to the other side", does not tell the whole story: if we were to attribute human agency to our feathered agent, then a complete answer would be couched in terms of her beliefs and desires. Because we are social animals, these reasons sometimes will involve several agents. In order to reason about interactive behaviour amongst such groups of agents, we have recourse to *higher-order* beliefs (which are beliefs about beliefs). For example, if Ann knows that Bob knows that the party starts at 8, then she usually won't tell him, unless she wants to make sure that he knows that she knows it. Indeed, how could one reason about such a situation without talking about higher-order beliefs? Even if Paul Churchland were correct, and one's talk of beliefs were unscientific nonsense, still it is pervasive and has some explanatory force. If people engage, consciously or unconsciously, in reasoning about beliefs, then a natural way to simulate some aspects of human behavour, particularly those aspects involving interaction, is to engage in reasoning about beliefs.

[1] ILLC, University of Amsterdam, and CWI, Amsterdam, The Netherlands, e-mail:`{awitzel,jonathan}@illc.uva.nl`

[2] We view "belief" as a more general notion than "knowledge": knowing something implies believing it. Concepts are defined in analogous ways for belief and knowledge, and we will use "epistemic" to mean "of/about beliefs/knowledge". In the more technical parts we will for simplicity focus on knowledge.

An important goal of interactive fiction (IF) games or, more generally, computer role playing games (RPGs) is to simulate convincingly social situations within a virtual world. We believe that to this end epistemic reasoning is crucial. It would therefore be natural for a programmer to be able to describe the behaviour of computer-simulated, non-player characters (NPCs) using rules that contain explicit knowledge statements, including higher-order ones. However, as we will see in Section 2, in current computer games this epistemic aspect is surprisingly overlooked.[3] To strengthen our argument, in Section 3 we will give some scenarios, involving higher-order epistemic reasoning, that we find plausible to occur in RPGs.

In [28] we described an approach for providing knowledge statements on the level of a programming language, and for proving that such statements are evaluated faithfully with respect to a formal definition of knowledge. That approach is modular in the sense that all aspects of epistemic modeling can be implemented in a dedicated knowledge module. In the context of computer games, an NPC scripter could thus use high-level tools and would be alleviated from having to keep track of these things explicitly and manually. We will briefly describe that approach, which uses ideas from epistemic logic, in Section 4.

We will conclude by raising some open issues in Section 5. These are concerned with making the knowledge module tractably implementable by restricting the possible inputs and queries. The basic questions are:

- What kinds of events should be used as inputs to change the state of the module, and how to formalize them? Even a simple event, like the player picking up an object, may have an effect on some agents' (higher-order) knowledge. These effects will depend on subtleties of the wider situation.
- What kinds of queries to the module are relevant? Not all theoretically possible knowledge statements will occur or even be realistic to compute. Since the goal is to simulate real-life social interaction, to answer this question one has to look at human higher-order reasoning.

2 State of the Art

We will briefly review the current state of the art with respect to epistemic modeling in computer games, both in the "real world" and in (academic) research.

2.1 Real World

The state of the art in commercial computer games is not easy to judge, since computer game companies are not very keen on publish-

[3] When we refer to "computer games" in this paper, we have games like RPGs and IF in mind. For other games, like poker, epistemic aspects have indeed been considered.

ing the details of their AI implementations. So if one doesn't want to rely on the enthusiastic slogans from marketing departments, then the best sources of information about computer game AI are private web pages like [29] where observations and analyses from playing, interview quotations, and possibly the website creator's own knowledge and experience as AI programmer are carefully collected and presented. For an extensive list of online resources, see [23].

From these resources it becomes evident that epistemic reasoning is definitely not in the focus of existing computer game AI, and we did not find any mention of higher-order reasoning. For example, the highly acclaimed Radiant AI engine is used in the RPG *The Elder Scrolls: Oblivion*™ [6] to make the game more lifelike. The following quotation from an interview [5] during the testing phase of the game AI illustrates its functioning:

> One [non-player] character was given [by the testers] a rake and the goal "rake leaves"; another was given a broom and the goal "sweep paths," and this worked smoothly. Then they swapped the items, so that the raker was given a broom and the sweeper was given the rake. In the end, one of them killed the other so he could get the proper item.

Obviously, the characters didn't mutually know their interests, or they couldn't make use of that knowledge.

To us it seems natural that one would use a logic-based approach in order to effectuate epistemic reasoning. Indeed, a logic-based formalism is an important part of what we will suggest in Section 4.

