# Complex Construction Kits for Coupled Real and Virtual Engineering Workspaces

Wilhelm F. Bruns

artec – Research Center for Work, Environment, Technology
Bremen University
Enrique-Schmidt-Str. 7, Bremen D-28334, Germany
Email: bruns@artec.uni-bremen.de

Published in: Norbert A. Streitz et al (1999): Cooperative Buildings. Lecture Notes in Computer Science 1670. Springer, Berlin, pp 55-68

**Abstract.** A concept of complex objects, being artifacts that have one real physical part and several virtual parts representing certain aspects of the object, is introduced. These parts are coupled by bi-directional double links of control and view, enabling a synchronous update of all part, if one of them is changed by user action or internal events. With a construction kit, being a set of compatible complex objects for a certain engineering or office application field, it is possible to build a system in reality with real parts, generating synchronously the assembly of virtual parts by means of a universal graspable user interface. The bi-directional double links allow the control of virtual parts by grasping and pointing on real parts and view the virtual parts by light projection into the real scene and vice versa, that is, control and view real parts by grasping and pointing on virtual parts. The concept is being demonstrated with prototypes for the application areas of pneumatic circuit design and flexible assembly systems.

**Keywords.** coupling reality and virtuality, simulation, cooperative system design, universal graspable user interface, augmented reality, tangible objects, complex objects

## 1 Introduction

Coupling tangible objects of real work spaces with information spaces of digital representation has been subject of increasing interest during the last decade. Weiser (1991) set up the vision of a room with information and action generated by a ubiquitous computer. Wellner et al (1993) emphasized the paradigmatic shift of computer-augmented environments: back to the real world. Fitzmaurice et al (1995) lay the foundations for graspable user interfaces. Resnick (1993) introduced behavior construction kits based on real objects. Since then, many prototypical applications have been published. To name only a few: Kang & Ickeuchi (1994) proposed a concept of programming robots by concrete teaching, the MIT Media Lab is hosting a strong research group working on tangible objects (Ishii & Ullmer, 1997), Suzuki & Kato

(1995) use real AlgoBlocks for programming, Rekimoto (1998) developed intelligent rooms and a series of workshops now has a focus on the integration of information into real Buildings (Streitz et al., 1998).

All these attempts to couple real tangible objects with digital representations only support one-way-links (manipulating digital representation by concrete handles) or a projection of digital representations into the real scene. In engineering applications, mainly in the area of design of automation devices and systems, it is extremely interesting to have an easy access to both sides of a system, the real physical environment of actors, sensors with their electro-mechanical mechanisms, and its digital representation used in simulation models and its driving control algorithms. We therefore introduce a concept of tight bi-directional coupling to bridge these two worlds.

Some prototypes we developed for the cooperative design and simulation of automation systems, namely flexible assembly systems driven by PLCs (Programmable Logic Controller), robotics and pneumatic circuits are presented and generalized. The derived concept aims at a new kind of distributed work space for systems design in production automation. However, it is also intended to yield a new kind of learning environment allowing the switch between concrete and abstract views on physical and work phenomena to be made easily and quickly.

## 2 Basic Concept

In several industrial simulation projects in the area of production and logistics, we learned from experience, that physical models are very helpful for a common understanding in multidisciplinary design teams and improve the understanding of difficult technical matters. This proved to be true not only for the specification of geometrical and topological features, but also for the intended dynamical behavior of devices or systems. We therefore developed the concept of complex objects, having one real part and several corresponding virtual parts (computer internal representations). In Fig. 1 two different kinds of complex objects are presented, one for a pneumatic cylinder (left) and one for a conveyor belt (right). Computer based links between real and virtual parts ensure the synchronization of their states. They can be realized by video-image-recognition or, as shown in Fig. 1, by data glove tracking. Starting from a reference situation, changes of state are sensed by a graspable user interface and used to update the complementary part (Bruns 1993). The term *complex object* is an allusion to the mathematical notion of complex numbers. Similar to complex numbers, having a real and an imaginary part, the complex object contains an abstract object with enriched possibility of mathematical treatment and behavior (algorithms, data-structures) and the controlled automation device as its projection into reality. Of course, this is only a limited metaphor because we allow different levels of abstraction and perspective for one real part.

