A4 Digital Building: Extensive Computer Support for Building Design, Construction, and Management

Ludger Hovestadt

Institut für Industrielle Bauproduktion
Universität Karlsruhe (TH)
7500 Karlsruhe 1, Germany

The integrated design, construction, and management of buildings are described as being of unlimited complexity. The data structures required to support these tasks cannot be predefined and have to be worked out during the design process. An instrument that integrates weakly and strongly structured data is necessary. A4 proposes – as a minimal structuring mechanism – the position of information in a dataspace. It offers diverse additional and optional structuring mechanisms. Examples from different domains show the particular strengths of the A4 integration model.

Keywords: architecture, intelligent building, CAD, multi-media, hypermedia, active objects, virtual reality, multi-user, expert system, case-based reasoning, communication, data-mining.

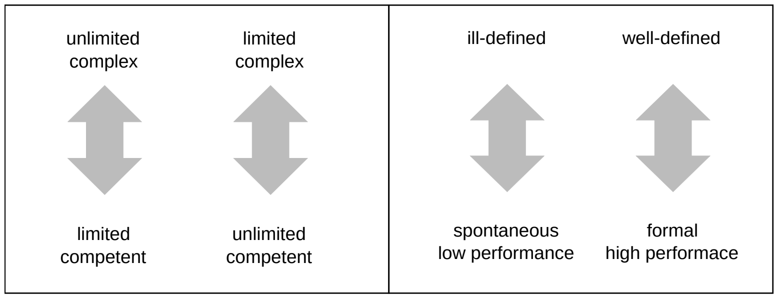
1 Introduction

The information needs of the building process are extraordinarily complex. Millions of information-processing steps occur, which can be organized in dozens of decompositions (function, components, space, costs, time …) (Piel, 1978; Richter, 1988). There may be up to a thousand participants, who work on dozens of physical sites and act all asynchronously with long, data-intensive, and widely overlapping transactions (Katz, 1985). Their work is exploratory (Mueller, 1989; Smithers, 1992), and they need loosely-structured data in a variety of versions. A physical building may have 100,000 sensors (as the airport of Munich), which produce asynchronous data for building control. Some of them may need security, and information may need to be generated and accessed in real-time. Data may have to be stored for a long time over the life-cycle of a building (which spans often more than 100 years). But the data will change during this life-cycle in response to reconstruction, adaptations, or renovations. Unexpected new techniques will be applied to the building, and thus unexpected new data may have to be integrated into the database.

A4 assumes that a uniform integration of these data is practically (and probably even theoretically) impossible. It therefore considers a building model as having unlimited complexty. At the same time, it considers a designer and/or computer program as having limited competence (see Figure 1). To use computers successfully, even in areas of

U. Flemming and S. Van Wyk (eds.), ***CAAD Futures '93****, pp. 405-421.*

unlimited complexity, one has to view them primarily not as automatons but as imagination amplifiers (GMD, 1992). They enable humans to deal with more information, with greater complexity, with more competence, and in less time.



Figures 1, 2.

It is therefore the aim of A4 to give the persons involved with buildings an instrument that enables them to define and extend cooperatively the needed support structures in a fast and competent way. Consequently, A4 needs information techniques that allow for a continuous transition from unstructured to highly-structured data. Unstructured data can be created spontaneously – for example, in the early stages of design – but perform poorly, whereas structured data are costly to edit – for example, in the late or established stages of design – but perform well (see Figure 2). Any addition of structure potentially increases performance, but decreases flexibility. Currently, there exists no instrument that allows for a continuous transformation from weakly- to strongly-structured data. Different attempts have been made in this direction (e.g. (Moerkotte and Zachmann, 1993; Maurer and Pews, 1992, 1993; FABEL, 1992)), and A4 stands in this tradition.

A4 has its origin in the architectural work of Fritz Haller (Haller, 1988), especially the component based building system MIDI (Haller, 1974) for multi-storey complex buildings like schools, laboratories or office buildings. The feature of MIDI is the integration of construction and service systems. The generic installation model ARMILLA (Haller, 1985) is an answer to the special complexity of this integration. Since now four software prototypes were developed on the base of ARMILLA: A1 shows, that objects and rules are an adequate representation for ARMILLA. It became obvious, that even with great care it would not be possible to implement design automatisms even for very small subproblems. Therefore A2 moves the focus from automatisms to the interactive control of complex structures. A2 supports a user in his multi-level and iterative design process of MIDI/ARMILLA buildings by multiple expertsystems coordinated by a blackboard (capacity up to 500 building objects). We tried to transfer the concepts of A2 to a multiuser environment by A3. This system became too complex, because the user inputs were no longer sequential but parallel and asynchronuous. A4 is the fourth attempt in this line. It is no longer fixed to MIDI and ARMILLA, can be used in other environments and can deal with 20,000 objects at minimum. A+ is another current prototype based on constraints. For more detailed descriptions on the different prototypes see: (Mathis, 1988; Raetz, 1989; Hovestadt, 1990, 1991; Gauchel et al., 1992a; Drach et al., 1992; Bhat et al., 1993; Hovestadt, 1993).

This paper presents the A4-Model of organizing data in a multidimensional dataspace in section 2. The coordination and cooperation of the often very heterogeneous data is described in section 3. Section 4 concentrates on the navigation of users in the dataspace. Section 5 illustrates the special capability of the A4-Model by different examples.

2 The A4-Model

The design and control of a building's life-cycle uses many different media : text, graphics, drawings, sketches, sounds, images, videos, tables, phone, faxes, discussions, conferences. … It is not the aim of A4 to deal with these media itself. This is the task of specific software. A4 encapsulates the different media by the concept of a datacontainer. Users in A4: 1) are able to run the data of a container by the corresponding software outside of A4, and 2) can establish correlations between the datacontainers by different techniques inside A4. The minimum structure for a container required in A4 is a position in a 3-dimensional dataspace (see section 2.1). If this spatial position is not sufficient enough to describe the intended correlation of the containers, A4 offers other optional techniques: additional dimensions of the dataspace, attributes and values, relations, rules and constraints (see section 2.2).

2.1 Weakly-structured Data

At minimum the datacontainers have to have the x-, y-, and z-coordinates in the multidimensional dataspace. Thus building components or events, which are located or take place in physical space, can have a directly corresponding datacontainer in the dataspace (see Figures 3 - 8).

