Synchronous Model Creation in Real and Virtual Spaces

Eckhard Meier
GMD – German National Research Center for Information Technology
Schloss Birlinghoven
53797 Sankt Augustin
GERMANY
Tel: +49 (-2241) 14 29 51
E-mail: eckhard.meier@gmd.de

Wilhelm F. Bruns
artec – Research Center for Work, Environment, Technology
Bremen University
Enrique-Schmidt-Str. 7
D-28334 Bremen, Germany
Tel: +49 (-421) 218 42 06
E-mail: bruns@artec.uni-bremen.de

ABSTRACT
In this paper an augmented reality system is introduced that is designed to support synchronous model creation in real and virtual spaces. Based on the approach of complex objects, which represent artifacts composed of real and virtual parts, real building blocks are used to simultaneous create physical and virtual models. By tracking the movements of a user's hand, the system generates a virtual reproduction of the original construction. To ensure extensive conformity of the real and virtual models, the system transfers the substantial behavior of physical building blocks to their virtual counterparts.

KEYWORDS: Augmented Reality, Tangible User Interfaces, Direct Manipulation, Geometric Modeling

INTRODUCTION
Physical models may be helpful for a common understanding in multidisciplinary design teams but may also improve the mental ordering and understanding of difficult technical matters. This is not only the case for the specification of geometry and topology, but also for the dynamic behavior of systems. On the other hand, virtual models are relatively free from physical restrictions, giving more possibilities for creative experimentation. Therefore, we combine physical with virtual modeling. The concept of *complex objects*, having one real part closely coupled to corresponding virtual parts has been applied in several modeling environments [3]. Real and virtual parts can be synchronized if we know the exact position and orientation of the real part. This can be achieved mounting sensors on the real part, on the modeling hand or by using a camera. This paper concentrates on sensorizing the user's modeling hand by the use of a data-glove and a tracking system.

Several prototype applications have been developed that use *complex objects* to construct a system in reality and synchronously generate a corresponding virtual model, which can be tested, analyzed and transmitted to remote places. Technical details of the implementation of this concept have been described elsewhere [4]. The use of sensors on our hand as a manipulator of physical objects in a real environment requires gesture recognition algorithms. We used those based on statistical methods [2]. The raw interface data are analyzed and gestures, grasps and user commands recognized. Based on this information, the virtual representation of the physical environment is updated. Even though the interface of the system becomes a passive observer of user activities, information about the real, as well as the virtual part of an object has to be handled system internally. Due to this direct interrelation of

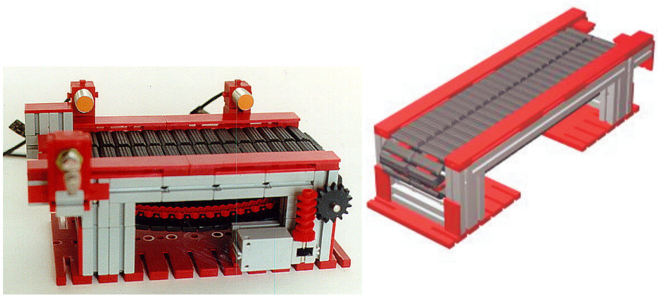


Figure 1: Physical and virtual model of a conveyor belt

the real and the system-internal part of an object, we call them *complex objects*.
Complex objects are one of the fundamental elements of our concept. As a matter of principle they are composed of a real and a corresponding virtual part. In order to be able to instantiate a complex object, a discrete element for each component must be provided. The physical part of a *complex object* must be constructed by using technical construction kits, wooden bricks or other materials,

whereas the virtual counterpart has to represent its appearance and dynamics as show in figure 1. Some prototypical applications proved some major advantages of the approach:

- The real and virtual object’s affinity enhance the spatial and dynamic orientation in complex systems. Physical laws are complied by design.
- Physical models instantly can be viewed from different perspectives, without the use of technical means like head mounted devices. The overall context is always preserved.
- Users senses the hardness and heaviness of complex objects and uses them intuitively.

The power of this concept lies in its orientation towards all human senses during the modeling process, especially to the haptic. The concept of utilizing the hand instead of sensoring each object, yields universality, since arbitrary objects of our surrounding can be used as part of the interface.