With construction kits, containing sets of these complex objects for specific application areas, it is then possible to construct a system in reality and synchronously generate a corresponding virtual model, that can be tested, analyzed and transmitted to remote places (Fig. 2).

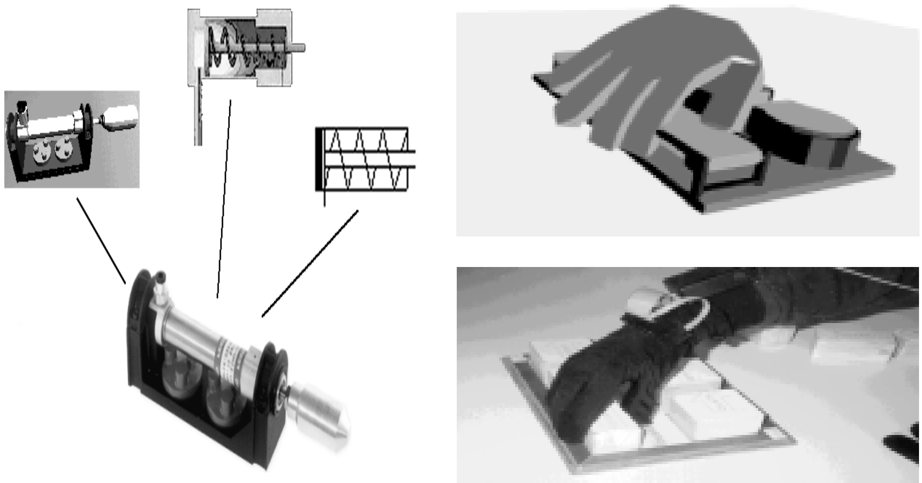


**Fig. 1.** Complex Objects with real tangible parts and various digital representations

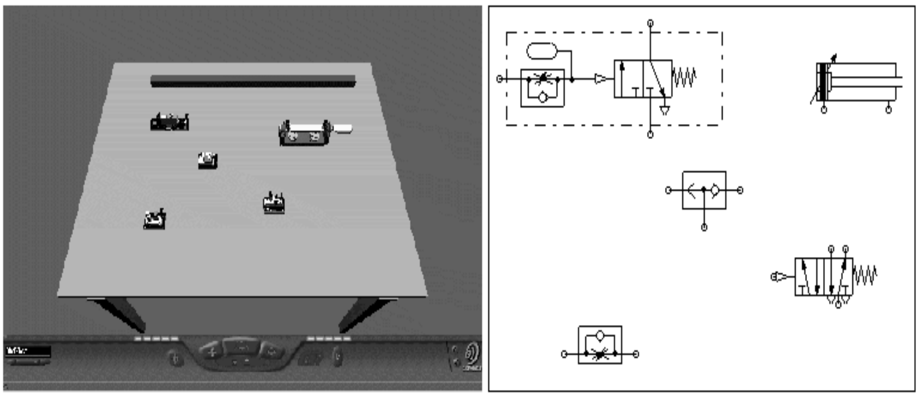


**Fig. 2**. Synchronous Generation of Virtual Reality and Simulation Models

Technical details of the implementation of this concept have been described elsewhere (Bruns 98) but are summarized for convenience. One main characteristic of the approach is the use of our hand as a manipulator of physical objects in a real environment. Appropriate interface devices like data gloves and tracking systems are used to capture the user's hand movements and finger flexions. With the help of gesture recognition algorithms, based on statistical methods (Brauer 94), the raw interface data is analysed and gestures, grasps, or user commands are recognised by the computer in real time. Working with physical objects while being linked to a computer has a certain analogy to the well known Drag & Drop principle of GUIs. When the grasp of an object happens, all following data of the Drag-Phase are recorded. This phase terminates when the user places the object at another location and releases it (Drop). Now, the physical object has a new position and due to this, the virtual computer internal model of the physical environment is immediately updated. By giving the user an acoustic feedback in the moment of grasping and releasing, the graphical output on a display becomes obsolete. So the user can work distinct from the encumbering aura of the monitor, the keyboard and the mouse. The interface becomes a passive observer and is ideally not noticed by its users. This is achieved by linking physical objects to their virtual representation. Because of maintaining objects having both a physical and a virtual representation, we call them *complex objects*.

