Affordances and level-of-detail AI for virtual humans*

Cyril Brom[1], Jiří Lukavský[2], Ondřej Šerý[1], Tomáš Poch[1], Pavel Šafrata[1]
brom@ksvi.mff.cuni.cz, jirilukavsky@gmail.com
[1]Faculty of Mathematics and Physics, Charles University in Prague, CZ
[2]Institute of Psychology, Academy of Sciences, Prague, CZ

ABSTRACT

Anyone who aims at developing a virtual reality application featuring a large artificial world inhabited by intelligent virtual humans will face two problems—the problem with simulation speed, and the problem with adding new components to the simulation both during the development and after the release. We have developed a framework that copes with these issues. The solution is based on augmentation of the level-of-detail AI technique and theories of affordances and practical reasoning. Contrary to existing approaches, our solution is theoretically well-founded, robust and deals with both these issues at once. In this paper, we present the key concepts of our framework and evaluate a test scenario.

Keywords: affordance, intention, virtual human, virtual world, level-of-detail AI

INTRODUCTION AND MOTIVATION

This paper concerns itself with simulations of large virtual worlds inhabited by intelligent virtual humans. Above all, we think of the worlds of role-playing computer games and interactive virtual storytelling applications.

By *virtual human* we mean a piece of software that imitates behaviour of a human in a virtual world and that is equipped with a virtual body visualized by a graphical viewer. We are focused on *intelligent* virtual humans, by what we mean that they carry out more complicated tasks than just walking, object grasping or chatting in Elisa-like manner. (The word “virtual” will be abbreviated as “v-”, e.g. a v-human.)

What sort of problems do we have with a large v-world? Originally, we aimed at developing a large long-lasting interactive computer game emphasising a story. Without any trouble, we have prototyped behaviour of several game actors in a toolkit, which we developed formerly (Bojar et al., 2005). However, not surprisingly, it has happened that it had been almost impossible to run all the actors on a single PC because of limited computational and memory resources. These actors had to be intelligent; they should have performed in real-time complex tasks like gathering the harvest, having fun in a pub, or travelling. As an example, consider a v-merchant riding with its donkey within a v-kingdom. Yet assume there are tens of such v-humans important for the course of the game.

There is yet another problem with large v-worlds. Typically, one needs to supplement the simulation with new components easily. Inserting of new objects and actions should be allowed anytime both during the development and after the release. However, using classical symbolic representation of the v-world, v-humans must be equipped with a learning algorithm (which teaches v-humans what are the symbols

* This work is partially supported by the Program “Information Society”; the project 1ET100300517.

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

denoting the new objects like "gun" good for), or modify the control algorithm of each v-human for every extension. While the former option is like a sledgehammer to crack a walnut, the latter is asking for a trouble with unmanageability.

To sum up; there are two problems with large v-worlds we have stumbled on:

1) Easy extensibility must be allowed, but without use of any learning algorithm.
2) Speedy simulation must be allowed, in spite of the limited resources.

We have aimed to develop a framework, or say architecture, for applications featuring large v-worlds, that would cope with these two issues. The framework has been called IVE (*intelligent virtual environment*). In seeking for solution we decides to take advantage of three successful concepts: psychological theory of affordances (Gibson, 1979) for the perception of v-humans, the theory of practical reasoning (known also as BDI; Bratman, 1987) for their decision making, and level-of-detail technique for controlling the overall simulation—but contrary to its typical use in computer graphics, we have used is for AI (LOD AI). These three main points help us to simulate the v-world efficiently and to facilitate its design. In particular, the simulation is simplified on unimportant places automatically, and new actions and places can be loaded into the v-world of IVE as plug-ins and v-humans are able to adapt to them without using any machine-learning algorithm. Hence, the two aforementioned issues are solved and we are planning to use IVE in our next project. The framework code, a test-case scenario and the documentation can be downloaded at: `http://mff.modry.cz/ive`.

The goal of this paper is to explain the key theoretical concepts underlying the IVE framework. We will start with setting the ground of related works concerning our project. Then, we describe the main concepts of the framework. Finally, we present the evaluation and discuss the future works.

RELATED WORKS

As far as we know there is no work that combines all of the three tenets of our framework: affordances, intentions and LOD AI. Notable exception could be *The Sims,* but it unfortunately cannot be compared, since there is no published paper on it. This section describes the most relevant works concerning the main principles of IVE.

