International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

IVE: Virtual Humans' AI Prototyping Toolkit

Cyril Brom, Zuzana Vlčková

***Abstract*—** IVE toolkit has been created for facilitating research, education and development in the field of virtual storytelling and computer games. Primarily, the toolkit is intended for modelling action selection mechanisms of virtual humans, investigating level-of-detail AI techniques for large virtual environments, and for exploring joint behaviour and role-passing technique (Sec. V). Additionally, the toolkit can be used as an AI middleware without any changes. The main facility of IVE is that it serves for prototyping both the AI and virtual worlds themselves. The purpose of this paper is to describe IVE's features in general and to present our current work - including an educational game - on this platform.

***Keywords*—** AI middleware, simulation, virtual world.

I. Introduction

In the last years, the field of virtual storytelling and computer games has developed rapidly. We are interested in particular research and development in this area - we focused on how to control virtual humans in large environments like role-playing game worlds. Our developmental activities include an educational storytelling game and a simulator of a society. This game was also our original motivation behind our past [2] and current research.

Specifically, by *virtual human* (or *actor*) we mean a piece of code that simulates a human-like behaviour. Virtual human is equipped with a virtual body and carries out more complex tasks than just walking, object grasping or chatting in an ELIZA-like way. *Large world* represents a spacious artificial environment – not a single room, but a village or a region. In this paper, we will focus on AI issues concerning virtual humans, not on computer graphics.

From the gaming AI point of view, creating the behaviour of a single actor is now basically an engineering issue provided that he has no extraordinary requirements. A reactive planning technique and the A* algorithm are acceptable solutions for standard issues. However, there are at least two new problems stemming from large environments – environments are so large that they cannot be simulated on a single PC due to enormous costs of computation, and they are inhabited with tens or hundreds of virtual actors, which means that we must handle the design complexity of their behaviour.

Hence, after we had successfully prototyped the behaviour of several individual actors in our former tool [2], we focused on these "large world" issues. We started our development by working on theoretical solutions. Since our former tool could not cope with the large worlds, we tried to find a new toolkit that would facilitate our creation of a large world case-study. However, we couldn't find any. We tried Jade [22], an agent development platform, but found it too slow for our purposes. We tried BDI platforms (such as Jam [14]) and Soar [17], but realized that they are mostly stand-alone languages and not toolkits supporting virtual world developments. Then we ventured on a computer game engine, such as UT [9], but since the game environment architecture was always firm, these open engines would be too restrictive for our purposes.

Finally, we had to create our own toolkit, IVE - Intelligent Virtual Environment [15] (Fig. 1). IVE has been developed in Java and its latest version (1.1) was released in April 2006. The main feature of IVE is that - unlike other tools - it presents a platform for prototyping not only behaviour of virtual actors, but also of virtual worlds.

In this paper, we first devote to the speed and complexity problems, and then describe IVE in general, discussing its potential as a developmental, educational and AI prototyping platform. Then, its features coping with "large world" issues will be highlighted. Finally, we introduce our current work exploiting IVE and discuss the toolkit's restrictions.

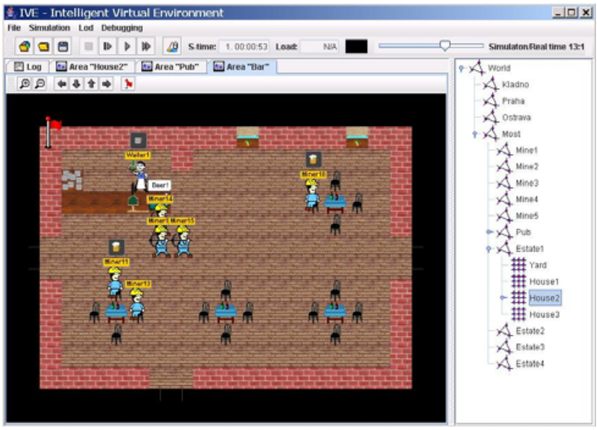

Fig. 1. A screenshot from the test-scenario. On the left, there is a restaurant with seven actors. On the right side, there is a tree of a world's locations

II. Problems and Solutions in Detail

We suggested that the developer of a virtual reality application which features a large world inhabited by tens of actors, would face two problems – the problem with simulation speed and the problem with handling the behavioural design complexity.

