An Architecture for Co-Habited Virtual Worlds

Christoph G. Jung*, Jürgen Lind, Christian Gerber†,
Michael Schillo, Petra Funk, and Alastair Burt
MAS Group, DFKI GmbH, Im Stadtwald, D-66123 Saarbrücken, Germany

Keywords: Virtual Worlds, Intelligent Agents, User Interface

Abstract

As a recent trend, research in *Distributed Artificial Intelligence*, *Distributed Interactive Simulation*, and *Virtual Reality* shares a new centre of interest which we call *co-habited virtual worlds* (CHVW). Herein, synthetic *agents* and human-controlled *avatars* interact in a globally networked setting. Possible CHVW applications range from lifelike conferencing, virtual marketplaces, and entertainment up to intuitive visualisation of complex *multi-agent systems*. This paper presents the *Effector-Medium-Sensor* (EMS) architecture for building open CHVW. EMS introduces a form of *virtually acting and sensing* which bases on common standards, such as JAVA™, RMI, VRML'97, and CORBA. Our implementation, the *Social Interaction Framework*, demonstrates the ability of *rapid-prototyping* multi-agent testbeds while not loosing sight of the general applicability.

1 INTRODUCTION

New developments in *Virtual Reality* (VR) have brought a qualitative change to human-computer interaction, in the form of *co-habited virtual worlds* (CHVW). In such worlds, synthetic *agents* and *avatars* (agents that are controlled and supervised by human users) interact in a globally networked setting. Applications of CHVW are, for example, virtual conferences where lifelikeness and interaction of avatars is a key issue. One could also think of virtual marketplaces incorporating electronic salesmen agents and customer avatars as a platform for future *e-commerce*. Similarly, the entertainment issue of interactive, virtual theatre requires mixed populations of synthetic characters and half-human, half-computationally steered personae.

In order to build reasonable CHVW, the three strands of technology associated with VR, *Distributed Interactive Simulation* (DIS), and *Distributed Artificial Intelligence* (DAI) have to contribute. DAI caters for the computational means to build (semi-)autonomous, interacting agent programs which are situated in an environment via well-defined *perception* and *action* channels. DIS researches the modelling of the (dynamic, kinematic) evolvement of such an environment, including "virtual bodies" for agents. Finally, VR techniques deliver an integrative frame for visualising the simulated world and interfacing human users.

DAI is not only providing methods to build CHVW, but is also, as a science, a potential user of CHVW because of requiring an intuitive framework for complex, distributed simulations. For example, issues of adaptivity and scalability in *multi-agent systems* (MAS) cannot to be investigated without the use of large-scale simulations. Using a human-oriented, realistic model of the environment allows a suitable embodiment for evaluating *intelligent agents*. Indeed, other sciences, such as the social sciences, are as well becoming aware of the usefulness of co-habited computer simulations. A common architectural framework for CHVW is thus reasonable and must satisfy the following requirements:

Multiple Human Users CHVW should allow multiple users to survey and influence the state of the virtual world. An intuitive visualisation as well as a corresponding user interface is required. From an observer's standpoint, user-controlled avatars and agents should not be distinguishable. From a user's perspective, an avatar should be semi-autonomous, i.e., its lifelike low-level behaviour should be guidable by frequent user adjustments.

Broad Platform In general, CHVW become the more attractive the more people they can reach, the more existing systems they can interface. Common and open standards, such as those based on JAVA™, RMI, VRML'97, JAVA3D, and CORBA are the key to connect users working on heterogeneous platforms and to connect agents built with differing tools and languages. Although the mentioned standards are not immediately designed with respect to the demands of CHVW, the widespread interest into them by software companies and researchers will result in a similar effectiveness than home-grown, special-purpose solutions.

*supported by the "Deutsche Forschungsgemeinschaft" (DFG).
†supported by Siemens AG, Germany

Autonomy and Transparent Distribution Agents (avatars) are autonomous (semi-autonomous), i.e., they are able to freely compute their activity (in given bounds). In AI, this is usually described by the mutual interplay of two well-defined kinds of interfaces [15]:

> *An agent is an entity that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.*

Hereby, the virtual world is responsible for "outputting" on the sensor interface as being fed by effector data. In turn, the effector data is independently derived by the agent, i.e., in a different computational background, from "inputting" the sensor data. When applied to a networked setting, this notion allows agents and avatars to safely reside and decide on local computers while perception and action are transparent mediators to the central, global world.

