Semantic Modelling for Virtual Worlds
A Novel Paradigm for Realtime Interactive Systems?

Marc Erich Latoschik*
FHTW Berlin
Germany

Roland Blach†
Fraunhofer IAO, Stuttgart
Germany

1 Introduction

Engineering systems plays a central role in the development of successful Virtual Reality (VR) and Augmented Reality (AR) applications. Increasing computational resources are utilized to build increasingly complex artificial environments and extensive Human-Computer Interaction (HCI) systems. These types of Realtime Interactive Systems (RIS) establish a closed HCI loop. They are characterized as systems continuously analyzing users' input while concurrently synthesizing appropriate output for several of the human senses in real-time.

The **expectation** is that such systems will significantly improve HCI since they provide *natural* modes of operation, e.g., using artificial agents as cooperation partners or systems' proxies. Hence, such systems offer the potential to reduce the cognitive workload of users during systems' operations. They use their computational resources to adapt to the human interaction modes instead of forcing users to adapt to theirs. Such novel forms of HCI require a strong conceptual and technical base.

Research on basic AR/VR and HCI principles, as well as of cognitive capabilities and cognitive behavior is a multidisciplinary endeavor currently considered a *hot topic*. But the underlying **complexity** of building systems which are capable of handling this challenge is manifold. The state-of-the-art in this field is rapidly **changing**, constantly creating new results, insights, and models from behavior science, cognitive science, psychology, neuro sciences, linguistics, artificial intelligence (AI) and computer science in general–to name just a few.

Still, despite its complexities and challenges, RIS development is sometimes not even considered as science. In contrast, some–most often the initiated, the engineers themselves–consider it as an engineering art or sophisticated craftmanship. The success of building RIS often depends on the implicit individual competences and expert knowledge. We believe that both is true. While building complex RIS certainly requires artistic competences, it is at least partly due to a lack of scientific methods applied to the field. That is, the methods and techniques require classification, analysis and evaluation to compare them to the advances of software engineering in computer science in general, and the specific AR/VR requirements in particular.

*e-mail:Marc.Latoschik@fhtw-berlin.de
†e-mail: Roland.Blach@iao.fhg.de

2 Software Engineering

Application design is at the heart of the software engineering process. Multiple paradigms, principles, and methods exist to keep up with the increasing challenges during system design. Modularization, functional decomposition, object oriented programming, pattern oriented design, waterfall process, rapid prototyping, or extreme programming — to name a few — all tackle the rising complexity of today's software which is considered the main reason for the so-called *software crisis* as initially coined by F. L. Bauer in 1968 and later noted by Dijkstra [Dijkstra 1972]:

"[The major cause of the software crisis is] that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem."

The overall problem is approached from two directions which interact with each other: From the one side, project management sets the organizational frame. From the other side, the project engineering task identifies appropriate methods and techniques to be applied to the different problems and tasks which arise during software design. Clearly, choices made from one side directly influence the other side even if external constraints do not vary or change, an assumption which almost never holds in real world situations. Software engineering tries to cope with that problem by defining a set of different concepts, methods, and techniques dynamically applicable to varying tasks and changing constraints. But – the sometimes quite novel – methods put into practice always seem to slightly lag behind the growing overall complexity of nowadays systems driven by high expectations in a changing environment.

3 Realtime-Interactive Systems

Visual perception is considered a primary human sense. Therefore multiple AR/VR or in general Mixed Reality (MR) systems were often based primarily on interactive graphics systems. Dating back to early work by [Sutherland 1968] on a head mounted display with hidden line graphics, the majority of interactive graphics systems tend to center around the graphics generation and representation.

Initial approaches for interactive graphics encompass work on light-material interaction, rendering, and shading algorithms on the one side and on attendant application programming interfaces (API) and design metaphors to make them available to application developers on the other side. GKS, OpenGL®, DirectX®, or current shader languages like OpenGL's GLSL, HLSL by Microsoft®, or Cg by NVIDIA are typical examples for basic rendering and shading APIs. The low-level graphics APIs were augmented by high-level spatial data structures, the **scene graphs**. Tools like OpenGL

Performer™, Open Inventor, Open Scene Graph, OpenSG [Reiners et al. 2002], or X3D still center around an hierarchical scene structure. They additionally provide performance optimizations, e.g., for picking, culling or state sorting.

