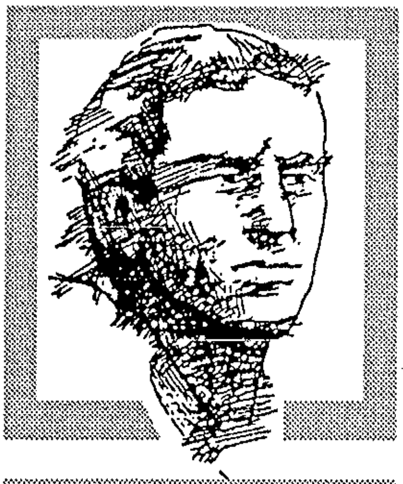
FORSCHUNGSNETZ

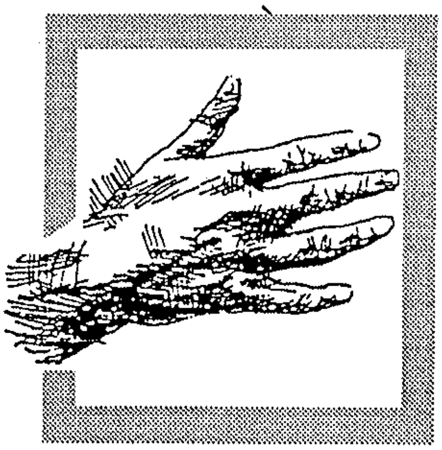
ANWENDUNGEN DER KI IN NRW

Situated Space Agent for 3-D Graphics Design

Yong Cao / Ipke Wachsmuth

Arbeitsgruppe Wissensbasierte Systeme
(Künstliche Intelligenz)
Technische Fakultät
Universität Bielefeld





PROJEKT VIENA

KI-NRW 93 - 13

Situated Space Agent for 3-D Graphics Design

Yong Cao and Ipke Wachsmuth
Faculty of Technology, University of Bielefeld,
P.O. Box 100131, 33501 Bielefeld, Germany
{yong | ipke}@TechFak.Uni-Bielefeld.DE

1 Introduction

In recent years some initial work has been reported where virtual reality systems are supported by artificial intelligence technology. For instance, the IBIS system (Feiner *et al.* 1992) has a knowledge base that includes different parts of information about physical objects to be illustrated, so that the system can design its own pictures which satisfy a specified communicative intent. Emhardt (1993) has integrated aspects of route planning and user guidance in a virtual environment.

Other work on synthetic agents can be connected to ideas from artificial intelligence. While a main aspect of the work of Badler *et al.* (1991) is the simulation and visualization of synthetic agents subject to human constraints and restrictions, they also use agents for task-oriented manipulations carried out in a virtual environment. Several mechanistic agents can act in the virtual world and change it, but only by being controlled by an intentional agent or from the outside world. Agent mediation has also been proposed for the human-computer interface (Laurel 1990). Recently, the development of knowledge-based agents for interaction with virtual environments has been recognized as a focus research theme (Bishop *et al.* 1992).

In our work, we explore the idea of communicating language and gesture commands to a virtual design environment via agents. We perceive an agent – which may consist of a number of subagents – as an "intelligent mediator" for interacting with a virtual environment. Having internal access to scene models, such an agent communicates and cooperates with a human user (who can see the visualized scene) in an overlapping perceptual situation. That is, human user and "situated" agent have a shared virtual world. The agent's task (or the agents' task, resp.) is to support the user's actions in the virtual environment. The agent thus integrates skills of situated perception, action, and communication to achieve an adequate system behavior with respect to our application domain, the design and manipulation of a virtual environment (Wachsmuth & Cao 1993).

We think that ultimately a synthetic agent graphically visualized could be used to place the designer's eye in the virtual environment and to allow the use of verbal language and gestures in interactive modeling. The idea of gesture-driven interaction with a virtual environment has been discussed in the literature (e.g., Böhm *et al.* 1992). In this paper, we concentrate on the aspect of language interaction. After discussing the possibility and necessity of using language to interact with a virtual scene, we explain how language input could be dealt with in interactive scene modeling. Particularly relevant in design is language that involves reference to spatial relations and details. Our core idea is to exploit the actual situation to the extent possible for computing the meaning of verbal input by way of a situated space agent.

2 Interfaces between Human and Virtual Environment

Natural language is an important media in human-human communication. In such communication certain cognitive abilities have played a non-ignored role, in particular, spatial cognition is very important for interpreting and reasoning about qualitative spatial relations between objects in an environment. For example, one uses spatial cognitive abilities to interpret which chair has been mentioned in the sentence "the chair left of the table" if more than one object is near the table.

Unlike the human being, a computer can only deal with exact quantitative information. In a virtual environment the geometrical data such as position and form are also exactly represented by coordinates or can be calculated directly. Is it useful to deal with qualitative spatial relations in a virtual environment? To answer this question, let us illustrate at first how a qualitative spatial relation is translated to internal operations.

Interactive exploration. Our first example is a "cybernaut" equipped with eye-phone and data-glove who explores a virtual environment. By way of data-glove interaction the cybernaut can navigate (left, right, up and down) in the environment. A tracking device in the eye-phone makes it possible that the left part of the scene is imaged if the cybernaut looks to the left, and the cybernaut can manipulate virtual objects in sight by means of the data-glove. The Virtual Reality system and its interface need not be able to interpret the cognitive meaning of "left of". It is the cybernaut who must understand the phrase "the chair left of the table" and make a selection by pointing to the chair with the data-glove. But what happens when the cybernaut wants to instruct the system, "turn the chair left of the table"?