RELATED WORK

Augmented reality [1] is not a principally new technology, but is one main research area that aims to combine real and virtual information spaces and for this purpose incorporates tangible interfaces.

The Bricks prototype [5] uses physical items for the manipulation of synthetic objects. These artifacts can be temporary attached to system internal entities and hence can act as specialized, space-multiplexed input devices. The realized prototype of an “Graspable User Interface” uses two Ascension Flock of Birds receivers as graspable bricks, operating on a rear projected desktop surface called “Active Desk”.

The BUILD-IT [8] system is based upon the concept of a “Natural User Interface”, that is to empower computer interaction in a natural way using all of the users body parts. The BUILD-IT application supports engineers in designing assembly lines and building plants. The technical infrastructure is based upon a tabletop interaction area, enhanced by a projection of a 2D computer scene on the tabletop. Additionally, a video camera is used to track manipulations of a small, specialized brick, that can be used as an “universal interaction handler”.

DigitalDesk [11] is one of the first augmented reality environment systems. Its main approach is to shift functionality from a workstation onto a desk instead of adding further desktop properties to a workstation as it is done in traditional graphical user interfaces. The DigitalDesk application is focused on direct computer-based interaction with selected regions of paper documents.

The Studierstube system [10] developed at the Technical University of Vienna uses light-weight head-mounted displays to project artificial 3D objects into the real world. The approach introduces the Personal Interaction Panel [9] as a new input device for augmented reality. Virtual objects such as maps, button, or sliders are then projected onto the panel and can be selected by the pen. While the use of the PIP shows that augmented reality can be used to realize new interaction mechanisms, the interaction techniques are basically the same as used within conventional immersed VR environments.

The MIT Media Lab [6] introduces the vision of Tangible Bits [7] that allow users to grasp and manipulate digital information by coupling them to everyday physical objects and environments. The design of Tangible User Interfaces and ambient media is focused on foreground activities of the users as well as peripheral background information. Several prototype systems such as metaDESK, transBOARD and ambientROOM have been developed, that take the concept of a Tangible User Interface into account.

The 3D modeling system presented is based on the concept of *complex objects*. Its general approach is to enable users to simultaneously assemble complex static models in real and virtual space by using elements of a construction kit. By tracking the user’s hand movements, the system generates a three dimensional virtual reproduction of the original model. In addition to the management of discrete model components, the system performs an analysis of the current scenery. The spatial layout of individual building blocks is interpreted in such a way that interrelated elements are grouped according to their structure and are combined to interconnected units

SYSTEM SPECIFICATION

The 3D modeling system presented is based on the concept of *complex objects*. Its general approach is to enable users to simultaneously assemble complex static models in real and virtual space by using elements of a construction kit. By tracking the user’s hand movements, the system generates a three dimensional virtual reproduction of the original model. In addition to the management of discrete model components, the system performs an analysis of the current scenery. The spatial layout of individual building blocks is interpreted in such a way that interrelated elements are grouped according to their structure and are combined to interconnected units.

A system that aims at the construction of complex models should realize a certain degree of flexibility to ensure its universal applicability. To meet diverse application specific requirements, some general assumptions should apply to the modeling system.

Different modular construction systems can be used in a wide field of applications thanks to their multifaceted structures. The choice of an appropriate construction system particularly depends on the model’s type of construction and its purpose of representation. An architectural model for instance would be build by a construction kit that allows to create flat, closed surfaces, since the model should give a realistic visual impression of the final product to its viewers. On the other hand the main focus of a mechanical or electronic model is more likely to represent operational aspects. Consequently construction kits that emphasize technical aspects and functional interrelations would be the preferred choice in such cases. A system that aims at the support of synchronized model assembly must be capable of supporting a variety of heterogeneous modular construction systems to ensure its applicability in a wide field of application.

Granted that model construction is a highly interactive, dynamic and creative procedure, it is important to perpetuate the user's flow of work. Users should be able to concentrate on their modeling activities without getting interrupted by the system. The system's user interface should passively operate in the background without getting noticed by the users explicitly.

