3D Immersion in Virtual Agents Education

Cyril Brom, Jakub Gemrot, Ondřej Burkert, Rudolf Kadlec, and Michal Bída

Charles University, Faculty of Mathematics and Physics
Malostranské nám. 2/25, Prague, Czech Republic
brom@ksvi.mff.cuni.cz,
{jakub.gemrot,ondrej.burkert,rudolf.kadlec,
michal.bida}@gmail.com

Abstract. Many projects featuring intelligent virtual agents have emerged in last years, but not many reports on their advances in education. This paper presents the curricula of a university course on Modelling Behaviour of Human and Animal-like Agents, including a seminar in which students develop their own virtual agents using a toolkit we have developed. This course has been also scaled for a workshop with computer science high-school students. An evaluation of the course is presented and main lessons learned overviewed. The paper also explicitly formulates the teaching methodology underpinning the course and outlines several general questions hoping to start a broader discussion on educational issues related to the field of intelligent virtual agents.

Keywords: Intelligent virtual agents, storytelling, education, course curriculum.

1 Introduction

The field of intelligent virtual agents (IVAs) and interactive storytelling (IS) is becoming matured. As the community is growing, an issue how to educate its new members, in most cases undergraduate students, starts to demand attention. Educational issues seem to be well addressed by neighbouring disciplines, namely computer graphics, robotics, and software agents; consider e.g. the number of summer schools and introductory books available for these domains, the RoboCup platform, or the special educational track at the Eurographics conference. While the classes directly focused on IVAs have already started to emerge, and some of the class materials are available on-line, e.g. [2, 6], the reports on advances in education directly related to IVAs and IS are scarce.

In this paper, we report on two things. First, the paper describes the curricula of a one-term theoretical course on *Modelling Behaviour of Human and Animal-like Agents* [3], highlighting main lessons learned from teaching perspective. The course was created in 2005 primarily for undergraduate computer science students, but it was also adopted for a high-school workshop and for a summer school. While primarily focussing on modelling behaviour of IVAs in the context of interactive applications such as serious games and virtual storytelling, the course also provides students with limited knowledge of cognitive and behavioural sciences, including emotional, perception, and memory theories, natural neural networks, and basics of ethology, presenting a starting point for students considering studying these subjects in the future. As far as we know, there is no single book covering the whole curricula.

U. Spierling and N. Szilas (Eds.): ICIDS 2008, LNCS 5334, pp. 59–70, 2008.

Second, the paper describes integration of this theoretical course with a practical seminar, developed in 2008, during which students learn to program their own virtual characters. Although there is an excellent entry-level tool for building simple agents and running social simulations that is used for seminars concerning classical software agents, NetLogo [19], we do not find it suitable as a main tool for students attending an IVA/IS course. It does not allow for creating agents with complex behaviour and possesses no complex 3D environment. This prevents students from exploring advanced applications such as virtual storytelling or 3D games, which, in effect, reduces their motivation. Computer game development courses tend to capitalise on 3D game engines like Counter Strike, Quake or Unreal Tournament (UT). However, coding the behaviour of a character directly in the language of a game is cumbersome for beginners. They must focus on low-level issues related to communication, navigation etc.; consequently, after dozens of hours of work, their agents "still don't do anything cool", which can be quite discouraging. In our seminar, students develop their characters for Unreal Tournament 2004 (UT04) [7], but instead of using Unreal Script, the native scripting language of the UT04, they use a special purpose toolkit Pogamut 2 [12] connected to UT04 via a Gamebot-like [1] interface. Pogamut 2 was developed to facilitate start with IVAs, both for educational and research purposes, and it is freely available to download[1]. Importantly, Pogamut 2 allows for rapid development of simple "NetLogo level of complexity" agents as well as challenging, complex agents with deliberative capabilities, thereby supporting a two-stage process of education: "doing simple things quickly" as well as "exploring the more advanced stuff".