Our solution to the speed problem takes advantage of a level of detail technique (LOD - Sec. VI) at virtual space layer and of the actors' AI (contrary to its common use in computer graphics). The LOD technique allows us to simulate in detail only the parts of the world which are in the centre of user's attention. To utilise the LOD technique, a hierarchical

Cyril Brom is with Charles University, Faculty of Mathematics and Physics, Malostranské nám. 2/25, Prague, Czech Republic (e-mail: brom@ksvi.mff.cuni.cz)

Zuzana Vlčková is with Charles University, Faculty of Mathematics and Physics, Malostranské nám. 2/25, Prague, Czech Republic (e-mail: zuzana.vlckova@gmail.com)

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

representation of the virtual space and a hierarchical kind of action selection mechanism must be used.

We have two solutions for complexity problems. First, an architecture for virtual worlds allows us to represent the actors' behaviour in a way that facilitates us to control actors both autonomously and in the centralised way (i.e., more actors can be driven by one control algorithm), enhancing the level of their coordination. Secondly, this architecture allows loading new actions and objects as plug-ins, but without using any machine-learning method at the side of actors' "minds".

Both the LOD technique and the complexity solutions stem from Bryson's hierarchical reactive planning [8], Bratman's theory of practical reasoning [3] and Gibson's perception theory of affordances [11]. These solutions will be described in sections IV - V.

III. IVE in General

IVE is a virtual environment simulator. Users are allowed to simulate their own virtual world with own actors - they can define objects, actions, spaces and AI for the actors. The set of actions and objects is easily extensible. After a simulation is started, user can interact with it, speed it up, slow it down and save/load it anytime. IVE also includes some limited debugging tools, which serve for monitoring properties of actors' and objects' properties (e.g. actor's "mind").

Level of abstraction. IVE works with environments which are discrete in space and pseudo-continuous in time. It means that the space is divided into square tiles or atomic places organised in a graph structure at the ground level of abstraction and that actors can perform atomic actions like "step" or "take an object", which can last any amount of time. Actors can interact among themselves and with objects.

Case-study world. IVE was tested with a scenario comprising about 100 virtual humans acting in four virtual villages (most of them were miners); each with a restaurant, 5 mines, and 12 houses. If all the world is simulated (without LOD), the simulation is still real-time (it takes about 5-8% of processor load at Pentium 4,3 GHz). With just one restaurant simulated in full detail and the rest of the world in lower detail, the load decreases below 1% of processor usage (GUI excluded).

Hierarchical space representation. The space is represented hierarchically: tiles are grouped into areas, areas into larger areas etc. Each layer forms another level of complexity. This representation is used because of the LOD technique. The number of layers depends on the virtual environment, in our case-study world, we use 5 layers.

Action selection mechanism. Let genius be a component encapsulating the action selection mechanism. Every actor has its own so-called *basic genius*. These genii utilise a hierarchical reactive planning that is inspired by Bryson's POSH [8] and that also resembles classical BDI implementations (e.g., [14]). Every genius has a set of goals. It adopts one goal as its present-directed towards-a-goal intention and finds an activity that accomplishes the goal. The genius commits itself to this activity (i.e., formulates a towards-a-means intention). Further, the activity can break down to sub-goals that can be consequently adopted as towards-a-goal sub-intentions. This adopting continues until an atomic action is performed. Then, independently of success or failure, the decision mechanism is started again. The activities as well as goals are predefined by the world designer, and searched from a "behavioural library" in the process. Fig. 2 depicts the hierarchy of committed towards-a-goal and towards-a-means intentions for a genius that drives a virtual miner.