The key functionality of a physical construction kit must not be restricted by its virtual counterpart. A system has to completely reproduce the constructional methodology of a modular construction system in order to prevent principal limitations of its element's combinational properties on a structural level.

The use of construction kits is characterized by a continuous variation of its element's spatial arrangement. This process of physical model creation is under total control of the participating users and should not be derogated by the system. Thus the system's main task of mapping physical model modifications to its virtual complement has to be performed in real-time in order to avoid synchronization delays of the real and virtual scenery.

SYSTEM DESIGN

To make a system meet the general requirements described above, it principally has to be capable to generate a virtual reconstruction of a physical model by tracking user activities. In general this problem domain of real and virtual model synchronization can be subdivided into two distinct categories:

- On a functional level the system has to reproduce the model assembly process comprehensively. Its main task is to recognize modifications of the physical model and to map each alteration to its virtual complement.
- On a structural level the system has to completely reproduce the principal design characteristics of a modular construction system to be able to reproduce the connecting behavior of each element.

Because the synchronization process of the real and virtual scenery is based on the tracking of the user's hands movements, the replication of the physical model's topological structure solely relies on the system's internal description of the modular construction system's structure. Accordingly a comprehensive definition of a construction kit's constructional methodology is an essential precondition in order to perform a proper interpretation of the model assembly process. The system presented specifically is designed to handle construction kits that are based on plug connections.

Structural System Design

In general the modeling system's structural design reflects each structural component that is significantly involved in the process of model assembly reproduction. Each of those components can be assigned to one of two mutual exclusive categories distinguishable by their level of abstraction. While components of the one category more or less bear upon reality, the others are purely fictional constructs that are only relevant for the virtual simulation of the physical model's topology.

Conceptual objects define abstract structures that are of major importance for the constitution of a modular construction system, but beyond they are not directly involved into the process of model assembly reproduction. The general purpose of conceptual objects is to define the *concept* of a modular construction system's functional principle and to make it available during model assembly processing. In general conceptual objects recreate causalities and behavior patterns that in reality are naturally given.

Representative objects reference aspects of the physical environment. A plenty of environmental items, persons and processes are of importance in the context of physical model construction, and thus have to be *represented* within the modeling system to make it capable to transfer the an model's formation to its virtual counterpart. Accordingly appropriate structural components are needed that adequately reproduce the actual circumstances.

Each structural component presented in the following is uniquely assigned to one of these categories.

Modular Construction Systems generally define and enclose the fundamental static functionality of a construction kit. They consist of a set of basic elements, which follow a homogeneous combinational principle and as a whole determine the constructive potential of a construction kit.

Within the scope of a modeling system's structural design, modular construction systems belong to the category of conceptual objects, since they themselves are not directly involved into the concrete process of model assembly. In fact the modeling process rather depends on the topological structures provided by the modular construction system, because the reproduction of the model's construction is impracticable without fundamental information about the combinational characteristics of the underlying construction kit.

Element Types uniquely identify a specific kind of building block and define its elementary static attributes. Since the individual appearance of each element type also determines its structural connecting properties, it also vitally affects the substantial behavior of a building block.

In principle each element type determines which building block interconnections can be established under what circumstances and thus directly influences the general construction methodology of the modular construction system it is part of.

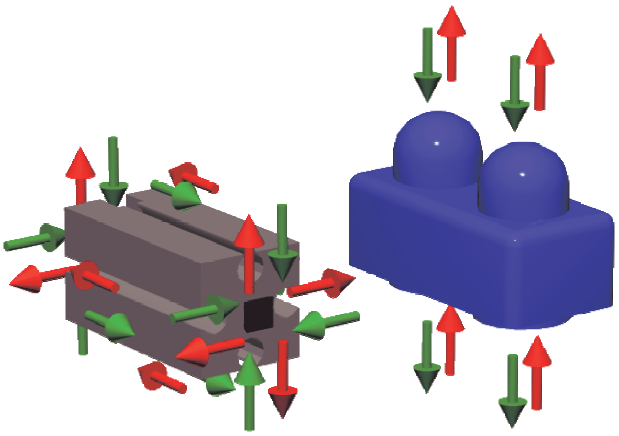


Figure 2: Connecting properties of different element types.