The primary goal of this paper is to discuss the curriculum of the course in order to facilitate either development of a similar course or adoption of our toolkit for a course that is already running. In this respect, the paper is most similar to the work [8], which reported on augmentation of a course on *Autonomous Agents and Multi-agent Systems* with a practical seminar using Counter Strike 3D game. We depart from this work in that our course is directly focused on *IVAs* behaviour, and we use a toolkit featuring not only a multi-agent library but also an integrated development environment tailored to support the development of IVAs behaviour. The number of supplementary features of Pogamut 2 and its support of education distinguishes it from similar freely available tools, including solutions capitalizing on Gamebots [1], as discussed in depth in [12]. On a more general level, the aim of this paper is to start a broader discussion on educational issues related to the subject of IVAs and IS.

The paper proceeds as follows: Sec. 2 details the course's curricula, including main lessons learned concerning individual course's parts. Sec. 3 introduces Pogamut 2 and explains how it is used in the seminar. Also a small evaluation study is presented. Sec. 4 discusses a teaching methodology underpinning this course, several general lessons learned, possibility of transferring the course to another teaching context, and also opens the general discussion on education of the subject of IVAs and IS.

2 The Course

The course was created at Charles University in Prague, the computer science study program, in 2005. Its theoretical part comprises 13 lesson units (1 unit = 90 minutes).

[1] The toolkit is available for download at http://artemis.ms.mff.cuni.cz/pogamut

In 2008, the practical seminar has been added (6 units). At the end, each student must create his/her own IVA, which presents about half of the student's evaluation.

The course is tailored to computer science students at least in their fourth term of bachelor studies, after they attended several courses on programming (10.5[2]), mathematics (17), general IT skills (8.5), and algorithms (5.5). Most of them have only limited knowledge of other topics. Many are recruited from technical high-schools. Every year, the course is attended by about 30 – 50 students (about 15% of the total number of students of one grade).

Objectives of the course are (1) to introduce the field of interactive applications featuring IVAs, (2) to teach students to develop behaviour of IVAs, (3) to boost their interest in related disciplines, namely in artificial intelligence, autonomous agents, and behavioural sciences. There are many other courses where the students can learn about other aspects of IVAs, most importantly courses on computer graphics, computational linguistic, artificial intelligence (AI), autonomous agents (AA), and computer games development (see [3] for details). While there is no given order in which to attend these courses, the questionnaire administered this year showed that our lecture is the entry-level point for the subjects of AI and AA for 60% students (either the first or the only lecture related to AI). Computer graphics, on the other hand, is studied mostly earlier than or in parallel to our course. From the point of view of IS and IVAs, some courses that would give students a broader context are missing; most notably social sciences and human-computer interaction.

Curriculum. Conceptually, the course comprises an introductory lecture, three theoretical blocks, and the practical seminar. The curriculum of the theoretical part is overviewed in Tab. 1. In general, the course starts with relatively concrete models for controlling behaviour and proceeds to more abstract ones. The course has more possible orderings of lectures, specifically within the block on advanced topics (Fig. 1). The content of the course is detailed in the supplementary material to this paper [3]. The rest of this section highlights main lessons learned. The motivation for the gross structure of the course is discussed in Sec. 4.

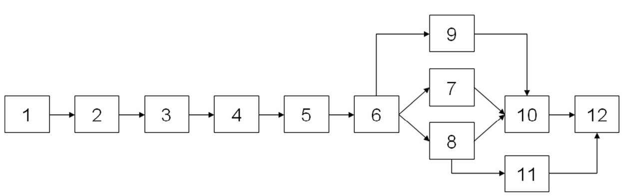

Fig. 1. Possible ordering of lecture parts

Introduction. The knowledge of many students on IVAs is minimal. They typically think that IVAs belong only to the domain of computer games. Thence, it is crucial to demonstrate the broad scope of the field. Students most welcome live state-of-art demonstrations. In this way, interactive drama, serious games, cultural heritage and

[2] Normalised number of courses on a given topic; "1 course" equals 13 lesson units (i.e. 13 x 90 minutes). The course presented in this paper, with the practical seminar, amounts to 1.5.

film applications, cognitive science research and therapeutic applications, technical applications, and computational ethology should be introduced. The last two examples are important for drawing distinction between *plausibility* in the sense of a natural science, e.g. physics or ethology, and *believability*. The NetLogo examples help with clarifying the notion of *emergence*. The notion of *autonomy* has to be stressed. The survey of main issues, such as navigation, action selection, emotional modelling, story generation, helps to outline the content of the rest of the course.