Interactive design. Now, let's look at another example: interactive design with a 3-D graphics system such as Softimage or Inventor. Usually, the designer uses mouse and menus to communicate with the technical system. To select an object and change its position, form, or orientation the designer can (1) use the mouse to manipulate objects directly, or (2) use a dialog window to pass exact geometry information to the graphics system. When having to communicate ideas of complex form to a technical modeling device, both ways may face crucial obstacles in the process of designing. If the designer could use simple language together with a gesture input device to communicate with the technical system, it might be possible to change and illustrate the design according to the "thinking aloud" of the designer.

From these two examples we conclude that qualitative language input to a virtual environment appears a useful extension in the human – virtual environment interaction. The question is, how can a virtual environment understand what we mean? We envision an intelligent mediator system which can interpret and translate not only gestures but also simple natural language instructions to internal operations of the virtual reality system (or graphics modeler). In our example a *space agent* could take the part to translate the qualitative relation "left of" under the situation to according scene coordinates. The space agent might cooperate with other agents to find out which objects are in the scene, where the table and the speaker's eye are positioned, and where the speaker looks to. To do so it is necessary to represent some common-sense knowledge for spatial reasoning in the system in order to support intelligent mediation.

3 Spatial Representation in a Virtual Environment

Today, most virtual environments are still simple, and they do not know much about the spatial structure of their illustrated worlds. Previous work in this area has mostly aimed at high-end productions but not at imaginative abilities of virtual reality systems. It has been recognized that artificial intelligence techniques of "physical modeling" need to be included for the effects of laws of nature such as gravity, resistance of solid bodies, etc. The inclusion of verbal communication ability would require representations for spatial perception and spatial orientation to be available to the system. As an important prerequisite for comfortable mediation of interaction, the system is to know about the spatial structure as perceived and experienced by the human user.

Anthropomorphic features. Besides of an implicit topological structure such as "... a part of ..." or "... in ..." space takes on anthropomorphic features if a person is present in a scene. The metrical structure of space does not only depend on geometric relations but also, for example, on how far one could reach from a position. The visual perception system constructs spatial relations between objects and the observer's body (Bryant 1991, 1992). Different reference structures have been proposed to represent such spatial relations: (1) the egocentric reference structure defined by the three body axes: head/foot, front/back, left/right (only the left/right axis is biologically symmetric); (2) the allocentric reference structure defined by orthogonal axes outside of the observer. Allocentric reference can rely on a prominent land mark in the environment or it can employ the compass direction system (Cao 1993).

Gravity. Some features of human space perception depend on the presence of gravity. Friederici (1989) points out that astronauts, in zero gravity, have problems to identify "up" and "down" because in gravity the head/foot axis is usually identified with the up/down axis if the observer is in upright position.

In a virtual environment the same situation may occur as long as gravity cannot be modeled. That is, a cybernaut in a virtual environment has in principle the same problem as an astronaut in a space shuttle: the up/down axis is not determined. The modeled objects and their relations in a virtual environment are seen as they appear in the physical world, and the user is likely to deal with them by using experiences acquired from the physical world. In a virtual environment such as an office room or a building, spatial axes can be determined according to experiences in the physical world. For example, the head/foot axis of a visualized agent in upright position should be parallel to the up/down axis of the environment.

Qualitative reference. Instead of using exact coordinates of objects in space one can often use prominent spatial objects as reference objects (Habel 1988&91). In most cases one understands a location by a qualitative relation deduced from a coordinate system (e.g., "Kiel is north of Hamburg") rather than by exact geographical coordinates (Cao 1993). However, in contrast to exact coordinates such qualitative spatial relations are not only vague but they also depend on the frame of reference which may change from situation to situation.

Different perspective. Qualitative spatial relations may depend on different perspectives: from the observer's point of view (deictic perspective) or from the point of view of an object which has a prominent front, e.g., a desk (intrinsic perspective). In many cases, e.g., in room description, the deictic perspective is preferred (Ullmer-Ehrlich 1982). However, space may be modified by intrinsic features of the objects present in the scene (Levelt 1986; Lang 1989). If a second person is present in the scene, the observer can also use the perspective of this person to describe a spatial relation. In a virtual environment we can imagine a similar situation: The designer observes the scene, and an agent illustrated in the scene can be conceived either as the second "person" or as a personification of the designer (as a participant observer), that is, the agent has the same perspective and field of vision as the designer.

Relative size. The prototypical size of objects and the distance between objects play an important role in the qualitative spatial relations perceived between these objects (Miller & Johnson-Laird 1976, Pribbenow 1988). To carry out an instruction such as "move the chair to the front of the desk", the space agent must determine how far the chair can be away from the desk and which position is best to place the chair. The same can occur in an identification task, i.e., when responding to WHAT and WHERE queries. For instance, on the question "What is near the table?" some small objects will not be selected as answer but a larger object further away can be selected; and the location of a small object with respect to a reference object cannot be described by the same qualitative spatial relation as that of a larger object although the larger object is further away.

Concreteness. Spatial representation in a virtual environment has some special features that differ from those of a physical environment. First, it is often simple to calculate the concrete size and position of an object by way of its bounding box. In this way the situated agent can get more concrete (geometry) data about objects in space than a human can obtain in a physical environment. Thus a main difference between a human and a situated agent is that, while a person would have to estimate a location, the agent can calculate it by accessing the geometry data base. Second, because the results of the spatial reasoning can be verified (illustrated) in a virtual environment directly, it is important that each object must have a determined position at any time. In contrast, abstract spatial representation schemes (e.g., Khenkhar 1991) allow objects to be located in a vague region that satisfies the given qualitative spatial relation.