To preserve the conformity of this methodology, all element types of the same construction system must be equally dimensioned and have to follow a consistent combinational principle. This principle specifies, which building blocks can be combined in general and in which way a connection has to be established. As a result element types, seen as a single entity, compose a homogenous structural configuration characterizing the topological and functional aspects of a single modular construction system. Within the modeling system, element types represent conceptual objects that have the following attributes:

- Object geometry is a fundamental component to determine an element type's appearance and thus is an essential need to properly determine an element's substantial behavior. The system references external files to import appropriate 3D geometry descriptions.
- Scaling information is added to the element type's structure, to keep the dimension of a virtual model flexible. The system stores an element type's height, length, width and its local origin.
- Joint attributes of an element type define potential connections to other elements. Each element type can own a number of individual joint attributes, which as a whole define the element's individual connecting properties. Figure 2 shows two different element types and their specific connecting properties. During the term of system design it turned out that the reproduction of an element's connecting behavior in real-time necessitates a relatively complex description of a connection's attributes. Thus an additional conceptual structure was introduced in order to encapsulate this connection specific information. As a result the system only has to retain the number of potential connections and their identities.

Joint Structures define a set-up for potential element type connections. Since the configuration of a connection individually depends on a dedicated element type, each joint structure is directly associated to its parent element.

Three partially interdependent parameters compose a joint structure, which enable the system to reproduce and to validate an element type's physical connecting and disconnecting behavior appropriately. Figures 3 and 4 illustrates the joint structure's parameterization.

- The origin of a joint structure determines where at the element type a valid connection can be established. Its position is defined in the local coordinate system of the parent element type and is not principally restricted. Thus it even can lie outside of the element types geometry, as required in some cases.
- The *Direction of Connection* (DoC) in conjunction with the actual *Direction of Movement (DoM)* vector is used to determine on what terms a change of a connection state is possible. The *Direction of Movement* vector represents an element's actual path of movement, which in essence is the only real world information available to the system. If a correlation of the DoC and DoM vectors is determined by the system, an element type's connection can potentially by establish or revoked.

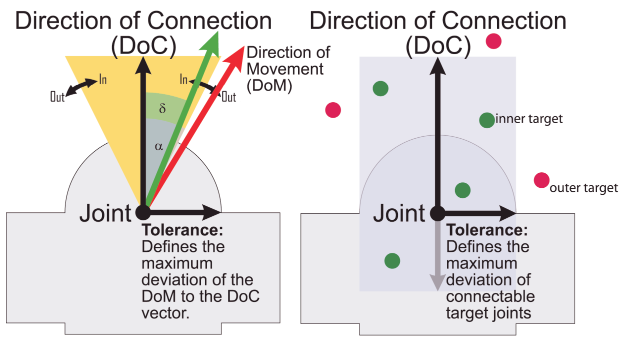


Figure 3: Graphical representation of a joint structure.

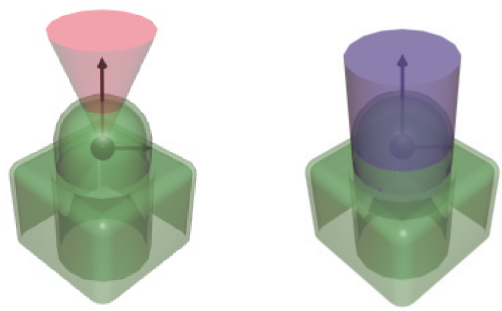


Figure 4: 3D visualization of a joint structure.

- A *Tolerance* vector perpendicular to the DoC vector is used to add robustness to the joint structure. Its main purpose is to eliminate discrepancies due to tracking data inaccuracies. In combination with the DoC vector it defines the maximum deviation of the DoM vector from the DoC vector. A joint remains connectable as long as the actual DoM vector lies within the cone spanned by the DoC and *Tolerance* vectors. Furthermore the *tolerance* and *DoC* vectors dimension a cylindrical region, which is used to determine connectable target joints (compare to figure …).