Table 1. The curriculum of the course

Block	**Topic**	**Units**
	(1) Introduction	1
Concrete models of action-selection.	(2) Reactive planning: hierarchical if then rules with priorities, hierarchical state machines, fuzzy rules, probabilistic state machines.	2
	(3) Neural networks, reinforcement learning, evolutionary algorithms.	2
	(4) Behavioural science models: "Psycho-hydraulic" model of K. Lorenz, Tyrrell's free-flow hierarchy, classical and operant conditioning.	1
	(5) Path-finding, steering, abstract terrain representations.	1
Conceptual notions, architectures, representation.	(6) Architectures: symbolism vs. connectionism, layered architectures, notion of deliberation. BDI, planning.	1
	(7) Multi-agent systems introduction: types of agents, communication.	1
	(8) Representation: logic, deictic repres., Gibson's affordances, smart objects.	0.5
Extras, broader context	(9) Artificial emotions.	1
	(10) Storytelling: emergent narrative, plots representation, level-of detail AI.	1
	(11) Perception and memory.	0.5
	(12) Unified theories of cognition.	1

Concrete models of action selection. This block helps students to see that some action selection issues can be promptly solved and how. Students appreciate concrete mechanisms and concrete solutions. Our experience is that they tend to regard these special-purpose mechanisms as abstract and general. While it is useful to give concrete counter-examples in which these methods fail, it is ineffective to give students any abstract, background knowledge, at this moment (more on this in Sec. 4). In fact, these concrete mechanisms will serve as a base upon which the students will represent in their minds more advanced and/or general knowledge.

An excellent material for introducing neural networks and evolutionary mechanism is the control architecture of animals from the computer game Creatures [10]. First, it helps to explain both topics. Second, the neural networks used in Creatures are more biologically plausible than typical artificial neural networks, which helps with introducing the distinction between biological and artificial neurons. It is also useful to introduce the action selection problem from the perspective of behavioural sciences. This demonstrates rich cross-fertilisation between disciplines, and reminds students of the believability—plausibility distinction. The excellent material is [18].

The lecture on path-finding helps with two additional things. First, the notion of representations can be stressed (with the help of terrain representations). Second, when drawing the distinction between low-level steering and high-level path-planning, it will become apparent that it can be helpful to think about IVAs at different levels of analysis - a bridge to the next part of the course.

Concept, architectures. Technically, so far, the students have been taught special-purpose solutions. Now, their knowledge should be unified within more abstract, conceptual frameworks. Notions of goals and intentions, basically Belief-Desire-Intention (BDI), should be introduced, reactive approach confronted with planning, symbolism with connectionism. A welcomed example demonstrating that BDI can be really implemented in various ways is the hybrid control architecture of Black & White creatures [9]. To frame the notion of IVAs in a broader context and to discuss the topic of communication, it is vital to introduce multi-agent systems.

Extras. Every year, many students are surprised that when developing an IVA, despite it is intended to be believable but not psychologically plausible, it might be a good idea to look at what psychologists say about real humans. For example, the students are amazed that mechanisms of attention can really help with limited computational resources. In sum, these lectures help the students to understand that a "mind" of an IVA comprises not only an action selection mechanism, but also various additional "circuits". Our experience is that it is better to give these lectures after the students are already familiar with IVAs architectures and action selection.

The high-point of the course is storytelling for many reasons. Most importantly, students start to understand that IVAs cannot exist without virtual worlds and vice versa, in other words that IVAs and virtual worlds constitute an intertwined couple, a sort of *marionette theatre* in a service of a user. Through this metaphor and the notion of *story manager* students can understand how to *systematically* abandon the concept of strong autonomy. These notions also help with introducing the technique of level-of-detail AI [4]. Additionally, the notion of *emergent narrative* reminds students of the idea of emergence and the notion pof *story construction* reminds them of planning.