Fine-grained Action and Perception Much often, pragmatic, visualisation-oriented simulations provide every agent with the same abstract (qualitative) sensor and effector interfaces. This means that perception already incorporates high-level information of which a commonly agreed semantics is seldom to be found in an open context. Similarly, actions are coarse patterns which denote complex, but uncontrollable behaviour in the virtual world. Especially for evaluating intelligent agents in a scientific context, but also for creating more lifelike scenarios, different quantitative modes of perception and action, such as touching, smelling, looking, communicating, etc., should allow the agent to classify objects and outcomes of actions from its personal, characteristic perspective in order to choose from a richer space of behaviour. Realistic embodiment requires various of those "non-labelled" methods of sensing and acting at the same time.

Rapid Prototyping What makes virtual worlds so attractive at all stages of scientific research and development , is the possibility for *rapid-prototyping*. This is because the granularity of the world is not determined in advance, thus the designer can incrementally refine the level of abstraction and thus gradually approximate real-world characteristics, such as continuity, dynamics, non-determinism, and inaccessibility, if needed. A CHVW architecture should support this process.

In this paper, the *Effector-Medium-Sensor* (EMS) architecture is proposed (Section 2). EMS has been designed to build CHVW with a special focus on DAI simulation. It meets the above criteria by introducing three architectural levels of *engine*, *view*, and *agent* which are interconnected by a method of *virtually acting and sensing*. For each of those levels of the architecture, we will discuss some implementation issues of our particular realisation of EMS, the *Social Interaction Framework* (SIF) [5] Sections 2.1–2.3). SIF is used to build several multi-agent testbeds, such as an automated loading dock. From these experiences, we conclude in Section 3 with future desiderata of CHVW and EMS.

Related Work Whereas the VRML'97 specification [19] focuses on animated, single-user 3D visualisation, the successful use of (proprietary) 3D-simulations in agent research has been shown by [17] and [1]. Similarly platform-dependent and non-interactive has been one of the first "uncaging" of autonomous agents in shared virtual worlds, [13], who modeled flocks of birds. Animated worlds present the problem of synthetic perception, where agents process the input they receive in the virtual world [18]. There is a strong and interesting link to research towards generic interaction languages [6].

Fisher's Menagerie [4] and Maes' ALIVE [9] introduce user interactivity via head-mounted displays or even full-body, wireless connections by video projection. The term *co-habited realities* and its application as a social interaction and communication tool is described in [20]. [12] emphasises the importance of virtual worlds in human-human interaction (*tele-presence* [8]); [10] takes an human-computer interaction perspective towards *interactive drama*. With respect to the history of VR, [14] gives a thorough overview on the mentioned threads. Networked simulation is a well-researched topic in *Distributed Interactive Simulation* (DIS) [3]. Its techniques for building complex worlds should guide future extensions of EMS.

2 THE EMS ARCHITECTURE

The *Effector-Medium-Sensor* (EMS, Figure 1) architecture stems from our efforts in building a generic toolkit for multi-agent simulations. Thus it starts with the aforementioned, minimal definition of [15]: In EMS, an *agent* is autonomously interacting via input *sensors* and output *effectors* with its environment, the simulation *engine*. The agent meets a decision as being *notified* by its sensors and by *activating* its effectors. The engine transfers effector activity back into sensor stimuli.

The first observation in EMS extending this simple picture is that, viewed from a particular level of abstraction, the environment offers several, independent channels of transfer — *media* — corresponding to different types of sensors and effectors: For example, agents have "mouth" effectors and "ear" sensors for verbal communication which is transmitted through the "air". This is independent of, e.g., emitting and detecting "infra-red light" and outpouring "odours" stimulating a "smelling" sense. An agent is characteristically equipped with a particular set of sensors and effectors which are able to interface just their respective media. Thus, each agent interacts only with a particular subset of the simulated world. For example, if an agent lacks an aural sense, it is not able to hear anything, but still able to behave reasonable. In turn, if the simulated world does not support a particular medium, e.g., it represents a vacuum and lacks "air", the respective agent interfaces, e.g., communication, are disabled.