4 A situated space agent

A main challenge for interacting with virtual environments is to translate qualitative language descriptions to exact geometrical models for navigation or model construction in a virtual environment. By projecting human biological and cognitive asymmetries into basic space, the human perspective could usefully be introduced in virtual space. For instance, this would allow to use situated references such as "front", "back", "right" and "left", especially if a synthetic agent is also visualized in the virtual environment. When the designer communicates with the agent about the scene, the agent should be able to determine which perspective the designer has used in the communication.

4.1 Context

The development of a situated space agent which is to mediate verbal interaction with a virtual environment is a core part of the VIENA project. The VIENA project (VIrtual ENvironments and Agents) has started in 1993 in a new research focus theme of "Artificial Intelligence and Computer Graphics" at the University of Bielefeld. The overall goal in VIENA is to enable an intelligent communication with a technical system for the interactive design and exploration of an environment visualized by way of 3D computer graphics. We have chosen interior design as an example domain. Instead of using mouse and menus to manipulate objects we develop a set of agents which altogether form an intelligent mediator. Our aim is to keep the user (designer) free from technical considerations such as planning of geometric details, etc. Simple written natural language is currently considered for the communication. In the future we also want to use voice input. A description of our project scenario is given in (Wachsmuth & Cao 1993).

Our current working environment includes a Silicon Graphics Indigo ELAN R4000 for the main project demonstrator. This machine supports the real-time hardware shading we make use of. We also have the stereo option, for better 3D impression, which we intend to use later. We do not include real-time texturing at this point which is not supported by our machine. Currently, we use the Softimage Creative Environment for scene modeling and rendering.

Besides this, several Sun SPARC stations are available for algorithm and model development. These are also the site for some of the agents, as we make use of interprocess communication. While we are not in the position, currently, to exploit the usefulness of our work in a more immersing environment (e.g., with a head-mounted display or a boom), we think it could be readily incorporated in such a setting at a later time.

We employ Artificial Intelligence techniques to make our interface agents become an "intelligent mediator". In a virtual environment all objects are represented by their geometry models, and the intelligent mediator can get all these models. Thus the intelligent mediator can do more things than can typically be done by symbolic reasoning. Common to all abstract representation formalisms is the difficulty of modeling *side effects* of actions in the modeled world, that is, making corresponding modifications in the database representing the state of the world (the "frame problem", cf. McCarthy & Hayes 1969). This can be better explained by an example of room reorganization. There is a desk in room and a lamp on the desk. The question is, when we move the desk, what happens to the lamp? We all know that a lamp in the physical world will be moved together with the table as side effect but that things not connected to the table will remain where they are. In a virtual environment it is possible to modify a scene selectively and leave uninvolved objects untouched. Of course, the mediating agents should be made smart enough to realize that things *on* the table should move with it.

4.2 System structure

In this section we give a brief overview of the VIENA system (cf. Figure 1). Our system acts as an interface between a human user (designer) and a 3D graphics modeler/renderer. The user communicates changes to the system by

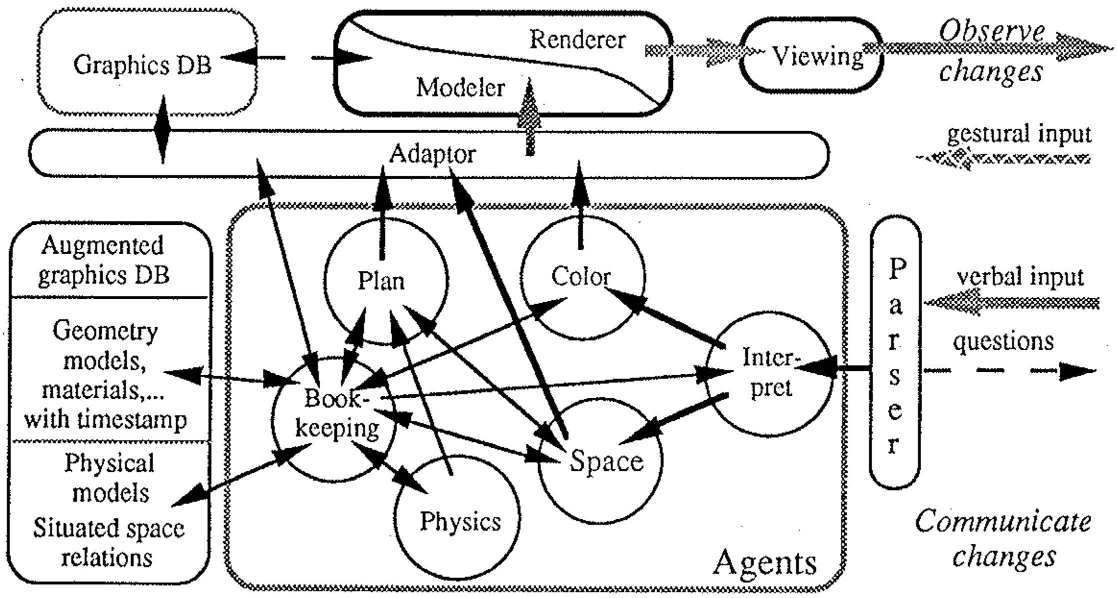


Fig. 1 Evolving architecture of the VIENA system: A multi-agent system mediates qualitative verbal instructions by translating them to quantitative commands that are used to update the visualization scene model (further explanation in text).

way of verbal instructions (which may later be accompanied by gesture input). A parser unit translates a user instruction to an internal surface representation which outputs to the mediating agents. The parser asks back if an instruction is not complete or if it is syntactically incorrect. The user observes changes from the Viewing graphical output medium.