3 The Seminar and the Toolkit Pogamut 2

The aim of the practical seminar is to give students an opportunity to exploit theoretical skills acquired during the course, i.e. to implement own IVA on a chosen topic, such as state machines or artificial emotions. As the course covers many topics, we looked for a toolkit having a potential to provide a base for as many topics as possible while not being too complex and becoming a burden to students. Such a solution wasn't available in previous years [12]. Therefore we have created the toolkit Pogamut 2.

Features of Pogamut 2. Pogamut 2 is designated for educational and research projects concerning modelling of behaviour of IVAs. It brings together a 3D simulator, an agent library, and a development environment (IDE)., The toolkit uses the environment of Unreal Tournament 2004 (UT04), an action game as the 3D simulator. The UT04 offers many maps of complex 3D words as well as human-like models of agents suitable for games and storytelling. The agent library, written in Java, allows students to quickly create new agents inside the UT04. It also enables utilisation of third-party libraries and planners (such as neural networks or POSH [5], which is already integrated). Rapid exploration is further facilitated by the IDE, which is based on the popular NetBeans platform [17]. It helps newcomers to get quickly comfortable with the process of agent development, namely:

- It offers agent project management tool along with several example projects.
- It has two types of view – Simple and Advanced. The simple view is used in the first lesson, allowing for quickly setting up and running example projects to motivate students. The latter one is used during the agent development.
- It gives contextual help during coding of the agent eliminating the necessity for consulting the manual frequently.

Additionally, the toolkit possesses many advanced features, such as a debugger, a variable manager, a simplified map of the environment facilitating orientation of the user, a support for GRID computing, etc. [12].

Table 2. Curriculum of the seminar. The two stages are separated by double line.

(1) Introduction, example projects	The toolkit and the sample IVAs were introduced. Students interactively explored the examples, meaning they made simple changes of the IVAs behaviour.
(2) Implementation of a simple IVA	Students were taught the agent library basics. Finally, they were able to implement their own agent, which was able to follow the player inside the UT04. Students were delighted that such behaviour can be achieved with merely 4 lines of code.
(3) Navigation	The topics: How to navigate in UT04 using way-points. How to detect obstacles using ray-casting. Finally, students were able to create an agent running around the environment, collecting weapons along the way, and performing dodges.
(4) Reactive planner POSH	Students learned how to code POSH-behaviour. Finally, they were able to reproduce several sample agents using POSH-rules.
(5) Items, weapons and weapon handling.	Students learned how to handle different weapons. Finally they were able to produce agent with basic combat capabilities. (Note, that the problem of weapon selection and usage is an issue of its own studied in gaming AI.)
(6) Practise	This lesson was reserved for student's projects.

Seminar topics. The seminar comprises six lessons at computers (Tab. 2). It capitalises on a two-stages teaching process: rapid start with simple things and slower exploration of more advanced stuff. Note, that the topics of the basic stage can be actually covered by NetLogo, contrary to the advanced stage.

Since the toolkit was introduced for the first time, we decided to keep its utilisation as simple as possible. Actually, we exploited the fact that UT04 is the underlying platform and focused mainly on gaming AI issues. Since we believe that these present only a part of what should be explored by students in such a seminar, we are presently adding a generic emotional module and a story manager to the Pogamut 2 library. These are intended to become the topics of the 6th lecture in the next year.

Concerning students' assessment, each student had to develop his/her own IVA. The default goal was to create an IVA for a death-match tournament. Nevertheless, many students came up with their own tasks. Ideas ranged from cooperating agents that would support themselves in the death-match over explorer agents building their own environmental representation to evolution of a neural network controller that would use ray-casting alone to navigate through the environment. As students were enthusiastic about their own ideas, it worked best to let them explore those ideas on their own even if their goal has apparently been too complex.

Evaluation. In order to evaluate the seminar and our approach, a pre- and post-questionnaires were given to 25 of our students (M=23, F=2), corresponding to 55% of total enrolled students. For brevity, only the main findings are highlighted.