Building Blocks embody the teeniest manipulable elements of a construction kit and thus can be seen as the essential physical parts within the model construction process. Several such elements can be used to compose variably complex arrangements, which hardly ever get completed as most often a crucial brick is missing. This chronically absence can be traced back to the fact that building blocks are representative objects and thus correspond to physically existent, quantitatively restricted elements.

Even though building blocks and element types belong to different structural categories, a close relationship between these components exists. Each building block is uniquely associated to an element type, which determines its static characteristics, and thus serves as a structural template. This concept of building block instantiation guarantees the persistency of the virtual model's topology, because all elements that are involved into the modeling process automatically rely on the same combinational principle predetermined by the underlying modular construction system.

Beside this structural relationship, building blocks have additional properties, which compose this type of object. These properties substantially apply to the dynamics of a building block during the term of model construction and essentially reflect a building block's spatio-temporal behavior and its state of connectivity at a given time.

To be able to represent the essential characteristics of a building block within the modeling application, the system-internal structure defines the subsequent attributes:

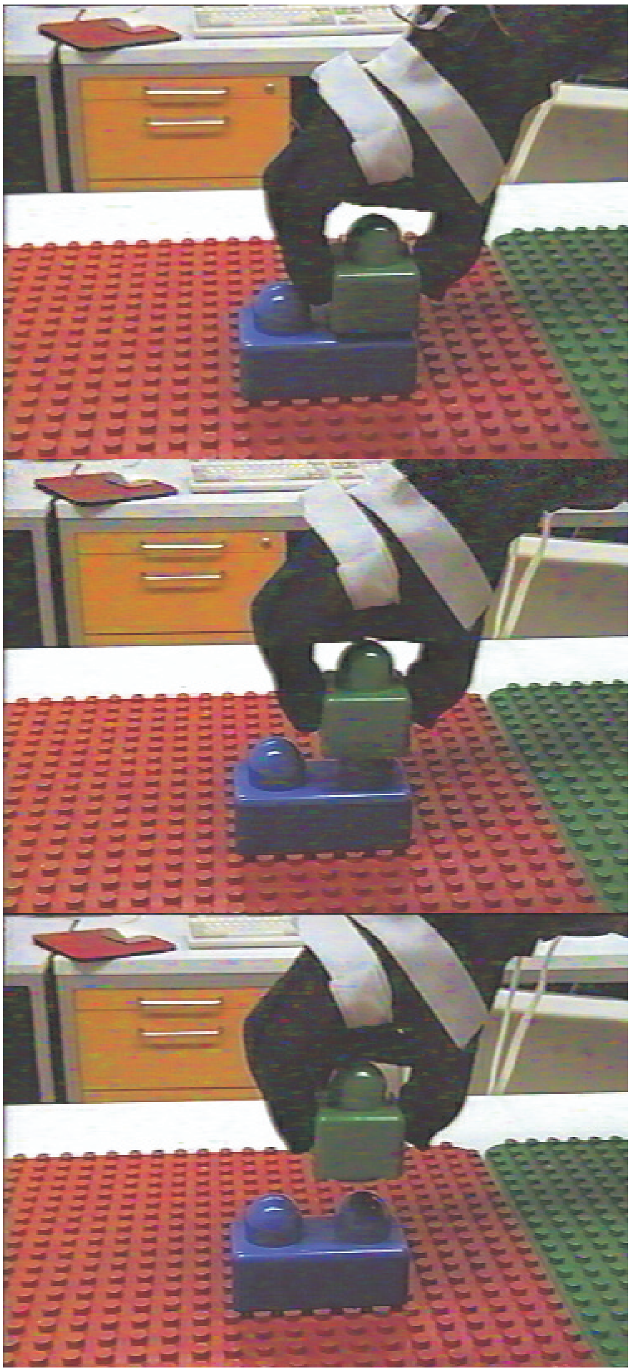
- Since the static structure of a building block is encapsulated in it's underlying element type, permanent and fast access to that element type has to be available. Thus each building block maintains a reference to its base element type.
- The actual position and orientation of a building block are of vital importance within the modeling process. Furthermore their reference frequency is superior to their modification frequency. Thus it is reasonable to store the position and orientation of each building block individually.
- Each instantiated building block imperatively belongs to a model. Because models are meta-structures that dynamically are created and destroyed at runtime, they cannot be uniquely identified by the system. In consequence each building block must be capable to autonomously identify its model affiliation by maintaining an appropriate model link.
- One of the most important properties of a building block is its connection status. This status is defined by the summation of each potential connection's state. In analogy to the joint structures of an element type, the modeling system treats these connection state information as independent objects. Consequently for each joint structure an equivalent joint object exists. Accordingly, each building block owns a fixed number of joint objects, which can be stored at the same place.