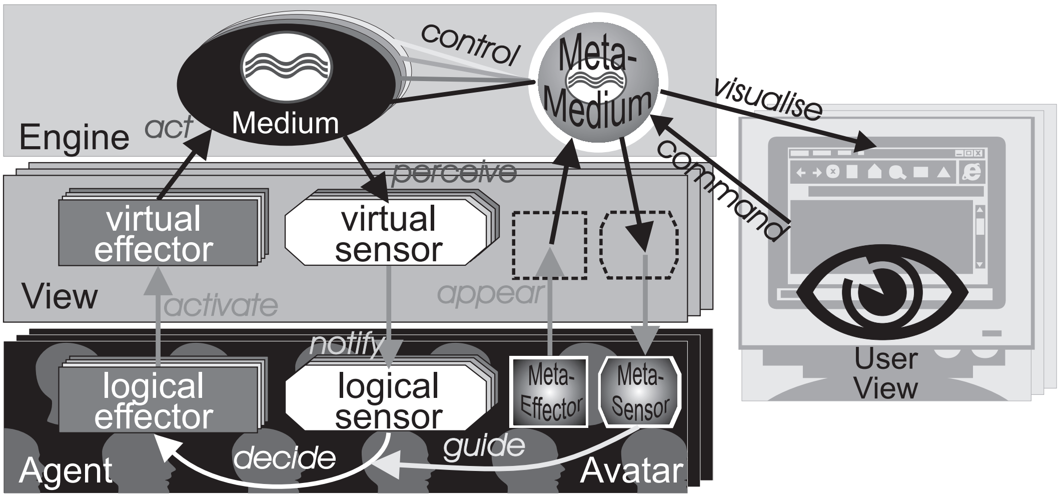

Figure 1: The EMS Architecture

The second observation of EMS moves away from regarding sensors and effectors as immediate windows from the external world state into the internal agent state. Both device types, such as mouth and eye, rather are themselves interpreting information; they comprise the *view* of an agent. For example, aural percepts by which an agent is notified normally become more and more noisy with increasing distance to the utterer. The degree of noisiness hereby depends on characteristics, i.e., the quality of the agent's ear. For this purpose, EMS-sensors and -effectors are split into a so-called *virtual* and a *logical* part. The virtual part interfaces the engine and serves as a preprocessing function. The logical part is tied to the agent and establishes a connection to its virtual complement.

The separation of media, views, and agents has been mainly designed in order to model perception and action in a more fine-grained manner. However, it also fulfils additional requirements of CHVW: the networked setting, rapid-prototyping of environments, and the multiple-user interface requirement. Views implement the perspective of an agent to the world locally. Therefore, the media are relieved from this computational burden. Virtual sensors and effectors can be easily distributed to the computation devices running the agents. This done by introducing open, medium-specific network interfaces rather than providing restricted, agent-specific connection methods.

EMS also provides the possibility to refine the complexity of the virtual world without having to change the agent implementation — a rapid-prototyping aspect. Refinement in modeling the medium in its continuity, dynamics, and nondeterminism, and the transition from perfect to imperfect sensors and effectors within the view are modular means of scaling the simulation complexity towards real-world aspects.

EMS also matches very well the special role of human users in CHVW. A user interface can be regarded as the view of a human in which a 2D/3D-browser serves as a virtual sensor (the display) and at the same time as an effector, e.g., by translating a mouse-click into a command for the world. However, as long as the full embodiment of humans, i.e., the immediate coupling of human senses to the computer, is not realisable or is too expensive, it is necessary to devise special agents, or avatars, to represent the users in the virtual world. Avatars are autonomous agents in that they (lifelikely) react to synthetic sensations. They are semi-autonomous in the sense that they are supervised or guided by humans.