The students were required to provide free report on their previous background, their motivation, and whether they think the course had helped them to understand the problematics. Table 3 summarises these data. It is apparent that the course was attended by students with various levels of previous knowledge both in AI and programming. Nevertheless, we found no correlation between this stratification and other reported data. Most students attended the course out of curiosity. This is an interesting finding for it suggests that the subject of IVAs/IS is still new and there is only limited previous common knowledge about it among students. Also it seems that not many students believe they can earn money by studying this topic (but can they?). Finally, the table shows that the students regarded the seminar beneficial.

In order to evaluate the platform, the students had to mark its features from 1 (worst) to 5 (best). The scores are high, except for the documentation, which we have started to improve.

Table 3. Summary of the main findings from the questionnaires

Question:	Category:	Nr.
Previous background in AI / programming?	Few (AI: 0 – 2 courses [normalised] / Prog.: 2 years studying)	15 / 8
	Some (AI: 3 – 5 courses / Prog.: 3 – 4 years studying, an internship)	5 / 4
	A lot (AI: 6 – 12 courses / Prog.: more years studying, at least several months of IT employment)	5 / 13
Why did you choose this seminar? (more answers are allowed)	Out of curiosity	21
	Fond of UT04	4
	I want to study Gaming AI / Entertainment applications	3
	Sounds like a good practical seminar	2
	To see AI in practise	1
Did the seminar help you to deepen your knowledge about the action selection mechanisms of IVAs?	Yes, it helped.	9
	It did not deepen the theoretical insight but it gave examples of practical problems the ASM must address.	11
	No, it didn't.	1
	Can't say.	4

Table 4. Evaluation of the features of the platform and comprehensiveness of the course

Easy to learn	3,82
Documentation	2,45
GUI and coding user-friendliness	3,91
NetBeans environment	3,64
Comprehensiveness of the theoretical course	3,82
Comprehensiveness of the practical seminar	4

Finally, we asked the question "Do you find Pogamut 2 toolkit suitable for practical needs of the course or should we look for something else?" Students assessed

Pogamut 2 as quite appropriate for the demonstration of practical implementation of IVAs behaviour. They really appreciated the fact that simple things can be expressed with minimum code, which encouraged them to experiment with the toolkit. Some students appreciated the UT2004 game and even suggested to connect Pogamut 2 with different environments as well. Others, especially the girls, demanded a non-violent environment. Examples of concrete reports follow:

"Pogamut 2 is the right tool for the course."
"Simple things in simple way, complex possible."
"It would be cool to connect Pogamut 2 with Operation Flashpoint to provide realistic maps and to try to implement some complex behaviour according to military books."
"Some realistic environment allowing better storytelling application would be nice."

We believe the emotional module and the story manager will help with the last point. We also look for another environment to connect the toolkit with, the Second Life [14] is one of the possibilities we are considering presently.

Case study. Adoption of the course for high-school students. An important feature of every course is whether it can be transferred to another learning context. In general, the issue of scaling of our course is discussed in Sec. 4. To back up this debate, it should be said that we have already transferred the course. Its theoretical part was adopted for a summer school (four 90 minutes lessons). The whole course was also scaled for high-school students of computer science (16, 17 yrs old, 2nd grade). This class lasted 90 minutes of theory and 90 minutes of practice with our tool. Despite the limited time, the students were given a compact introduction into the subject. The issues of path-finding, steering and emotions were discussed, and the links towards ethology were highlighted with the help of NetLogo. Navigation issues were solved with the usage of reactive planners and simple neural networks controllers thus giving concrete solutions for concrete issues. Finally, a brief introduction into Pogamut 2 was made, during which the students had an opportunity to implement a simple IVAs behaviour by means of a few if-then rules. The questionnaires were administered; the results are similar to the results of university students (Tab. 3, 4).

4 General Discussion

Many courses on computer games development and multi-agents systems capitalises on the same teaching methodology as our course. The objective of this section is to make this methodology explicit. This will help us to discuss structure of such courses in general and also highlight several findings concerning our course that have not been discussed yet.

