The design of buildings which have complex mechanical infrastructure using expert systems

Mihaly Lenart
University of Karlsruhe,
Institut für Baugeschäft,
Baukonstruktion und Entwerfen I.
Englerstr.7,
7500 Karlsruhe 1,
West-Germany.

This paper presents a project (for details see [5] and [6]) under development at the University of Karlsruhe (Institut fuer Baugeschäft, Baukonstruktion und Entwerfen I., West-Germany, under the leadership of professor Fritz Hailer) in which the author took part for two years. The aim of this project which was supported by the German Research Association (Deutsche Forschungsgemeinschaft) is to find better methods for the design of buildings having complex mechanical systems like laboratories, office buildings, schools, hospitals, etc. The design of the mechanical infrastructure in such buildings is as important as the design of other architectural or construction parts. The fundamental idea of the project is to consider design problems of the mechanical system as part of the design of the architectural and structural concepts of the entire building. This is based on the belief that the use of an expert system containing computer programs for the solution of design problems can support the whole design procedure and that the design of buildings having complex mechanical infrastructure can be qualitatively better and more efficient than the design with traditional methods.

1. What is the problem?
The usual way to design a building is the following: The architect, after outlining the architectural tasks, designs the building according to his image or idea about a solution. During the design of the building but more often afterwards, he consults the civil engineer. Then after almost every architectural and structural part is already designed the mechanical infrastructure will be designed. Therefore, engineers in charge of the design of the heating, ventilation, air conditioning (plumbing) and electric installation must find their way in this architectural and structural "maze".

In buildings with a complex infrastructure, the cost of the mechanical system ranges from 30% to 40%. The cost of the building structure itself is only about 25%. This suggests that, in such cases, the traditional sequence of design may lead to an undesirable result.

1 This work is supported by the German Research Association (Deutsche Forschungsgemeinschaft, West-Germany)
2 On leave, present address: Carnegie-Mellon University, Dept. of Architecture, Pittsburgh, PA 15213, USA.
The next problem is that different parts of the mechanical systems are designed by different engineers or even by different firms. Communication between the engineers in charge of the heating, ventilation, air condition (H.V.A.C.) or electricity and the architect is not always satisfactory. At this time no one is organizing the work of all these engineers, so there is no overall concept for the organization. Collision problems under or between pipe lines, mechanical equipment and construction parts are often unsolvable. In case of unsolvable conflicts redesigning the whole building or even parts of the building or construction is costly and should be avoided.

The mechanical systems of buildings are not flexible. This means a change after the building is completed is difficult or even impossible. Changes in the function of a building or in the size (it becomes larger) or just in the mechanical system (new technology or equipment) may also cause difficult, if not unsolvable problems.

There is a tendency in architecture for standardization. Like other fields, architecture has become standardized due to industrialization. It is an economical principle to have as many compatible parts as possible, and to prefabricate them. But although the technical parts of a building, most importantly mechanical equipment, have been already standardized (electric switches, plugs, taps, pipe line connections, etc.) there is no standardized, prefabricable mechanical system corresponding to architectural needs and vice-versa: there is no building system corresponding to the mechanical infrastructure.

2. Purpose of the project
The long term aim of the Karlsruhe Project is to find a general method solving the problems mentioned earlier. The immediate aim is to develop programs for the design of mechanical systems. These programs are part of an expert system representing our design philosophy, which is:

- to develop a general design method for designing buildings which have complex mechanical infrastructure using the knowledge of previous projects (experience) and
- to develop expert programs for designing mechanical systems as part of an expert system for designing the entire mechanical infrastructure. This expert system is the realization of the design method mentioned earlier and the final aim of the project.

