

Integrated Real and Virtual Prototyping

F. Wilhelm Bruns
University of Bremen
Research Center artec
Bibliothekstr. MZH
D-28334 Bremen
bruns@artec.uni-bremen.de

Abstract - A Modelling Environment for the geometrical, topological and functional specification of mechatronic systems will be presented. The close coupling of real model components and their virtual counterparts allows a synchronous construction of real and virtual prototypes and their evaluation.

I. Motivation

In several industrial simulation projects we made the experience, that vividness, based on computer generated images alone, does not support the diversity of perspectives and creative design ideas we get if we are able to grasp and manipulate concrete parts of a complex system with our hands, be it a conveyor system, a flexible assembly system, a pneumatic or electrical circuit. Users often preferred physical models to communicate with each other, to understand complex details and to keep the overview. We also encountered this behaviour in the planing of complex assembly of process plants, where interdisciplinary groups were discussing on the basis of a physical model. Three-dimensional computer-aided modellings of dynamic processes supported by virtual reality techniques such as 3D-stereo viewing and n-D manipulation devices do not reach the usability (ease, concreteness, intuitiveness, directness) that we experience in modelling with real physical objects. Pure virtual prototyping may reduce cognition and communication. This personal experience, certainly not claiming for a generalisation (some gifted person may be as powerful and creative without such sensual feedback), motivated us to build a new learning and prototyping environment. We therefore developed the concept of a Graspable Real Reality User Interface which bridges two modelling worlds: the real world of physical objects and the virtual world of signs and images. Sensorized user hands couple physical objects of the real world to virtual objects, thus allowing fairly unrestricted manipulation and expression. In this way modelling with real physical objects can create an abstract virtual model. This can further be used for systems analysis and control optimisation.

In several projects we are working on the demonstration of the feasibility and the advantages of this new kind of learning and working environment for production engineering. Based on the combination of real world phenomena and virtual world simulations through a **Graspable User Interface** it is possible to freely change

between operations on real physical objects and their virtual counterparts. These **Twin-Objects** (Fig. 1), real and corresponding virtual components, compose a new kind of **complex construction kit** which represents a new kind of simulation technology: **Real and Virtual Reality**. In this complex construction kit, real physical parts have an adequate functional, structural and behavioural description, to build from these components a composite system which again shows the correspondence between the physical and the virtual system. This new environment allows the composition of a real system and the concrete demonstration of its behaviour by human hand operations (sensed by a scanning device), synchronously generating the control software which assures this behaviour.

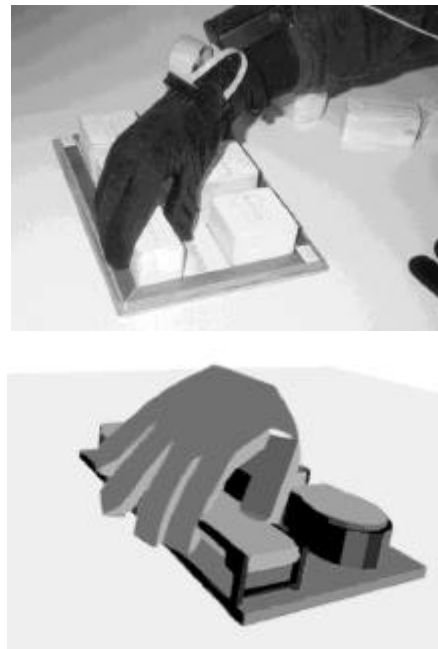


Fig. 1: Complex Twin Objects with real and virtual part

A prototype of this environment has been developed. Different real construction kits and simulators for electric, pneumatic, electro-pneumatic, hydraulics and robotics (Construction kits: Fischertechnik and Festo Didactic, Simulators: DigSim, FluidSim, PneuSim, Cosimir, Simple++) have been connected to a modeller for Virtual Worlds (Sense8 World Toolkit) via our Real Object Manager (ROM) [1][2][3].

II. Basic Concept

A new kind of user interface in shop floor and handicraft working has been described earlier [4][5]. The main characteristic of this "Real Reality User Interface (RR)" is the use of the user's 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 [6] 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 is 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 term *Real Reality* ironically reflects the term *Virtual Reality*, where the user immerses in and is surrounded by the interface. Real Reality means experiencing the real world, with all human senses being attracted, and communication within groups being encouraged. 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 these objects are called “twin-objects”.