The core system is designed to be portable, hence a special Adaptor unit is used to convert data back and forth between the mediating agents and the technical modeler. The Adaptor also establishes a 'pipeline' to the modeler/ renderer unit so that modified scene data can be visualized instantaneously. The graphics data base is mirrored in the mediating system in an augmented graphics data base which is local to a "Bookkeeping" agent. All data about the scene can only be modified via the Bookkeeping agent. Besides the viewing data, the augmented data base holds information about scene objects that are needed for situated communication.

An internal surface representation of a verbal instruction will be elaborated by an "Interpret" agent which transmits commands to further agents. As a verbal command input may refer to a previous one (e.g., "a little less" following "move the table to the right"), the system must be able to exploit the situation history to elaborate a complete command. This will involve the interpret agent to cooperate with the bookkeeping agent. To this end, the history of changes (data with timestamp) must be kept in the augmented data base so that the situated communication can successfully arrive at offering a new resulting scene.

We have planned two types of changes so far, spatial and material, which will be dealt with by the "Space" and the "Color" agent. As a core part for the intelligent mediator we build a space agent incorporating rules for reasoning about spatial relations between scene objects. The space agent has the following responsibilities: (1) to identify a mentioned object and to determine how to display the object so that the designer can understand its relative spatial relation to another object, (2) to determine where and how an object will be moved, in relation to another object, in order to satisfy a spatial relation. When establishing a new spatial relation, the augmented graphics data base must be updated accordingly. In a simple case the intelligent mediator transmits only the target image model for visualization. In case of a more complex scene modification, a "Plan" agent will become active and cooperate with a "Physics" agent to generate a series of images.

In the following subsections we explain some of the tasks the space agent is to deal with. In the last section, we give a brief state report of the work so far underway.

4.3 Deictic and intrinsic reference

One of the responsibilities of the space agent is to support commands that make reference to the virtual presence of the user (e.g., exploit anthropomorphic features), or commands that incorporate qualitative reference to other scene objects. As mentioned above, there are different reference frames to describe spatial relations.

The intrinsic reference frame can only be selected if the reference object has a prominent orientation such as a front (e.g., a desk). So the intrinsic reference frame will be derived from the prominent orientation of the reference object and is independent of the position and orientation of the camera. To deal with information with respect to this reference frame the prominent orientation of an object must be supplemented in the augmented graphics database.

The deictic reference frame is usually defined by the observer's body. As long as the designer does not appear in a virtual environment, the designer can only see what the camera picks up. In this case the deictic perspective can be derived from the camera position and its orientation. However, the (fictitious) virtual position of the user (determining an implicit up/down axis) must be discriminated from the direction to which the user looks. This information can be inferred from the camera position and orientation. Analogously to the three spatial axes determined by the human body the camera has also three axes (cf. Figure 2). One axis is directed towards the point of interest. It determines which objects will be shown in the visualization. The other two axes are used for orientation. Usually the camera must be so positioned such that a special perspective of a 3D scene can be picked up. In most situations the three axes of the camera, especially the axis that corresponds to the up/down axis of human spatial orientation and the axis toward the point of interest, are not parallel to the coordinate system used in the geometry scene model.

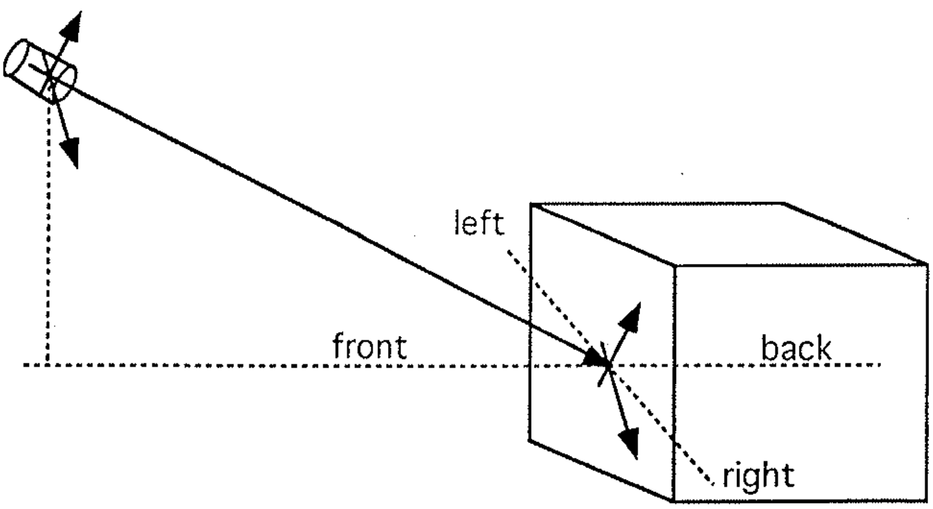


Fig. 2 The camera view extends to the synthetic environment with a certain angle. For describing spatial relations in the environment, this angle will not be considered.