Complex objects are one of the basic elements of our concept. For both kinds of object representation a number of instances must be available. This means to create a virtual representation consisting of the object's geometry and algorithms describing the dynamic behaviour. The geometric description contains the object's size (length, width, height) and its surface shape. The dynamic behaviour is specified by application specific languages or general purpose descriptions like Petri-Nets. It may be predefined and fixed or taught by concrete demonstration (see below). On the other hand, the physical part of an object must be constructed using technical construction kits, wooden bricks or other materials. The object representations may vary in shape, size and level of detail. In initial state, the objects are located in an object box which has a predefined position on the tabletop, such that for each object in the box the position can be computed. A model is created stepwise by taking objects out of the box and putting them on the model ground. This way, several models can be managed synchronously, providing different views on the system to be built. With the help of 3D visualisation software, the geometrical representation can be displayed on a monitor screen. With the help of an application specific simulator, the symbolic, functional and behaviour representation can be displayed in a circuit diagram, a program source-code or a projection of its dynamics on the screen or table. Although the visual feedback is not necessary for those who model with the physical model, it is used for replaying the actions recorded during a modelling session. Furthermore people working in remote locations can observe a modelling process via a network connection to the *Real Object Manager* running as Server.

Working synchronously with two models requires sophisticated communication structures between several software modules in which each of them is responsible for a specific task. These tasks are:

- maintaining a virtual model, keeping track of the actions performed with the complex-objects,
- recognising grasp and gesture events,
- recording data of the object movements,
- abstracting a general description of the recorded data,
- visualising the modelling process and
- persistent storage of data in files.

According to this allocation of tasks, a general software architecture was designed (Fig. 3). A central component of this architecture is the Real Object Modeller (ROMAN). This module maintains an object database which contains geometric object descriptions, keeps track of the state of the model, provides information for the visualisation in a 3D graphical model, handles model-files, and provides an interface for the dynamic data exchange (DDX) with external real or virtual processes. Via DDX and appropriate communication protocols, a connection of external processes, running on different machines, may be established. This is the case for the Gesture-Server task which handles the data glove and sends gesture event messages to the ROMAN. Another example is the Simulation-Converter which acts as a mediator between the ROMAN and standard simulation software products. Fig. 3 shows bi-directional connections between the DDX interface and simulators. This indicates that a model-file (SML) is downloaded from the ROMAN, translated with a converter to a simulator-specific data format, and is then simulated. The dynamic simulation yields to changes in the model, for example a container is transported by a conveyor belt to a new location. These changes are immediately transferred via DDX to the ROMAN where the virtual model is updated. We developed visualisation clients which can be connected to the ROMAN via DDX. By using standard TCP-IP protocols for data exchange a connection via the Internet to the ROMAN is possible. This architecture has several advantages: computational power of various computers becomes available, different hardware and operating system platforms may be used, and remote access and visualisation of dynamic changes to the model is provided.

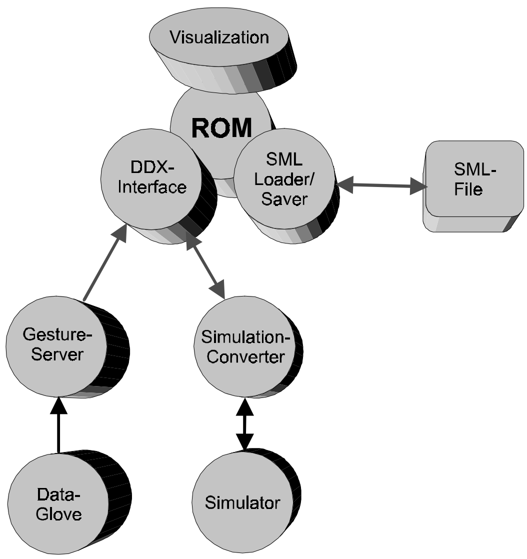



**Fig. 3.** System Architecture of the Modeller

The gesture and grasp recognition algorithm is based on techniques of statistical multi-variate discriminance analysis (Brauer 94). Different grasp- and gesture-patterns can be defined using a teaching software. The user teaches the algorithm with some examples and for each example a set of characteristic features is computed. While acting on the complex objects, the gesture server continually tries to match an actual feature vector with one of the taught patterns in the n-dimensional feature space. In case of a match, the object is bound to the position-path of the data glove. If the bending values of the fingers change significantly (the hand opens), the recognition indicates a release event and the object is left at the actual position. The statistical recognition algorithm performs well and runs in real time on a dedicated 4/86 CPU. The modularity of our concept allows the integration of further improvements in pattern recognition.