Purpose-built VR development tools extend these concepts with VR-specific key features: First, input/output device customizability and embedding [Preddy and Nance 2002] is mandatory, see, e.g., AVANGO [Tramberend 1999], Lightning [Blach et al. 1998], VR Juggler [Bierbaum et al. 2001] or commercially available ones like the WorldViz™or EonReality™. Second, network distribution features are commonly integrated, e.g., in AVANGO [Tramberend 1999], MASSIVE3 [Greenhalgh et al. 2000], DIVE [Hagsand 1996; Steed and Frécon 2004] or Net Juggler. They either allow distributed rendering on cluster architectures, hence again output device support, or to develop shared virtual environments. Third, application programmers often require an entity centered access to world states or world logic which is often realized using specific event mechanisms as e.g. data-flow.

Descending from a different line of research, a principle found in Intelligent Virtual Environments [Luck and Aylett 2000] (IVEs) is a semantic representation of scene content [Soto 1997; Soto and Allongue 2002; Peters and Shrobe 2003; Latoschik and Schilling 2003; Kalogerakis et al. 2006; Lugrin and Cavazza 2007] to provide knowledge driven access to the scenes' entities. A semantic representation of scene content or — more general — of applications' entities has proven to be a necessity for several kinds of intelligent applications. That is, applications which provide computer simulated intelligent behavior, e.g., by incorporating AI methods. It is specifically true for systems which simulate human like capabilities in one or the other way. Typical examples include multimodal interactions, e.g., for the simulation of human like agents, or systems which surrogate simulation aspects using knowledge based approaches. The requirement of a semantic content representation is also the core idea of next generation's web efforts [Berners-Lee et al. 2001].

Semantic models have also gained interest in object oriented programming paradigms [Meseguer and Talcott 2002] as an abstract description for object reflection. On a fine grained implementation level, reflection provides support for extensible and portable software designs, e.g. for dynamic programming approaches. Reflection provides meta-access to an object's API during runtime which enables calling objects to automatically query target objects' capabilities and adjust to their interfaces. The combination of reflection based on semantic models has lately been explored as a novel development paradigm for RIS architectures [Latoschik and Fröhlich 2007].

4 Issues

Scene and data propagation graphs as provided by interactive graphics systems initially seem to offer useful abstractions layers. Complete simulation applications can be build around the graph structures utilizing node inheritance, routing, and scripting methods. While well motivated in the beginning, such designs lead to close couplings between the applications' content and the specific design tools. This is a source of several drawbacks now known for some time [Bethel et al. 1999; Arnaud and Jones 1999].

Intelligent Interactive Systems ask for far more than just image generation and basic input handling. They require diverse simulation aspects which — from the engineering point of view — will depend on the integration of various additional software modules [Kapolka et al. 2002]. A simple decomposition into stages for input processing, internal "logic", and output generation seems inadequate. For example, input processing can range from simple WIMP interfaces to full body movement tracking and gesture analysis as well as to speech understanding and multimodal interpretation. Internal "logic" may be based on simple decision trees but may as well incorporate large-scale physics simulation or advanced AI methods. Output generation may encompass stimuli for all kinds of human senses, from visual input, to sounds, tactile feedback, or even smell etc.

Technically, these aspects can not be separated following a simple stage decomposition since they overlap. For example, a module for decision making may be required during output generation, a module for physics calculation may be required for sound simulation or agent animation, a module for AI-based classification may be required for input processing and gesture analysis. Hence, we will have to cope with various modules which have to be integrated into a continuous process and data flow to produce coherent user experiences.

Two different integration approaches can be identified: Modules are either included on a case by case base, or they are integrated a priori into holistic architectures as found in many 3D game engines like the Doom 3 Engine, the Unreal Engine 3, the Source Engine, the C4 Engine or the CryENGINE™. Both, the case-by-case as well as the holistic architecture approaches, have their drawbacks when it comes to application customization and long-term reusability, persistence and portability. The first one requires a deep understanding about the internal algorithms and data structures of the utilized tool. It requires extensive low-level implementation efforts to customize or exchange a specific module, e.g., if a certain software library is no longer available or if it is not available on a given operating system. In contrast to that, holistic approaches often don't support extensive modifications to the underlying base system at all.

These problems are well recognized and they are not specific to RIS developments. In fact, they are precisely caused by the same characteristics as the roots of the software crisis in software engineering: complexity, expectation, and change. The underlying technical nature of RISs just makes them highly prototypical candidates for such engineering problems.