In the A4-Model, the dataspace substitutes for the directory structures of conventional operating systems. The software for navigation through this space corresponds to conventional browsers. The metaphor of the dataspace replaces the metaphor of a desktop.

2.2 Additional Structures

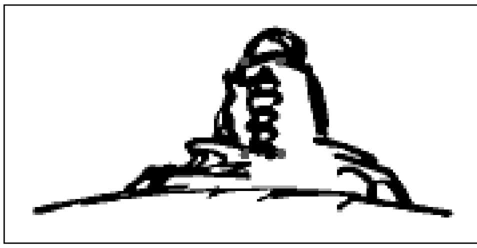
If one wants to position all data of a building, the dataspace needs more dimensions to get sufficient capacity. Section 2.2.1 describes eight additional dimensions. Some other well known techniques can be used as well: frames, relations, rules and/or constraints (see Section 2.2.2 ff).

2.2.1 Additional Dimensions of the Dataspace

Time: In addition to their position on the x-, y-, and z-axis, data also can have a position on a time axis. Thus, succession of construction, organization of the site or facility management tasks can be described easily.

Precision: On this axis, imprecise data (e.g. sketches) can be separated from precise data (e.g. details). Figure 11 shows an overview of the construction of a building on a sketch level, Figure 12, the lower left corner in more detail, and Figure 13, the very detailed level of this corner.

Morphology: Most of the objects located in x, y, z, time and aspect can be differentiated e.g., in 1) the intended use of the building at this location, 2) the aggregates, necessary, and 3) the supply systems necessary for the aggregates. Figure 14 displays the intended climate zones of a building, Figure 15 the rough position of the supply air outlets, necessary to achieve the intended climate, and Figure 16 the ducts necessary for server to outlets. In general, the axis morphology allows various descriptions of a system: functional, logical, structural, etc.



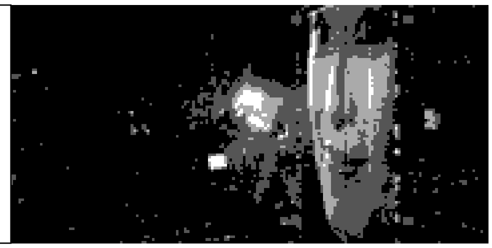
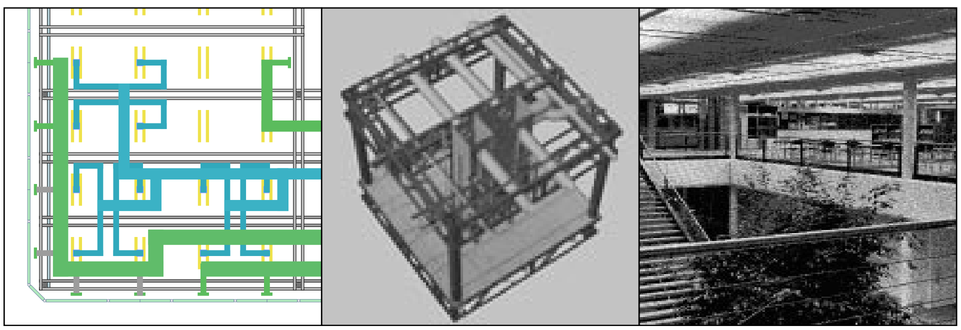


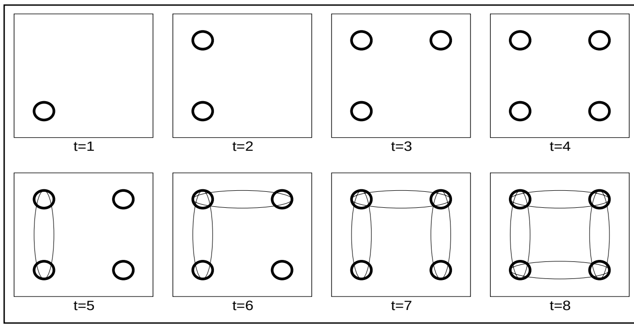
Figure 3. The example of Mendelsohn's manual sketch of the Einstein Tower (1920) demonstrates that data can be intuitively arranged, found and edited in space without any additional structuring. In conventional systems, components have to receive a unique name and have to be organized in some semantical hierarchies (shape grammars are an exception).

Figure 4. This snapshot of the science fiction film "Blade Runner" develops a scenario, where architecture is completely imbued with computer techniques. Therefore, this film is a very detailed vision of virtual dataspaces.



Figures 5, 6, 7. Data organized space corresponds directly to building components.

Figure 8. Different experiments to achieve a better density of information and user orientation by projecting 2-D computer interfaces by 3-D visualization techniques (Clarkson, 1991).



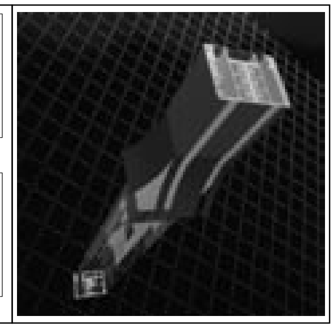
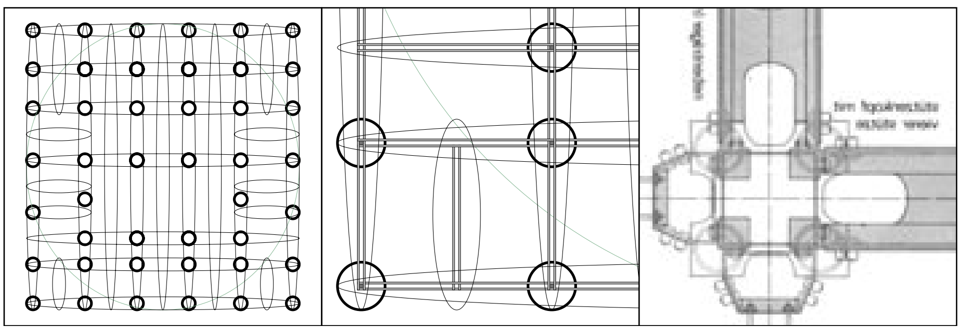
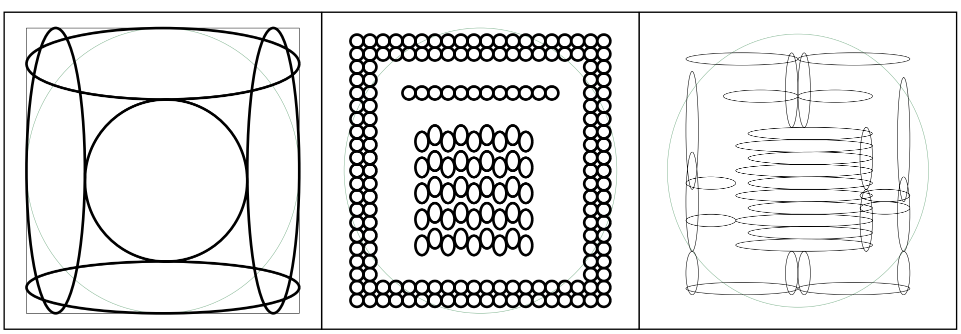


Figure 9. Different positions on the time-axis.