More precisely, we want to have a design method and proper tools which:

1. allow parallel design of a building, i.e., the architect, the structural engineer and the engineers of the mechanical system to be working on the project at the same time, intensively communicating with each other,
2. allow the mechanical system to be an integrated part of the building. The design of all parts of a building should be mutually related,
3. allow the mechanical system to be as flexible as the building itself, i.e., if the building is expandable or can be modified, then the mechanical infrastructure could follow such changes without any problem.
The basic idea behind the above design method is the following:

The common way architects design a building is to start with an abstract building with approximately fixed empty spaces corresponding to the building functions. During the planning procedure, they fill the abstract spaces step-by-step with content (details) modifying the previous concept. The civil engineers do the same; they develop first a construction scheme or skeleton of the building which they improve until all needs are met. Similar to the design of the construction or the entire house, we can design the pipe-line system like a skeleton and refine it step-by-step as in the two previous cases, too.

We consider the architectural, the structural design and the design of the mechanical infrastructure equally important parts of the same design procedure and believe that ignoring one of them results in poorly designed buildings. Several projects have shown that mechanical infrastructure designed carefully, together with other architectural and structural parts of the building, results in a better performance of the building. In a pilot project, which was a training center for the Swiss Railway designed by Professor F. Haller, several principles for the design of mechanical systems were derived from the structural and architectural concepts and vice versa. These principles led eventually to solutions which the participants never before thought of and which probably never before occurred in a building. This was a result of the design philosophy mentioned earlier and intensive communication between the project participants. (I will return to some of these principles in the last section.)

3. Using an expert system

Design problems in architecture are often considered to be different from other design problems. Simon [13] divides problems into well- and ill-structured problems and refers to the building design as a typical example of an ill-structured problem. He suggests, however, that ill-defined problems do not need to be treated differently (in AI) than well defined ones. Here I will not elaborate on this fundamental question, but we considered it very important for our project to adopt the largely accepted view of architects about their design activity rather than the view of other people working with different kinds of design problems; in other words, we did not want to change the attitude of architects, how they look at and how they solve design problems.

Some common properties considered typical in building design are listed below:

* Not all of the design parameters are known at any particular stage of the planning.

* Many of the parameters depend on the decision, or, in other words, the architect's choice of a (sub)-solution.

* The specification of a design task usually contains uncertain data.
• In many cases, the solution can not be derived by a formula (i.e., there is often no algorithm generating a solution).

• For the many exceptions it is not worth developing a strategy or algorithm even if that were possible.

On the other hand, expert systems provide a tool for solving problems having the above properties. Expert systems are not only capable of efficiently solving design problems, but they also have features of a human expert:

1. They can explain and justify why the system has provided a certain answer or solution.

2. They are flexible in the sense that modifying and adding new knowledge is possible.

3. Their knowledge is easily understandable, since it is based on usually simple rules or on experience of human experts.

4. The user's communication with expert systems is in a language which is easily understandable for humans (or at least for professionals of the given area).

5. They can also handle uncertain and incomplete informations.

In the first two years of the project, we acquired knowledge about the design of mechanical systems and formalized rules for the design of air conditioning, sewage, water supply, electric network, etc. The aim was to develop computer programs solving certain well-defined subtasks of the design process. The programs we created are either regular programs, like pipe-sizing or heat-loss calculations, or expert systems, which solve complex design problems like the design of a pipe-line layout or energy analysis. Expert systems themselves use regular programs, but most of the programs developed in the project are not yet connected to each other. Hence, the design procedure at present is entirely guided by the user.

The design parameters, facts, and so on are stored in the data base. The data base contains two parts: a part independent of the project and a part dependent on the project. The independent part is used for a larger class of projects than the dependent. (For the organization of the data base see [3].) The other part of the system, which I call the knowledge base, has access to the data base and can retrieve information (see Figure 3-1).

The advantages of such a system are:

• The completeness of the knowledge and data base is not required, i.e. we have to specify the function of a program but we do not have to worry about other parts of the system.

• We can also add unsolved problems to the knowledge or data base (if we can formalize them).
The scope of the database can vary. We can even change the entire database according to the current design task. (Which is known in AI as multiple context.) Therefore we can save memory space and use it for other purposes.

We can change programs any time we need something new or we do not need an old version, or when, for example, a standard is changed or we have new knowledge about a (sub) problem.