Virtual affordances. The concept of virtual affordances enables us loading new components into the v-world as plug-ins. It resembles so-called *smart object* approach to interactions of v-humans with v-objects (Kallmann, 2001). A smart object is an entity with the ability to describe in detail its functionality, its possible interactions as well as behaviour of an interacting v-human. We can say, that the purpose of a smart object is "engraved" in it directly. Using smart objects, the v-world can be described in the terms of a purpose-oriented language, and since v-humans can directly perceive this purpose, they do not need to infer it from a symbolical representation. It follows that an object can be loaded into the v-world as a plug-in and v-human can interact with it automatically. However, smart objects encapsulate only low-level "graphical" information like a v-human's position during execution of an action, or a desired hand-shape. Ciger (2005) extended smart objects so that they can pass on planning operators to a v-human, which can use a planning algorithm to generate a more complicated sequence of actions. This work actually equipped smart objects with "AI", however, since planning suffers from combinatorial complexity, it does not fit well to the domain of applications with highly dynamic v-worlds as computer games

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

are. We present a concept similar to smart objects, the concept of "smart materialisations", which, however, encapsulate AI that is rather reactive. In addition, in IVE, new actions and places (not only objects) can be loaded as plug-ins.

In many existing computer games the problem of meaningless v-worlds is tackled by adding some semantic marks (e.g. firing positions). Contrary to IVE, these ad hoc solutions are not theoretically founded, and cannot cope with aforementioned issues.

Level-of-detail AI. In a classical simulation, the v-world is simulated in its entirety. The idea behind LOD for AI is not to compute such details that a user cannot see, or that are otherwise unimportant for the overall course of the simulation. Sometimes, computer games reflect this fact, but only to a limited degree: creatures out of the sight of the user are simulated not at all typically. This approach "all – or nothing" would cause scenic inconsistencies in a game emphasising the story. Moreover, it allows for only one important place in the v-world; the place observed by the user. The method "see – not see" brings a problem with places that are important at a given instant, but unobserved. Contrary, our method allows for (a) simulation of all places that are important (not only the observed ones), (b) partial simulation of partially, and important places, and (c) gradual simplification of simulation complexity between an important and an unimportant place.

Relatively robust approaches have been presented in (Brockington, 2002; Champandard, 2003). Our approach resembles most the work of Champandard, who used hierarchical state machines for controlling behaviour of v-humans to simplify the simulation smoothly, but the work has remained in a sketch, not further explored.

Intentions. A huge amount of works exploiting BDI exists; this architecture is becoming nearly an industrial standard for modelling of action-selection. In our framework, we use probably its most traditional version described e.g. in (Wooldridge, 2002), but we augment it with affordances and level-of-detail.

THE FRAMEWORK OF IVE

In this section, we describe the basic features of IVE; virtual affordances, the intentions, the hierarchical mechanism of action selection, and the LOD technique.

Virtual affordances and virtual world architecture

Affordances have been 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 simple 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* (Gibson, 1979, p.127)".

He wanted to say, metaphorically, that we directly perceived the meaning of the objects—what they were good for—not the objects themselves. Of course, a human or an animal had to "pick up information" about the meaning somehow, and this "information pick-up" was carried out by the animal's brain and its moving body. However, the cognition was fold into the perception according to the theory. Gibson adopted a different level of description of reality—the ecological level of description.

Fundamental properties of an affordance are its relativity to a particular actor and its independence of actor's ability to perceive it. To Gibson, affordances are relationships. They exist naturally: they do not have to be perceivable, known or desirable. Thus, a potty affords sitting to a child, not to an adult, because the potty is

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

not “sittable” for adults. However, some children may not be able to perceive this possibility, even if they are technically able to sit on the potty.

The theory both has been criticized and has inspired a lot of researchers, who have brought new interesting experimental results. Here, we are interested in the theory’s implementation potential. The level of description in the theory is close to the level of representation of a v-world, therefore the theory can underlie a development of a large application featuring v-worlds and v-humans. The advantage of the concept is its close relationship to the v-humans’ actions. If we allow a v-human to directly perceive its possible actions, there is no need for emulating the cognitive processes in its mind, which are still rather unclear, in spite of the psychological and AI research.