Technically, the choices between goals and activities are made according to reactive rules with priorities. However, the advantage of IVE is that these rules can be easily replaced by another hierarchical mechanism. For example, we are now augmenting IVE with HTN planning [11]. The evaluation of rules' expressions is performed in a lazy and Rete-like way [10], which means that we cache results from previous computations, and evaluate the same sub-expressions only once. This technique speeds up the rules evaluation rapidly. When the case-study world is simulated in a full detail, the actors are driven by maximum 5 000 rules together.

Notable exception to this mechanism is path-finding, which is performed by hierarchical A* algorithm.

The IVE contribution is that the basic mechanism described here is extended to cope with the complexity problem, as will be detailed in Sec. IV and Sec. V.

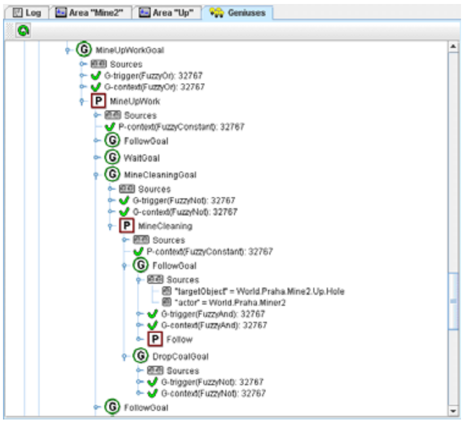

Fig. 2. The detail of a miner's "mind" (in the mine at the moment). (G) denotes goals, [P] activities.

Sensory perceptions. Every genius perceives its surrounding through a set of sensors, which can be defined by a world designer, and manages its own "short-term memory" consisting of actual percepts. These percepts are actually some index-functional (or deictic) entities [1]. Sensors can be used both in an active way (i.e., genius initiated) and in a passive way (i.e., driven by environmental events). In our case-study world, we have implemented only a simple eyeSensor, which perceives everything from the area the actor is located in.

GUI. Since IVE has been intended for AI issues, its GUI is rather simple. However, the design of the GUI - world interface allows an easy replacement of a particular GUI.

Interaction. In our case-study scenario, user does not control an avatar in the environment, but she can influence the state of the world directly (Fig. 3). However, the avatar can be developed easily if needed.

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

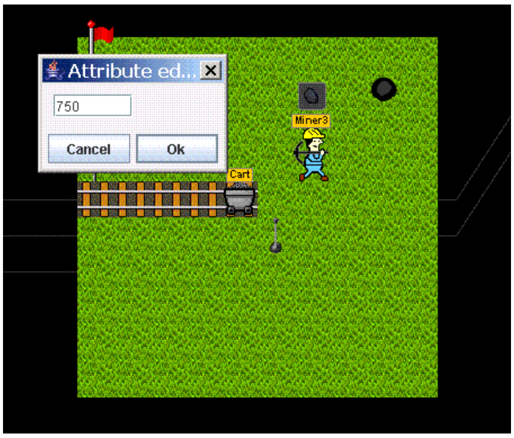

Fig. 3. The miner points to the hole to throw away the stone he holds. Meanwhile, a user is adding 750 new stones to the cart.

How can we use it? There are at least three ways how IVE can be used - for case-studies prototyping, for education and as a middleware.

First, since we are allowed to create a new environment and actors in IVE, the toolkit can be used as a platform for prototyping almost any kind of virtual world with a discrete space. IVE has been particularly intended for using large environments, where we can benefit from the LOD technique. User simply loads the configuration files, starts the simulation and he can monitor actors behaviour, interact with them, explore the log files etc. She can also change the mechanism of action selection and sensing easily, provided that she keeps its hierarchical nature that is needed for LOD.

Second, since it has a specious GUI and some debugging tools, it can be used during a lecture for a live demonstration of certain techniques. In particular, we use IVE in this way for demonstrations of a reactive planning, LOD and some issues concerning representation of virtual environments.

Finally, since the code is open and it is possible to modify the IVE core, the toolkit can be used as a middleware for applications featuring large environments based on discrete space worlds. In this fashion, we use IVE for our educational game and the for society simulation.