Supervision by humans, thus their special 2D/3D-perspective onto the world, is indeed a "meta"-functionality provided by the so-called *meta-medium*. On the one hand, the meta-medium comprises data from all object-media and delivers them to the user view for visualisation and interaction purposes. Actions of the human user, on the other hand, are not object-level actions, but guidelines for their avatars or even commands for controlling the overall simulation. The meta-medium manages these commands and transmits behaviour guidelines to the avatars through *meta-sensors*. Agents and avatars also own a *meta-effector* to transmit their appearances, i.e., visualisation data, to the meta-medium at startup.

In the following, we will discuss the different architectural levels of engine (Section 2.1), view (Section 2.2),

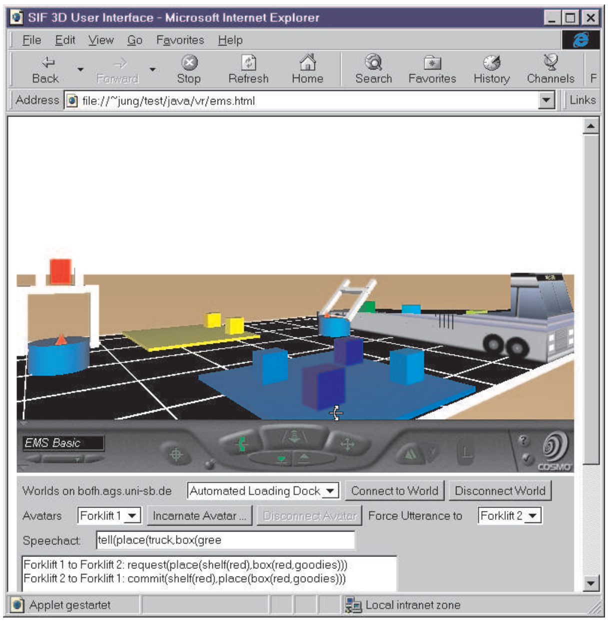

Figure 2: User View: An Automated Loading Dock

and agent (Section 2.3) more detailed at hand of our current implementation, the *Social Interaction Framework* (SIF). SIF is a toolkit to build CHVW and bases on common standards, such as JAVA™, RMI, VRML'97, and CORBA in order to make it a globally networked and broad platform. While some parts of SIF, such as network code, 3D-user interface, and advanced types of media, are not yet fully completed, the tool is already functional to a wide extent. We are currently realising several multi-agent testbeds, amongst them an automated loading dock (Figure 2) in which forklift robots transport and exchange boxes of different categories from (to) a truck to (from) the appropriate shelves.

2.1 Engine

The SIF-engine and embedded media are implemented as a library of JAVA™-classes. A particular virtual world is described in a *script file* which determines a selection of media and constrains their initial state. At startup, the script file is interpreted by the engine and appropriate instances of media are constructed. For the duration of the simulation, the engine dynamically manages the connections to the views (Section 2.2).

Media are derived from a generic medium class; there are already predefined media available, such as the *air* for communication, the *grid* for a discrete 2D-geometry, and the *meta-medium* for incorporating human user access. This set is currently extended, e.g., to support continuous sceneries, such as the automated loading dock.

Each medium contains an *action manager* which registers and unregisters virtual effectors, queues incoming action requests, and elaborates the queue by changing the

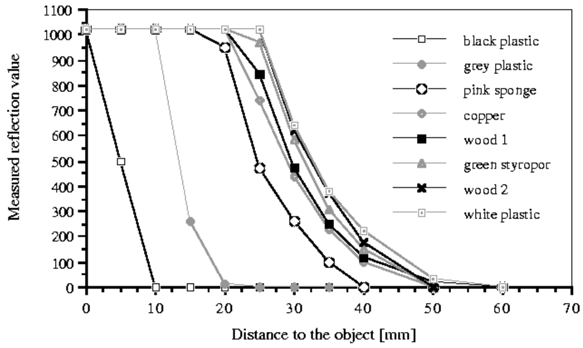

Figure 3: Virtual Infra-Red Sensor

medium state accordingly. The medium state can be regarded as a set of *features* with dynamically changing *values*. Depending on the medium, value updates can be local, such as in the case of an agent changing its position in the *grid*, they can also be more global, in the case of simulating fine-grained kinematics.