In our effort at framework allowing for easy extensibility and LOD AI we have refined Gibson’s theory to allow for its implementation. In IVE, a typical v-human is not perceived as intelligent and autonomous (i.e., with its own “AI algorithm inside its mind”). Instead, it is navigated by *intelligent environment*. Nevertheless, the illusion of intelligence of these dummy actors is retained. The refinement constitutes a ground for the architecture of v-worlds in IVE, and is described in this subsection.

Topology. In IVE, *way-places* are the ground constituting a v-world. Each way-place is a place that affords standing to a v-human, and that cannot be further divided. Way-places are represented as nodes of a *topology graph*. An edge of the graph affords crossing to a neighboring way-place. Way-places resemble the concept of way-points, however, contrary to way-points, which are used only for path-finding, in IVE, the way-places constitute the “physical” reality of the v-world. Way-places are organized in a hierarchical structure, which is always a tree. All way-places are leaves of the tree, a non-leaf node is a *location*, and the root represents the whole world. See Fig. 2.

Objects. IVE distinguishes two types of “physical entities”: *objects* and *actors*, where actors are v-humans’ bodies. Typically, each object and an actor are located in a way-place. This can be violated during a so-called LOD contraction as described later. An object or an actor has so-called *attributes*, which represents its physical properties.

Actions. From the AI perspective, a v-world from an application featuring v-humans is viewed as a set of objects that are represented symbolically, and as a topology structure. Actions are then perceived as transformations of the objects’ attributes. Each v-human must decides which action to perform in order to satisfy its goals, or say, intentions. V-human’s decision is based on a manipulation with the symbols denoting the objects and their properties, e.g. in the course of look-ahead-planning or evaluating of options. After the action is chosen, its name is passed on to the server of the v-world, which performs the transformation of the attributes of affected objects. In IVE, we adopt the same view, with one notable departure from it. Actions, in IVE, have similar status as objects—they are entities located in the v-world (at a way-place or a location), entities that can be perceived directly. We call them *materialisations*.

Direct perceiving of an entity means (in our terms) that the entity can be accessed in a linear time. In the case of materialisations, it means that having an intention, a v-human’s control algorithm is able to find a materialisation that accomplishes the given intention in $O(n)$, where n is the total number of materialisations that accomplish the intention. No problem-solving algorithm suffering from combinatorial complexity need to be executed in order to infer the name of the action.

A materialisation is equipped with *slots*. Relation between a materialisation and an object is represented by so-called *reference* from the object to a slot. The slots are like parameters of a function of a programming language. Similarly to materialisations,

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

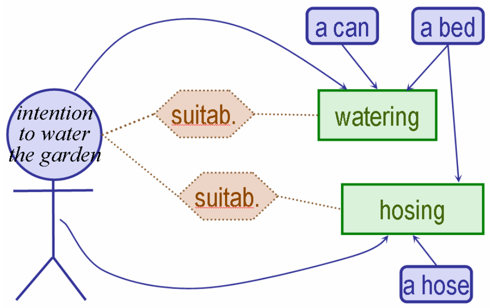

Fig. 1: An actor directly perceives two possibilities in a garden. Legend: objects are blue, suitabilities are red, materialisations are green.

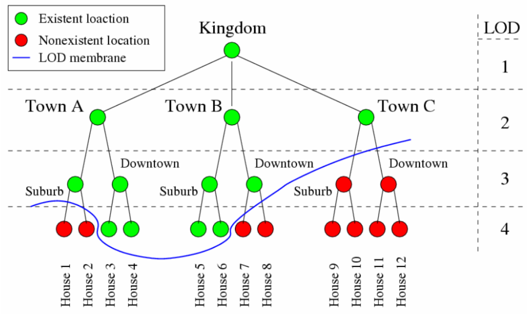

Fig. 2: A hierarchical structure of a v-kingdom. While all actions in Town C are simulated only partially, Houses 3–6 are simulated in detail.

references and slots can be perceived directly. As an example, consider watering a garden bed by a can held by a v-human. First, a materialisation accomplishing the intention of watering a bed is to exist in the given v-world. Second, the materialisation will be equipped with at least three slots referenced by three types of objects: something that is to be watered (i.e., the "what-affords-waterability"), something that is able to water (i.e., the "what-affords-performing-of-watering"), and something that can be watered with (i.e., the "what-affords-watering-with-possibility"). The first parameter can be a garden bed. The second one can be a v-human (but not a v-frog). The last parameter is whatever that can be used for watering, e.g. a watering can or a bucket. See Fig. 1. Notice, that in IVE, the v-human does not perceive the symbol of "can_12", or "garden_bed_15", but the materialisation, their slots and the respective references. Only through the references, it can perceive the objects and their attributes.