Figure 10. Any combination of three axes of the dataspace can be displayed and edited by conventional 3-D techniques. This figure shows the x-, y-, and time-axis to present an interface to the construction process of a plug-in sanitary cell of a high-rise office building. The design with no physical axis of the dataspace (no x-, y-, or z-axis) is an interesting problem for research in virtual reality.

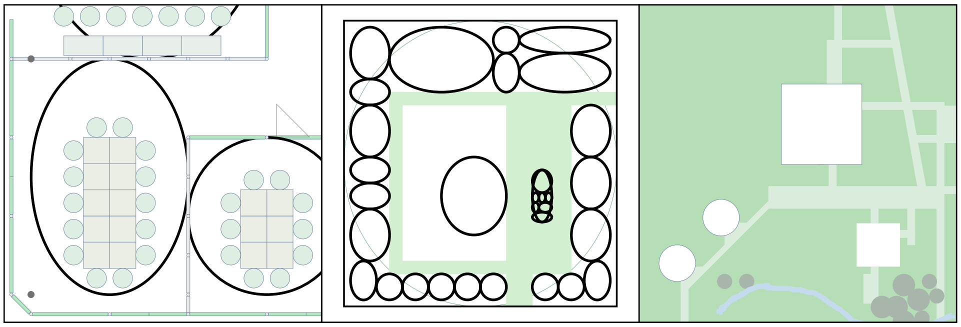


Figures 11, 12, 13. Different positions on the precision-axis.



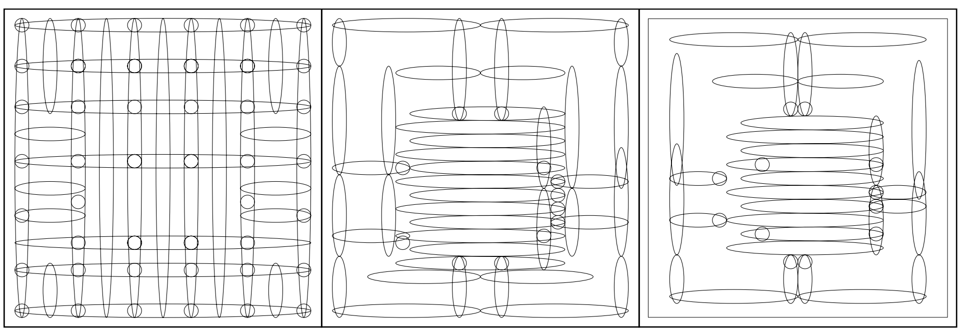
Figures 14, 15, 16. Different positions on the morphology-axis.

Size: This axis distinguishes data of different absolute size, e.g., the design of a detail (like a doorknob) is relatively independent of the design of a building component (like the door) (see Figure 17), which again is relatively independent from the building layout (see Figure 18), which again is independent from the layout of buildings on a campus (see Figure 19) etc. Thus, the absolute size of an object is a good starting point for the additional structuring of the dataspace.



Figures 17, 18, 19. Different positions on the size-axis.

Aspect: The next axis of the dataspace recognizes the fact that architects, the different engineers, and users deal with different parts of a building. Figure 20 gives a sketch 2-D view on x and y axis to the construction, Figure 21 a comparable view on the return-air system, and Figure 22 to the supply-air system. Each of these views are focussing on the same location in x, y, z and time of the dataspace, but on different positions on the aspect-axis.



Figures 20, 21, 22. Different positions on the aspect-axis.

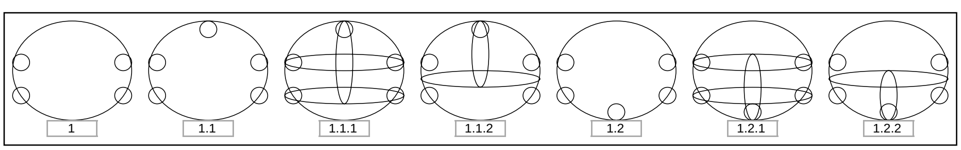


Figure 23. Different positions on the alternatives-axis.

Alternatives: Especially during the design phase, a range of alternatives should be explored. A separate dimension of the dataspace provides an efficient and simple means for dealing with this issue. The scale on this axis represents a tree structure.

Timetag: By adding creation, modification, and destruction times to data, mechanisms for operations like recording, backtracking, undo, trace, etc. can be installed easily.

Users: This axis is the platform for multiuser environments in the dataspace. The example in Figure 24 displays the working area of three different users inside the building concurrently. They share their data and are able to communicate, if their working areas overlap each other, as the two lower left users do. It is obvious, that transaction management, network traffic, etc. are strongly influenced by the spatial organization of data.

The above list of dimensions of the dataspace is not closed and can be augmented, if required. The idea of organizing data in a multidimensional dataspace is used by (Rivero, 1977) and (David, 1987) in the domain of design, by (Lane, 1990) in the domain of software engineering, by (Minkoff, 1992) in the domain of visualization, and by (Stonebraker and Dozier, 1991) in the domain of database management.

2.2.2 Frames

Data positioned in the multidimensional dataspace can be supplemented in A4 by additional structures in the form of attribute/value pairs. In this way, the dataspace can be coupled with frames (Minsky, 1975) or scripts (Shank and Abelson, 1977; Schank, 1982), which have been developed in artificial intelligence. The relevance of frames for architectural design is discussed in (Sowa, 1984) and (Gero, 1987).