Although the same situation can be found in human perception, only the vertical body axis is used to establish the deictic reference frame. For example, a prototypical "front" direction of a reference object does not point towards the observer's eye but rather straight towards his body (Fig. 2). This means, the deictical front/back, left/right and up/down axes are, from a cognitive point of view, not the same as the three camera axes, but they can be calculated with respect to these axes.

4.4 Object identification

Another responsibility of the space agent is to support object identification, i.e., working out a response to a "where is object xyz" query. In three-dimensional space an object can be occluded by another object in two different situations. The first situation is that one object is fully included in another object, e.g., a ball in a box. The second situation is that an object is currently not visible for the camera because something is in between.

The first case, that is, showing an object included in another object, has been dealt with in the context of equipment maintenance, for identifying the location of an object to be repaired (cf. Feiner *et al.* 1992). A similar approach is used in the Hyper-Renderer (Emhardt *et al.* 1992) which can display selected objects enhanced, and all unselected objects appear in a ghost form. In these solutions the camera position and orientation are kept unchanged; only the object in the vision field is highlighted or marked for rendering to allow it to be seen through the objects occluding it.

In interior design, the second situation is more likely to occur. In case objects are not including one another, one can always find a camera position and orientation that shows these objects of interest in the same vision field. For instance, the camera can be moved to a position to display a chair behind a desk. Thus the space agent could be employed to find out ranges of an according camera position and orientation and to calculate the trajectory of a camera move for the renderer. Although the camera picks up the scene from another position and orientation now, the designer can understand the relative spatial relations between objects by translating the camera's movement to a spatial relation by way of experiences from moving in the physical world.

5 State of work

Work so far has concentrated on the technical environment in which mediating agents can come to bear. That is, we have experimental implementations for an Adaptor unit (specific to the Softimage modeler/renderer) and a Parser unit, and we are currently conceptualizing an augmented graphics database (cf. Figure 1). Based on this evolving working environment, we have begun to specify agents and agent interactions for intelligent mediating.

5.1 Adaptor

As mentioned above, we use Softimage for scene modeling and rendering. The Adaptor must translate data from Softimage's data base to an augmented data base and filter modified data to the renderer. In our experimental implementation we manipulate 3D scene objects by way of channels in the Softimage motion environment, and we have also started to use a custom script language provided with Softimage (version 2.6.1) environment for interfacing.

Figures 3 and 4 show how a scene is changed based on input from a simple command interface external from the Softimage environment. In Figure 3, we see a scene with a desk and a chair before a command is issued. In Figure 4, the scene has been changed according to the command "move_the_chair_to_the_left". The command is directly hooked – for demonstration – to precalculated target positions for the chair model. Our next goal is to have mediating agents calculate, for a *language* command input, which scene detail has to be changed and in what respect, and then pass according specific information to the Adaptor.

For our purpose, Softimage's channel driver interface has many limitations. First, there are only 256 channels and each channel can be mapped exactly onto one pre-defined feature. A maximal number of 21 objects per channel driver can be controlled. Second, one can transfer data to the environment via channels but it is not possible to acquire scene data from the environment by way of channels. Thus we are pursuing the use of custom scripts which provide another means of interfacing user-defined functions with Softimage. Using the Developer Kit, all types of scene data can be read and modified by way of appropriate script commands.

5.2 Parser

While the Adaptor unit constitutes an interface between the agent system and the visualization, the parser unit interfaces between verbal input and the mediating agent system. In our setting we think it more important that user input is kept to the minimum significant information. So we are not trying to process very complex sentential structures. However, the system should be able to process discourse that negotiates the semantics of instructions. That is, if the effect of an instruction does not satisfy the user's expectations, the parser should be able to accept short corrective statements and analyze them based on previous discourse.

Our parser ("PARSY") can translate simple qualitative English instructions to a qualitative surface representation which outputs to an Interpret agent. PARSY analyzes an instruction (such as "move the table to the right") and translates it to a representation with an operator ("move"), object ("table"), degree (" "), and a location ("right") specification. The same works for changing material features of scene objects such as color. PARSY is also able to resolve simple ellipses such as "a little less". In Figure 5, a sample discourse and generated surface representations are shown.