Let us perceive the course as a learning process. Now, we can say that we engaged our students in such a process, in which they could systematically explore the field of study, incrementally constructing their personal representations of knowledge, that is, a) objectives of the fields of IVAs/IS: motivations, goals, applications, b) technical issues stemming from these objectives, c) models/algorithms/mechanisms of IVAs and virtual environments that cope with these issues, i.e. solutions (Fig. 2). Students were not only to mentally represent this knowledge, but also to develop new mental processes to operate with this knowledge. Their general ability to think about algorithms and models from different perspectives was intended to increase. Importantly, this ability is quite general, applicable *outside* the field of IVAs.

What were the stages of this learning process? The retrospective analysis suggests that they were three, roughly corresponding to the blocks of the course, helping the students to build their knowledge *incrementally* (Fig. 3). First, the students were confronted with different technical issues of the IVAs and several "middle-level abstraction" mechanisms that can solve these issues, such as prioritised if-then-rules, finite state machines, reinforcement learning, steering rules etc. Second, they were to understand that these models are embedded within general, let us say conceptual or "black-box", frameworks, such as rule-based systems, first-order logic, or BDI. They were required to "zoom through" different levels of description and be able to mentally substitute one model for another. For example, they had to "see" that neural networks controlling animals from the game Creatures [10] are conceptually similar to fuzzy if-then rules. As computer science students, they also had to acquire the ability to bring their mental models to "life" in the digital media, i.e. to be able to "zoom in" at the code level of description. At the same time, they had to start to understand that the whole point of this enterprise is not the issues, but the objectives. The third stage even strengthened this notion by means of providing the students with additional objectives, issues and solutions, and links with other disciplines to help them to connect their new knowledge with what they have already known and with what they might learn in the future. To help the students to create this "semantic network" (Fig. 3(3)) was the primary objective of the course.

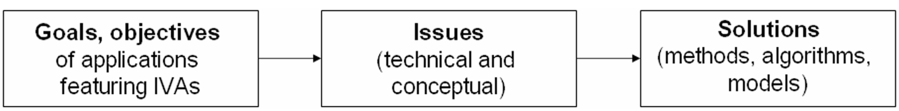

Fig. 2. The conceptualisation of the causalities behind the course

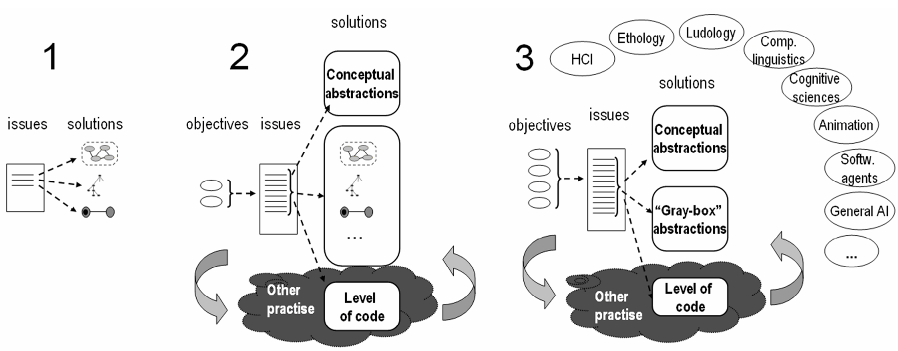

Fig. 3. Schematic depiction of knowledge students are intended to represent within their minds during the three stages of the learning process

This would not have been possible, however, without providing *feedback*, through which the students could test their ideas. This feedback was given via two mechanisms: through the practical seminar and by demonstrating various applications during the course, which the students should have further explored during self-stydying (e.g. Façade, NetLogo).

This teaching framework is actually situated within the *constructivist* learning approach [15]. The structure of knowledge the students should build can be conceived as a kind of mental models of Johnson-Laird [11]. It is natural to suggest a metaphor of a *non-linear game* for courses capitalising on this framework. The non-linear aspect stresses that students should be allowed to approach the field from *different perspectives*, visiting the topics in order and at speed given by their individual preferences and background knowledge. The *gaming* aspect stresses that the process of exploration should be *interactive*.