2.2.3 Semantic Nets

Another important means to structure data in dataspace are explicit connections, which can be used to build semantic nets. Elaborate semantic nets have been developed in architecture (Waard and Tolman, 1991), in software engineering (Maurer and Pews, 1993a, 1993b), and as a platform for massive parallel computing (Sapaty, 1990).

2.2.4 Behavior

Objects also have behavior, and one can distinguish between rules and constraints that affect this behavior:

- A rule is a uni-directional connection between two objects: (A -> B). (Coyne, 1988) gives an overview of the importance of rules in architecture and design. Figure 30 and Figure 33 give examples of rules within the context of A4.
- A constraint is a bi-directional connection between objects. It can be represented by a name, a definition, and variables (e.g., "adder," "A + B = C," [A, B, C], which means: A + B -> C and C - B -> A and C - A -> B). Constraints that share the same variables are part of a constraint-network. Design can be considered as the search for a solution within a set of possible solutions. The solution aimed at by a designer has to fulfill specific conditions that may be expressed as constraints. Well known examples of systems based on this premise are LOOS/ABLOOS (Flemming, 1978, 1980, 1989), (Coyne, 1991), MOLGEN (Stefik, 1981), and GARI (Descotte and Latombe, 1985).

3 Coordination

An environment in which different objects with different behaviors using different AI-techniques coexists requires a model for their coordination. Blackboard systems deal with this problem (Engelmore, 1988). They traditionally install a blackboard on which different knowledge sources (implemented by different AI-techniques) can work cooperatively. Coordination is achieved by a supervisor, which inspects the data written on the board and decides which knowledge source can work on this data next (see Figure 24). Several blackboard systems have been proposed in the domain of architecture: e.g. ACCORD (Hayes-Roth et al., 1986), ICADS (Pohl, 1988), AEDOT (Stratton and Jarrell, 1990).

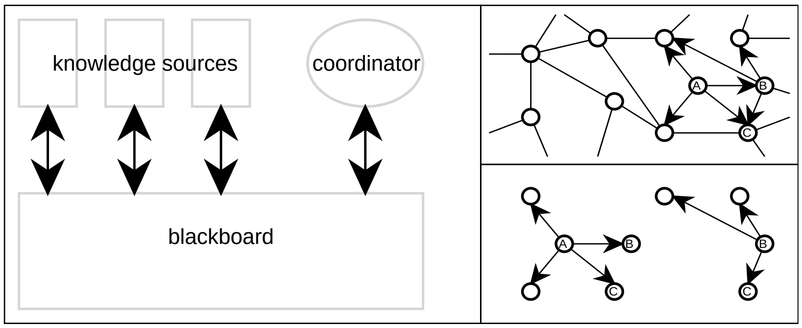


Figure 24. A typical blackboard architecture.

Figures 25, 26. The A4-model of autonomous objects.

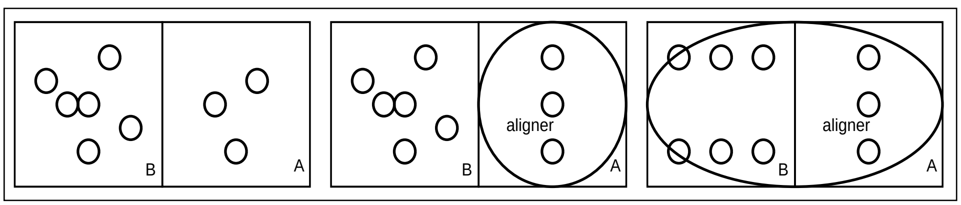


Figure 27 a, b, c. The spatial coordination of autonomous objects. Objects A and B have an interest in a specific number of circles inside their spatial area of influence or work areas (see Figure 27a). Therefore, one cannot, e.g., remove or add a circle inside the working area of object A without protest from this object. But it is possible, e.g., to move or align the circles inside the object's working area. This can be done either by a user or another object, which is called "aligner" in this example. If someone (a user or another object) places the aligner at the position of object A, the three circles lie inside the working area of object A, which looks at the number of circles and the aligner, which in turn looks at their adjustment (see Figure 27b). It is important that a user is able to control the objects' interdependencies by their relative spatial position. To expand the functionality of the aligner, e.g., to the working area of object B, the aligner simply has to be enlarged in the dataspace (see Figure 27c).

A4 works with an alternative, more distributed approach (Hovestadt, 1991; Gauchel et al., 1992a, 1992b; Hovestadt, 1993; Bhat et al., 1993). It assumes that there is a pool of objects, each of which manipulates several other objects (see Figure 25). An object's behavior is described only inside that object (see Figure 26). Objects have a position in the multidimensional dataspace and potentially can see and manipulate only objects with which they collide in that space (that is, they have to touch at least on one axis, which has a different meaning for the different axes of the dataspace). Given the concept of a spatial "area of influence" for objects, a central coordinator can be distributed in space as well: The coordination of several objects can be achieved by another higher-level object. In this way, hierarchies of control structures that diversify in space can be built. The final control of this system of self-organizing objects is left to the users based on another important implication of spatial arrangements of objects: they can easily be displayed, distinguished from each other, and manipulated graphically (see Figure 27).

A discussion of self-organizing, autonomous objects can be found in (Agha, 1986; Kemper et al., 1990; Minkoff, 1992).

4 Navigation

Users are not able to coordinate all objects at the same time. Therefore, they have to move around in the mulitdimensional dataspace; that is, in order to see specific objects, they have to forget others. The two primitive commands `moveTo` and `forget` allow users to navigate in the dataspace (see Figure 28).

5 Examples

To illustrate the special capabilities of the A4-Model of autonomous objects located in a multidimensional dataspace, some ensembles of these objects from different areas in a building's life cycle are presented in Figures 29 through 35.

6 Implementations

All implementations at the Institut für Industrielle Bauproduktion are prototypes. The versions A1, A2 and A3 were written in CommonLisp and the shell KnowledgeCraft. A4 has three parallel developments: The fastest and most experimental prototype is the computer animation with MacroMind Director on a Macintosh (Hovestadt et al., 1992). The next level is an interface to the A4-Model developed on a NeXT workstation with Objective_C. Everything, which was discussed in this paper can be done by hand with this tool. On the third and slowest level there are multiple experiments in the dataspace using AI techniques. They usually are written in CommonLisp.