Suitabilities. Apparently, in real world, a human waters rather with a watering can than with a bucket, even if the bucket is suitable too. It is also obvious that a child will prefer watering with a smaller can than an adult, while the adult's preference is the other way round. In addition, more ways of accomplishing a given intention can exist: e.g., a hosing can compete with watering provided that a hose exists in the garden.

In IVE, each materialisation is equipped with a *suitability*, an entity pre-given by a designer, which is a function that on the basis of attributes of objects referring to the materialisation (including the actor) computes how convenient the performance of the materialisation would be in the given context for the given actor. Suitability is a part a part of the v-world, with a status similar to the status of materialisation. An actor can perceive the suitability on a given materialisation directly.

To recapitulate, there are five important entities comprising the "physical reality" of a v-world in IVE. These entities are way-places (and locations), objects (together with actors), materialisations, references, and suitabilities. We differ from typical v-world architecture because the last three entities are directly a part of the v-world, they are not to be inferred by the actors' minds. We can say that materialisations and suitabilities mediate perception of the objects in a meaningful way to the actors. Their role in perception of v-humans is similar to the role of a graphical viewer in perception of a human observer. The purpose of a viewer is just to mediate symbolic objects to the observer by means of a neat graphical representation. This is actually a consequence of refinement of Gibson's theory in an implementable way.

Since all materialisations and suitabilities are a part of the v-world, and not a part of the actors' minds, the v-world can be easily extended. Adding a new actor entails

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

just equipping the actor with a set of intentions linked with the materialisations; adding a new object is just about creating its references to respective materialisations. Below, there is a sketch of the algorithm of perceiving the environmental possibilities.

Algorithm 1. Perceiving of possibilities.
The algorithm is performed whenever the v-human aims to accomplish an intention.
Input: An actor with an intention situated in a given location.
Output: The materialisation that is most suitable for the actor in a given context.

1) $M \leftarrow$ set of all applicable materialisations that accomplish a given actor's intention
2) $M^* \leftarrow$ set of all applicable instantiated materialisations (based on M)
3) $m_{best} \leftarrow$ the most suitable materialisation from M^*, or 0, if no materialisation exist

Subsequently, the actor is allowed to pass on m_{best} to the server of the v-world, which will perform the desired transformation (e.g., it will cause that the garden bed is wet, the can is empty, and the v-human becomes more tired).

Notice that a part of the step (2) is a so-called *instantiation* of a materialisation. It is a process of assignment a set of particular objects. Hence, in the step (3), the set M_* can contain for example: (a) the materialisation of watering with the small can, (b) the watering with the big can, and (c) the hosing with the hose.

Intentions, genii, and intention-action selection

We have described how a v-human of IVE perceives environmental possibilities in order to perform a simple action. We now turn to a problem of accomplishing a task, that is a chain of actions, and a problem of selecting a task to perform. We outline a theory of practical reasoning we use, and on the basis of it, we present the second part of the algorithm that controls behaviour of v-humans in our framework.

According to the theory of practical reasoning (Bratman, 1987; Wooldridge, 2002), the state of each rational human can be viewed as a triple $<B, D, I>$; where B is a set of present human's beliefs, which in fact constitutes the memory; D is a set of present human's desires; and I is a set of intentions. Intentions are the states that guide the human's present and future conduct—they are desires that are *committed* to be performed. Their important feature is their persistence. Once a rational human has committed an intention, it will act in order to accomplish the intention, until the intention is satisfied, or it is believed that it is impossible to achieve the intention.

The process of adopting new intentions is called *deliberation*. Deliberation is not carried out in every instant, but on certain occasions only. The already committed intentions play a significant role in one's deliberation, since one would typically not adopt a new intention that is in disagreement with an already committed intention, unless a committed intention is being reconsidered.