To make the computer redo or even derive programs from what was previously demonstrated by the user, is an old issue in human-computer interaction. This research is currently focused on the conventional WIMP (Windows, Icons, Menus and Pointers) interface styles. The 3D interface provided by our concept offers new possibilities for specifying dynamic model behaviour by demonstration. The investigation of this potential is one of our main goals of research.

In a first approach, a scenario out of the domain of production and logistics was constructed. A conveyor belt delivers containers of different types (represented by differently coloured blocks). A robot has to transport these containers to one of three outgoing conveyor belts which convey them to further places of processing. The assignment of a specific container to a target conveyor depends on its type (colour). For a concrete situation these assignments have to be specified by the user. Additionally, a control program for the robot which picks up the containers and places them on the target conveyor must be created. In the following it is described how these tasks can be performed with the interface.

The initial scenario described above was created with our modeller. The containers are located in the object box which in this case is simply a dedicated area on the tabletop. Now, the user takes the containers and moves them through the system on individual paths. While putting a container from one conveyor to another the user plays the role of the robot that picks, transports, and releases containers. Furthermore, the modeller recognises the assignment of a specific type of containers to a target conveyor belt. According to our philosophy the movement paths are recorded, can be saved and animated. A path which bridges a gap between different conveyors may be refined with a path editor, and a basic version of a robot control program can be generated.

In addition to continuous path-control programs, rules for the distribution of the typed container belts within the system are generated, for example: *„put green containers always on conveyor A“*. The rules and control programs can be used for simulation. Randomly created containers are moved through the virtual conveyor system according to the taught set of ramification rules and paths. This way, experimental changes of the material flow through the system can be easily and intuitively analysed and optimised. Furthermore, by scaling the model and the paths to the size of a real plant, the control programs for the robot can be simulated. For this purpose a robot simulator (COSIMIR) is employed. It offers various types of robots contained in a library. This simulation tool provides the functions to make unreachable co-ordinates visible and to optimise transport curves.

The overall behaviour of the system can be simulated with a universal simulation tool (simple++). The controlling algorithm for the behaviour of the components can be taught by concrete demonstration of hand movements using real tokens, generating Petri-Nets. These tokens are placed on certain points to mark relevant states of the system (Fig. 4).


**Fig. 4.** Generating Petri-Nets by Concrete Demonstration

This control algorithm can then be used to drive the real model, enriched by actors and sensors, see Fig. 5.

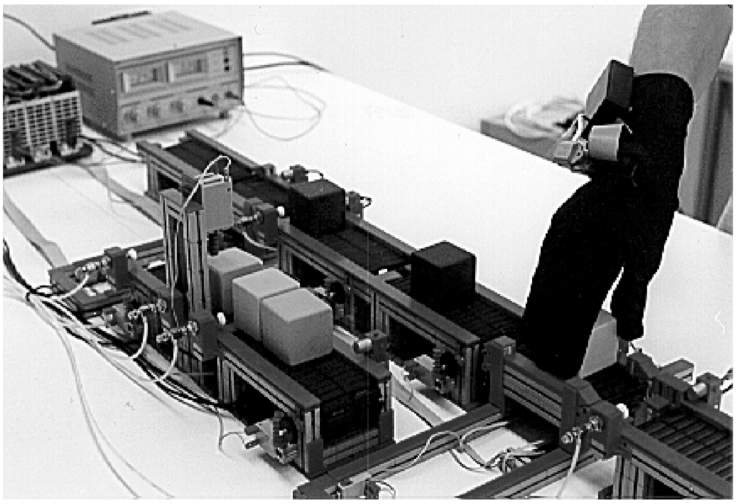


**Fig. 5**. Model with actors and sensors driven by a Programmable Logic Controller (PLC)

In our modelling environment, physical and virtual objects are tightly coupled by sensored user hands. The experience with prototypical applications shows some major advantages of this concept:

- The similarity between real and virtual objects supports the spatial and dynamic orientation in complex systems. Physical laws are carefully respected (spatial extension of bodies, steadiness of motion, friction, acceleration, synchronisation).
- The physical model can be viewed from different perspectives, without additional technical means like head mounted devices. The context as a whole is always preserved.
- The user senses the hardness and heaviness of the complex objects and uses them intuitively.