7 Acknowledgements

Thanks to Fritz Haller for his long support of this project, and to Ulrich Flemming for his comments and help in preparing the manuscript. A4 currently is founded by the German BMFT as part of the Verbundvorhaben FABEL (1992).

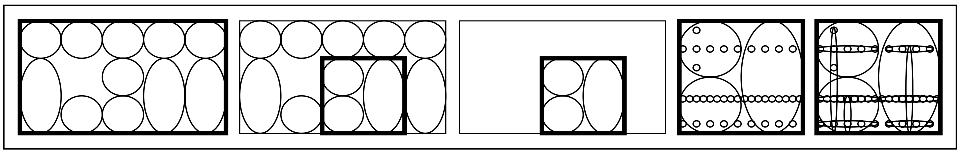


Figure 28a, b, c, d, e. This figure illustrates the navigation on 3 axes of the dataspace: x, y, and morphology. Figure 28a shows the spatial arrangement of nets for returnair (ellipses) inside a service space of a building (rectangle). The position of the user in Figure 28a is: x=0, dx=108, y=0, dy=108, aspect=returnair, morphology=net, size=4, resolution=sketch. Figure 28b shows the working area of the user after the command: moveTo x=72, dx=144, y=0, dy=72. Figure 28c dissolves the objects outside the user's working area by the command forget. Figure 28d shows additional objects by moving the user's working area on the morphology axis to the position fixture (command: moveTo morphology=fixture). Figure 28e shows the ductwork of the returnair net after another move on the morphology axis (command: moveTo morpholgy=ducts).

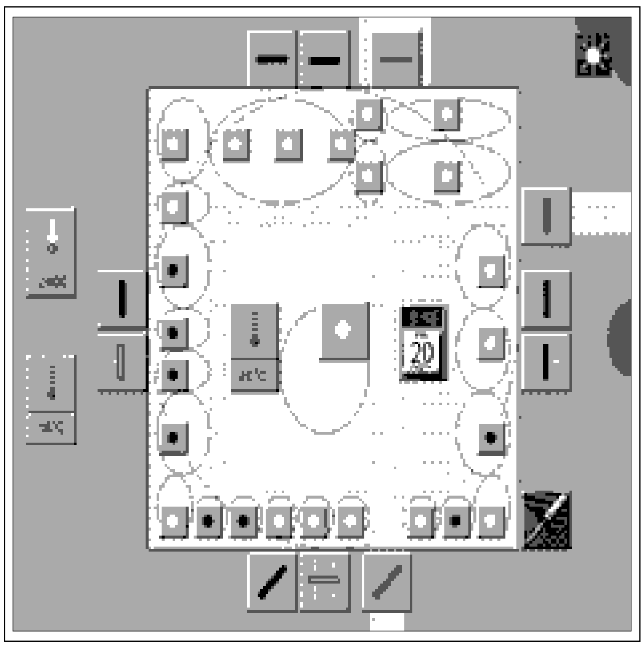


Figure 29. Naive Interfaces – Especially in the domain of facility management and building control, simple interfaces for non-expert users are necessary. In so-called Intelligent Buildings with a huge variety of electronic control systems, the spatial arrangement of display and control elements on top of a schematic floor plan is applicable.

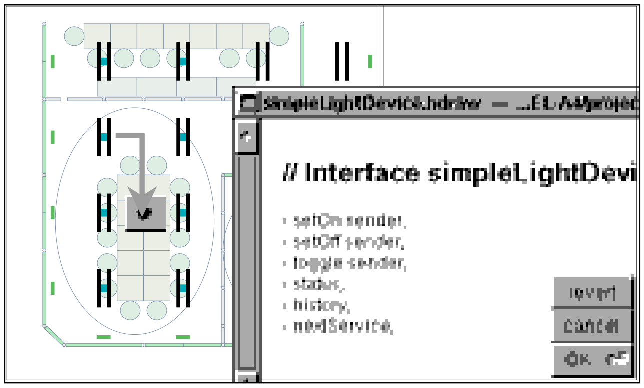
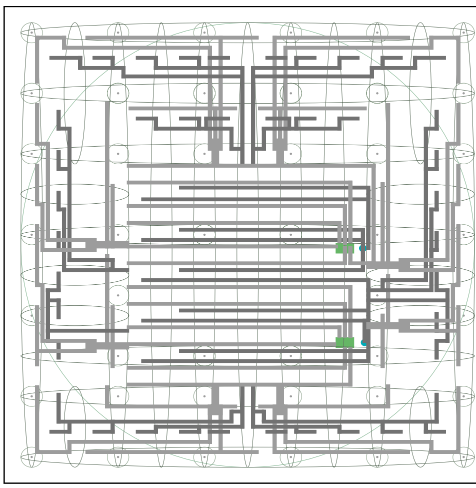


Figure 30. Interface Builder for Buildings – The behavior of many building objects can be developed by modern software engineering tools. This figure shows an interface builder for buildings used to connect a switch to light sources of a room and the corresponding source code.



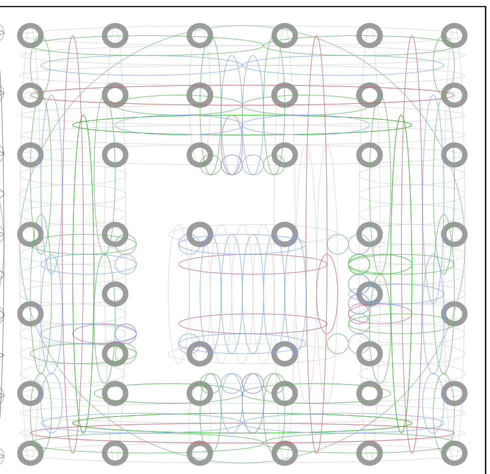


Figure 31. Vertical Coordination – This figure shows different technical systems (construction, HVAC return and supply air ducts) on different levels of abstraction (bounding boxes (rectangles) and sketches (ellipses)). The graphical representation of sketches by ellipses (they represent their spatial area of their bounding box) gives a very high density of information because the objects overlap each other only at a few points. This is the graphical key to the coordination of objects of different levels of abstraction.

Figure 32. Horizontal Coordination – This figure shows the objects of different engineers at the sketch level of design. It again illustrates the power of the graphical representation by ellipses.

