Communicating with Virtual Environments
A Survey of Recent Work at the University of Bielefeld

Ipke Wachsmuth, Britta Lenzmann, Bernhard Jung

Faculty of Technology, University of Bielefeld
ipke|britta|jung@techfak.uni-bielefeld.de

Abstract

This survey, compiled from our recent research reports, describes research accomplishments in interactive design and assembly with 3D computer graphics environments, carried out in the AI & Computer Graphics Lab at the University of Bielefeld. As a means of communicating with such environments, agent techniques and dynamic knowledge representations are used to process qualitative verbal instructions to quantitative scene changes. A key idea is to exploit situated 'perceptive' information by inspecting the computer graphics scene models.

CR Descriptors: I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods — *dynamic knowledge representation; hybrid representation; spatial reasoning*; I.3.2 [**Computer Graphics**]: Graphics Systems — *interactive graphics; virtual environments*; I.6.7 [**Simulation and Modeling**]: Simulation Support Systems — *virtual design; virtual assembly*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces — *intelligent user interfaces; interface agents; adaptive interfaces; situated verbal interaction*;

1 Motivation

The ubiquity of machine-generated virtual environments and the attractiveness of exploring complex virtual worlds by eye inspection have launched a great demand for sophisticated application. Computer-based presentations of synthetic geometry data, transformed to visual surface structures by way of rendering techniques, are of growing importance in the design and construction areas. To make better profit of this new technology, new ways of human-computer interaction are called for. Previous work has concentrated on walk-through type interaction and on interacting with the environment by direct manipulation, for instance, by using the data glove.

An alternative way we explore at the AI & Computer Graphics Lab is to use verbal instructions to communicate alterations. These are put in effect by a mediating system which changes the arrangement or assemblage of scene objects. Our general goal is to use AI to establish a communication link between humans and multimedia. In particular, our aim is to keep the user free from technical detail such as geometry planning in a responsive virtual environment. This is, however, possible only when the image of a design or an assembly is not a meaningless visual presentation, but is coupled to an internal semantic representation of the presented images.

We have concretized our approach in two examples of responsive virtual environments, one in virtual design and one in virtual assembly. It is our aim that user and system can communicate about a *dynamic* scene. To this end, we have developed dynamic knowledge representations to account for changing conceptualizations of objects and scene, and we use software interface agents which inspect graphic scene descriptions (geometry models, materials, etc.) and keep track of varying situation parameters.

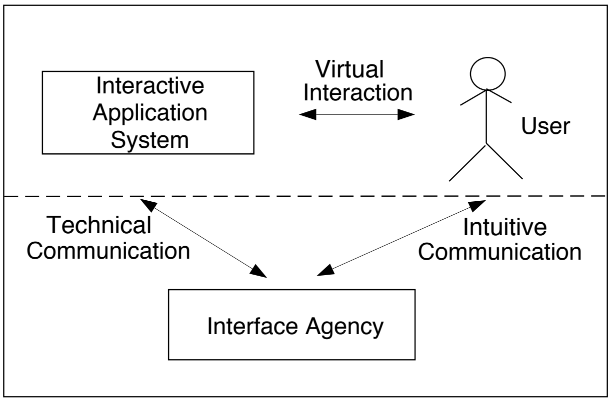

Figure 1: Agent-mediated interaction

The user can instruct the application system by way of abstract commands (virtual interaction) and an interface agency interprets them (intuitive communication) and transmits the results to the application system via technical communication (see figure 1). Qualitative communications of the user are evaluated so as to produce appropriate quantitative changes in the scene description which is then visualized for the user. Since the user perceives the scene by eye inspection, system and user can communicate about scene details from the same "point of view." We call this *situated verbal interaction* [CJW95].

Discourse in simple written natural language is used for communication. Very recently we have started to use voice input and simple directional hand gestures.