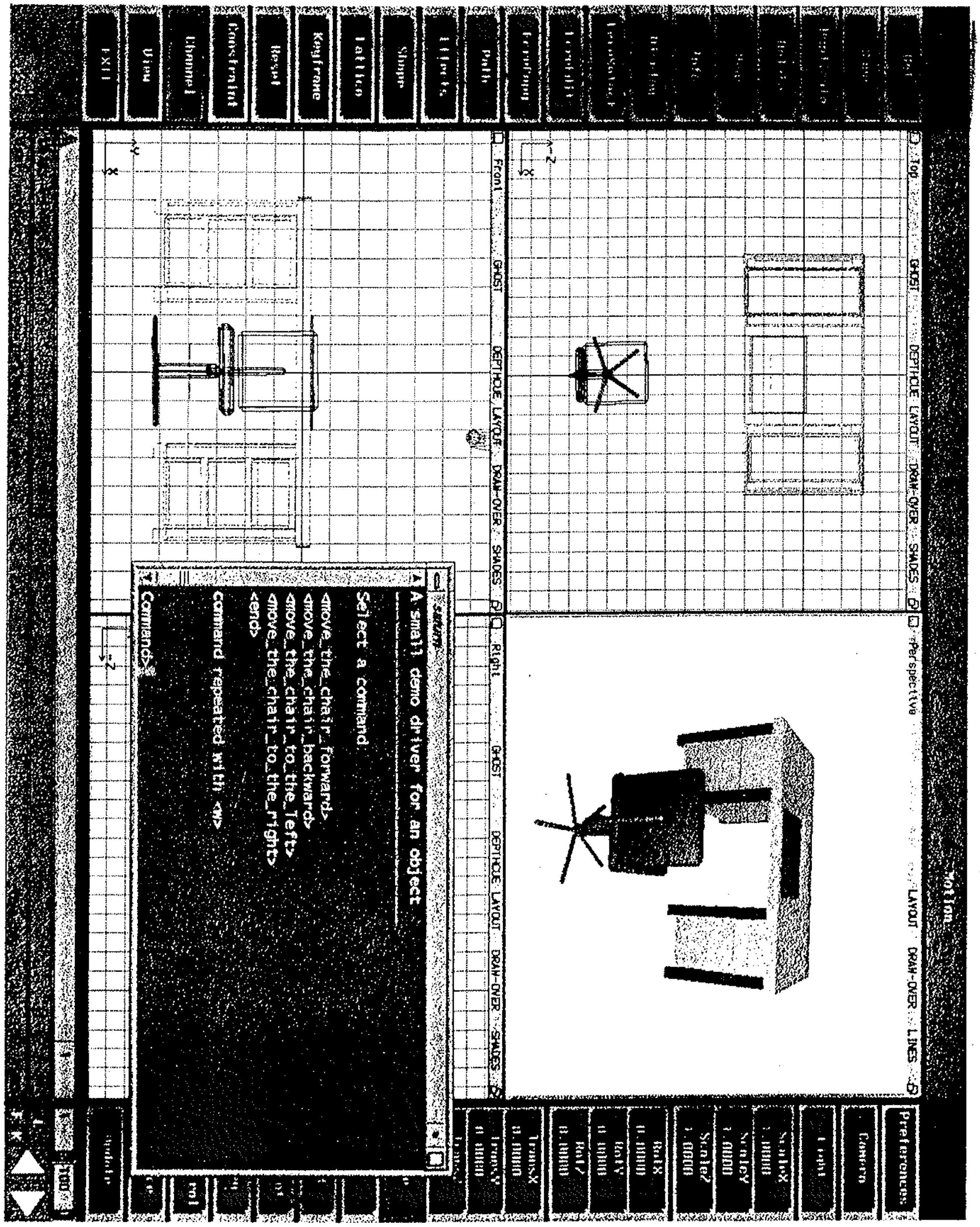

Top
GHOST
DEPTHCUE
LAYOUT
DRAW-OVER
SHADES
Perspective
LINES
Front
Right
A small demo driver for an object
Select a command
<move_the_chair_forward>
<move_the_chair_backward>
<move_the_chair_to_the_left>
<move_the_chair_to_the_right>
<end>
command repeated with <w>
Command>
Motion
EXIT
View
Channel
Constraint
Reset
Keyframe
Lattice
Shape
Effects
Path
Preferences
Camera