Twin-objects are one of the basic elements of the RR 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 attributes describing the dynamic behaviour. The geometric description contains the object’s size (length, width, height) and its surface shape. On the other hand, the physical objects may be constructed using technical construction kits, wood or other materials. The object representations may vary in shape, size and level of detail. In initial state, a model is empty. The objects are located in an object box which has a predefined position on the tabletop (Fig. 2). Thus, for each object in the box the position can be computed.

After preparing the modelling elements and the data glove, the RR software is initialised and ready to use. A model is created stepwise by taking twin-objects out of the object box and putting them on the model ground. Because two models are managed synchronously, the RR systems provides two views on the same model. With the help of 3D visualisation software, the virtual representation is displayed on a monitor screen. 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 Reality Model Server.

Fig. 1 shows the virtual representation of a hand reaching for a twin-object contained in the object box, in this case a conveyor. It is taken out of the box and is placed to a location near another conveyor that was inserted into the model in a preceding step (Fig. 2). A grasp is recognised if a valid grasp-pattern occurs within the close vicinity of an object. If this happens a *grasp*

event is generated and the subsequent co-ordinates measured with the tracking system are recorded. By opening the hand the object is released, a *release event* is generated and the recording of the co-ordinates is terminated. By this procedure all manipulations, done with the objects are captured and ordered in time.

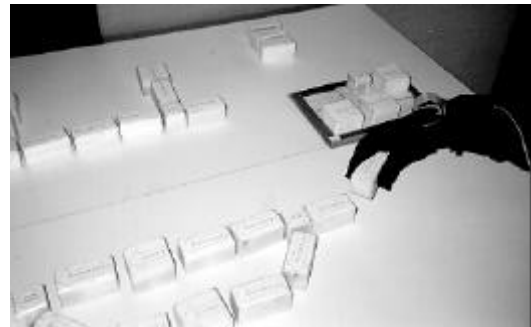


Fig. 2: Building a Model with Twin-Objects



Fig. 3: Synchronous Modelling of a Conveyor System

By pointing at a specific object the user can access information about it, while creating a model. The information is displayed on a monitor, projected on the tabletop or on a wall (Fig. 4).



Fig. 4: Modelling in Reality and Virtuality

When the model is complete, it is saved. For this purpose a special file format, the Simulation Modelling Language (SML), was developed. It contains the model's geometric description (position and orientation of each object) as well as the recorded actions (paths) performed on them. The SML representation is used subsequently for analysing the model, and also serves as a documentation of the modelling process which can be later recalled and graphically animated.

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 twin-objects,
- recognising grasp and gesture events,
- recording data of the object movements,
- visualising the modelling process and
- persistent storage of data in SML files.

According to this allocation of tasks, a general software architecture was designed (Fig. 5). A central component of this architecture is the Real Object Modeller (ROM). This module maintains an object database which contains geometric object descriptions, visualises a 3D graphical model, handles SML files, and provides an interface for the dynamic data exchange (DDX) with external 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 GestureServer task which handles the data glove and sends gesture event messages to the ROM. Another example is the SimulationConverter which acts as a mediator between the ROM and standard simulation software products. Fig. 5 shows bi-directional connections between the DDX interface and simulators. This indicates that an SML model is downloaded from the ROM, 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 with a conveyor to a new location. These changes are immediately transferred via DDX to the ROM where the virtual model is updated. With this link a dynamic visualisation of the experiments done with the simulators is achieved. We also developed visualisation clients which can be connected to the ROM via DDX. By using standard TCP-IP protocols for data exchange even a connection via the Internet to the ROM 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.

For the development of application software, the ROM may be used in two different modes. Firstly, it can act as a server which manages the basic RR features. In this case an application process which implements the application-specific functionality is connected via DDX to the ROM. The ROM passes all messages to the application where they are handled depending on the application's logic. Secondly, the ROM software may be used as an appli-

cation framework. Programmers may use the implemented software classes as a basis for their own developments and extend it with additional functions.