The database is an interactive program used for several purposes rather than just serving the expert system.

There certainly are some disadvantages of this program organization, too:

- Each program (expert) has to fit together with other programs (this is called the interface problem). If we connect a new part to the system, we have to check its compatibility (i.e., we have no automatic compatibility check). Since the design sequence is not fixed, it is difficult to see which parts of the system are affected by the new part.

- Inconsistencies are possible in both, the knowledge and database and have to be controlled by the user. (However, this is not purely a consequence of the organization.)

The expert systems in our design system are production systems currently generating pipeline networks for sewage, water, air conditioning, and electric supply. Production systems have three parts: working memory, production memory, and inference engine (see Figure 3-2).

The working memory represents the current state of the design procedure, i.e., facts, data, and statements about the design task achieved so far in the procedure. The production memory is a set of production rules in the form of implications. The implications have a left hand side and a right hand side which are the condition
part and the action part of the rule, respectively. Both parts contain a (non empty) set of conjunctions of statements. The inference engine seeks all rules which are satisfied by the current state of the working memory, selects one rule and executes it. After execution (firing) the same sequence will be repeated until the system achieves a terminal state. (For a more detailed explanation see [10] and [11].)

Our system is implemented in FRANZLISP and in the frame representation language, PEARL (see [2]). However, in the following part of the paper I will discuss future developments rather than implementation tasks.

4. Expansion of the current project

Up to now the Karlsruhe Project has analyzed and formalized certain subtasks of the design process and if possible, it solved these by computer programs. The next step in the project will be to generate a sequence of subtasks leading from the initial design state to a solution, i.e., from a given building and other mechanical systems parameters (constraints) to a mechanical infrastructure meeting all the requirements. Therefore, on the one hand, we will try to complete our program library for solving subtasks (the knowledge base) covering the whole domain of mechanical system design, and on the other hand, we will try to put together logically dependent subtasks. The putting together of subtasks to form larger subtasks of the design procedure replaces previous decisions of the user (we call this synthesis).

After analyzing the design process and developing design methods, we can also put this knowledge into an expert system. This expert not only knows about the design procedure but also coordinates interactions between other experts. Such systems for guiding and coordinating independent experts are called blackboard systems and are used for a wide range of design tasks (e.g., for the design of metal alloys, integrated circuit layouts, mobile robots). The design rules of the blackboard systems are either general or domain specific. The kernel of the system contains the general design rules maintaining the blackboard itself and communicating with the user. Domain specific rules can start the work of experts or retrieve information from the data.
We can modify and expand the blackboard system according to the knowledge or experience we acquire in the projects developed using this system. We also plan to provide our system with two other important features:

- to develop a data base to store (sub)solutions which can be also used in other projects, and
- to add (meta) rules to the blackboard kernel system which can subtract knowledge from previous projects, i.e., the system will have learning capability.

Our problem solving procedure which we use for modifying and/or expanding our system is a loop (see figure 4-2):

1. Step: We solve a problem using the expert system.

2. Step: If possible, we generate a new expert or a new rule, or just store the data of the problem and go back to step 1.

5. Examples for developing and using experts

As I mentioned in the previous section, we do not have the whole system yet, but we have developed programs or experts which are part of it. The best way to demonstrate how the system works is to show an example. The example below demonstrates how the expert system can be developed and used for designing the mechanical system of a given building. That means, we assume that a certain part of the design is already done and we want to develop a mechanical system using the information from the previous steps of the
design. Unlike in the traditional method, in the expert method (as with CAD methods in general) we can easily design alternatives and check all consequences of a given solution (which are known to the experts). As the entire building is in a "fluid" stage, we can easily change every component in the plan.

In the following I will describe a design process leading from the initial stage of the design to the complete design of the mechanical infrastructure. The process contains a sequence of design tasks. The design tasks of this sequence represent tasks that our expert systems can handle. The sequence itself represents one possible order of steps which the future blackboard system will generate (or will help to generate interactively). Given a building (or usually a rather incomplete description of the building), the first step is to decide the organization concept of the mechanical infrastructure. There are three possibilities:

1. Most of the mechanical parts (pipelines, equipment) are packed into a separate room (usually an entire floor) of the building. Most of the connections are vertical pipelines.