The second part of reasoning is the *means-ends* analysis. In a nutshell, it is a process of elaborating how to accomplish a given intention. Intentions can be hierarchically nested. More specifically, in order to accomplish an intention, one can adopt a sub-intention as result of the means-ends analysis. For example, in order to make a trip to one's cottage, one should adopt a sub-intention of taking the key. Sub-intentions of more intentions can be interleaved; that means, for example, that one can take the key (in order accomplish a sub-intention of the cottage-trip-tomorrow), and then buy a plane-ticket (in order to accomplish a sub-intention of a business-travel-after-cottage-trip), and then drive a car to the cottage. Sub-intentions can be adopted continuously, filling one's partial plans about the future.

In our framework, the theory is used in the intention-action selection algorithm. This algorithm is performed by *genii* that are bodiless entities guiding one or more

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

actors. In a basic case, each v-human is a couple <*actor*, *genius*>; it is driven by just one genius.[1] The algorithm of intention-action selection commits the v-human's desires as intentions according to the v-human's internal drives, schedules pre-given by a designer, or artificial emotions. At a given instant, one intention is chosen as *active* (i.e., present-directed in the terms of Bratman); this intention serves as the input for perceiving the environmental possibilities in Algorithm 1. When the active intention is attained by a materialisation, a new intention becomes active.

Algorithm 2. Intention-action selection (flat).
This algorithm is performed by a genius of a v-human in the simple case, i.e. when the v-human is a couple <actor, genius>.
Input: An actor in a given location, and its genius with the set of desires and intentions.

1) if a genius of a v-human wants to deliberate, then $I \leftarrow$ set of all adopted intentions (based on desires, internal drives, past failures, and artificial emotions)
2) $i_a \leftarrow$ active intention (from I)
3) $m_{best} \leftarrow$ Algorithm 1 (based on i_a)
4) if m_{best} is *0*, then i_a has failed, and go to the step 1
5) pass on m_{best} to the server of v-world
6) if m_{best} succeeded, go to the step 1;
   otherwise, go to the step 3 and look for a new materialisation.

This algorithm is modified and extended version of classical BDI-action-selection algorithm (e.g., Wooldridge, 2002). Notice, that the algorithm joins deliberation and means-ends analysis. However, it is "flat"; no sub-intentions can be adopted.

Hierarchical action-selection mechanism

We now extend Algorithms 1 and 2 in a hierarchical manner. It is just this extension, which makes IVE unique among other architectures of v-worlds, and which actually combines intentions with affordances and enables for the LOD AI technique.

In IVE, each materialisation is not only coupled with its suitability, but also with a *breakdown*. Let us have a v-human *v* executing a materialisation *m* achieving an intention *i*. We say, that breakdown of *m* is a set of pairs <*sub-intention*, *advise*>. A *sub-intention* is an intention that *v* has to commit itself to in order to accomplish a part of *m*. An *advise* is a rule that *v* should follow in a future in Algorithm 2 in the step (2) for choosing the active intention. Following advises, *v* will be activating sub-intentions of *m* in such order that, finally, original intention *i* will be attained. Essentially, *m* will be performed by execution of some other materialisations that accomplish the sub-intentions. For example, sub-intentions of a breakdown of watering a garden, can be (a) finding a "what-affords-performing-of-watering" (a can), (b) filling it, (c) finding a "what-affords-waterability" (a bed), (d) cleaning-up the "what-affords-performing-of-watering". In accord with advises, (c) will be performed in a loop, for each bed once. See Algorithm 3 below.

In the step (2) of Algorithm 3, the genius chooses the active intention on the basis of all advises of sub-intentions passed on from the v-world, and on the basis of the v-humans' internal drives, schedules, and emotions. The step (6) presents means-ends analyse, however, it is performed instantly due to the breakdown. The sub-intentions

[1] Our framework allows for more complicated cases, where one actor is guided by more genii. In addition, one genius can guide more actors. For example, in a pub, so-called *genius pub-specialist* can guide behaviour of v-humans in a central manner. In this case, a basic genius of a v-human, just before entering the pub, is allowed (but not required) to pass on the actor to the pub-specialist. Then, both genii will control the actor simultaneously. This technique is called *role-passing*. It is implemented in IVE, as detailed in the extended version of the paper (Brom at al., 2005).