Now, what does this tell us? We will start with concrete points and proceed to more general ones.

1. Bottom-up approach. The described teaching process is a bottom-up strategy: to start with a concrete and to end up with an abstract. One year we swaped the first two stages depicted in Fig. 3 and the interviews showed that this was not a good idea: the students forgot or did not grasp the abstract without the concrete at all.
2. Transfer of our course to another learning context. Basically, the course can be scaled along two axes: (A) background knowledge of the students and other courses available, (B) available time. Along both lines, the following key points should be preserved: the availability of feedback and incremental knowledge development.
   (A) We have three suggestions. (i) To change the feedback according to previous students' knowledge. Practise of our students is focused on programming, but for non-IT students, the advanced programming block (see Sec. 3) can be replaced e.g. by developing Machinimas [13] or building virtual worlds in Unreal Editor. (ii) Links to other science disciplines should anticipate what knowledge students are most likely to need in the future. (iii) While some concrete topics can be suppressed in Stage 1, e.g. neural networks, some extras can be added in Stage 3, e.g. agents' social interaction.
   (B) Again, we have three suggestions. (i) In Stage 2, building knowledge of conceptual models can be suppressed. Our experience is that without enough time available, the students are not able to grasp the abstract knowledge anyway. (ii) In Stage 1 and 3, many concrete issues and solutions can be omitted (e.g. probabilistic state machines, Tyrrell's free-flow hierarchy, perception and memory). Nevertheless, it is important to preserve the line "concrete issues → concrete solutions" during the Stage 1 at least to some extent to boost up students curiosity and engagement. (iii) Discussions about connections with other sciences can be shortened, but again, at least some links should be provided.

   Along these lines, the course was actually scaled for a summer school and a high-school course, as detailed in Sec. 3. Many points of this "transfer schema" are quite general and can work for similar courses as well.
3. Availability of materials. Today, predominantly, research prototypes, applications, or at least videos, are *not* published on the internet by their authors. As these materials are essential for "providing feedback", this trend should be changed. This "call for teaching resources" is even amplified when considering

correlations between availability of materials and community growth (think of open-source communities or communities of gamers emerging around partially-opened games such as Unreal Tournament or Counter Strike).

4. Another open issue is a schoolbook that would cover the topics of the fields of IVAs/IS. Though there are collections of papers, e.g. [16], these cannot compensate for a coherent learning text. Neither can this be done by general AI, gaming AI, software agents or computer graphics textbooks. We argue that interactive applications featuring IVAs is a subject of its own and requires its own schoolbook covering basics ranging from animation through action selection and virtual storytelling to social and behavioural sciences, giving the reader a consistent and a broad view on the field.
5. Longitudinal evaluation. Currently, there are only very limited data available on how many students of interactive technologies actually use their knowledge in practise. Similarly, it is not yet known how many students really keep in their minds the knowledge they gained during curricular learning: longitudinal studies are missing. E.g. does the proposed 3-staged teaching process increase retention of knowledge over long periods? How many students earn money doing interactive applications comparing to let's say databases? We have to start to ask such questions.

5 Conclusion and Future Work

This paper started with drawing attention to education of new members of IVAs and IS community, an issue that has not been much studied yet. From this perspective, the contribution of this text is twofold: first, the curricula of a university course on *Modelling Behaviour of Human and Animal-like Agents* has been overviewed and the main lessons learned (from the educational perspective) detailed. Second, the teaching methodology behind this course and similar courses has been verbalised to facilitate thinking about such courses and about their transfer between different learning contexts. A small evaluation study has suggested that the course is effective and a case-study demonstrated that the transfer of the course is possible. Nevertheless, a rigor longitudinal evaluation is missing.

The paper also introduced the toolkit Pogamut 2, a training vehicle in which students can practice development of IVAs using the environment of Unreal Tournament 2004. Actually, the toolkit can be used for research purposes as well.

Presently, the Pogamut 2 is being augmented with a generic emotional module and a story manager, two components that we believe will contribute most to the learning experience of students. We are also considering connecting Pogamut 2 with another environment, e.g. Second Life [14].