2. Most of the mechanical parts are packed into the ceiling using the free space given by the construction parts (beams, slabs, etc.).

3. A combination of the previous two concepts.

Our project concentrates on the second organization concept and it provides for orthogonal building systems models or schemata of the entire mechanical infrastructure (see [6]). The schemata are either general or object specific schemata. A general schema is a collection of design rules taking no concrete design parameter (as building construction or floor plan) in account. An object specific schema however, is the application of a general schema for a given design task. That means, it is a modified general schema according to the given design parameters. On one hand, if we fix design rules (by fixing schemata) then we restrict ourselves by prohibiting alternatives which contradict the rules. On the other hand, thoroughly chosen rules guarantee that typical design problems (like collisions of pipelines) will not occur in the design process. Therefore the designer can concentrate on problems of higher priority. It is outside of the scope of the present paper to show the proposed concept, but the example below assumes that we have such a concept.
Furthermore, we assume that the underlying modular grid of the building structure is a regular, orthogonal grid. We imagine also that an outline of the general floor plan is given. (For our purpose it is irrelevant what this floor plan looks like or what the function of the building is. However, we assume, that the function of the building implies a complex mechanical infrastructure.) Our concern is the design of the mechanical infrastructure belonging to the part of the building which is represented by the given floor plan.

We can narrow the problem, by making the task simple and more general using the following assumptions:

- We want to design just two systems, let us say the air conditioning and drainage systems. (Obviously these are the biggest and most complicated parts of the installation. Additionally, in order to show common design problems we need at least two mediums.)

- The calculation of volumes and measurements for air, waste water, pipelines are already done and suitable air conditioning and drainage systems chosen (by using our expert systems).

- We restrict possible positions of the vertical pipelines to the middle of the building which has obvious advantages. These positions are by a not further specified general schema given.

- From the function of the rooms and from the construction schema we can also deduce the possible or necessary places of the connections (stubs) of the sanitary water and of the air conditioning pipelines.

Now we can start to solve the problem, namely to link the connection points such that general requirements are satisfied. While we generate a solution, we generate expert systems or regular programs which solve a certain problem and we generate general rules and data. By the assumptions above we reduced the task to the design of pipe line networks connecting vertical pipelines with connection points of intake/output-air and sanitary water. The general schemata for air conditioning and drainage networks provide informations about pipe line organization, possible locations of pipe lines and possible locations of connection points. Our task is to modify these schemata according to the given design parameters.

Before considering this task we have to look at the structure of a pipe line network. Each network has the same hierarchy shown in Figure 5-1: vertical main pipes (1), horizontal main pipes (2), main branch pipes (3), main connection pipes (4), branch pipes (5), branch connections (6) and end-connections (7). The design process of pipe line networks can follow this hierarchy but the order of the design steps usually depends on the medium (air, water, sewage, etc.) and on other design parameters.

Let us start with the design of the horizontal main pipe lines and ask whether a general schema we have chosen is applicable.

**Design Task #1:** What is the minimal height in the horizontal construction (ceiling) which we need for the
Hierarchical structure of pipe lines

How can we reduce the height of the building by minimizing the space for horizontal pipelines?

If we take the diameter of the biggest pipeline (usually the ventilation duct) to be the height of the installation space then we can not have intersections of pipelines of different networks. A solution exists if the graph (network) of the pipelines is embedable into the plane. This leads to a fundamental problem in the graph theory treated in several graph theory books (e.g. in [7]). Figure 5-2 demonstrates two networks (let us say the networks of cold and hot water) connecting the same points (taps).