References in these directions are scarce. In [18], it is suggested to use logic for NPC scripting; however, higher-order epistemic reasoning is not considered, and the article seems to have been left at a brainstorming stage and not followed up on. The clearest statement promoting the use of epistemic reasoning comes from the famous IF writer Emily Short [25]:

> *Abstract Knowledge.* One of the artificial abilities we might like to give our NPCs, aside from the ability to wander around a map intelligently and carry out complex goals, is the ability to understand what they are told: to keep track of what items of knowledge they have so far, use them to change their plans and goals, and even draw logical inferences from what they've learned.

It is not clear whether this refers to higher-order knowledge, or whether "abstract" just is meant to imply that the implementation should be generic and encapsulated in a knowledge module; in any case, the currently existing IF implementation of such ideas [11] is restricted to pure facts and does not include any reference to the possibility of higher-order knowledge.

2.2 Research

Computer game AI is slowly becoming accepted as an area for serious academical research [15, 13]. In recent years, conferences and other scientific meetings on the interface of artificial intelligence and computer games have emerged [24, 4, 17] and special issues of magazines have appeared [1, 2, 3].

Again, where knowledge is considered, the concern seems to be exclusively domain knowledge, or knowledge about facts in the game world, as in [22, 26]. A more general approach of using agent programming languages to script NPCs (e.g. [16]) inherits the epistemic reasoning facilities of such languages – which tend to focus on facts. The closest in spirit to higher-order modeling are attempts to detect the general attitude of the human player (for example, aggressive or cautious) and to adjust the game AI accordingly. But we could find no references to explicit higher-order epistemic modeling.

The ScriptEase system [8] is an academic approach to NPC scripting, which was motivated by the insight that the scripting process needs to be simplified. It provides a graphical interface for generating behaviours of characters in a commercial RPG. However, knowledge statements to steer the behaviour are not considered.

A very interesting approach, described in [9], uses deontic logic to specify NPC behaviour in a rule-based fashion. While epistemic issues are not considered there, a fusion of these two aspects could provide a highly suitable system for scripting believable social agents.

Some literature on higher-order reasoning in multi-agent systems that does *not* focus on computer games is also very relevant. In [10], the specific problem of agent *communication* is studied, in which agents weigh costs against expected benefit of communication. The authors point out the importance of using higher-order reasoning, in the form of beliefs about beliefs, when agents make such assessments. Their particular interest is in formal representation of belief "*abduction*". We do not consider abductive reasoning here, but we recognise that it is also important in our settings.

We also note that higher-order reasoning is discussed in [30] in the context of a Petri Net method for designing "intelligent team training systems". The authors suggest that using Petri Nets can help to overcome tractability issues in epistemic reasoning. However, they note that communication, an important ingredient in the kind of social interaction we wish to simulate, "is more complicated than Petri Nets can represent". We do not consider the Petri Net formalism further, but if progress is made in this area it could be of relevance.

3 Scenarios and Examples

Having seen that higher-order epistemic reasoning is not currently considered in the context we are interested in, we will now describe a few small scenarios of social interaction naturally involving this kind of reasoning. We believe that by simulating the aspects of the real world that are highlighted by these examples, computer game worlds will become more convincing.

Scenario 1. If Ann is (openly) present when Carl tells Bob about some fact, then she won't tell the same fact to Bob again a minute later. Indeed, doing so under usual circumstances would be puzzling and inexplicable behaviour.

Scenario 2. If Ann gets an advantage from lying to Bob, and knows that he doesn't know the truth then she might indeed lie; if she knows he does know then she usually won't; if she doesn't know whether he knows then her decision may depend on other circumstances.

Scenario 3. Part of being a doctor is the obligation to help people around you whom you know to be sick.[4] Imagine Ann is a doctor who unfortunately would profit of getting rid of Bob. Not only will she take care that no-one sees her putting the poison into his glass, she will also make sure (e.g. by immediately going on holidays) that no-one knows that she knows that Bob is dying, because otherwise not saving him would make her suspicious.

The simple rule of pragmatics in Scenario 1 (that things which are commonly known[5] aren't usually worth stating) makes essential use

[4] This scenario is inspired by a discussion from [20].

[5] Something is common knowledge among a group of agents if all agents know it, know that they all know it, and so on ad infinitum.

of epistemic reasoning. Moreover, whatever character type Ann in Scenario 2 is supposed to have, implementing it should be facilitated by epistemic operators. The last scenario may seem a bit contrived, but on the other hand it is very common for game characters to have professions, and in a world of intrigue and adventure having to get rid of someone is also quite conceivable. It manifestly also involves epistemic reasoning.

The variety and generality of these scenarios illustrates how applicable the underlying ideas are in many different situations which could occur in IF or RPGs.