Changing values trigger the *sensor manager* which generates new percepts and sends them to registered virtual sensors. Since not all features are accessible this way and since we allow for randomisation in state transitions, actions are seemingly (for the agents) non-deterministic. Note that action manager, state transition, and sensor manager operate asynchronously in order not to block the autonomous computations in the views and the agents.

The meta-medium plays an outstanding role as its state controls the operation of all other media in order to maintain 2D/3D-visualisation information which updates the users' virtual sensing, i.e., connected graphical browsers and interaction tools. In turn, effector commands from user tools have to do with controlling the engine, e.g., by starting and stopping particular media. User actions also serve to control the behaviour of avatars. Depending on user permissions managed by the meta-medium, the *meta-sensor-manager* sends guidelines to avatar's virtual meta-sensors. The appearances of agents in user views are given to the meta-medium through virtual meta-effectors when agents initialise.

2.2 View

Agent & Avatar View The agent and avatar part of the SIF view is a library of JAVA™-classes for virtual effectors and sensors according to the media already implemented in the engine. The view itself is a facility to create and discard particular virtual effectors and sensors on request of the agent. Virtual sensors can be regarded as interpretation functions (Figure 3 shows an example) which convert method calls by the medium's sensor-manager (containing feature-value pairs, e.g., material-blackplastic distance-4.0) into method calls for the agent's logical sensors (transmitting sensor *readings*, e.g., 600). Comple-

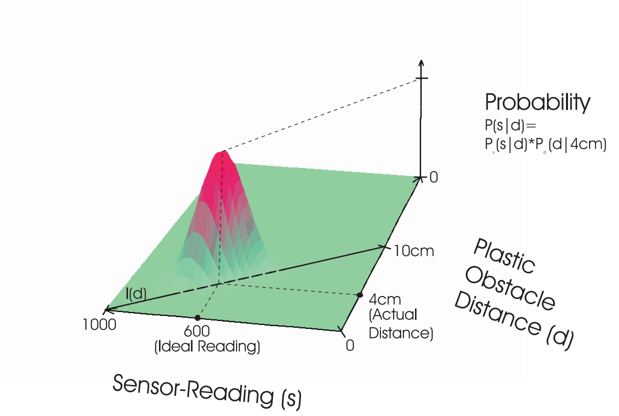

Figure 4: Noisy Virtual Sensor

mentary, virtual effectors inter-operate with logical effectors and the medium's action manager. Virtual meta-sensors/meta-effectors are just canonical mappings of visualisation data and commands (see Sections 2.1 and 2.3).

On instantiation, virtual effectors and sensors connect to the engine via *Remote Method Invocation* (RMI). This is a JAVA™-specific distribution facility to transparently invoke methods of remote objects. The engine routes virtual sensors and effectors to the managers of the respective medium (if it exists), i.e., view and medium now operate in a bidirectional fashion (sensor-manager to sensor, action-manager to effector) as so-called *stationary remote objects*. If there is no corresponding medium in the engine, virtual effectors and sensors turn into dummies, i.e., they are "muting" incoming method calls.

The rapid-prototyping aspect of incrementally refining the noisiness of a view is demonstrated in Figure 4. Without having to change medium or agent, the perfect sensor specification of Figure 3 delivers a probability distribution to simulate imperfect readings.

User View SIF is equipped with two optional user views in 2D and 3D. Both interface the meta-medium via RMI. The 2D version is purely JAVA™-*Abstract Window Toolkit*-based and is suitable for grid-like worlds. The 3D-version (see Figure 2) runs in standard WWW-browsers using VRML'97 technology. The *Virtual Reality Modeling Language* was proposed as a standard for describing single-user, animated 3D worlds focusing on authorability, composability, extensibility, and performance. Basic building blocks of VRML'97 are hierarchically organised 3D-objects, or *nodes*, through which activity, *events*, is routed. VRML'97 is interactive in that special *sensor nodes* coupled to user navigation and pointing operate as event sources. The multi-user aspect and the network connectivity that we require for SIF can be integrated into *script nodes* and the *External Authoring Interface*. They allow to extendedly control the VRML'97-scene via an attached JAVA™-machine.