Joint objects are representational objects that control a building block's possible connection. Their internal structure essentially rests upon their base joint structure, which characterizes a potential connection's fundamental behavior. In contrast, a joint itself represents the state of a connection at a given time. Thus joints do not have a direct physical counterpart, but rather reflect the real world status of a building block. To be able to represent a building block's connection status comprehensively, a joint object has to define the following attributes:

- The general static configuration of a connection is defined by a joint structure. To gain access to this basic structure, an appropriate reference to the joint structure exists.
- A connection basically can show two distinct states. A connection is closed as long as two different joints are linked together. Otherwise a connection is open. Within the model assembly process the connection state of a joint frequently changes and must be updated accordingly.
- Because a joint without a relation to a building block is useless, a joint always defines a reference to its owner.
- Target joints are stored in dependency of a joint's connection state. As long as a joint's connection is established, a single target joint must be uniquely specified to keep the consistency of the system-internal model representation. Vice versa, while a joint is not in a connected state, a list of possible targets is maintained. In the moment of connection establishment, this list serves as the general basis of final target joint determination.

Models can be defined as a complex spatial formation, which is shaped by individual building blocks in the context of a predetermined set of rules. This individual constellation of elements as a whole can be understood as an independent representative object.

In the context of the construction process however the view of a model as an autonomous object is not useful. Instead it is more reasonable to interpret it as a dynamic arrangement. From this point of view, models basically represent a group of interconnected discrete elements, which generate topological structures of superior order. Within the modeling process, these building block configurations are subject to frequent changes, since new items often are added or removed. These dynamics have to be adequately reproduced by the system in order to ensure a precise synchronization of the real and virtual model's structure. In doing so some general axioms must be considered:



- Each model consists of a number of building blocks, which must be an interconnected network. If an element is disassociated from this network, a number of new models inevitably emerge.
- A building block that does not belong to such a network itself represents an autonomous model.
- Models are dynamically changing constructs, i.e. the number of a model's items is not constant. Thus no unique identifier of a model exists. It is exclusively identifiable on the basis its constituents.
- The topology of a model results from the existing connections between individual building blocks.