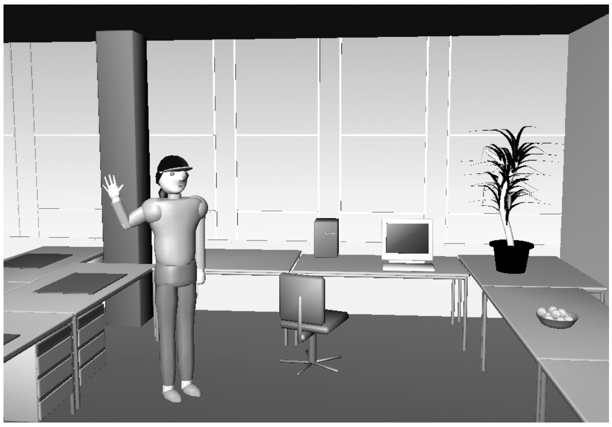

Figure 2: Anthropomorphic virtual interface agent

2 Virtual Design Environment

In the VIENA project ("Virtual Environments & Agents"), we have chosen interior design as an example domain. Instead of using the complex mouse-and-menu commands to manipulate objects we communicate with the system by way of natural language. We have developed a set of software interface agents, realized as autonomous Unix processes, which altogether form an intelligent mediator agency. Taking special responsibilities in processing verbal instructions, the VIENA agents cooperate with each other and with the user to offer a goal scene meeting the user's wants. For instance, a space agent translates qualitative relations such as left of to appropriate scene coordinates. The offer can be changed in further interaction, that is, semantics construction is situated in the visible scene and can be negotiated in discourse [WC95]; [WLC95]; [LWC95].

The VIENA system is tested in a prototype scenario with various items of furniture as well as color and light impressions of a virtual office room which can be changed interactively. As a further feature, models of individual objects can be exchanged; e.g., "real" CAD models of a kitchen manufacture company were imported to probe adaptability of our prototype system to more realistic conditions.

2.1 Anthropomorphic Interface Agent

In our ongoing work, we manipulate the virtual environment via an anthropomorphic virtual interface agent (VIA) which is present in the virtual scene (cf. figure 2). As an "embodied opposite" the VIA, named "Hamilton" in our scenario, is a means for enhanced and extended ways of interactive manipulation and exploration. Furthermore, it gives rise to more flexible ways of communication which allows utilization of various kinds of spatial reference schemes.

The agent can be instructed to change its position, look left/right, or to point to specified positions (like *"point to the palmtree"*), etc. By this, more natural ways of locating objects or positions are possible by referring to the position of the VIA in the scene, or by using pointing gestures that help to process otherwise underspecified deictic instructions like *"move the chair there"* (cf. figure 3). As a next step, we are planning to mouse-control the pointing arm of the VIA, or make Hamilton point to certain positions by indicating them from the outside, using a data glove.

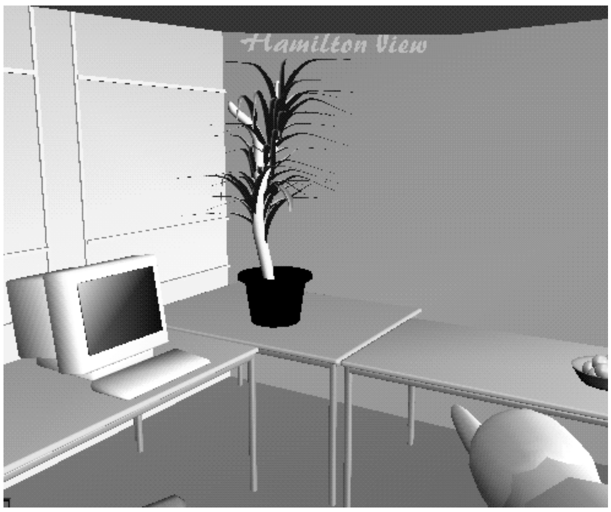

Figure 3: Pointing gesture from Hamilton's view

Through the presence of the anthropomorphic interface agent, several spatial reference schemes become available. Besides of the egocentric reference frame of the external user (identified with the position and orientation of the virtual camera), an allocentric (Hamilton-anchored) reference frame is given. Therefore, the user can direct the agent through the virtual world in order to make judgements with respect to anthropometric features of the agent from the outside. Or, the user can take on the perspective of the virtual agent to explore the scene and act in it from an involved view. By instructing the agent to *"be smaller/taller,"* the height (point of view) of the agent can be adjusted; it is also possible to give an explicit height [JöLW95].