5 Panel Discussion

Semantic techniques, ontologies, or semantic models so far only had partial impact on RIS development. IVEs require semantic description of scene content, but the technical base architectures have not benefited from semantic techniques to a higher degree. Hence, basing the development of real-time interacticve applications on semantic virtual worlds might turn out to be a new devopment paradigm to master the complexity. The core method, semantic modeling, reflects ongoing advances in the area of software engineering, i.e. current object oriented programming approaches.

The panel will give different developers and researchers the opportunity to present his/her viewpoint on the following issues:

- Are we ready for more semantics (content, system or both)?
- What are the costs (performance, learning curve, etc.)?
- Do we need a common language and if so, how to establish?
- What are the possible limits?
- Who will pay for it?

With this panel we also want to facilitate the discussion and exchange between academia and industry. We think that for RIS engineering a strong link to real world application is necessary. Serious complex applications can not completely be emulated in research

labs and driven by research projects, i.e., projects which do not emphasize RIS engineering as a central topic.

Acknowledgements

The **S**oftware **E**ngineering and **A**rchitectures for **R**ealtime **I**nteractive **S**ystems (SEARIS) group as well as the participants of the first workshop at IEEE VR 2008 in Reno and the 2008 Dagstuhl seminar *Virtual Realities* for providing an exchange platform, foster intensive and fruitful debates, and ensuring us to continue the discussion on realtime interactive systems' engineering.

6 Organizers

Marc Erich Latoschik

Marc's research area are Intelligent Virtual Environments (IVEs). Interdisciplinary devoted to Artificial Intelligence, real-time 3D computer graphics and Cognitive Sciences, he works on game development, Virtual Reality, multimedia and simulation, language and gesture processing, and novel human-computer interaction methods. Marc studied mathematics and computer sciences at the University of Paderborn, the New York Institute of Technology and the Bielefeld University. He received his PhD in 2001 in the area of multimodal–gesture and speech–interaction for Virtual Reality. He headed the AI & VR Lab of the AI group at the Bielefeld University until 2007 and is now a professor for media informatics at the University of Applied Sciences (FHTW) in Berlin.

Roland Blach

Roland is a senior scientist at the Competence Centre for Virtual Environments of Fraunhofer IAO in Stuttgart. He finished his studies of control engineering at the University of Stuttgart in 1992 and works in the field of VR since 1993. He is one of the architects of the VR system Lightning and has participated in many industrial and research projects. His research interests are software architectures for interactive realtime systems, 3D interaction, projection based display systems and immersive information vizualization. He actively supports the scientific exchange as organizer and reviewer for major VR related events, e.g., for EGVE, IEEE VR, or the IEEE 3DUI Symposium. He is the responsible technical manager of the european INTUITION network of excellence on VR/AR technology and for the network's VR-technology working group.

References

ABRAN, A., MOORE, J. W., BOURQUE, P., AND DUPUIS, R., Eds. 2005. *Guide to the Software Engineering Body of Knowledge - SWEBOK*. IEEE Computer Society.

ALLARD, J., GOURANTON, V., LECOINTRE, L., LIMET, S., MELIN, E., RAFFIN, B., AND ROBERT, S. 2004. Flowvr: a middleware for large scale virtual reality applications. In *Proceedings of Euro-par 2004*.

ARNAUD, R., AND JONES, M. T. 1999. Innovative software architecture for real-time image generation. In *Proceedings of the I/ITSEC Conference*.

BERNERS-LEE, T., HENDLER, J., AND LASSILA, O. 2001. The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American 284*, 5, 34–43.

BETHEL, W., BASS, C., CLAY, S. R., HOOK, B., JONES, M. T., SOWIZRAL, H., AND VAN DAM, A. 1999. Scene graph apis: wired or tired? In *SIGGRAPH '99: ACM SIGGRAPH 99 Conference abstracts and applications*, ACM Press, New York, NY, USA, 136–138.

BIERBAUM, A. D., JUST, C., HARTLING, P., MEINERT, K., BAKER, A., AND CRUZ-NEIRA, C. 2001. VR Juggler: A Virtual Platform for Virtual Reality Application Development virtual platform for virtual reality application development. In *IEEE Virtual Reality 2001 conference proceedings*, IEEE Press, Yokohama, Japan, 89–96.

BLACH, R., LANDAUER, J., RSCH, A., AND SIMON, A. 1998. A Highly Flexible Virtual Reality System. In *Future Generation Computer Systems Special Issue on Virtual Environments*. Elsevier Amsterdam.

CALERO, C., RUIZ, F., AND PIATTINI, M. 2006. *Ontologies for Software Engineering and Technology*. Springer.

DIJKSTRA, E. W., 1972. The humble programmer.