Planarity problem for networks

Although each network is embedable into the plane, we can not find a planar network containing both
networks. An expert can decide whether such a network exists at all. If no solution is possible, then the expert can list connections (line segments) which prevent the realization. If the graph is embeddable, then the expert can draw several embeddings generating alternative solutions. Even if a planar network exists, it does not necessarily suggest an economical solution as it is shown in figure 5-3. In fact, there usually are a large number of non-economic solutions of no interest. The solving algorithm must attempt to generate a minimal configuration.

Figure 5-3: A possible non-economical planar network

Design Task #2: At least how much space or how many levels do we need if a planar network is not possible or not economical? Try to find out the optimal number of levels.

It is easy to show that we can connect an arbitrarily large number of pipelines using just two levels (supposing that we have enough space in the horizontal direction). Figure 5-4 demonstrates such a configuration of pipelines where the pipelines are connected pairwise. But the construction shows that we need to order the pipelines if we want to avoid conflicts (collisions).

Figure 5-4: Pipeline configuration connecting each pipeline to each other

Design Task #3: How should the network be organized?

Since the orthogonal system is organized in three perpendicular directions x, y and z, the pipe lines should be organized in the same way, so that on each level a pipe segment can lie in the x and/or y direction. If we
allow pipelines to lie on the same level in both directions, then we have the same problem as before, namely the network may be non-planar. Using one level for pipelines lying in the x and the other level for pipes lying in the y direction, we can avoid such problems. Conflicts and conflict free solutions are shown in figure 5-5.

![Figure 5-5: How can arise and avoid conflicts between pipelines](image)

**Design Task #4:** Find a suitable network pattern for horizontal networks.

There are typical layout patterns for horizontal networks. The suitability of a pattern for a concrete network depends primarily on the medium and on the floor plan. (For example, the comb-pattern is suitable for drainage and the cross-pattern for air-conditioning systems.) The four most common patterns are depicted in Figure 5-6.

**Design Task #5:** How should the connections be organized?

After laying down the organization principles for the horizontal pipe lines we can organize the branch connections to the equipment. We have a similar problem as before, but we have fewer directions because these pipelines connect equipment using shortest ways either to the top or to the bottom, but never to both. According to the size of the pipe lines and the number of connections in a limited area we face different connection problems. Figure 5-7 shows conflicts and conflict free solutions for the branch connections. In the project a general connection schema was developed providing solutions for a large number of connection problems (see [6]). Rather than digress here to show this schema we can assume having or developing a connection schema for our design task.
Figure 5-6: Patterns for horizontal networks

Figure 5-7: Problems and solutions of pipe line connections

**Design Task #6:** A network connecting reference points of vertical pipelines and connection points (pipeline endings) should have minimum length.

It is enough to consider the problem in the plane, since having the solution, i.e., the network for each level,
and by connecting these networks we get a three dimensional network of minimal length. The problem in this case is to find two closest points in the networks, each of them belongs to different networks (see figure 5-8). However, the problem is not as simple if we have more than one shortest network connecting the same points in the plane and there is no vertical straight line connecting two such networks in different planes.

![Figure 5-8: Connection of shortest planar networks](image)

The problem, which is to find a shortest planar network connecting given points in the plane (or in a planar network) is a well known problem in mathematics, called Steiner problem. We have a set of points in the plane with a given metric. Metric means the definition of the distance between two points and if we have the Euclidian metric, then the distance is the length of a straight line connecting the two points.

We have here the Minkowski, taxi cab or Manhattan metric, i.e., the distance of two points, a and b, is the length of the semiperiphery of the rectangle parallel to the axes x and y spanned by a and b. We know the solution for a small number of points, but M.R.Garey and D.S.Johnson [4] have shown that this problem, called the rectilinear Steiner tree problem, is NP-complete. That is, we can not hope to find an efficient algorithm generating Steiner trees for a larger number (let us say 80-100 points, see also [1]). Efficient algorithms providing an approximate solution would be satisfactory, but no algorithm is known to have the following properties, important in the context of building design (see [9] and [8]):

- The length of the Steiner tree provided by the algorithm should be not much larger than the length of the optimal Steiner tree. (The error factor should be less than a certain value).
- Hindrances like stairs or elevators should be possible to be consider.