Algorithm 3. Intention-action selection (hierarchical).
This algorithm is performed by a genius of a v-human in the basic case.
Input: An actor in a given location, and its genius with the set of desires and intentions.
(1) if a genius wants to deliberate, then $I \leftarrow$ set of all adopted intentions, and sub-intentions
(2) $i_a \leftarrow$ actual intention (from I)
(3) $m_{best} \leftarrow$ Algorithm 1
(4) if m_{best} is *0*, then i_a has failed; and go to the step 1
(5) if m_{best} is an atomic act, then
    pass on m_{best} to the server of v-world
    if m_{best} succeded, go to the step 1
    otherwise, go to the step 3 and look for a new materialisation.
(6) otherwise, $I \leftarrow I \cup m_{best}[breakdown]$; and go to the step 1

are perceived directly like affordances. It helps with extension of v-world. Consider a v-human with a top intention of "having fun" activated in the evenings according to an internal schedule. According to Algorithm 1, the v-human will perceive directly all materialisations accomplishing the intention. Let us assume they are "gossiping at a square" and "visiting a concert". However, a designer may want to add "enjoying in a pub". In IVE, she is allowed to load it as a plug-in comprising a v-pub and a "recipe" how to act in it (parts of the recipe are materialisations, suitabilities, sub-intentions, and v-objects). The v-human will perceive the new possibility of "enjoying in a pub" directly. Further decision of the v-human will be carried on in accord with the loaded sub-intentions from the recipe, even if the v-human has not learned what the symbols "pub", "bar", or "beer" mean. The v-human will act upon them automatically. Moreover, still, suitabilities will allow a v-gossip to prefer the gossiping at a square, while a v-worker to prefer the newly added pub.

Level-of-detail AI
Remember that the idea behind LOD AI is not compute the details that a user cannot see, or that are otherwise unimportant for the overall course of the simulation. The technique of LOD AI can be applied in our framework in a robust and yet straightforward way due to the breakdown structure of materialisations.

To describe our approach we use a membrane metaphor—imagine an elastic membrane cutting through the hierarchy of way-places and locations (see Fig. 2). Only the locations above this membrane do currently exist and only these locations are simulated. During the development, so-called *LOD complexities* are assigned to the levels of hierarchy of materialisations by a designer. For example, LOD complexity of watering a garden can be 3, while watering an individual garden bed can be 4. Then, in the course of simulation, on the basis of important objects (e.g., a human observer), and events (e.g., a story-generated local uprising), in every instant, a *LOD values* are assigned to all locations. The values denote the desired complexity of the simulation in the given location; the values shape the membrane. In addition, the step (5) of Algorithm 3 is refined as follows:

if m_{best} is an atomic act, or $m_{best}.LOD\_complex$ = *location.LOD_value*, then ...

It means that the breakdown of m_{best} will not be passed on to the genius, provided the LOD value is too low. Instead, the m_{best} will be performed as it was atomic. For example, in the case of watering a garden atomically, the execution will mark the whole garden as being watered, and will reduce amount of water in a barrel. On even higher level, the garden may exist not at all, and only a materialisation of growing will be executed at the level of a village. Hence, in the course of partial simulation, some

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

objects or actors may cease to exist, and the other objects will be located in localities, not in way-places (e.g., the barrel will stay in the garden “somewhere”).

LOD values are assigned so that the values of neighbouring locations differ at most by one. The algorithm of values assessment is detailed in (Šerý at el., 2006).

IMPLEMENTATION AND EVALUATION

The described framework is implemented in Java and tested with a scenario comprising about 60 actors acting in four villages; each with a pub, four mines, and nine houses (see Fig. 3).

Direct perception requirement is achieved due to exploitations of hash-maps. Advises are implemented as fuzzy if-then rules with lazy evaluation. The overall simulation uses discrete simulation paradigm. The benchmarking results suggest, not surprisingly, that the LOD simulation acceleration is significant. The exact numbers depend, of course, on the tasks performed. The highest speed-up is achieved in LOD contraction of materialisations performed in loops (e.g. loading a cart in a mine). IVE overhead on LOD management takes about 5-10% of simulation time. Implementation and evaluation are detailed in the extended version of the paper (Brom et al., 2005).

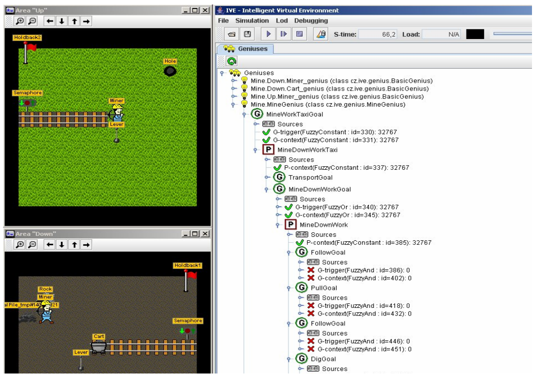

Fig. 3: Mine scenario, a screenshot.