Probably the main limitation of IVE is that there is no user-friendly editor. A world model and actors' behaviour must be currently specified in external xml and Java files. Our experience is that (even though we do not have an editor), IVE can be used by an undergraduate IT student. The editor is one of our current works in the process.

IVE was programmed in Java. It can be used perfectly well as a middleware, but in spite of LOD only for those applications that are not time-critical (in other worlds, a commercial game should be developed e.g. in C++). We also remark that IVE is focused on large environments. If one wants to develop just a simple application with one or two actors, she may find that to use IVE for this purpose is like using a sledgehammer to crack a walnut.

IV. J. J. Gibson and Smart Activities

In this section, we discuss how IVE challenges the first part of the complexity problem – extensions managing. In fact, we will describe a refinement of the action selection mechanism from Sec. III. The solution stems from the affordance theory.

Affordances were introduced by J. Gibson, a perceptual psychologist, in so-called ecological theory of perception. He claimed that we tend to perceive what the environment offers us rather than physical properties of objects. The environmental offers were called affordances. "...the affordances of the environment are what it offers the animal, what it provides or furnishes [12]".

In our effort to develop an extensible architecture offering LOD AI, we have to refine a Gibson's theory. As mentioned above, a typical virtual human in IVE is driven by its own genius, but still it is not fully autonomous. That is because genii are given information about how to drive the actors. How does it work? Up to some level, IVE approach resembles the concept of smart objects [16] used often in computer graphics and games, e.g. [19]. *Smart objects* are entities providing their detailed functionality description, possible interactions as well as the behaviour of a potential interacting actor. Smart objects encapsulate a script that has to be executed by an interacting actor. Using smart objects, the world can be described in terms of a "purpose-oriented" language, and therefore an object can be loaded into the simulation as a plug-in and an actor can interact with it without any learning.

However, the original smart objects have some limitations we were able to overcome. Most notably, we have pushed the concept towards *smart activities*, which are abstract entities that describe the interaction among more actors and more objects (think of the following situation: which object shall offer a) closing a pen: the pen, or the cap? b) two actors dancing in a couple?). Smart activities are localized in the environment and genii can perceive them. More specifically, since we linked activities with genii's intentions, a genius has perceived only activities that could satisfy its present-directed towards-a-goal intentions.

Moreover, each activity contains a *smart suitability* - a function that computes how convenient would it be to execute the activity for given actor in the current context. (Apparently, a human prefers to water with a watering can to a bucket, even if the bucket is suitable too. Moreover, a child would prefer watering with a smaller can than an adult. Additionally, the garden might be also hosed. Decision among the possible alternatives for every particular actor depends on the solution suitability.). Finally, not atomic smart activities offer sub-intentions and smart advices to genii. *Smart advices* help genii to decide which sub-intention should be adopted in pursuing the current (non-atomic) activity.

In other words, thinking in terms of our action selection mechanism (Sec. III), the genius has a set of top-level goals, but when it wants to accomplish them, it must "look around" the environment for a set of smart activities for finalizing the goal. Based on the suitabilities, the environment even tells the genius how each particular activity is proper in the current situation. The genius directly perceives suggestions what to

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

do. Here we see the Gibson's concept in its most clear form. Further, if the activity is not atomic, the genius was to adopt sub-goals offered by the activity and according to the given advice, commit itself to those sub-goals (Fig. 4). In yet other words, rules and priorities for making decision are represented within the environment, not in the actors' "heads". Note that in IVE, the genius is also allowed not to follow advices and suitabilities, hence the framework allows actors to be truly autonomous.