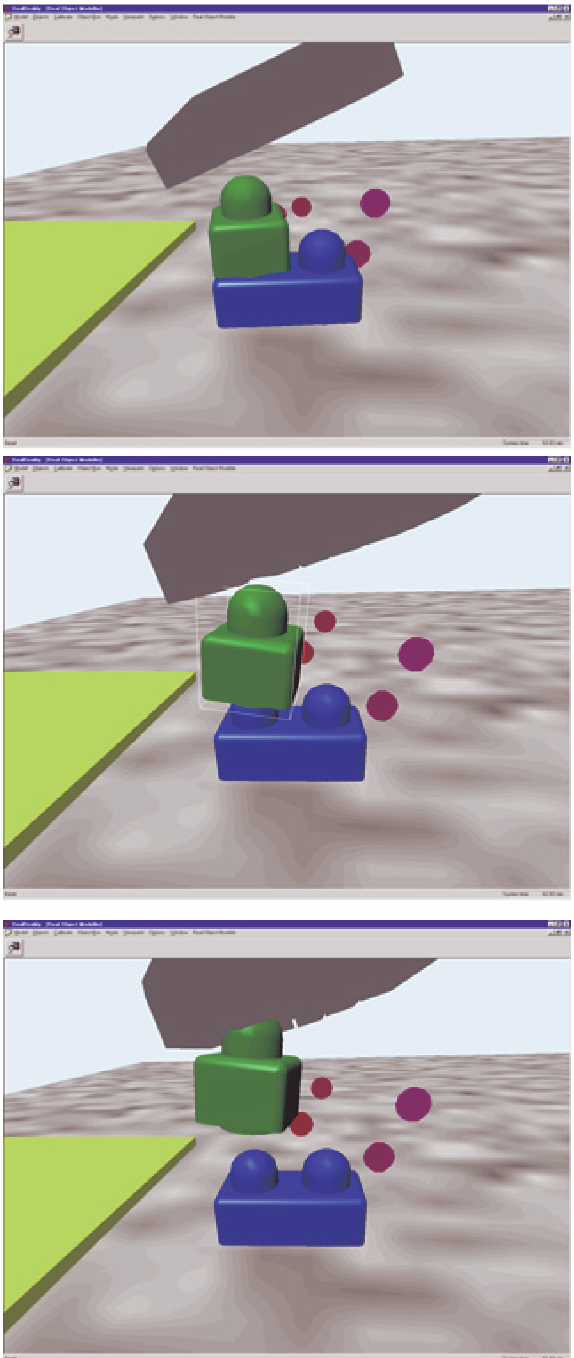


Figure 5: Illustration of the grasping phase.

Model Assembly Dynamics

The object classifications presented in the preceding section already show that the procedure of model construction is subject to a whole set of generalities and general conditions. These guidelines must be considered within the process of physical model synchronization, in order to be able to exclude misinterpretations of the real situation as far as possible. Thus the system's main task within that process is to recognize modifications of the physical model and to map each alteration to its virtual counterpart.

As already described, models consist of a number of individual elements, whose composition is determined by the user's imagination. This procedure of composing a model is an interactive, dynamic and time-progressive process that has to be reproduced by the application in real-time, but is under total control of the users. Those exclusively decide about a modification of a model by connecting, separating or repositioning individual items or whole groups of elements. Each step of modification therefore causes an alteration of the model's topological structure and results in a new model iteration. In consequence the overall constructional procedure can be seen as a sequence of distinct model states as shown in Figure 6.

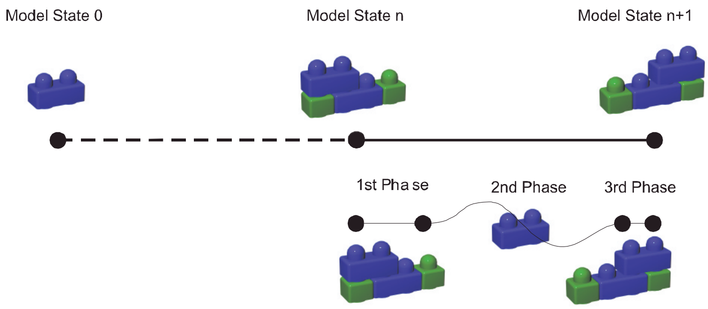


Figure 6: Model Assembly Dynamics.

But after careful consideration, it shows up that model state information do not allow any reasonable conclusions of the construction process time flow. Model states rather represent exactly those spatial layouts that have to be reproduced by the application. Instead, the in-betweens of each model state are of vital importance to the system, since these iterations are decisive for a structural alteration of a model. In consequence a precise analysis of consecutive model iterations is essential, in order to be able to properly reconstruct each model state.

Each iteration period again is subdivided into three successive sub-phases, which serve to precisely reproduce model alterations.

The 1st phase is the grasping phase that marks the beginning of a model alteration and is illustrated in Figure 5. A user automatically initiates this phase by grasping a building block. This action notifies the system that this building block presumably will be taken off the model by the user, in order to placed it somewhere else. In consequence, the system starts tracking the user's hand movements and in dependency of the initial brick's motion revokes each disconnectable joint of the grabbed element. Accordingly this separation process will subdivide the original model into a number of autonomous model subsets. The grasping phase directly terminates with model disassembly completion.