2.2 User-Adaptive Interface

Since the interface system must meet varying conditions to enable an effective human-computer interaction, incorporating adaptation facilities becomes necessary. We have started to work on adaptation in respect to individual differences among users by applying machine learning techniques.

In our approach, we consider a system of interface agents which adapts to user preferences by learning from direct feedback. The user gives feedback by way of correcting solutions offered by single agents until the agent generating the preferred solution is dominant in the system-user interaction. The core idea is that agents which were successful in meeting the user's expectations are given credit while unsuccessful agents are "discredited." By this, the overall behavior of the interface agency gradually adapts to the individual user as the session is proceeding.

For example, spatial transformations of scene objects are communicated by way of qualitative spatial descriptions, as in *"move the palmtree to the left."* The semantics of such spatial instructions may depend on different perspectives: from the user's point of view (deictic perspective) or from the point of view of an object which has a prominent front (intrinsic perspective). Figure 4 illustrates the two alternative solutions when an object located on a desk is to be moved to the left.

In an empirical study we found out that, depending on their individual preferences, users may choose one of either perspective. Consequently, we have implemented one space agent embodying the user's reference frame (deictic reference) and one space agent embodying an externally anchored reference frame (intrinsic reference). Instructing the system with a spatial transformation, one of both space agents offers a possible solution. In case the visualized solution does not meet the expectation, the user can correct the system (*"wrong"*). The other space agent then generates an offer which modifies the previous

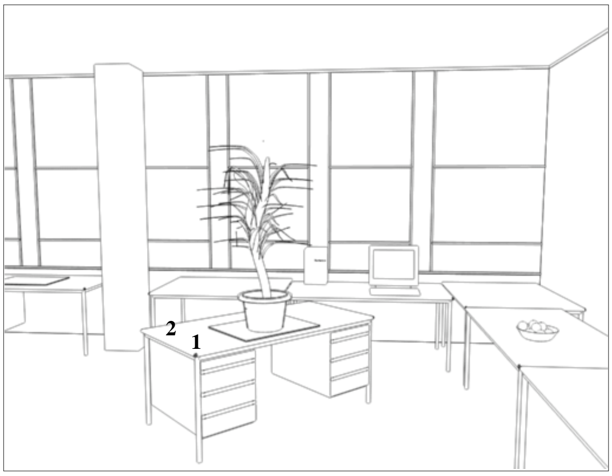

Figure 4: Example scene from the VIENA test application: The palmtree located on the desk can be moved to the left from an intrinsic perspective (1) or from a deictic perspective (2).

solution. Depending on their success in the preceding session, agents adjust their credit values dynamically to meet the user's expectations. The system's knowledge of users' preferences is expressed in adjustments of agents and is distributed among agents. Thus, user adaptation is achieved without accumulating explicit user models.

As further preferences which the agent system could adapt to, differences in color perception as well as differences in strength regarding transforming or scaling objects will be investigated [LW95].

3 Virtual Assembly Workbench

In the CODY ("Concept Dynamics") project, we develop a knowledge representation scheme for dynamic object conceptualization in assemblage. In our testbed scenario, we assemble a toy airplane and similar constructs from building blocks on a virtual assembly bench (figure 5). The user can inspect the graphics scene from different perspectives and change the building blocks' configuration by way of natural verbal communication. Instructions can refer to visual object properties, such as their location and color, but also to assembly grouping structures superimposed dynamically on the geometry models.