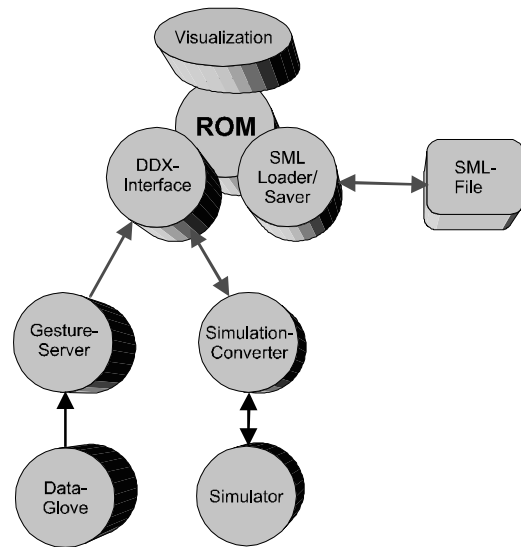


Fig 5: System Architecture of the Real Reality Modeller

The gesture and grasp recognition algorithm is based on the technique of statistical multi-variate discriminance analysis [6]. 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 (a feature vector) is computed. While acting on the twin-objects, the gesture server continually tries to match an actual feature vector with one of the taught patterns in the n-dimensional feature space.

The recognition function returns a value that indicates the distance of a given feature vector from the mean feature vector calculated from the set of examples of a specific gesture. If the distance is smaller than a threshold, a grasp or a gesture is recognised. If this happens, the gesture server subsequently tries to classify only this gesture. If the bending values of the fingers change (the hand opens), the recognition will fail and a release event is indicated. For a stable grasp, the releasing distance threshold may be greater than for recognition. 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 the RR 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 (Fig. 3). The conveyor in the foreground delivers containers of different types (here represented by differently coloured blocks) in the plant. A robot has to transport these containers to one of the three conveyors in the back-

ground 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 RR interface.

The initial scenario described above was created with the RR 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 RR application recognises the assignment of a specific type of containers to a target conveyor. According to the RR 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 is generated.

In addition to control programs, rules for the distribution of the typed containers within the system are generated, for example: *“put green containers always on conveyor A”*. The rules and control programs also can be simulated.

Firstly, 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. Secondly, 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 the 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, generating petri-nets that are generated by hand movements using real tokens. These tokens are placed on places with relevant state of the system, Fig. 6.



Fig. 6: Generating Petri-Nets by Concrete Demonstration

This control algorithm can also be used to drive the real model, enriched by actors and sensors, see Fig. 7.



Fig. 7: Computer Controlled Real Model

In Real Reality Modelling, physical models and virtual models are tightly coupled through sensed 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 twin-objects and uses them intuitively.

The power of this concept compared with conventional Graphical User Interfaces lies in its orientation towards all human senses during the modelling process, especially to the haptic. Instead of sensing each object, the concept of utilising the hand, yields a universality, because we can use all familiar objects of our surrounding as the user interface. The independence from conventional user interface devices supports a cooperative and communicative work in teams.

III. Further Applications

Currently we are extending this concept to pneumatic and electrical circuit design Fig 8ff. The pedagogical objectives are to improve (1) the understanding of theoretical and practical interdependencies in mechatronic systems, (2) the competence of action in electro-pneumatics and (3) the communication about them on the level of skilled workers and engineers. The new **intuitive User Interface** is expected to improve the learning process in production automation by supporting action oriented learning, social verbal and non-verbal communication and the grasping (understanding) of physical phenomena by **really** grasping them. The insight into the correspondence between concrete real and abstract virtual structures is expected to promote adequate mental models for the handling of complex technical systems [7][8].