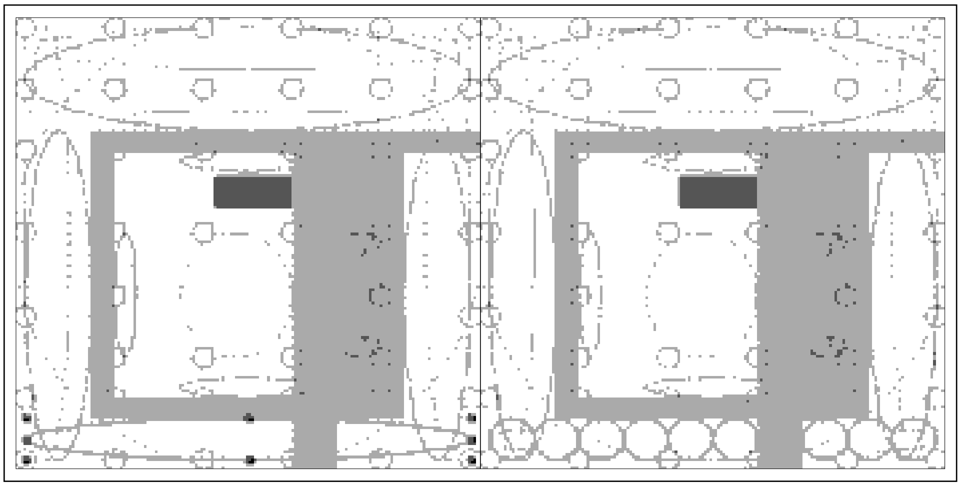


Figure 33 a,b.

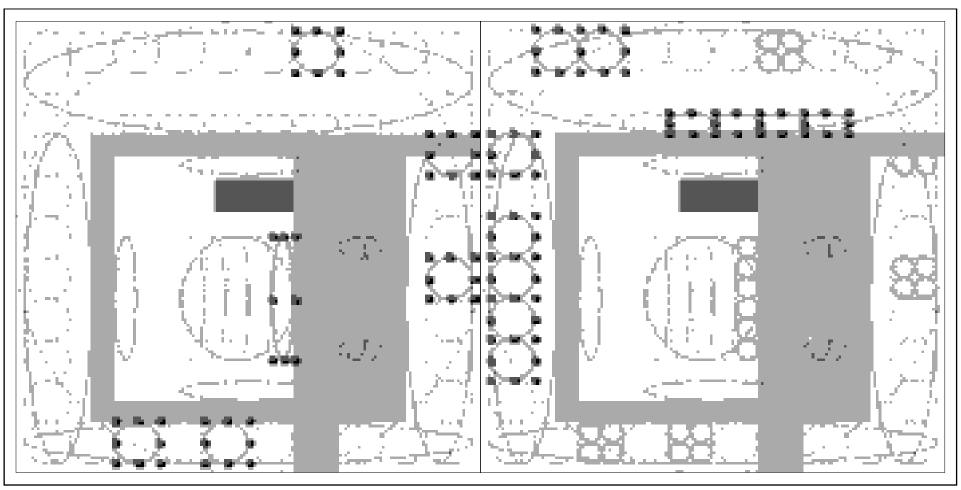


Figure 33 c, d. Expert systems for design support (construction) – This sequence shows how objects arranged in the dataspace and equipped with knowledge about objects of the next detailed abstraction level can work cooperatively and develop a building design step-by-step. The sequence shows also how users are able to monitor and control multiple processes with a unique graphical interface. The objects with handles are activated; the objects in bold are newly created (this direct correspondence only exists between Figures a/b and c/d).

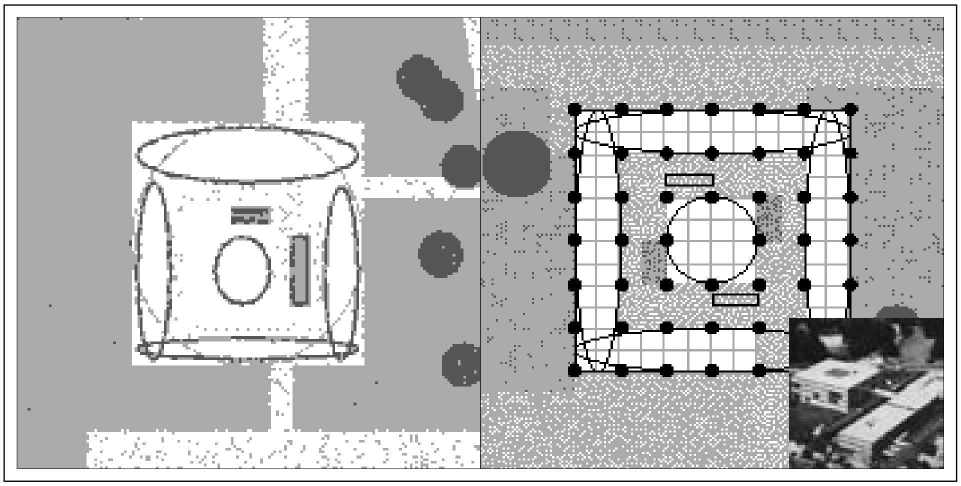


Figure 34 a, b.