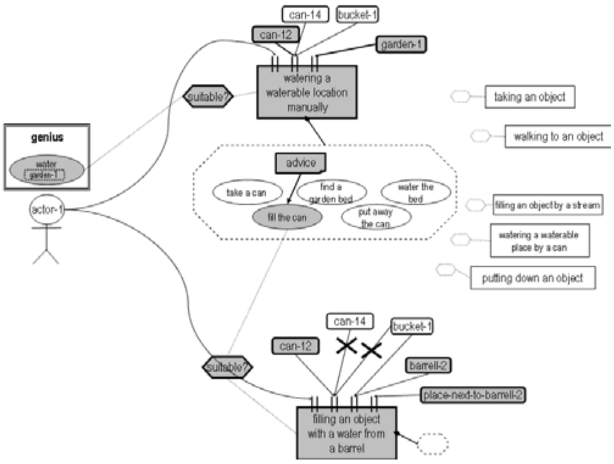

Fig. 4. Representation of behaviour. The actor-1 has a "water garden-1" intention, has committed itself to "watering garden-1 manually with the can-12", has "fill the can" sub-intention, and choose the sub-activity "filling the can-12 from the barrell-2 at the place-next-to-barrell-2". Other sub-activities for watering the garden, which are also depicted, can be perceived by the actor's genius.

This concept actually solves one part of the complexity problem – we can simply manage extensions both during the development and after the release (Fig. 4). Actors can be though of as being navigated by highly modular intelligent environments (hence the acronym IVE). Our representation is detailed in [6].

V. Role-Passing

Central control of actors is common in many applications. e.g. in strategic games. It is also possible to trace role-passing as layering of new reactive plans on actor's plans. Role can also contain new character's qualities and rules for their changes. For example, in "bar-guest" role, we can specify that after passing the role, a level of "thirst" property is watched [23].

In this apparatus, we can switch roles (switch on grounds of norms [24]). Technically, norms are rules or evaluating functions determining when switch roles. In IVE, it is an analogy of suitability.

IVE supports combinations of central and autonomous control in the following way: An activity (perceived by an actor's own genius), like behaviour in a pub, can be linked to a so-called *genius specialist* - the bar genius for example. The actor's genius is then allowed to pass its actor to the specialist to perform the chosen activity. Hereby, the specialist can carry out some special-purpose reasoning, e.g. how to assign the actors to tables not allowing two actors to collide. The specialist can pass on an actor further to another specialist. This process is called *role-passing*.

Finally, the specialists and the basic genius control the same actor together. Each of them manages its own set of goals, while at each particular moment genius that holds the goal with the highest priority, drives the actor. If a goal of another genius becomes the highest prioritised one, this new genius gently overtakes the control (gently means that the interrupting genius waits until the interrupted genius performs some clean up). For example, in Fig. 5, all actors in the row at the bar are controlled by three genii - the basic one, the bar specialist; and the queue specialist and actors sitting at the tables are managed only by the first two genii. If a drinking actor wants to go to the toilet, the toilet goal (managed by the basic genius, contrary to drinking) interrupts the drinking, but the restaurant-specialist is allowed to let the actor to put the beer on the table first (and not to take it to the toilet).

We think that this combination of semi-autonomous and centralised controlling is vital for any application featuring more virtual humans. For example, Trip and Grace from Façade [18] are controlled in this way up to some level as well as squads in the game F.E.A.R [19].

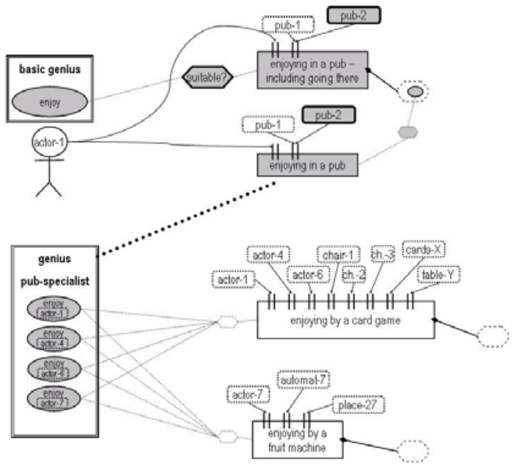

Fig. 5. Role-passing in IVE. Actor-1 is delegated by its basic genius to the genius pub-specialist, who can manage several actors together and choose activities for them (e.g. card playing)

VI. Level of Detail for AI and Space

There are many works describing LOD in computer graphics, but just a few are related to LOD for virtual humans and virtual space controlling. Probably only areas, where ad-hoc application of this technique are used, are computer games and experimental large worlds simulation in virtual reality.