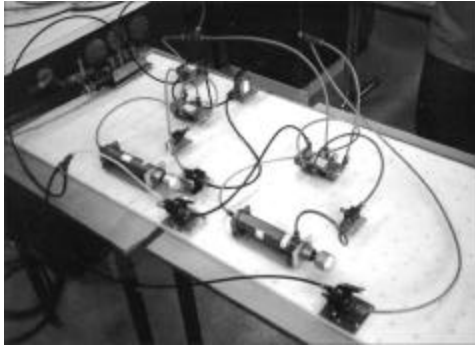


Fig 8: Real Pneumatic Circuit

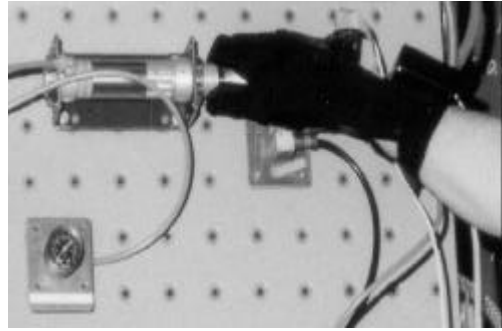


Fig 12: Grasping a Cylinder

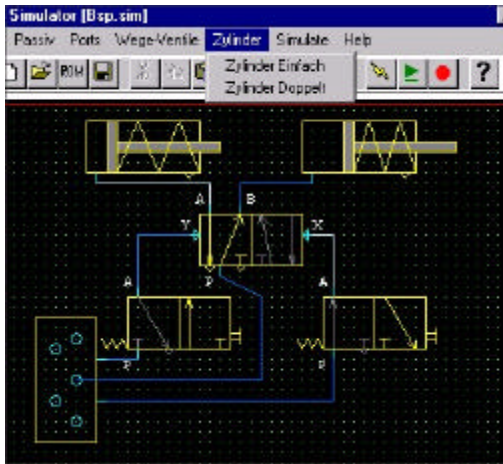


Fig 9: Pneumatic Circuit Simulation



Fig 10: Building Circuits in Reality

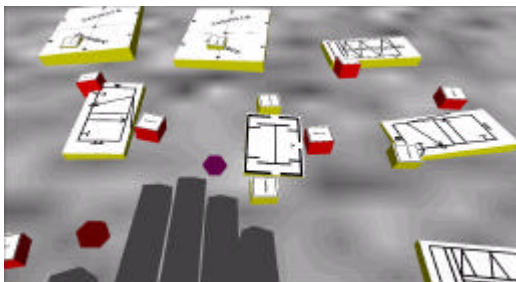


Fig 11: Circuits in Virtuality

IV. Related work

Several efforts have been undertaken to improve the concreteness of modelling and the ease of machine usage.

Weiser [9] introduced the concept of ubiquitous computing. Computational functionality is embedded in physical artefacts and spread throughout the environment. A behaviour construction kit [10] allows building models, consisting of computerised LEGO pieces with electronic sensors. These can be programmed using LEGO/Logo and are then spread throughout the environment, where they can interact with users or other physical objects. An open tool for situative learning is proposed by Suzuki and Kato [11]. They developed a graspable programming language, called AlgoBlock. The language consists of several commands having a physical representation. The physical blocks are equipped with an electronic interface. This way the computer can keep track of which blocks are connected. By connecting block by block a sequence of commands can be created and the result is graphically displayed on a monitor. This arrangement supports learning of abstract issues in groups by manipulating concrete artefacts.

Another main stream is Virtual Reality (VR) where the aim is to get a better view and feeling of virtual objects by using interface devices like data gloves, head mounted displays or CAVEs. Improving the appearance of the virtual by immersing users in a computer simulated environments has some severe disadvantages relating interpersonal communication, is very expensive, and thus is not our aim. Another approach frequently referenced to is Augmented Reality (AR) where the user's view of his physical environment is merged with computer-generated images [12]. The emphasis of AR lies on enhancing reality with information stored in the computer but not on creating and modifying models synchronously in two different worlds of representation. Although the idea of combining physical entities and virtual objects is investigated by some researchers, their aims as well as their areas of application differ from ours crucially.

Fitzmaurice et al. [13] propose a Graspable User Interface. They use tracking sensors as physical handles (bricks) for controlling virtual objects. The bricks are located on a flat screen and are logically linked to their visible virtual counterparts, thus moving a brick yields to a translation of the attached object. They propose a new kind of drawing program with the option of using several

bricks simultaneously. At MIT's Media Lab this approach will be improved in the Tangible Media project [14]. Instead of flat 2D models they additionally use stereo vision and 3D geometrical models, displayed on a desk-like device. The movements of physical handles such as cubes or pyramids are mapped to their graphical counterparts. By doing so, graphical user interfaces are enhanced with physical embodiments of their elements.

Most of these approaches have a sensorisation of physical objects in common. The implementation must provide a model of how to react on the changes, sensed by the objects. Furthermore, these projects employ visual cues displayed on monitors or special surfaces for giving the users feedback of their actions. This is different from our approach where the user's hands are sensed and tracked, and a model of how the hand changes the environment is implemented – as the user acts on physical objects which are self-contained and themselves represent the interface, a technical and perceptual dualism is avoided. Keeping the users in a common real space while their actions are seamlessly recorded by the computer has some tremendous advantages: keeping face-to-face communication, natural orientation in the physical space, and intuitive handling of the interface. From the recording of the users' actions performed with the interface objects, abstract descriptions such as rules and action patterns are derived and translated into formal representations. These data serve as the basis for the creation of simulation models and can be recalled any time if saved persistently in appropriate file formats. Sophisticated network functionality provides access for remotely working people. Several communication channels such as voice, gesture and graphical visualisation are used to exchange ideas, experiences, opinions, models and data, thus enabling computer-mediated Tele-learning and Tele-cooperation over large distances and across borders on a broad action-oriented and perceptual base.