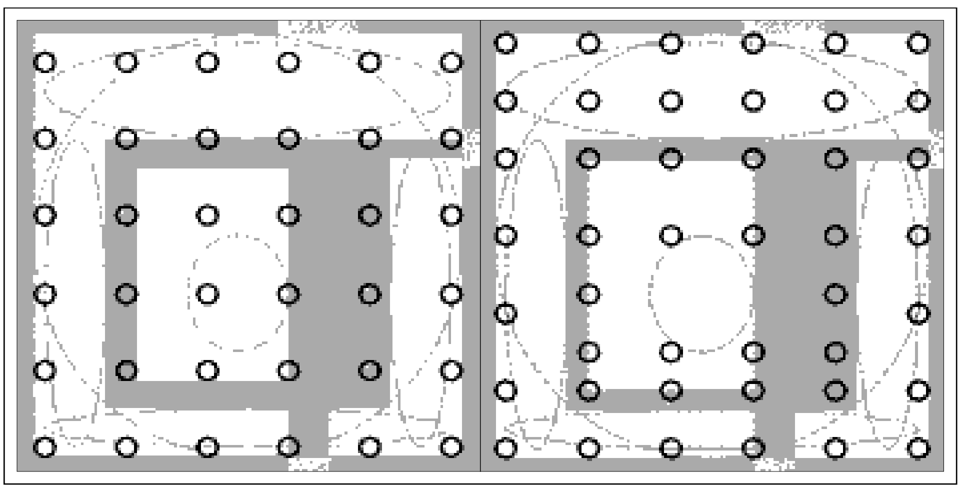


Figure 34 c, d. Case-based reasoning (CBR) – This sequence shows how a scene in an actual design can be interpreted as a problem and used as a request to a case library (a). The library answers with one or multiple comparable problems of other designs (b) (episodic knowledge) or catalogs (generic knowledge). A user or specific application is able to navigate though the milieu of these destination problems and able to collect data (e.g., solutions) useful for the actual design. These data can be copied (c) and adapted (d) to the actual design. If one was able to solve a problem successfully, e.g., by the use of CBR techniques, the problem with its solution can be added to the case library and is accessible to future requests (FABEL, 1992). CBR in architecture: (Domeshek and Kolodner, 1992).

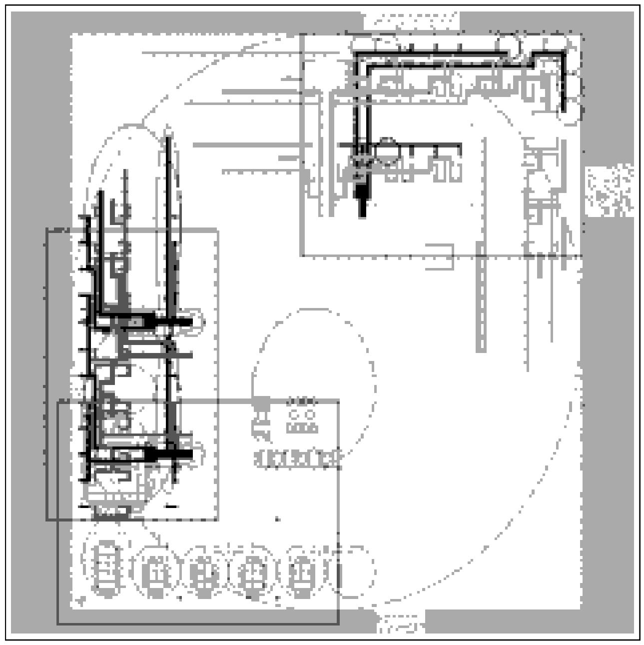


Figure 35. Multiuser – The working areas of multiple users in A4 can be organized in space as well. This figure shows three users with different locations in the multi-dimensional dataspace. Like all other objects, the users are able to communicate if their working area overlaps, i.e. if they meet each other in the dataspace. The spatial location of users and objects is a very simple criterion for distributing the objects on the different hosts.

References

Agha, G.A., 1986. *ACTORS: A Model of Concurrent Computations in Distributed Systems*. Cambridge, MA: The MIT Press.

Bhat, R.R., Gauchel, J., and Van Wyk, S., 1993. "Communication in Coorperative Building Design," in *CAAD Futures '93*. Netherlands: Elsevier Science.

Clarkson, M.A., 1991. "An Easier Interface," *BYTE* 2, pp. 277-282.

Coyne, R.C., 1988. *Logic Models of Design*. London: Pitman.

Coyne, R.F., 1991. *ABLOOS: An Evolving Hierarchical Design Framework*. Ph.D. Dissertation, Department of Architecture, Carnegie Mellon University, Pittsburgh, PA.

David, B.T., 1987. "Multi-Expert Systems for CAD," in P.J.W. ten Hagen and T. Tomiyama (eds.), *Intelligent CAD Systems I*. Berlin: Springer-Verlag.

Descotte, Y., and Latombe, J., 1985. "Making Compromises among Antagonist Constraints and a Planner," *AI Journal* 27, pp. 183-217.

Drach, A., Langenegger, M., and Heitz, S., 1992. "Flexible Environments for Integrated Building Design," in Krause et al., (eds.),*CAD'92 - Neue Konzepte zur Realisierung anwendungsorientierter CAD-Systeme*. Berlin: GI-Fachtagung.

Domeshek, E.A., and Kolodner, J.L., 1992. "Toward a Case-Based Aid for Conceptual Design," *International Journal of Expert Systems*.

Engelmore, R., and Morgan, T., eds., 1988. *Blackboard Systems*.

FABEL, 1992. *Integration von Modell- und Fallbasierten Entwicklungsansaetzen fuer Wissensbasierte Systeme*. Verbundvorhaben des BMFT unter der Leitunge der Gesellschaft fuer Mathematik und Datenverarbeitung, Bonn.

Flemming, U., 1978. "Wall Representation of Rectangular Dissections and their Use in Automated Space Allocation," *Environment and Planning B: Planning and Design* 5, pp. 215-232.

Flemming, U., 1980. "Wall Representations of Rectangular Dissections: Additional Results," *Environment and Planning B: Planning and Design* 7, pp. 247-251.

Flemming, U., 1989. "More on the Representation and Generation of Loosely Packed Arrangements of Rectangles," *Environment and Planning B: Planning and Design* 16, pp. 327-359.

Gauchel, J., Hovestadt, L., Van Wyk, S., and Bhat, R.R., 1992a. "Modular Building Models," in *Design Decision Support Systems '92*, Eindhoven.

Gauchel, J., Van Wyk, S., Bhat, R.R., and Hovestadt, L., 1992b. "Building Modeling Based on Concepts of Autonomy," in *Proceedings*, AID '92 - Second International Conference On Artificial Intelligence In Design, Pittsburgh, PA.

Gero, S.J, 1987. "Prototypes: a new schema for knowledge-based design," working paper, Architectural Computing Unit, Department of Architectural Science, Sydney.

GMD, 1992. "Imagination at Work. Ein Forschungsprogramm der GMD," manuscript, Sankt Augustin, December 1992.

Haller, F., 1974. *MIDI - ein offenes system fuer mehrgeschossige bauten mit integrierter medeininstallation*. USM baussystme haller, Muensingen, Switzerland.

Haller, F., 1985. *ARMILLA - ein installationsmodell*. Institut fuer Industrielle Bauproduktion, Universitaet Karlsruhe, Germany.