DISCUSSION AND FUTURE WORK

We aim to use the framework, or at least its concepts, in a larger virtual-storytelling application. The story will be generated and adjusted continuously by means of a special story-generating genius. This kind of genius is similar to so-called *drama manager* of Mateas (2002). The v-humans will be equipped with an episodic memory for direct access of objects’ positions (which is not included in the current version).

We plan to use two main types of actors; dummy actors, and persistent actors. While the former will exist only for a particular period and carry out only a specific task, the latter will carry out complex tasks for a longer period, and will be allowed to increase LOD value in the locations they will be located in.

We remark, that even if we disregard the role of symbolic learning and problem solving in large applications featuring v-humans, we do not deny them. For example, in our framework, each genius is allowed to use (or even develop) its own suitabilities, regardless of the environmental ones, which may lead to starting an inappropriate process, *e.g.*, watering with a pen. This feature is vital for modelling of symbolic trial-error learning. Nevertheless, we think we hardly use it in our application.

There are several notable limitations of IVE in the current version. First, each v-human can perform just one materialisation in a given instant, e.g. eating during walking is not allowed. Second, the innate structure of materialisations is hierarchical, what denies the use of non-hierarchical action-selection models. Other limitation, but merely terminological, is the fact, that all materializations must be performed intentionally. As Bratman (1987) put it, some actions are not performed with an intention, consider reflexes as an example.

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.

CONCLUSIONS

In this paper, we have presented the main principles behind IVE, the framework of large v-worlds featuring intelligent v-humans. The principles were virtual affordances, intentions, and LOD AI. The combination of these three allows us to achieve the two goals of the project: enable easy extensibility of a v-world, and enable speedy simulation by means of smooth simplification of simulation detail on unimportant places. Contrary to similar approaches, the framework copes with both issues at once, additionally allows for a role-passing technique, and is robust and theoretically well-founded. In particular, it augments the theory of practical reasoning with the theory of affordances refined in an implementable manner. Details can be found at: `http://mff.modry.cz/ive`.

REFERENCES

Brom, C., Lukavský, J., Šerý, O. (2005) “Affordances, level-of-detail AI, and role-passing for virtual humans”. Technical Report 2005/6. Department of Software and Computer Science Education, Charles University in Prague, Czech Republic.

Bojar, O., Brom, C., Hladík, M., Toman, V. (2005) “The Project ENTs: Towards Modeling Human-like Artificial Agents”, In: SOFSEM 2005 Communications. Slovak Republic, pp. 111 – 122.

Bratman, N. (1987) “Intention, plans, and practical reason”. Cambridge, Mass: Harvard University Press.

Brockington, M. (2002) “Level-Of-Detail AI for a Large Role-Playing Game”, In: AI Game Programming Wisdom, S. Rabin (Ed.). Charles River Media.

Champandard, A. J. (2003) “AI Game Development: Synthetic Creatures with Learning and Reactive Behaviors”. New Riders, USA.

Ciger, J., Abaci, T., Thalmann, D. (2005) “Planning with Smart Objects”, In: Short papers of WSCG 2005. Pilsen, Czech Republic, pp. 25 – 28.

Gibson, J.J. (1979) “The Ecological Approach to Visual Perception”. Boston: Houghton Muffin.

Kallmann, M. (2002) “Object interaction in real-time virtual environments”. PhD Thesis. EPFL, Switzerland.

Mateas, M. (2002) “Interactive Drama, Art and Artificial Intelligence”. PhD Thesis. Department of Computer Science, Carnegie Mellon University, USA.

Šerý, O., Poch, T., Šafrata, P., Brom, C. (2006): “Level-Of-Detail in Behaviour of Virtual Humans”, In: Proceedings of SOFSEM 2006. Czech Republic (to appear)

Wooldridge, M. (2002) “An Introduction to MultiAgent Systems”. John Wiley&Sons.

This work was presented at Game Set and Match 2 Conference, Delft, The Netherlands.
Published by Episode-Publishers (c) in the conference proceedings.