The power of this concept compared to conventional Graphical User Interfaces lies in its orientation towards all human senses during the modelling process, especially to the haptic. Instead of sensoring each object, the concept of utilising the hand, yields a universality, because we can use all familiar objects of our surrounding as user interface.

Conceptually our work can be seen as an extension and application of the Model-View-Control concept of Smalltalk80 (Goldberg & Robson 1983). The Model-View-Control concept separates model functionality from the user interface (Fig. 6). Whenever the model changes, it broadcasts a message ("I have changed") to all dependents and they take whatever action is appropriate. The controller tells the models what to change, the view displays the current state of the model from one perspective. We introduce double valued bi-directional links between real objects and their virtual counterparts, mediated by ROMAN, a real object manager, implemented as a software-component. These links are double valued as they allow the submission of state-information (view) and control data (controller) between the virtual and the real world (Fig. 7). They are bi-directional in the sense, that they can be used from the real and the virtual side. From the reality side, one can point on a certain element of the system and get a a video projection of information into the scene (R->V-view) or one can start a simulation in virtuality (R->V-control) having the visualization again being projected into the scene (Fig. 8). From the virtual side, one can point on a certain virtual element of the system and get the video picture of the real system (V->R view). Starting a control program of the generated type, one can not only see the simulation on the screen or projected into the scene but also drive the real actorized model (V->R control).