4 Making Knowledge Explicit

If more attention should be paid to epistemic reasoning in simulating human interaction in IF and RPGs, a natural question is: how should this be done? In this section we propose an approach for tackling the problem of programming a simulation of higher-order epistemic reasoning.

The approach is based around epistemic logic.[6] That is, we propose using some formal language in which epistemic statements can be formulated and evaluated. The formulae of such a language might for example be built recursively from "atoms", which are non-epistemic facts, by using propositional connectives, like $\vee$ (or) and $\neg$ (not), and knowledge operators K_a, one for each agent a being simulated. We will call these "epistemic formulae". So for example if p were an atom, then $K_a K_b p$ could be an epistemic formula with the intended reading "a knows that b knows that p".

In [28], we described a simple and preliminary implementation to provide statements involving explicit knowledge formulae on the programming language level, and proved the implementation to be sound with respect to a formal notion of knowledge defined on the level of the underlying process calculus. We took a modular approach, writing a knowledge module that is instantiated for each process. We will briefly review this work in the following.

In our particular implementation, the events that were used as inputs to the knowledge module were always synchronous communications between two of the processes concerning the values of some bits. In general though, an event can be anything which would have epistemic effects. The idea is to pass to the knowledge module of a process a the events that a 'observes'.

The queries to which the knowledge module can respond are epistemic formulae. We used a formal language with atoms $p_{x_0}, \neg p_{x_0}, p_{x_1}, \neg p_{x_1}, \ldots$ for each of the bits $x_0, x_1, \ldots$, and a knowledge operator $K_a, K_b, \ldots$ for each of the processes $a, b, \ldots$.[7] Then, as an example, the formula $K_c K_b \neg p_{x_2}$ means that process c knows that process b knows that x_2 has the value 0. If the knowledge module of process a were passed this formula it could respond by saying "yes", "no" or "don't know". If the module were to respond "yes", then this should be interpreted as (the agent which is modelled by process) a knowing that c knows that a knows that x_2 is 0. A programmer can use such queries, for example, in conditional statements and thus have the program flow depend on the process's knowledge.

Even with a simple implementation, it was desirable to prove that it was in some sense "correct". Thus we used a formalism from the literature on epistemic logic, namely Kripke models. The argument for correctness of the implementation then proceeds in two steps, which can roughly be stated as follows:

- Argue that a particular model $\mathcal{M}$ represents faithfully the intuitive situation which we intended to capture.
- Prove that knowledge formulae are evaluated in the same way by process a after the sequence of events σ as they are by agent a in the model $\mathcal{M}$ after the same sequence of events.

One criticism that one can make of using a Kripke model formalism as an intermediate step is that that formalism itself can appear unintuitive. However, we know of no more philosophically grounded and mathematically robust formalism with which to work in the context of reasoning about higher-order knowledge. (In order to deal with various phenomena like so-called "explicit belief", or inconsistent beliefs, many other models have been proposed, but these are all essentially refinements or variations of Kripke models – see [19, Sections 2.4 to 2.13] for a selective survey.)

The Kripke model that we specified for the particular implementation we had in mind resembled an "interpreted system" from [12]. It was not our aim to implement directly an entire interpreted system, which in this case is an infinite structure. Even if it is finitely representable, we might only be interested in certain parts of it.

In general, an implementation can be simplified by considering a subclass of the formulae that would be in the full logical language which the model could interpret. As it happened, for the particular implementation we had in mind (a distributed implementation of an algorithm for eliminating dominated strategies in strategic games), it was only necessary to consider formulae from the very simple epistemic language that we have described.

In the case of simulating realistic *human* agents (so in particular within IF and RPGs), the limits to human cognitive faculties should be taken into account. So for example, would it make sense to allow as queries to the knowledge module epistemic formulae involving complex iterations as in, "Ann believes that Bob believes that Carl doesn't believe that Ann believes that Derek believes that it's raining"?

5 Open Issues

This brings us to the question of what inputs and queries to the knowledge module ought to be allowed. The two parts of this question are to some extent independent.

The first part concerns the events in the game world that should affect the knowledge states of the characters. In a way the issue of what events should be taken into account is up to the designer of the world. We think it is difficult to make a general statement about which kinds of events matter and which don't.

Once the events that the virtual world generates are more or less decided, we need abstract representations of them and specifications of their exact conditions and epistemic effects. Again it is difficult to make general statements, because they depend on the specific event. For example, if an event consists in the player picking up an object, the simplest approach would be that it becomes common knowledge between all present agents that the player possesses that object. But there is some freedom in the degree of fine-grainedness and detail in which this event should be represented and processed. Should simply all agents in a certain radius gain common knowledge, or should it be taken into account in which direction they are currently looking and whether everyone is mutually visible?