Haller, F., 1988. *Bauen und forschen*. Dokumentation der Ausstellung. Solothurn, Switzerland.

Hayes-Roth, B., Vaughan, J. M., Garvey, A., and Hewett, M., 1986. "Application of the BB1 Blackboard Control Architecture to Arrangement Assembly Tasks," *The International Journal for Artificial Intelligence in Engineering* 1 (2), pp. 85-94.

Hovestadt, L., 1990. *The ARMILLA Film* (MacroMind Animation for Macintosh), Institute fuer Industrielle Bauproduktion, Universitaet Karlsruhe, Gemany.

Hovestadt, L., 1991. "A4 - A Model for an Extensive Use of Computers in Architecture," in *Proceedings*, Second International Workshop on Computer Building Representation for Integration, Aix-les-Bains, France.

Hovestadt, L., 1993. *A4 - Digitales Bauen - ein Modell für weitgehenden Computereinsatz im Bauen*. Dissertation, Institut fuer Industrielle Bauproduktion, Universitaet Karlsruhe, Germany (forthcoming).

Hovestadt, L., Ziegler, Ch., and Hovestadt, V., 1992. *A4 - Digitales Bauen*. Video zu (Hovestadt 1993). Institut fuer Industrielle Bauproduktion, Universitaet Karlsruhe, Germany.

Hovestadt, V., 1992. *Zeitraumplanung - Entwurf von Instrumenten und Darstellungstechniken für das Planen in Zeiträumen*. Diplomarbeit, Lehrstuhl für Konstruktives Entwerfen, Prof. P. Kulka, RWTH Aachen.

Katz, R.H., 1985. "Information Management for Engineering Design," *Surveys in Computer Science*.

Kemper, A., Lockemann, P.C., Moerkotte, G., Walter, H.D. and Lang, S.M., 1990. "Autonomy over Ubiquity: Coping with the Complexity of a Distributed World," *Proceedings*, International Conference on Entity-Relationship Approach, Lausanne, Switzerland.

Lane, T.G., 1990. *Studying Software Architecture through Design Spaces and Rules*, Technical Report CMU-CS-90-175, Carnegie Mellon University, Pittsburgh, PA.

Mathis, Ch., 1988. *Installationsplanung mit einem wissensbasierten System*. Dissertation, Institut fuer Industrielle Bauproduktion, Universitaet Karlsruhe, Germany.

Maurer, F., 1992. “Wissensbasierte Systeme in der Betriebswirtschaft - Anwendung und Integration mit Hypermedia,” in J. Biethahn, R. Bogaschewsky, U. Hoppe, and M. Schumann (eds.), *HyperCAKE - Ein Wissensakquisitionssystem fuer hypermedia-basierte XPS* .

Maurer, F., and Pews, G., 1993a. “Hypermedia als Zwischenrepraesentation bei der Expertensystementwicklung,” in *Proceedings*, Hypermedia 93. Berlin: Springer-Verlag.

Maurer, F., and Pews, G., 1993b. “Validierung von konzeptuellen Modellen,” in *Proceedings*, XPS 93. Berlin: Springer-Verlag.

Minkoff, M., 1992. “The FERN Model: An Explanation with Examples,” Human Interface Technology Laboratory of the Washington Technology Center, University of Washington, Seattle, WA.

Minsky, M., 1975. “A framework for representing knowledge,” in Winston (ed.),*The Psychology of Computer Vision*. New York: McGraw-Hill.

Moerkotte, G. and Zachmann, A., 1993. “Towards More Flexible Schema Management in Object Bases,” Fak. fuer Informatik, Universitaet Karlsruhe, Karlsruhe, Germany.

Mueller, J., 1989. “Moeglichkeiten und Ergebnisse der Analytischen Darstellung Konstruktiver Entwurfsprozesse im Aktivitaets- und Ereignisorientierten,” *Graph. Konstruktion* 41 , pp. 25-34.

Pawley, M., 1990. *Theory and Design in the Second Machine Age*. Oxford: Blackwell.

Piel, R., 1978. Ordnen, Suchem, Finden - Bauinformationen mit dem BRD/SfB-System. Koeln, Germany.

Pohl, J., and Reps, I., 1988. “ICADS: An Integrated Intelligent CAD Environment,” in *Proceedings*, 4th International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany.

Raetz, P., 1989. XNET: *Ein intelligentes CAD-System fuer die Planung von Lokalen Netzwerken in Gebaeuden*. Dissertation, Institut fuer Industrielle Bauproduktion, Universitaet Karlsruhe, Germany. Fortschritt-Berichte VDI, Reihe 20: Rechner-interstuetzte Verfahren, Nr. 19, Duesseldorf.

Richter, P., 1988. *Entwicklung einer integrierten Informationsstruktur fuer relationale Datenbanken im Bauwesen*. Dissertation, Gesamthochschule Kassel, Germany.

Sapaty, A., 1991. “The WAVE Model for Advanced Knowledge Processing,” Alexander von Humboldt Fellowship Programme, FRG and Department of Engineering Science, University of Oxford.

Schank, R.C., and Abelsonn, R.P., 1977. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures. Hollsdale*, N.J.: L. Erlbaum Associates.

Schank, R.C., 1982. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge: Cambridge University Press.

Smithers, T., 1992. “Design as Exploration: Puzzle-Making and Puzzle-Solving,” position statement at the *AID '92 - Second International Conference On Artificial Intelligence In Design* , Pittsburgh, PA.

Sowa, J.F., 1984. *Conceptual Structures: information processing in mind and machine*. Reading, MA: Addison-Wesley.

Stefik, M, 1981. “Planning with Contraints, MOLGEN Part 1; Planning and Meta-Planning, MOLGEN Part 2,” *AI Journal* 16, pp. 111-169.

Stonebraker, M., and Dozier, J., 1991. “Large Capacity Object Servers to Support Global Change Research,” Technical Report Nr. 91/1, Electronic Research Laboratory, University of California, Berkeley, CA.

Stratton, R.C., and Jarrell, D.B., 1990. “Towards the Development of Multilevel-Multiagent Diagnostic Aids,” chapter in *Intelligent Diagnostic Systems*, London.

de Waard, M., and Tolman, F., 1991. “A Model to Describe Residential Buildings,”in *Proceedings*, Second International Workshop on Computer Building Representation for Integration, Aix-les-Brains, France.