The domain of production engineering is increasingly requiring a merge of different qualifications: knowledge and experience about physical and material processes, logical and algorithmic control mechanisms, social and organisational structures. Only a learning and working environment supporting the acquisition of this mixed qualification will meet future needs. Isolated learning with a support of conventional multimedia technology, strictly separated theoretical and practical learning in lectures and exercises, strict separation of vocational school and working place may be overcome with our concept.

V. ACKNOWLEDGMENT

I gratefully acknowledge the contributions of Volker Brauer, Dieter Müller, Martin Faust, Kai Schäfer, Eva Hornecker, Bernd Robben and Ingrid Rügge to this concept. Our work is gratefully being sponsored by the German Research Community (DFG Br1556/2-3, Br 1556/3-2) and the European Community (TAP MM1002).

VI. REFERENCES

- [1] F. W. Bruns and V. Brauer, "Bridging the Gap between Real and Virtual Modeling - A New Approach to Human-Computer Interaction", 2. *IFIP 5.10 Workshop on Virtual Prototyping*, Arlington, Texas, 1996
- [2] V. Brauer, "Simulation Model Design in Physical Environments", *Computer Graphics ACM Siggraph* vol 30, no. 4, 1996, pp. 55-56
- [3] F. W. Bruns, K. Schäfer and V. Brauer, "A new Approach to Human-Computer Interaction – Synchronous Modelling in Real and Virtual Spaces", *Proceedings of Designing Interactive Systems (DIS '97)*, Amsterdam, 1997
- [4] F. W. Bruns, F. W., "Zur Rückgewinnung von Sinnlichkeit - Eine neue Form des Umgangs mit Rechnern", *Technische Rundschau* vol 29/30, 1993, pp. 14-18
- [5] F. W. Bruns, A. Heimbucher, D. Müller, "Ansätze einer erfahrungsorientierten Gestaltung von Rechnersystemen für die Produktion", *artec-Paper 21*, Bremen, 1993
- [6] V. Brauer, "Feature-basierte Erkennung dynamischer Gesten mit einem Datenhandschuh", *Diplomarbeit*, Universität Bremen, 1994
- [7] E. Hornecker. Coupling Physical Artifacts and Abstract Representations. ED-Media 98, AACE, Freiburg, 1998
- [8] B. Robben and I. Rügge, "Real Reality: Computer(be-)nutzung nach der Desktopmetapher" In: W. Bruns, E. Hornecker, B. Robben, I. Rügge, *Vom Bildschirm zum Handrad - Computer(be)nutzung nach der Desktop-Metapher* - artec-paper 59. Universität Bremen, 1998, pp. 159-174
- [9] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing", In: *Communications of the ACM*, 36/7, 1993
- [10] M. Resnick, "Behavior Construction Kits", *Communications of the ACM*, 36(7), pp. 64-71
- [11] H. Suzuki and H. Kato, "Interaction-Level Support for Collaborative Learning: AlgoBlock - An Open Programming Language", In: *Proc. of the Computer Supported Collaborative Learning (CSCL) Conf.*, University of Indiana, 1995
- [12] Milgram, Drascic, Grodski, Restogi, Zhai & Zhou "Merging Real and Virtual Worlds", *Proc. of IMAGINA '95*, Monte Carlo, Feb. 1995
- [13] G. W. Fitzmaurice, H. Ishii, W. Buxton, "Bricks: Laying the Foundations for Graspable User Interfaces", *CHI'95 Mosaic of Creativity*, 1995, pp. 442-449
- [14] H. Ishii, B. Ullmer, B., "Tangible Bits: Toward Seamless Interfaces between People, Bits and Atoms", *CHI'97*, Atlanta, Georgia, 1997, <http://tangible.www.media.mit.edu/groups/tangible>

<http://media.mit.edu>

<http://www.uni-bremen.de/~artec/rugams>