The SIF-VRML'97-browser is given 3D-descriptions of world and agents by the meta-medium in the form of *Uniform Resource Locations* (URL's)[1] to distributed VRML'97-files. These files are stored near the code of their corresponding agent on a local computer of their user, accessible over the web from remote user views. VRML'97-files not only contain pure 3D-geometries, but also the complete animation code (including sound) to visualise particular actions of an agent. Each SIF-VRML'97-browser is independently able to manage and synchronise these animations being notified by the meta-medium at simulation time. This means that there is no large-bandwidth transmission of whole 3D-animations, rather a small-bandwidth transmission of interrelated animation triggers. The SIF-library provides JAVA™-classes with which existing VRML'97-descriptions and animations can be easily made compatible to the SIF-browser.

2.3 Agent

Agents are characters, i.e., they exhibit personalities in making particular decisions from given stimuli. This decision making process can be implemented on various, heterogeneous platforms depending on the envisaged functionality. Typically, simple or communicating agents are written in standard imperative languages, e.g., C or JAVA™. Intelligent agents often use declarative AI languages, e.g., Lisp or PROLOG. This set of agent platforms constantly increases.

For EMS-agents, the set of their abilities in the form of a selection of sensors and effectors is also characteristic. Because of the increasing variety of agent platforms, we cannot provide implementations of all sensor and effector types. In order to retain an open solution, the view and virtual sensors/effectors are speaking a platform-independent language, in our case CORBA[2] [7], and provide particular services in that language. For example, the view offers to create and discard virtual effectors and sensors from a range of types. Virtual effectors provide an activation service based on typical CORBA-data structures and depending on the intended medium, e.g., the virtual meta-effector deals with URL's of JAVA™- and VRML'97-code.

Thus the agent designer can easily implement his own logical effectors and agents by using simple CORBA-calls; his logical sensors just have to identify themselves as a tiny notification service. For JAVA™, we are already providing a library of agents and logical devices according to the available media. In the near future, we will also cover intelligent INTERRAP-agents [11] to describe reactive, planned, and social motivational behaviour [2]. They are based on the state-of-the-art AI language OZ [16].

[1] The meta-medium itself gets URL-references by meta-effector's and the script file — see Sections 2.1 and 2.3

[2] Actually, CORBA is not a language, but a ISO/IEC-standardised platform-independent distribution facility quite like RMI is for JAVA™.

3 CONCLUSION

We have presented an architecture for co-habited virtual worlds, EMS, and its implementation, SIF, as a broad and reasonable simulation framework for multi-agent systems. The key design aspects have been the fine-grained virtually acting and sensing, the ability of rapid-prototyping simulated worlds, a transparent network access, a broad implementation platform, and the possibility of supervising avatars by multiple human users. The testbeds we are currently developing with our tool, for example the automated loading dock, confirm the applicability of our approach. We will come up with a fully implemented system in the near future.

Future work will focus on how to integrate the important capability of *migration* into EMS (Section 3.1) and how to gain performance from the distributed architecture given the possibilities of VRML'97 and JAVA^{3D}(Section 3.2). An open question remains a standard *interlingua* between human and synthetic agents.

3.1 Migration

Migration has two meanings when applied to CHVW. On the one hand, it denotes the possibility of mobile agent computations. This is already supported by RMI and CORBA, thus EMS. All that is necessary is to turn stationary objects within agent and view into mobile objects.

On the other hand, migration also denotes agents/avatars/users which migrate between virtual worlds, e.g., connected by doors. This will have a greater impact onto EMS, since multiple meta-media in different engines need to be able to coordinate in order to remove an agent in a source world, build up a clone with a coherent state (e.g., its velocity) in the destination world, and link its view to the new media.

3.2 Distributed Simulation

Media are *per definitionem* independent dimensions of the virtual world, thus an ideal candidate for gaining performance by further distribution. A particular problem of this definition is however posed by inherent dependencies between features regarding realistic simulations: the engine could collapse into a single medium as a single dynamic and kinematic equation. DIS research shows that an appropriate separation of interacting parts of the simulation can be given for real-world models. The notion of media should be changed accordingly in the future.