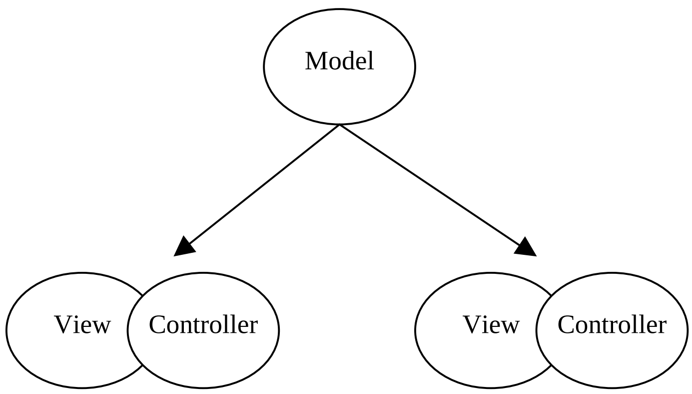


**Fig. 6.** Model-View-Controller Concept of Smalltalk

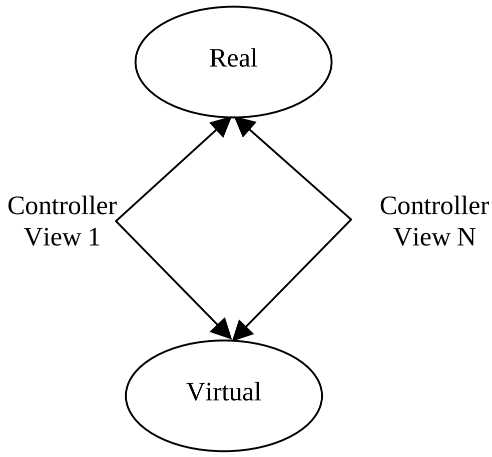



**Fig. 7.** Bi-directional Model-View-Controller Concept

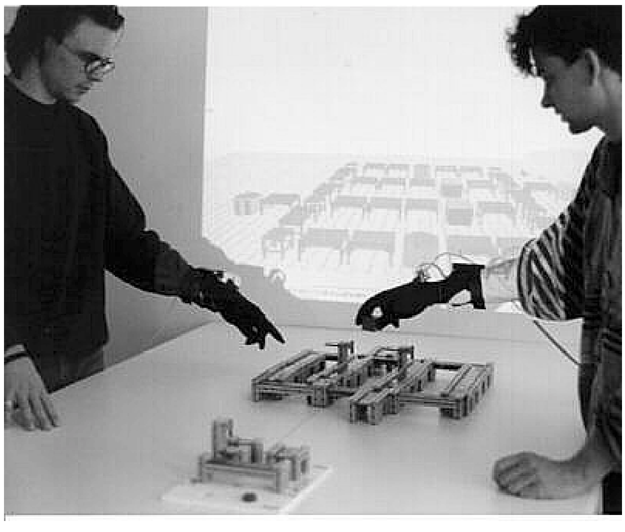


**Fig. 8.** Projection of Information into the Scene

## 3 Applications

In three projects, two funded by the German Research Community (RUGAMS and EUGABE) and one by the European Community (BREVIE) we demonstrate the feasibility of the above concept.

*Project RUGAMS*

First prototypes of a flexible assembly systems designer have been improved in the project RUGAMS (*Computer supported Crossing between concrete and abstract Models of Production Systems).* With a simple data glove and feature based versions of grasp recognition, a "brick world“ and a virtual world can be geometrically manipulated as shown in fig. 3 (Bruns 93, Brauer 96, Bruns et al. 97). In addition to geometrical modeling it is now possible to teach dynamic behavior and decision rules by concrete demonstration. Individually taught behavior patterns are abstracted and then used to create machine control programs. The prototype has been demonstrated for a Conveyor System of the type being subject to considerable investigations in the European Simulation Community (Krauth 1992).

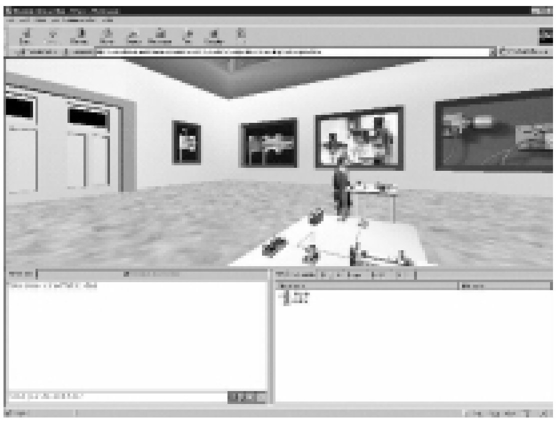
For each conveyor belt type we construct a geometrical and functional virtual representation with conventional modeling tools. These virtual building blocks are then imported into the Real Reality modeler. Using a data glove, the user teaches his way of grasping the real objects. He thus associates a grasp pattern to each type of object. After that, the user places the conveyors on the model ground, thus building a conveyor system. Conveyor belts may be connected or gaps between them may be bridged by a robot system. Now, different types of containers can be taken out of their starting position and moved through the system of conveyors on individual paths. These paths are recorded and abstracted to a parametric representation which then can be interactively edited in the virtual scenario. From this internal representation we are able to generate program code for the control of industrial robots and PLCs. The virtual model is now used for systems analysis. Randomly created containers enter the system and pass their way through it, activating the relevant robot programs according to the taught set of ramification rules and pathways.

With this prototype we can demonstrate a new and efficient way of system specification, programming, testing and optimization. Our reality oriented method is especially suited for new forms of communication between customer and system developer.

*Project EUGABE*

In project EUGABE (*Experience oriented Bridges between real and virtual Modeling-worlds for vocational Training),* we apply the idea of coupled modeling to the area of pneumatics and vocational training. This field is difficult, because pneumatic circuits in reality are very disordered and have crossing air tubes connecting the cylinders, valves and switches; furthermore they are flexible, not rigid like conveyors, and the physical laws of pneumatics are complicated enough so it is always possible to find a perspective, where the real and the virtual model differ (Fig. 9). Many students prefer to build a real physical model to understand the principles. On the other hand, the advantages of virtual and abstract modeling show up very clearly as soon, as the models are getting complex enough. We found, that it is a very interesting, yet not enough investigated question, how and when thinking styles may switch from physical to logical orientation and vice versa, depending on the problem and the representing modeling languages. Further pedagogical and psychological research will result from this project.