The second part of the question, which is clearer than the previous part and which we believe may be answered empirically, concerns

[6] The research field of epistemic logic can be said to have been initiated with [14].

[7] Note that here we are not using the richer language that could be built using also the connectives $\neg$ and $\vee$ mentioned above.

the class of knowledge statements that matter and that the knowledge module should be able to handle. Here again the fact can be used that the computer games we have in mind want to realistically model human social interaction. The question, which is also of independent interest, then becomes: What class of epistemic formulae are evaluated by humans in everyday life, consciously or not?

Some results from experimental game theory about levels of strategic thinking [7] can be interpreted as being relevant to this question. However, these experiments do not focus on everyday social interaction. For example, the Beauty Contest game mentioned in the previous reference might invoke conscious and explicit reasoning about the other agents, while we believe that, through years of experience in social interaction, the requisite higher-order reasoning processes may also occur on a more intuitive and reflexive level.

Furthermore, the experimental designs are in general not specifically concerned with knowledge, so that at best the results can give us hints about the nesting depth of knowledge operators. Clearly other criteria might define the class of knowledge formulae that are of relevance in social interactions.

For example, from Scenario 2 it is clear that formulae like

$$K_a \neg K_b p$$
$$K_a K_b p$$
$$\neg(K_a K_b p \vee K_a \neg K_b p)$$

matter. However, that doesn't necessarily hold for all formulae of knowledge operator depth 2. Also, human reasoning capabilities may not be monotonic with respect to this complexity measure. For example, for a special concept like common knowledge, which in theory involves infinite depth, we might want to assume that humans are able to cope with it, while this does not hold for "intermediate" depths like 10000.

The main issue thus remains: How can we define this class of relevant formulae?

In [27], results from experiments suggest that subjects use first-order Theory of Mind (beliefs about others' beliefs), but not "all kinds of reasoning possible and useful in this context". This supports the claim that the depth of knowledge operators is not the only relevant criterion. It is further reported that some subjects use second-order Theory of Mind, which corresponds to third-order epistemic formulas.

While it is interesting to look at such questions in experimental setups, an alternative approach could shed additional light on these issues, namely to come up with realistic social situations and think about what kinds of reasoning processes go on. This is basically what we did in Section 3, and work by Parikh is an excellent source for enlightening examples (see e.g. [21]). Such thought experiments or observations from real life can be convincing enough to remove the need for abstractions and reproducible lab conditions as provided by experiments, for the benefit of being set in more natural environments, where human reasoning capabilities possibly profit from experience and training in specific social situations.

6 Summary

The main aim of this paper was to raise awareness of higher-order knowledge as an issue in simulating realistic non-player characters in computer games. Section 2 surveyed existing games and literature and found that this has not yet been done. Section 3 gave some illustrative and motivating examples of what we mean by higher-order knowledge, in situations that could plausibly occur in IF and RPGs. Section 4 gave an example of an implementation of explicit knowledge programming. One suggestion of this paper is that this is a sensible and realistic approach to implementing higher-order knowledge reasoning, and therefore to simulating some interactive aspects of human behaviour. Finally, Section 5 described a number of issues which arise when carrying out such an implementation. We raised issues which go beyond the scope of the present motivation of IF and RPGs, and are of general interest.

Acknowledgments

We would like to thank three anonymous reviewers for their very helpful comments.

This research was supported by a Marie Curie Early Stage Research fellowship in the project GLoRiClass (MEST-CT-2005-020841).

REFERENCES

[1] *IEEE Intelligent Systems*, **17**(4), (Jul/Aug 2002). Special Issue on Interactive Entertainment.

[2] *IEEE Intelligent Systems*, **21**(5), (Sept/Oct 2006). Special Issue on Interactive Entertainment.

[3] *Science of Computer Programming*, **67**(1), (June 2007). Special Issue on Aspects of Game Programming.

[4] *Proceedings of the 2005 IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games*, eds., David W. Aha, Héctor Muñoz-Avila, and Michael van Lent, 2005.

[5] Wikipedia article. The Elder Scrolls IV: Oblivion. `http://en.wikipedia.org/w/index.php?title=The_Elder_Scrolls_IV:_Oblivion&oldid=96078473#Radiant_A.I.`, December 2006. (The relevant section was removed in later revisions, since it concerned "Radian AI's behavior prior to the game release. Listing all these examples is beyond the scope of this article." We did not succeed to find the original interview from which the section was taken.).

[6] Bethesda Softworks. The Elder Scrolls: Oblivion. `http://www.elderscrolls.com/games/oblivion_overview.htm`, 2006.