An interesting issue is also the full use of VRML'97- and JAVA^{3D}-capabilities[3]. The event-routing model could be used to efficiently implement a part of the functionality of the medium distributed to the local computers of the users. We will investigate how a consistent, global state can nevertheless be retained by reconfiguring the simulation topology.

[3] Though being similar, JAVA^{3D} is probably the more promising standard since the strategical reorientation of Silicon Graphics.

References

[1] S. Benford, J. Bowers, L. Fahlén, and C. Greenhalgh. Managing multual awareness in collaborative virtual environments. *Virtual Reality Software & Technology*, 1994.

[2] A. Burt. Modelling Motivational Behaviour in Intelligent Agents in Virtual Worlds. In *Proc. of the 1998 Conference on Virtual Worlds and Simulation*, 1998.

[3] IEEE Standards Department. Ieee standard for distributed interactive simulation — application protocols. Technical report, Piscataway New Jersey, 1994. Standards Proposal P1278.1 ballot draft.

[4] S. S. Fisher, S. Amkraut, M. Girard, and M. Trayle. Menagerie: Designing a virtual experience, stereoscopic displays and virtual reality systems. In S. S. Fisher, M. T. Bolas, and J. O. Merritt, editors, *Stereoscopic Displays and Virtual Reality Systems*. 1994.

[5] P. Funk, C. Gerber, J. Lind, and M. Schillo. SIF: An Agent-Based Simulation ToolBox using the EMS Paradigm. In *Proceedings of the 3rd International Congress of the Federation of EUROpean SIMulation Societies (EuroSim)*, 1998.

[6] D. Gelernter and N. Carriero. Coordination languages and their significance. *Communications of the ACM*, 35(2):97–107, 1992.

[7] The Object Management Group. The common object request broker architecture. http://www.corba.org/.

[8] R. M. Held and N.T.Durlach. Telepresence. *Presence: Teleoperators and Virtual Environments*, 1:162–174.

[9] P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE system: Wireless, full-body interaction with autonomous agents. *ACM Multimedia Systems: Special Issue on Multimedia and Multisensory Virtual Worlds*, Spring, 1996.

[10] K. Meyer. Dramatic narrative in virtual reality. *Communication in the Age of Virtual Reality*, 1995.

[11] J. P. Müller. *The Design of Intelligent Agents: A Layered Approach*, volume 1177 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, December 1996.

[12] M. T. Palmer. Interpersonal communication and virtual reality: Mediating interpersonal relationships. *Communication in the Age of Virtual Reality*, 1995.

[13] C. W. Reynolds. Flocks, schools and herds: A distibuted behavioral model. In *Proceedings of SIGGRAPH'87*, 1987.

[14] H. Rheingold. *Virtual Reality*. Summit, New York.

[15] S J. Russell and P. Norvig. *Artificial Intelligence, A Modern Approach*. Prentice Hall, 1995.

[16] G. Smolka. The Oz Programming Model. In Jan van Leeuwen, editor, *Computer Science Today*, Lecture Notes in Computer Science, vol. 1000, pages 324–343. Springer-Verlag, Berlin, 1995.

[17] S. Strippgen. Insight: A virtual laboratory for looking into behavior-based autonomous agents. In W .L. Johnson, editor, *Proc. of the First International Conference on Autonomous Agents*, pages 474–475, Marina del Rey, CA USA, February 1997. ACM Press.

[18] D. Terzopoulos, T. Rabie, and R. Grzeszczuk. Perception and learning in artificial animals. In *Artificial Life V: Proc. Fifth Inter. Conf. on the Synthesis and the Simulation of Living Systems*, 1996.

[19] The VRML consortium. *The Virtual Reality Modeling Language, ISO/IEC DIS 14772-1*. 1997. http://vrml.sgi.com/moving-worlds/spec.DIS/index.html.

[20] Walter Van de Velde. Co-habited mixed realities. In Fumio Hattori, editor, *Proceedings of the IJCAI'97 workshop on Social Interaction and Communityware*, Nagoya, Japan, 1997.