GREENHALGH, C., PURBRICK, J., AND SNOWDON, D. 2000. Inside massive-3: flexible support for data consistency and world structuring. In *Proceedings of the third international conference on Collaborative virtual environments*, ACM Press, 119–127.

HAGSAND, O. 1996. Interactive MultiUser VEs in the DIVE system. *IEEE Multimedia Magazine 3*, 1.

ISO/IEC, JTC 1/SC 24. 1994. Graphical Kernel System for three dimensions (GKS-3D). Tech. Rep. 7942-1:1994, ISO/IEC.

ISO/IEC, JTC 1/SC 24. 2004. X3d abstract. Tech. Rep. 19775-1:2004, ISO/IEC.

KALOGERAKIS, E., CHRISTODOULAKIS, S., AND MOUMOUTZIS, N. 2006. Coupling ontologies with graphics content for knowledge driven visualization. In *Proceedings of the IEEE VR2006*, 43–50.

KAPOLKA, A., MCGREGOR, D., AND CAPPS, M. 2002. A unified component framework for dynamically extensible virtual environments. In *Fourth ACM International Conference on Collaborative Virtual Environments*.

LATOSCHIK, M. E., AND FRÖHLICH, C. 2007. Semantic Reflection for Intelligent Virtual Environments. In *Proceedings of the IEEE VR 2007*, 305–306.

LATOSCHIK, M. E., AND SCHILLING, M. 2003. Incorporating VR Databases into AI Knowledge Representations: A Framework for Intelligent Graphics Applications. In *Proceedings of the Sixth International Conference on Computer Graphics and Imaging*, ACTA Press, IASTED, 79–84.

LUCK, M., AND AYLETT, R. 2000. Applying Artificial Intelligence to Virtual Reality: Intelligent Virtual Environments. *Applied Artificial Intelligence 14*, 1, 3–32.

LUGRIN, J.-L., AND CAVAZZA, M. 2007. Making Sense of Virtual Environments: Action Representation, Grounding and Common Sense. In *Proceedings of the Intelligent User Interfaces IUI'07*.

MESEGUER, J., AND TALCOTT, C. 2002. Semantic models for distributed object reflection. In *ECOOP 2002 - Object-Oriented Programming: 16th European Conference Malaga*, Springer Berlin / Heidelberg, Lecture Notes in Computer Science, 1–36.

MOORE, G. 1965. Cramming more components onto integrated circuits. *Electronics 38*, 8.

PETERS, S., AND SHROBE, H. 2003. Using semantic networks for knowledge representation in an intelligent environment. In *PerCom '03: 1st Annual IEEE International Conference on Pervasive Computing and Communications*, IEEE, Ft. Worth, TX, USA.

PREDDY, S. M., AND NANCE, R. E. 2002. Key requirements for cave simulations: key requirements for cave simulations. In *WSC '02: Proceedings of the 34th conference on Winter simulation*, Winter Simulation Conference, 127–135.

REINERS, D., VOSS, G., AND BEHR, J., 2002. OpenSG: Basic Concepts. www.opensg.org/OpenSGPLUS/symposium/-Papers2002/Reiners_Basics.pdf, february.

ROHLF, J., AND HELMAN, J. 1994. Iris performer: a high performance multiprocessing toolkit for real-time 3d graphics. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 381–394.

SOTO, M., AND ALLONGUE, S. 2002. Modeling methods for reusable and interoperable virtual entities in multimedia virtual worlds. *Multimedia Tools Appl. 16*, 1-2, 161–177.

SOTO, M. 1997. Semantic approach of virtual worlds interoperability. In *Proceedings of IEEE WET-ICE '97, Cambridge, MA*, IEEE Press, M. Capps, Ed.

STEED, A., AND FRÉCON, E. 2004. Construction of collaborative virtual environments. In *Developing Future Interactive Systems*, M.-I. S. Segura, Ed., no. ISBN 1591404126. Idea Group, November, 235–268.

STRAUSS, P. S., AND CAREY, R. 1992. An object-oriented 3D graphics toolkit. In *Computer Graphics*, vol. 26 of *SIGGRAPH Proceedings*, 341–349.

SUTHERLAND, I. E. 1968. A head-mounted three-dimensional display. In *Proceeding of the Fall Joint Computer Conference. AFIPS Conference Proceedings.*, vol. 33 of *AFIPS*, 757–764.

TRAMBEREND, H. 1999. A distributed virtual reality framework. In *IEEE Virtual Reality Conference*, 14–21.