Fig. 3

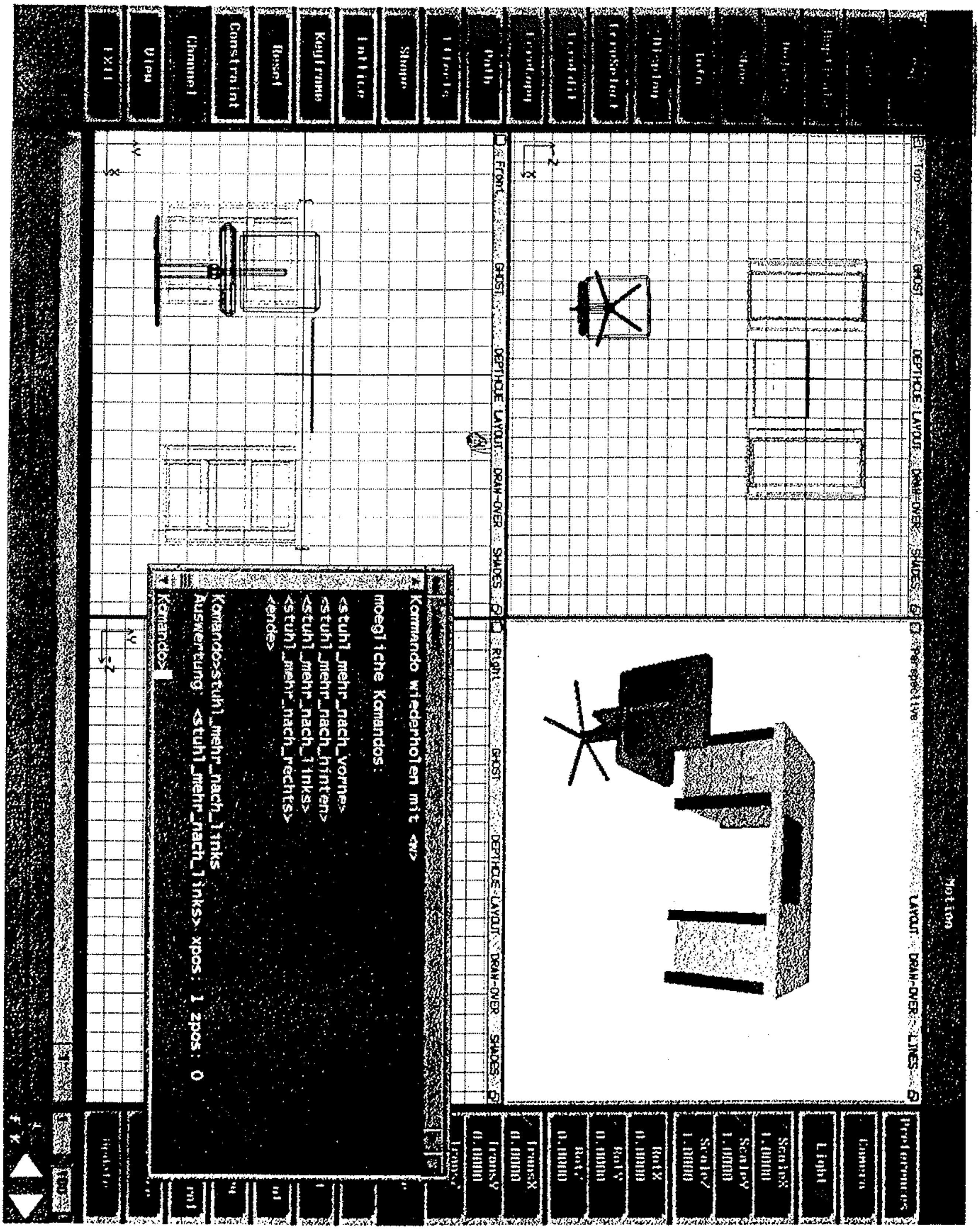

Top
GHOST
DEPTHCUE
LAYOUT
DRAW-OVER
SHADES
Perspective
LINES
Motion
Front
Right
Kommando wiederholen mit <w>
moegliche Komandos:
<stuhl_mehr_nach_vorne>
<stuhl_mehr_nach_hinten>
<stuhl_mehr_nach_links>
<stuhl_mehr_nach_rechts>
<ende>
Komandos-stuhl_mehr_nach_links
Auswertung: <stuhl_mehr_nach_links> xpos: 1 zpos: 0
Komando>

Fig. 4

```
saturn%parsy
PARSY> move the table to the right
       move(obj=table, deg=, loc=right)
PARSY> a little less
       modify(obj=table, deg=little, loc=negativ)
PARSY> make the chair red
       make(obj=chair, deg=, col=red)
PARSY> a little more
       modify(obj=chair, deg=little, col=positiv)
PARSY>
```

Fig. 5 Input and internal output (bold face) of the parser PARSY

Within a session discourse, an elliptic phrase ("a little less") will be represented as a modification of the previous instruction. (A modify-command will make agents compare the most current and the previous scene model by way of time-stamped entries in the augmented graphics data base; see below.) When analyzing an incomplete instruction, PARSY prompts the user for further input. Again, consecutive input is processed specifically (cf. Figure 6).

```
saturn%parsy
PARSY> move the chair
       You must enter a direction.
PARSY> left
       move(obj=chair, deg=, loc=left)
PARSY> end
       PARSY GOOD BYE
saturn%
```

Fig. 6 PARSY analyzes an uncomplete instruction

5.3 Augmented data base

The modeler/renderer has its own data structures in a graphics data base which holds information necessary for visualization. These data structures need to be augmented to be suitable for situated communication. That is, "non-visualizable" information needs to be added that enables the system, for instance, to make reference to intrinsic perceptive features of scene objects. Also, previous features of a changed scene model must be kept in order to evaluate elliptic discourse ("a little more"). To support such situated communication, the data structures in the augmented data base must accommodate multiple feature value entries which are tagged with a timestamp. Only the Bookkeeping agent, acting as a "gate keeper", is authorized to access and modify the augmented data base (cf. Figure 1).

5.4 Situated Space agent

The Space agent takes a main part in evaluating user commands. Qualitative "relative" instructions, output in surface representation from the parser, need to be processed to quantitative, absolute geometry data changes (e.g., translations in appropriate coordinates). As the system is able to take into account all situational data of a current scene, including previous discourse, ambiguity in instructions can be greatly reduced. If a system response does not meet the user's expectations, a follow-up instruction can build on the previous response.

To process a surface representation like **move(obj=table, deg=, loc=right)**, the Bookkeeping agent determines first which object named "table" is addressed by the instruction, and then reads the geometry data of this object to the Space agent. (If more than one object is named "table", the "table" object with the most current time stamp is selected.) The Space agent must find out which location is "right" in the current situation, how far "right" can range, and which moderate degree of a "right" move is offered in response to the command. The Adaptor transfers the change of "table's" geometry data to the renderer, and the Bookkeeper updates the augmented data base accordingly. To deal with a "modify" instruction, the Bookkeeper accesses according entries in the augmented data base and finds out the last change. A new change will be inferred based on the last recorded change and the current change command.

Other agents will enhance the systems ability to mediate verbal instructions. Next to be worked on is a Color agent which acts in a similar fashion as the Space agent: a qualitative instruction is processed to quantitative material data changes (e.g., changes of rgb-values). It may be sensible to conceptualize sub-agents of an agent to take special tasks. The idea is that each agent is just "smart" enough to meet its special responsibility. By this, modular system development is greatly supported.

Acknowledgment

Research in the VIENA Project is partly supported by the Ministry of Science and Research of the Federal State North-Rhine-Westphalia in the program "Artificial Intelligence and Computer Graphics" under grant no. IVA3-107 007 93. The ideas and conclusions expressed are those of the authors and do not necessarily reflect the views of the funding institution.

The authors are indebted to Norbert Siekmann, Britta Lenzmann, and Majid Amanzadeh for assisting during the research.