The 2nd phase is the moving phase that lasts as long as the user keeps holding the grabbed model. Within that period this model's position continuously changes. The system's main task is to constantly track the user's hand movements, to determine the actual *DoM* vector and to update the spatial arrangement of the virtual models. In a following step all potentially connectable joints of the grabbed model and corresponding targets of immobile models of the scene are determined in dependency on the *DoM* vector. Doing this preselection is of essential, since the termination of the moving phase by releasing the grabbed model is in total control of the user, and in consequence cannot be predicted by the system. Thus it is important to predetermine all relevant information for the subsequent assembly operation that depends on the grabbed model's motion.

The 3rd phase is the release phase that realizes the reconnection of the grabbed model to the stationary models of the scene. This process does not have a real world complement, due to the fact that the physical model immediately is in a valid static configuration. In contrast, this situation only has to be accomplished explicitly in the simulation. Since the formerly grabbed model still is an independent entity, it has to be integrated into the topological structure of the overall construction by linking all connectable joints. The successful completion of the release phase results in a consistent model state, which graphically and structurally represents the physical model adequately. Because at the end of the release phase no more changes of the model's topological structure are possible, the simulation process enters an idle state that keeps valid until revoked by the succeeding model iteration.

CONCLUSIONS

Coupling tangible objects of the real environment with digital information spaces let users profit from advantages of both areas. Making physical elements an integral part of the user interface, it significantly gets simplified since even unskilled and untrained users instantly know how to handle these items. Thus users can work in a very task oriented way, but at any time can exploit additional information provided by the virtual replica. In consequence the system's field of application directly depends on the attributes of the virtual elements, which again are defined by the construction kit in use. As a result, switching from one construction kit to another may also imply a change of application. Due to this flexibility the modeling system provides a basis for a wide field of spatial layout applications.

REFERENCES

1. Azuma, R. *A Survey of Augmented Reality*, Presence, Vol. 6, No. 4, ed. W. Barfield, S. Feiner, T. Furness III, and M. Hirose, MIT Press, Cambridge, MA, pp. 355-385 (1997).

2. Brauer, V. *Feature-basierte Erkennung dynamischer Gesten mit einem Datenhandschuh.* Universität Bremen, Bremen, 1994.

3. Bruns, F. W. Complex Construction Kits for Coupled Real and Virtual Engineering Workspaces. In *Norbert A. Streitz et al: Cooperative Buildings. Lecture Notes in Computer Science 1670*. Springer, Berlin, 1999, 55-68.

4. Bruns, F. W. Integrated Real and Virtual Prototyping, In: *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON 98)*, Aachen, 1998.

5. Fitzmaurice, G. W., Ishii, H., Buxton, W., Bricks: Laying the Foundations for Graspable User Interfaces, In *Proceedings of CHI'95 Conference on Human Factors in Computing Systems,* ACM, N. Y., 1995, pp. 442-449

6. Ishii, H. *Tangible Media Group Project List,* [www] media.mit.edu

7. Ishii, H., Ullmer, B., *Tangible Bits: Towards Seamless Interfaces between people, Bits and Atoms*, CHI'97, Atlanta, Georgia, 1997. [www] http://tangible.www.media.mit.edu/groups/tangible/papers/Tangible_Bits_CHI97/Tangible_Bits_CHI97.pdf

8. Rauterberg, M., Fjeld, M., Krueger, H., Bichsel, M., Leonhardt, U. & Meier, M. BUILD-IT: a video-based interaction technique of a planning tool for construction and design, In H. Miyamoto, S. Saito, M. Kajiyama & N. Koizumi (eds.) *Proceedings of Work With Display Units--WWDU'97*, pp. 175-176. Takorozawa: NORO Ergonomic Lab, 1997. [www] http://www.ifap.bepr.ethz.ch/~rauter/publications/WWDU97paper.pdf

9. Szalavári, Zs. and Gervautz, M., The Personal Interaction Panel - A Two-handed Interface for Augmented Reality, In *Proceedings of EUROGRAPHICS 1997*, 1997.

10. Szalavári, Zs., Schmalstieg, D., Fuhrmann, A., Gervautz, M. *Studierstube - An Environment for Collaboration in Augmented Reality*, Virtual Reality Journal, Springer, 1998.

11. Wellner, P. Interacting with Paper on the DigitalDesk, In *Communications of the ACM*, 36 (7), 1993, pp. 87-96.