3.1 Virtual Constructor

The CODY *Virtual Constructor* is a knowledge-based interface agent for the interactive assembly of

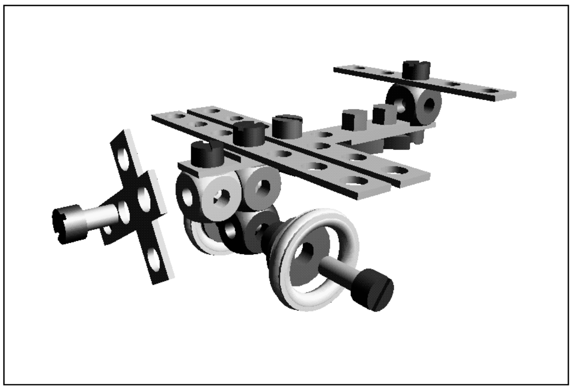

Figure 5: Airplane assembled from construction kit

complex aggregates [JuLW95]. The Virtual Constructor is able to simulate the following action types: assemble objects, disassemble objects, and rotate parts of assemblies. When processing a user instruction, e.g., *"attach the long bolt to the top of the undercarriage"* the Virtual Constructor accesses the geometric scene descriptions to evaluate spatial expressions (*"the top of"*), as well as additional conceptual (logical) representations containing knowledge about the currently assembled aggregates (like *"undercarriage"*).

The background knowledge of the Virtual Constructor is defined in four knowledge bases. The first one contains the building blocks' generic geometry models. They define the building blocks' wire frame models, center of gravity, and prototypical orientation. They also define the relative positions and orientations of the objects' *connection ports*. A second knowledge base defines several qualitative spatial relations over the geometry models, such as $\mathsf{parallel}_x$, $\mathsf{orthogonal}_z$, and touches. A third knowledge base defines conceptual knowledge about the the building blocks and their connection ports. Finally, a fourth knowledge base defines the airplane's structured assembly groups as well as the specific *roles* the building blocks can assume in assembly groups.

Dynamic conceptual descriptions are kept in working memory and linked to their corresponding geometry models. Every time objects are composed on the virtual assembly bench, logical relations such as connection are inferred from the geometric scene description. In an internal loop, these relations are propagated along the part-of structure in the logical data base. Unstructured aggregate representations are created dynamically and matched against the model knowledge base. The logical description is restructured when instances of assembly groups are recognized. Furthermore, functional roles and properties are ascribed to object representations depending on their aggregate context. For example, a bolt assumes the role of an axle when it is part of a half-axle-system (figure 6). Assembly operations, typically, are

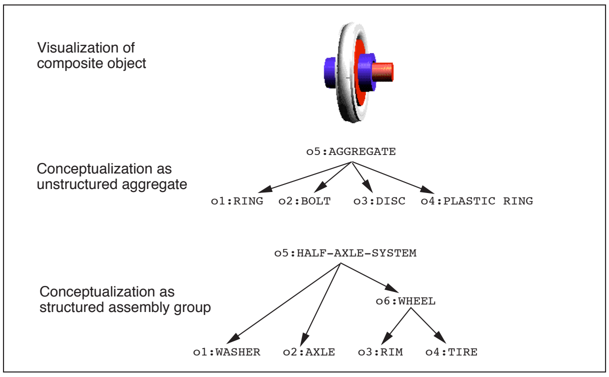

Figure 6: Assembly conceptualization

subject to side effects, i.e., they may affect objects not mentioned in instructions. The system detects these side effects and updates object representations accordingly.

3.2 Integrating Spatial and Conceptual Knowledge

The airplane's subassemblies usually require their parts to be arranged in a specific way, for example, the airplane wings must be attached *crosswise* to the fuselage; or, the wheels of the undercarriage must be in *parallel* planes. On the one hand, spatial information is necessary to adequately describe the airplane's assembly groups. On the other hand, this information can be inferred from the scene descriptions. Thus, the spatial knowledge of the geometric scene descriptions needs to be integrated with the conceptual knowledge representations.