The general idea behind LOD is not to compute such details that a user cannot see or that are otherwise unimportant in the given instant. To describe LOD in IVE, we use a membrane metaphor - imagine an elastic membrane cutting through the space hierarchy (see Fig. 6, 7), which is being reshaped in the course of the simulation. Areas (or atomic places) at this membrane exist as abstract points, whereas areas below the membrane do not exist in the instant of observing. The membrane also determines the complexity of actors' behaviour in each area. Technically, the membrane is a function setting a LOD values to areas.

How to simplify the complexity of a given behaviour automatically? We already mentioned that if an activity is not atomic, the genius should adopt sub-goals offered by this activity. The tricky part is just in here: if the LOD value for the location where the activity should be performed is low, the genius must not adopt sub-goals, it should execute the chosen activity atomically instead of that (Fig. 6, 7). In this way, for example, an actor can water the whole garden atomically, instead of finding a can and going from a flower-bed to a flower-bed. If the LOD value is too low, actors and objects even cease to exist, and they are not created again until the detail is increased. Of course, the outcome of a simplified simulation might differ from the outcome of simulation in full detail - but since the simplified place is not in the centre of attention, this does not matter.

IVE approach is very robust comparing to LOD AI techniques typically used in games, e.g. [4], [13]. Notice that IVE allows more places to be simulated in detail at any particular moment (even a place unobserved by any user, since there might happen something important from the point of view of the story). IVE also admits a gradual simplification of a simulation complexity between an important and an unimportant place (which keeps the overall load more consistent and controlled during shifts of the centres of attention) and areas to be simulated partially. Additionally, the hierarchy of abstractions may have any arbitrary depth. However, in practice, we would hardly need more than 4 or 5 levels of abstraction, and - naturally - some applications might benefit from a more lightweight LOD approach then we use in IVE (as discussed also in [13]).

We recognised several remarkable issues concerning our LOD approach. First, there can occur an 'oscillation problem' with a user furiously oscillating between two locations. IVE uses a garbage collector technique to cope with this. When a LOD value can be decreased, we still keep the simulation at the previous level of detail until we need the processor capacity somewhere else - only then the garbage collector decreases the LOD value.

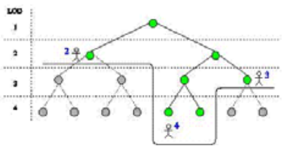

Fig. 6. The space hierarchy in IVE. Some parts are simulated in full detail (LOD 4), other in lower detail (LOD 2, 3).

Second, there are problems with increasing LOD during running some activity - for example when an atomic watering of the garden becomes non-atomic in the middle of process, one half of flower-beds should be already watered. We adopted some *ad hoc techniques* to deal with this problem, but we want to resolve it in a more robust fashion in the future.

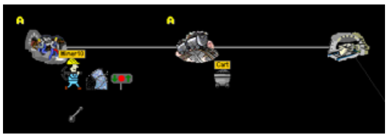

Fig. 7. Mine scenario in IVE. The mine comprises of three sublocations: the lower part, the tunnel and the upper part. There are two miners - one in the upper part and one in the lower part. The lower part and the tunnel are not expanded (here). The scenario illustrates an activity taking place in locations with different LOD (i.e., mining). The cart leaves the pit, which is simulated in lower detail (LOD is 4) and enters the upper part, simulated in full detail (LOD 5).

To sum it up, IVE presents a robust solution to LOD technique and is advisable as a platform for LOD research. However, for some applications, our technique might be too robust. In detail, the technique is described in [6], [21]. For example, in ALOHA system [23], role-passing (Sec. V) realises simple LOD for actors controlling - the role is not passed if actor cannot be seen.