The technological challenge of not being able to apply our grasp-pattern method to a flexible and bending object (how do you recognize the sliding of a tube in your hand?) may be solved by imposing procedural constraints on the user. When she connects two pneumatic devices she is not allowed to connect one end of a tube, leave the other end, do some other work and then decide to connect the other end. The user always has to do her work in the sequence: 1. grasping a tube, 2. connecting one end to one component and 3. the other end to the other component. This certainly is a restriction which is not acceptable for a free intuitive and experimental work. Therefore we left our pure hand orientation for this application type and switched to a video-image recognition of the modeling parts and actions. This solution is being further investigated in project BREVIE.



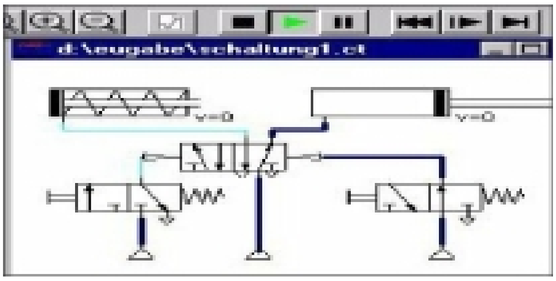
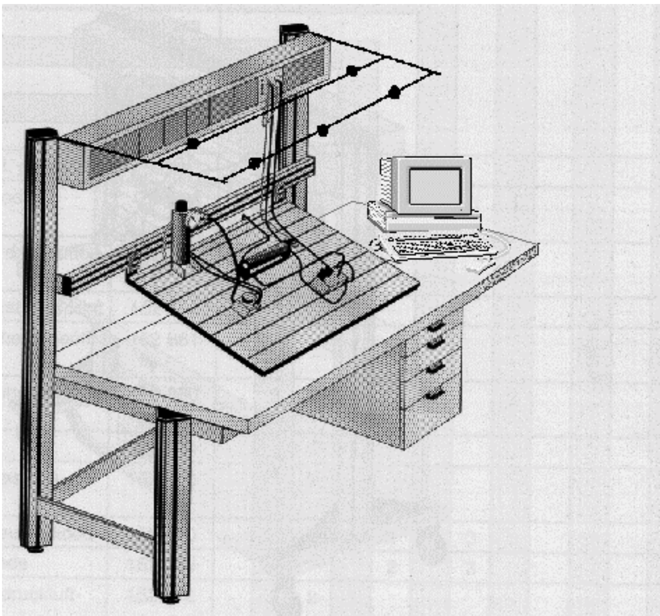



**Fig. 9.** Group oriented learning of Pneumatics

*Project BREVIE*

This European project *Bridging Reality and Virtuality with a graspable User Interface*[1] aims at the development of a product, that eventually may replace all conventional pneumatic learning environments. The new learning environment consists of a table for the placement of real pneumatic elements, a camera-system for the recognition of elements and actions and a PC for 3D-VR presentation, simulation, animation and multimedia support for the subject (video films, sounds, demo applications). The real and virtual worlds are coupled, so that the learner can switch between different views and action rooms (Fig. 10).



[1] Our partners in this project are Festo Didactic, Virtual Presence, Superscape, Stockport College, Friese Poort Drachten, Escola Superior Leiria, Schulzentrum am Holter Feld, Institut for Work-Psychology ETH Zürich

**Fig. 10**. Learning Environment for real and virtual Pneumatics

The Model-View-Control concept allows the distributed use of this environment. One Real Object Manager is assigned to one real workspace, but can offer its services for world wide Internet-Clients that understand VRML.

## 4 Conclusion and further Work

In this paper, I introduced a concept of complex objects and a kind of reality oriented modeling which is closely linked to virtual modeling. Compared to conventional Graphical User Interfaces the power of this concept lies in its orientation toward a variety of our senses during the modeling process, especially to the haptic one. By using direct manipulation of real objects as a computer interface and by integrating this principle into the learning process, as we do, a new approach to human-computer interaction is followed. It supports the process of concrete modeling and it constitutes a basis for cognitive abstractions, thinking, and the formation of concepts. Our approach raises questions of cognition and system theory. How do we grasp tools and parts? In which way are mental models affected by the acts of grasping and concrete manipulation? What working and learning styles are preferred by students or system designers if they can freely switch between different modes? Some of these questions are investigated in a current evaluation project where we observe three different groups of students: one group is learning pneumatics by blackboard teaching, one by simulator support and one with the new real-virtual modeling environment (Grund 99).