References

Adler, M., Durfee, E., Huhns, M., Punch, W., and Simoudis, E. (1992). AAAI workshop on cooperation among heterogeneous intelligent agents. *AI magazine 13*(2), 39-42.

Badler, N.I., Webber, B.L., Kalita, J. & Esakov, J. (1991). Animations from Instructions. In N.I. Badler, B.A. Barsky & D. Zeltzer (eds.): *Making Them Move - Mechanics, Control, and Animation of Articulated Figures* (pp. 51-93). San Mateo (Cal.): Morgan Kaufmann.

Bishop, G., Fuchs, H. *et al.* (1992). Research Directions in Virtual Environments: Report of an NSF Invitational Workshop (March 23-24, 1992, University of North Carolina at Chapel Hill). *Computer Graphics 26*(3), 153-177.

Böhm, K., Hübner, W. & Väänänen, K. (1992). GIVEN: Gesture driven interactions in virtual environments: A toolkit approach to 3D interactions. Proc. Conf. Interface to Real and Virtual Worlds (Montpellier, France, March 1992).

Bryant, D.J. (1991). *Perceptual Characteristics of Mental Spatial Models* (Dissertation). Stanford University: Dept. of Psychology.

Bryant, D.J. (1992). A Spatial Representation System in Humans. *PSYCOLOQUY 3*(16) space.1

Cao, Y. (1993). *Zur Darstellung und Verarbeitung von Wissen über Himmelsrichtungen – Geometrische und Kognitionswissenschaftliche Aspekte* (Dissertation am Fachbereich Informatik der Universität Hamburg). Sankt Augustin: Infix (DISKI-33).

Emhardt, J. (1993). Agentenunterstützte interaktive Exploration von virtuellen Gebäuden. In *Konferenzband Virtual Reality '93*, Fraunhofer-Gesellschaft, Stuttgart.

Emhardt, J. & Strothotte, Th. (1992). Hyper-rendering. In *Proc. Graphics Interface'92*, Vancouver, 37-43.

Feiner, S., MacIntyre, B. & Seligmann, D. (1992). Annotating the Real World with Knowledge-Based Graphics on a See-Through Head-Mounted Display. *Proc. Graphics Interface '92*, Vancouver, Canada, 78-85.

Friederici, A.D. (1989). Raumreferenz unter extremen perzeptuellen Bedingungen: Perzeption, Repräsentation und Sprachliche Abbildung. In Ch. Habel, M. Herweg & K. Rehkämper (Hrsg.): *Raumkonzepte in Verstehensprozessen* . Tübingen: Niemeyer.

Habel, Chr. (1988). Cognitive Linguistics: the Processing of Spatial Concepts. In: *A special issue of ATALA (Association pour le Traitement Automatique des Languages).*

Habel, Chr. (1991). Processing of Spatial Expressions in LILOG. In: O. Herzog & C.-R. Rollinger (eds.): *Text Understanding in LILOG.* Lecture Notes in Artificial Intelligence, 546. Springer-Verlag:Berlin.

Khenkhar, M. (1991). Object-Oriented Representation of Depictions on the Basis of Cell Matrices. In: O. Herzog & C.-R. Rollinger (eds.): *Text Understanding in LILOG.* Lecture Notes in Artificial Intelligence, 546. Springer-Verlag:Berlin.

Lang, E. (1989). Objektschemata und räumliche Konfiguration. In Ch. Habel, M. Herweg & K. Rehkämper (Hrsg.): *Raumkonzepte in Verstehensprozessen.* Tübingen: Niemeyer, 150-173.

Laurel, B. (1990). Interface agents: Metaphors with character. In B. Laurel (ed.) (1990): *The art of human-computer interface design.* Addison-Wesley.

Levelt, W. (1986). Zur sprachlichen Abbildung des Raumes: Deiktische und intrinsische Perspektive. In H.-G. Bosshardt (Hrsg.): *Perspektiven auf Sprache.* Berlin: DeGruyter, 187-211.

McCarthy, J. & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. In: D. Michie & B. Meltzer (eds.), *Machine intelligence 4.* Edinburg: Edinburger University Press, 463-502.

Miller, G & Johnson-Laird, P. N. (1976). *Language and Perception.* Cambridge University Press.

Pribbenow, S. (1988). Verträglichkeitsprüfungen für die Verarbeitung räumlichen Wissens. In: W. Hoeppner (ed.), Proc. GWAI-88. Springer-Verlag:Berlin.

Ullmer-Ehrlich, V. (1982). The structure of Living Space Descriptions. In: R. Jarvella & W. Klein (eds.) *Speech, Place and Action.* Chichester: John Wiley.

Wachsmuth, I. & Cao, Y. (1993). Virtual environments and situated agents. In W. Strasser & F. Wahl (eds.): *Graphics and Robotics.* Heidelberg: Springer (LNAI).

Authors' Addresses

Yong Cao, Faculty of Technology, University of Bielefeld, P.O.Box 100131, D-33501 Bielefeld, yong@techfak.uni-bielefeld.de

Ipke Wachsmuth, Faculty of Technology, University of Bielefeld, P.O.Box 100131, D-33501 Bielefeld, ipke@techfak.uni-bielefeld.de

Limited Distribution Notice

KI-NRW

KI-NRW is a long-term cooperative research initiative on "Applications of Artificial Intelligence in North-Rhine-Westphalia". It is organized informally as an association of research groups from different universities and research institutes in the Federal State of North-Rhine-Westphalia, Germany.