**Design Task #7**: Find a solution for the horizontal pipe line networks by analyzing and synthesizing previous subsolutions.

With the help of shortest connection networks and network patterns we can find a compromise for each
network. Pipe lines belonging to different networks can often be bundled providing obvious assembling, construction and other advantages. However, building bundles often contradicts other design principles, e.g., length minimality. The general schema restricts the number of pipe lines we can bundle. This number will be further limited by the space needed for the construction parts. We seek a balanced distribution of pipe lines which takes all these requirements in account. Changing pipe line directions also plays a central role in the network design (e.g., the pipe line for sanitary water should not build a siphon). In some cases minimizing of direction changes becomes an important design problem.

**Design Task #8:** Determine the connection points of the vertical pipe lines.

Each vertical pipeline serves a certain area in the floor plan, i.e., there is a circle in the Euclidian case and a parallelogram in the case of the Minkowski metric having as its center our reference point of the vertical pipe line. We will get the minimal number of vertical pipelines if the intersections of the circles or parallelograms are 0. On the other hand, the capacity of a vertical pipeline should be designed with regard to future connections in its serving area. Hence, solutions obviously contain intersections of serving areas which can be minimized. The boundary of the serving area depends on the chosen direction of the pipelines lying on the top or bottom level as it is demonstrated in figure 5-9.

![Diagram](image)

**Figure 5-9:** Serving area of the vertical pipelines

**Design Task #9:** We have to find a compromise between the geometry of the construction and the necessary space of the pipelines (or bundles).

The schema for air conditioning and drainage systems has to be modified by the given building construction. This means that construction parts occupy certain areas and we have to find a solution by either rearranging pipe lines (or bundles) or modifying the construction parts. It is useful to first develop a construction specific
schema and test the compatibility of the mechanical systems with the construction parts.

Throughout the design procedure of this example, we saw that every decision has a chain of consequences. Each decision opens and closes ways to several solutions. Therefore, it is not worth showing further details to make the example more or less general. If we continue this sequence of design tasks, then we achieve either a solution (complete design of the mechanical infrastructure) or a dead end [having no possible (sub)solution]. Evaluating and backtracking leads eventually to other solutions. After finishing a design problem, the design procedure itself also will be evaluated.

6. Conclusion
This paper describes a project at the University of Karlsruhe, in West-Germany. The purpose of the project is to develop a new method for the design of buildings which have a complex mechanical infrastructure. The research embraces four areas: a general model for the design of mechanical systems, specific models for the design of air conditioning, heating, etc. systems, rules describing the design procedure and a computer based design assistant. I described the fundamental questions and problems arising in this areas and the principles of the design process. The design process is supported by computer programs which are mainly expert (production) systems. Some parts of the design process can already be solved by the computer. However, currently used programs are not connected yet and an aim of the ongoing project is to develop a blackboard system guiding and coordinating the entire design process.

References

Rectilinear Steiner trees: efficient special case algorithms.

[2]  Deering, M., Faletti, J., Wilensky, R.,
PEARL - A package for efficient representation in LISP.
IJCAI, 1981.

[3]  Eastman, C. M.,
Database facilities for engineering design.
The rectilinear Steiner tree problem is NP-complete.

Planung, Betrieb und Unterhalt von Leitungsführungen in Hochinstallierten Gebäuden.

ARMILLA, ein Installationsmodell, Instrumentarium zur Planung von Leitungssystemen in hochinstallierten Gebäuden.

[7] Harary, F.
Graph Theory.
Addison-Wesley, 1969.

[8] Lenart, M.
An algorithm for the rectilinear Steiner Problem.

[9] Lenart, M.
Über kürzeste Leitungsführungsalgorithmen.

R1: A rule based configurer of computer systems.

A rule-based blackboard kernel system: some principles in design.

[12] Rychener, M.D.
Expert systems for engineering design: Problem components, techniques, and prototypes.

The structure of ill structured problems.