[7] Colin F. Camerer, 'Behavioural studies of strategic thinking in games', *Trends in Cognitive Sciences*, **7**, 225–231, (May 2003).

[8] Maria Cutumisu, Curtis Onuczko, Matthew McNaughton, Thomas Roy, Jonathan Schaeffer, Allan Schumacher, Jeff Siegel, Duane Szafron, Kevin Waugh, Mike Carbonaro, Harvey Duff, and Stephanie Gillis, 'Scriptease: A generative/adaptive programming paradigm for game scripting', *Science of Computer Programming*, **67**, 32–58, (June 2007).

[9] Flávio S. Corrêa da Silva and Wamberto W. Vasconcelos, 'Rule schemata for game artificial intelligence', in *Advances in Artificial Intelligence - IBERAMIA-SBIA 2006*, Ribeirão Preto, Brazil, (2006). Springer.

[10] Aldo Franco Dragoni, Paolo Giorgini, and Luciano Serafini, 'Mental states recognition from communication', *Journal of Logic and Computation*, **12**, 119–136, (February 2002).

[11] Eric Eve. Epistemology (Inform 7 Extension). `http://www.inform-fiction.org/I7Downloads/Extensions/Eric%20Eve/Epistemology`.

[12] Ronald Fagin, Joseph Y. Halpern, Moshe Y. Vardi, and Yoram Moses, *Reasoning about knowledge*, MIT Press, 1995.

[13] Chris Fairclough, Michael Fagan, Brian Mac Namee, and Padraig Cunningham, 'Research directions for AI in computer games', in *Twelfth Irish Conference on Artificial Intelligence and Cognitive Science*, pp. 333–344, (2001).

[14] Jaakko Hintikka, *Knowledge and Belief: An Introduction to the Logic of the Two Notions*, Cornell, 1962.

[15] John E. Laird and Michael van Lent, 'Human-level AI's killer application: Interactive computer games', *AI Magazine*, 15–25, (2001).

[16] João Leite and Luís Soares, 'Evolving characters in role playing games', in *18th European Meeting on Cybernetics and Systems Research (EMCSR 2006)*, volume 2, pp. 515–520, Vienna, (2006).

[17] Sushil J. Louis and Graham Kendall, eds. *Proceedings of the 2006 IEEE Symposium on Computational Intelligence and Games (CIG06), University of Nevada, Reno, campus in Reno/Lake Tahoe, 22-24 May, 2006*. IEEE, 2006.

[18] Sami Mäkelä. NPC Scripting and Reasoning about the NPC behaviour. `http://www.worldforge.org/project/newsletters/November2001/NPC_Scripting`.

[19] John-Jules Ch. Meyer and Wiebe van der Hoek, *Epistemic Logic for AI and Computer Science*, Cambridge University Press, 1995.

[20] Eric Pacuit, Rohit Parikh, and Eva Cogan, 'The logic of knowledge based obligation', *Synthese*, **149**(2), 311–341, (March 2006).

[21] Rohit Parikh, 'Levels of knowledge, games, and group action', *Research in Economics*, **57**, 267–281, (September 2003).

[22] Marc Ponsen, Pieter Spronck, Héctor Muñoz-Avila, and David W. Aha, 'Knowledge acquisition for adaptive game AI', *Science of Computer Programming*, **67**, 59–75, (June 2007).

[23] Craig Reynolds. Game Research and Technology. `http://www.red3d.com/cwr/games/`.

[24] Jonathan Schaeffer and Michael Mateas, eds. *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference, June 6-8, 2007, Stanford, California, USA*. The AAAI Press, 2007.

[25] Emily Short. Conversation. `http://emshort.wordpress.com/writing-if/my-articles/conversation`.

[26] Pieter Spronck, Marc Ponsen, Ida Sprinkhuizen-Kuyper, and Eric Postma, 'Adaptive game AI with dynamic scripting', *Machine Learning*, **63**, 217–248, (June 2006).

[27] Rineke Verbrugge and Lisette Mol, 'Learning to apply theory of mind', *Journal of Logic, Language, and Information*, (2008). To appear.

[28] Andreas Witzel and Jonathan Zvesper, 'Epistemic logic and explicit knowledge in distributed programming (short paper)', in *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, (2008). To appear.

[29] Steve M. Woodcock. The Game AI Page. `http://www.gameai.com`.

[30] Jianwen Yin, Michael S. Miller, Thomas R. Ioerger, John Yen, and Richard A. Volz, 'A knowledge-based approach for designing intelligent team training systems', in *Agents*, pp. 427–434, (2000).