Further technical work will be concentrated on possibilities of tele-modeling and reality oriented distributed cooperation. With devices of light-projections, it is possible to point into the real scene and synchronously into a virtual model at another place. This may considerably improve communication in service work for automation systems.

**Acknowledgments**

I would like to thank my colleagues for their valuable contributions: Volker Brauer, Hauke Ernst, Hermann Gathmann, Eva Hornecker, Jürgen Huyer, Dieter Müller, Mario Müller, Bernd Robben, Ingrid Rügge, Kai Schäfer, Kai Schmudlach and all of my group's graduate researchers: Cedrik Duval, Martin Faust, Mladen Ilic, Piotr Kaczmarczyk, Keith-Andre Piedboeuf, Sven Ruttkowski, Wolfgang Tieben.

Our research is being supported by the Deutsche Forschungsgemeinschaft DFG (G-Nr. Br 1556/2-3, G-Nr. Br1556/3-3) and the EU (MM1002).

## References


1. Brauer, V. (1994): Feature-basierte Erkennung dynamischer Gesten mit einem Datenhandschuh. *Diplomarbeit*, Universität Bremen
2. Brauer, V. (1996): Simulation Model Design in Physical Environments. ACM SIGGRAPH Computer Graphics, Vol. 30, No. 4, Nov.
3. Bruns, F. W. (1993). Zur Rückgewinnung von Sinnlichkeit – Eine neue Form des Umgangs mit Rechnern. *Technische Rundschau,* 29/30, 14-18
4. Bruns, F. W. (1998): Integrated Real and Virtual Prototyping. Proc. 24th Ann. Conf. IEEE Industrial Society, Aachen, Vol 4, 2137-2142
5. Fitzmaurice, G. W., Ishii, H., Buxton, W. (1995). Bricks: Laying the Foundations for Graspable User Interfaces. CHI`95 Mosaic of Creativity, 442-449.
6. Goldberg, A. Robson, D. (1983). Smalltalk-80: The language and its implementation. Addison-Wesley, Reading, Massachusetts
7. Grund, S. (1999): BREVIE Evaluation. Presentation at the Concertation Meeting of the EU, Brussels, 22. 1. 99 (soon to be published)
8. Ishii, H., Ullmer, B. (1997). Tangible Bits: Toward Seamless Interfaces between People, Bits and Atoms. CHI'97, Atlanta, Georgia. http://media.mit.edu/groups/tangible
9. Kang, S. B., Ikeuchi, K. (1994). Grasp Recognition and Manipulative Motion Characterization from Human Hand Motion Sequences. Proc. of IEEE Int. Conf. on Robotics and Automation, San Diego, Cal., Vol 2, 1759-1764.
10. Krauth, J. (1992): Comparison 2, Flexible Assembly System. EUROSIM Simulation News 2, May 1992.
11. Resnick, M. (1993): Behavior Construction Kits. Communications of the ACM. 36(7), 64-71
12. Rekimoto, J. (1998). Multiple-Computer User Interfaces: A Cooperative Environment Consisting of Multiple Digital Devices. In: Streitz, N., Konomi, S., Burkhardt, H.-J. (Eds.) see next. 42-52
13. Streitz, N., Konomi, S., Burkhardt, H.-J. (Eds.) (1998). *Cooperative Buildings - Integrating Information, Organization and Architecture.* Proceedings of the First International Workshop on Cooperative Buildings (CoBuild'98), Darmstadt, Germany (February 25-26, 1998). Lecture Notes in Computer Science, Vol. 1370. Springer - Verlag, Heidelberg.
14. Suzuki, H., Kato, H. (19195). Interaction-Level Support for Collaborative Learning: AlgoBlock – An Open Programming Language. *Proc. of the Computer Supported Collaborative Learning (CSCL) Conf*., University of Indiana
15. Weiser, M. (1991). The Computer for the Twenty-first Century. Scientific American, vol 256, (3), 94-104.
16. Wellner, P., Mackay, W., Gold, R. (1993). Computer-Augmented Environments: Back to the Real World. Communications of the ACM, 36, 7, pp. 24ff