The structured descriptions of the airplane's subassemblies are modelled in a concept definition language which builds on semantic network representations and which provides additional support for integrating spatial relations inferred from the geometry models (for detail cf. [JW95]). Concepts for assembly groups are defined by their parts and part-part-constraints describing necessary relations between them. We distinguish between *logical* constraints, e.g., connection which require, when tested, corresponding relations to be asserted between part representations, and *geometric* constraints, e.g., $\mathsf{parallel}_x$

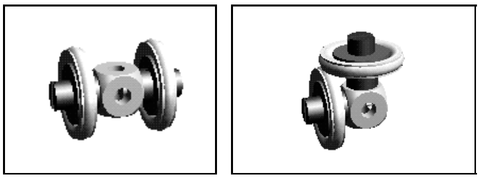

Figure 7: a) undercarriage b) no undercarriage

that trigger tests in the geometry models directly. E.g., the definition of an undercarriage, among other things, requires it to consist of exactly two half-axle-systems which must be $\mathsf{parallel}_x$ to each other. Figure left, an undercarriage was recognized. On the right, a complex aggregate which is not an undercarriage is depicted that consists of the same parts, but with the geometric constraint $\mathsf{parallel}_x$ violated.

4 Discussion and Future Work

Agent systems have proven useful in the design of more intelligent user interfaces. Acting as mediator between the user and the application system, they add comfort in human-computer interaction by allowing more human-like communication forms. We have successfully used interface agents for communicating with virtual environments.

A topic we are going to address next is multimodal input. Communication between people is so effective and flexible since they can simultaneously use different senses to receive or transmit informations. Thus, the interface agency should be able to understand and integrate user instructions of different modalities such that natural language input and simple hand gestures indicating a direction can be used. The problem of integrating informations of these two modalities shall be solved by a *multimodal input agency*. This agency consists of several mode-specific input agents, i.e., a speech listener agent and a gesture listener agent, a global input data structure, and a input coordinator agent. The listener agents are responsible for receiving and analyzing the sensor data and for sending them to the input coordinator agent which stores all incoming data in a global input data structure. By the integration of multimodal input, intuitive communication could be greatly improved.

Acknowledgment. The KI-NRW Project VIENA is partly supported by the Ministry of Science and Research of the federal State North-Rhine-Westphalia under grant no. IVA3-107 007 93.

CODY is a project in the collaborative research center "Situated Artificial Communicators" (SFB 360) at the University of Bielefeld and is in part supported by the German National Science Foundation (DFG).

References

[CJW95] CAO Y., JUNG B., WACHSMUTH I.: Situated Verbal Interaction in Virtual Design and Assembly, IJCAI-95 Videotape Program. Abstract in *Proc. Fourteenth International Joint Conference on Artificial Intelligence*, 1995, 2061-2062.

[JöLW95] JÖRDING T., LENZMANN B., WACHSMUTH I.: Ein anthropomorpher Interface-Agent für die Interaktion mit einer virtuellen Umgebung. In *KI-95 Activities: Workshops, Posters, Demos*, 182-183. Bonn: GI. Full paper: KI-NRW Report 95-02.

[JuLW95] JUNG B., LENZMANN B., WACHSMUTH, I.: Interaktive Montage-Simulation mit wissensbasierter Grafik. 8. Workshop Simulation und Künstliche Intelligenz: Virtuelle Realität in der Simulation, *ASIM Mitteilungen Heft Nr. 49*, 1995 (also SFB 360 Report 95/6).

[JW95] JUNG B., WACHSMUTH I.: Integrating Spatial and Conceptual Knowledge in Virtual Assembly. In *KI-95 Activities: Workshops, Posters, Demos*, 103-104. Bonn: GI.

[LW95] LENZMANN B., WACHSMUTH I.: A User-Adaptive Interface Agency for Interaction with a Virtual Environment. In *Working Notes of the IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems*, 43-46. AAAI Press, 1995. Full paper to appear in Springer LNAI series.

[LWC95] LENZMANN B., WACHSMUTH I., CAO Y.: An Intelligent Interface for a Virtual Environment. In *KI-95 Activities: Workshops, Posters, Demos* , 335-337. Bonn: GI. Full paper: KI-NRW-Report 95-01.

[WC95] WACHSMUTH I., CAO Y.: Interactive graphics design with situated agents. In W. Strasser and F. Wahl (eds.): *Graphics and Robotics*, 73-85. Springer, 1995.

[WLC95] WACHSMUTH I., LENZMANN B., CAO Y.: VIENA: A Multiagent Interface to a Virtual Environment. Extended Abstract in *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, 465. Menlo Park: The MIT Press.