VII. Limitations and Present Work

In this section, we discuss main IVE limitations and introduce our present work.

Editor. As already said, perhaps the main limitation of IVE is that there is no user-friendly editor. A world model and actors' behaviour must be specified in external xml and Java files. Worlds and behaviour editors belong among our current works:

Episodic memory. On the present, only short-term memory for actors is an architectural build-in in IVE. Episodic memory can be programmed externally. Without it, actors wouldn't be able to operate easily with the objects they currently do not perceive (e.g., take a hammer that is in the next-door room). Episodic memory is our second work-in-progress.

Planning. IVE action selection is based only on reactive planning. However, the action selection mechanism can be easily replaced. In the third work-in-progress, we focus on augmenting IVE with an HTN planner [11]. Every smart activity will offer not only reactive rules, but also PDDL operators [20] for a genius.

LOD research. We plan to continue in LOD AI research, as detailed in Sec. VI.

GUI. The GUI of IVE is not user friendly. However, IVE is designed in a modular way, therefore GUI can be easily replaced.

Discrete space. IVE was designed only for discrete, graph-describable worlds, not for 3D environments. IVE creates a discrete abstraction of a 3D world for the purposes of a high-level reasoning. 3D virtual worlds are mostly represented in way that they can be easily displayed, but on basis of this representation, a virtual human can't make decisions. We plan to interconnect IVE and 3D word in the close future.

Virtual company. We work on a society simulator (as proposed in [5]) using a combination of agent-oriented and process-oriented approaches. IVE is used here as a middleware. This tool is intended mainly for managers' education.

Educational game. We have already started to work on a simple storytelling game using IVE, which is intended for children education in civics. The idea is to play short social narratives in IVE that would allow children to interact in order to see "what happed, if..." For the purpose of this enterprise, we must primarily extend IVE with a drama manager, which would drive actors according to the given plot. We have developed the technique for driving stories and evaluate them in a dummy fantasy case-study [7]. The technique uses Petri Nets for plot specifications and admits describing non-linear plots and unfolding the story at several different places simultaneously, what is vital for our game.

VIII. Conclusion

In this paper, we have presented IVE, a toolkit for prototyping virtual worlds and virtual humans' AI.

The contribution of the toolkit is two-fold. First, it serves for prototyping both the AI and the virtual worlds. Second, it challenges the simulation speed problem and the behavioural design complexity problem. Particularly, IVE utilises a robust level of detail technique and a knowledge representation admitting simple extensions managing (i.e., objects and actions) and actors coordination (autonomously and also in a centralised way). There are many other prototypes and applications coping with similar problems, but IVE joins all mentioned solutions, and is designed and developed in the extensible way.

We have suggested that IVE can be used in many ways – e.g. as an educational tool, for case-studies prototyping, or as a middleware. We have also briefly presented our current work in progress, focused both on overcoming some limitations of IVE and on our practical projects, including and educational game and a simulator of companies.

The details of the projects can be found in [6], [7], [21]. IVE can be downloaded at [15].

Acknowledgement

The work was supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) "Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realisation" and GA UK No. 351/2006/A-INF/MFF. The work on company simulator is also supported by SoftDeC company: http://www. softdec.cz/. The research would not be possible without intensive work of tens of students from Charles University in Prague. The toolkit IVE was implemented by Ondřej Šerý, Tomáš Poch, Pavel Šafrata, Jan Kubr, Jiří Kulhánek and Zdeněk Šulc. The drama manager prototype was implemented by Adam Abonyi. The additional work is being developed by Daniel Balaš, Martin Juhazs, Ondřej Holeček, Jiří Vorba, Klára Pešková and Josef Reidinger.

Special thanks belong also to Daniel Ryšlink for his comments.