In general, the authors hope that this paper will help to start a broader discussion on educational issues related to the subject of interactive storytelling and intelligent virtual agents, including such topics as learning materials, a "required minimum" for a student of this field, or facilitation of team work among students with different backgrounds (e.g. programmers and artists).

The lecture materials, the extended version of this paper detailing the curricula and the literature used, and the Pogamut 2 are freely available for download [3, 12].

Acknowledgments. This work was partially supported by the Program "Information Society" under project 1ET100300517, by the Ministry of Education of the Czech

Republic (Res. Project MSM0021620838), and by two student grants GA UK No. 351/2006/A-INF/MFF, and 1053/2007/A-INF/MFF.

References

1. Adobbati, R., Marshall, A.N., Scholer, A., Tejada, S.: Gamebots: A 3d virtual world test-bed for multi-agent research. In: Proc. 2nd Int. Workshop on Infrastructure for Agents, MAS, and Scalable MAS (2001) (26.6.2008), http://gamebots.planetunreal.gamespy.com/
2. Aylett, R.: Slides for the course on Artificial Intelligence, Heriot-Watt University, Edinburgh (26.6.2008), http://www.macs.hw.ac.uk/modules/F23AD3/AIcourse.html
3. Brom, C.: Supporting material to this paper, extended version of this paper, slides for the course on Modelling Behaviour of Human and Animal-like Agents. Charles University in Prague (26.6.2008), http://artemis.ms.mff.cuni.cz/main/tiki-index.php?page=Teaching
4. Brom, C., Šerý, O., Poch, T.: Simulation Level-of-detail for Virtual Humans. In: Pelachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 1–14. Springer, Heidelberg (2007)
5. Bryson, J.J.: Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents. PhD thesis, MIT (2001)
6. Dignum, F., Westra, J.: Slides for the course on Games and Agents, University Utrecht (26.6.2008), http://www.cs.uu.nl/docs/vakken/gag/
7. Epic Megagames: Unreal Tournament (2004) (26.6.2008), http://www.unrealtournament.com/
8. de Melo, C., Prada, R., Raimundo, G., Pardal, J.P., Pinto, H.S., Paiva, A.: Mainstream Games in the Multi-agent Classroom. In: Proc. Int. Conf. on Intelligent Agent Technology (2006) (26.6.2008), https://fenix.ist.utl.pt/disciplinas/aasm
9. Evans, R.: Varieties of Learning. In: Rabin, S. (ed.) AI Game Programming Wisdom I, pp. 567–579. Charles River Media, Inc., Hingham (2002)
10. Grand, S., Cliff, D., Malhotra, A.: Creatures: Artificial life autonomous software-agents for home entertainment. In: Proc. 1st Int. Conf. Auton. Agents, pp. 22–29. ACM press, New York (1997)
11. Johnson-Laird, P.N.: Mental Models. In: Posner (ed.) Foundations of Cognitive Science, ch. 12. MIT Press, Cambridge (1993)
12. Kadlec, R., Gemrot, J., Burkert, O., Bída, M., Havlíček, J., Brom, C.: Pogamut 2 – a platform for fast development of virtual agents' behaviour. In: Proc. CGAMES 2007, La Rochelle, France (2007) (26.6.2008), http://artemis.ms.mff.cuni.cz/pogamut
13. Kirschner, F.: Movie Sand BOX (26.6.2008), http://www.moviesandbox.net/
14. Linden Research, Inc.: Second Life (26.6.2008), http://secondlife.com/
15. Pappert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York (1980)
16. Prendinger, H., Ishizuka, M. (eds.): Life-Like Characters: Tools, Affective Functions, and Applications. Springer, Heidelberg (2004)
17. Sun Microsystems, Inc.: Netbeans (26.6.2008), http://www.netbeans.org
18. Tyrrell, T.: Computational Mechanisms for Action Selection. PhD thesis, Centre for Cognitive Science, University of Edinburgh (1993)
19. Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University (1999) (26.6.2008), http://ccl.northwestern.edu/netlogo/