References

[1] Agre, P. E., Chapman, D. *Pengi: An implementation of a theory of activity.* In: Proceedings of the 6th national Conference on AI, Seattle, Washington, 1987, p. 196-201

[2] Bojar, O., Brom, C., Hladík, M., Toman, V. *The Project ENTs: Towards Modeling Human-like Artificial Agents.* In: SOFSEM 2005 Communications Liptovský Ján, Slovak Republic, 2005, p. 111-122

[3] Bratman, N. *Intention, plans, and practical reason.* Cambridge, Mass: Harvard University Press, 1987

[4] Brockington, M. *Level-Of-Detail AI for a Large Role-Playing Game.* In: AI Game Programming Wisdom, 2002

[5] Brom, C., Kocáb, P. *Virtual agents in a simulation of an ISO-company. A poster.* In: Proc. of Intelligent Virtual Agents, Springer, Berlin, 2005

[6] Brom, C., Lukavský, J., Šerý, O., Poch, T., Šafrata, P. *Affordances and level-of-detail AI for virtual humans* In: Proceedings of Game Set and Match 2, Delft, 2006

[7] Brom, C., Abonyi A. *Petri-Nets for Game Plot.* In: Proceedings of AISB - Artificial Intelligence and Simulation Behaviour Convention, Bristol, 2006, p. 6-13

[8] Bryson, J. *The Behavior-Oriented Design of Modular Agent Intelligence.* In: Müller, J. P. (eds.): Proceedings of Agent Technologies, Infrastructures, Tools, and Applications for E-Services, Springer LNCS 2592, 2003, p. 61-76

[9] Epic Megagames. *Unreal Tournament*, 2004, http://www.unrealtournament.com/

[10] Forgy, C. *Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem.* In: Artificial Intelligence 19, 1982, p. 17-37

[11] Ghallab, M., Nau, D., Traverso, P. *Hierarchical Task Network Planning.* In: Automated Planning: Theory and Practice, chapter 11, Morgan Kaufmann Publishers, USA, 2004

[12] 1Gibson, J. J. *The Ecological Approach to Visual Perception.* Boston: Houghton Muffin, 1979

[13] Grinke, S. *Minimizing Agent Processing in "Conflict: Desert Strom".* In: AI Game Programming Wisdom II, 2004

[14] Huber, M. J. *JAM: A BDI-theoretic mobile agent architecture.* In: Proceedings of the 3rd International Conference on Autonomous Agents (Agents'99). Seatle, 1999, p. 236-243

[15] *IVE project*, http://urtax.ms.mff.cuni.cz/ive/public/about.php

[16] Kallmann., M., Thalmann, D. *Modeling Objects for Interaction Tasks* In: Proceedings of EGCAS 98, Lisbon, Portugal, 1998, p. 73-86

[17] Laird, J. E., Newell, A., Rosenbloom, P. S.: SOAR: An Architecture for General Intelligence. In: Artificial Intelligence, 33 1, 1987, p. 1-64

[18] Mateas, M. *Interactive Drama, Art and Artificial Intelligence.* Ph.D. Dissertation. Department of Computer Science, Carnegie Mellon University, 2002)

[19] Orkin, J. *3 States & a Plan: The AI of F.E.A.R.* In: Game Developer's Conference Proceedings, 2006

[20] *PDDL, Planning Domain Definition Language.*, 2002, http://planning.cis.strath.ac.uk/competition/

[21] Šerý, O., Poch, T., Šafrata, P., Brom, C. *Level-Of-Detail in Behaviour of Virtual Humans* In: Proceedings of SOFSEM 2006: Theory and Practice of Computer Science, LNCS 3831, Czech Republic, 2006, p. 565 - 574

[22] Tilab: *Jade, Java Agent DEvelopment Framework.*, http://jade.tilab.com/

[23] McNamee, B. Dobbyn, S. Cunningham, P. and O'Sullivan *Men Behaving Appropriately: Integrating the Role Passing Technique into the ALOHA system.*, C. Proceedings of the AISB02 symposium: Animating Expressive Characters for Social Interactions (short paper) p. 59-62.

[24] Dignum F. *Autonomous Agents with Norms.*, AI and Law, 7, p. 69 - 79, 1999