**Zeitschrift:** IABSE reports = Rapports AIPC = IVBH Berichte

**Band:** 58 (1989)

**Artikel:** Development of an advisory system for site managers

**Autor:** Gehri, Markus

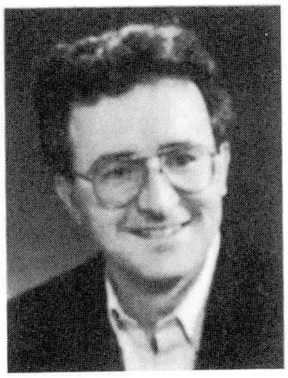
**DOI:** https://doi.org/10.5169/seals-44900

# Development of an Advisory System for Site Managers

## Développement d'un système conseil pour le chef du chantier

## Entwicklung eines Beratersystemes für Baustellenleiter

**Markus GEHRI**
Dipl. Bau. - Ing. ETH/SIA
Swiss, Fed. Inst. of Technol.
Zürich, Switzerland



Markus Gehri, born 1953, received his civil engineering degree at ETH in Zürich. He worked for two years as assistant site manager in Munich and later for five years as site manager in Nigeria. Three years ago he came back to the ETH to do research work concerning ComputerAided Site Management.

**SUMMARY**
By use of modern computer methods we want the site management to become more attractive and efficient. The article describes the development work of an advisory system for site managers which shall fulfill different tasks emanating from a knowledge-based daily report. The aims and procedures will be explained. The intendent use of a meta-expert-system has lead to difficulties. The inadequacy of traditional shells for this task will be discussed and the necessity of developing a data model will be considered. An alternative attempt which is object-oriented will be presented.

**RESUME**
Notre but est de rendre la gestion d'un chantier plus attractive et plus efficace à l'aide des méthodes informatiques. L'article décrit le développement d'un système d'assistance pour la gestion d'un chantier par le chef de chantier. Ce sysème doit remplir des tâches différentes dont la base est le rapport journalier preparé à l'aide d'un système expert. Les buts et les procédés sont expliqués. L'introduction d'un méta-expert-système a été difficile. L'incapabilité de shells traditionels est traitée et la nécessité de créer un modèle de base de données est envisagée. Une variante de mise en application, orientée objet, est égalment présentée.

**ZUSAMMENFASSUNG**
Durch den Einsatz moderner EDV - Methoden soll die Baustellenführung attraktiver und effizienter gemacht werden. Der vorliegende Artikel beschreibt die Entwicklungsarbeiten an einem Unterstützungssystem für den Baustellenleiter, dasausgehend von einem wissensgesteuerfen Tagesrapport verschiedene Aufgaben erfullen soll. Es werden die Ziele und die Vorgehensweise erläutert. Die vorgesehene Einführung eines Meta-Expertensystems hat zu Schwierigkeiten geführt. Unzulänglichkeiten traditioneller Shells für diese Aufgabe werden besprochen und auf die Notwendigkeit der Erstellung eines Datenmodells wird eingegangen. Ein alternativer, objektorientierter Ansatz wird vorgestellt.

*Preliminary Remark:*

*The following article describes a project at the Institute for Planning and Construction Management (Swiss Federal Institute of Technology Zurich) that is still in progress. At the time of writing this paper I am evaluating the software which is to be used. The essay describes therefore mainly the preliminary work and the experiences gained with expert systems and expert sytem shells.*

## 1. WHAT WE WANT TO DO

### 1.1 Goals

The building sites are generally managed by one person only (site manager or foreman) whose status of knowledge can vary according to the size of a site. The most important task of these managers is to keep their sites running. This leads sometimes to a neglection of the administrative and economic aspects.

Our basic idea is to run all the building sites of one contracter as independent profit or cost centers. This implements, that the building site manager should be aided in the best possible way by all available means to fulfill his work in the shortest possible time and with a basic competence.

The following of the site manager's jobs should be supported:

- *To gain and keep the general overview by concentration, registration and clear presentation of the necessary information.*
- *Analysis and decision supporting by comparing and evaluating different alternatives of acting.*
- *The disposition will be improved due to high flexibility without loosing the link to planning and economical constraints.*
- *The controlling task shall be systematisized by building in minimal requirements which can be varied by the site manager according to his needs.*
- *The time to do creative and demanding work shall increase by taking over as much as possible of the routine (administrative) work by the computer.*

Additionally we try to improve the access to existing applications such as bidding and project planning programms by providing a common user interface. By incorporation of new solutions (e.g. graphic, expert systems or simulation packets) the building site computer should be able to become the central element of the construction process concerning organisation and administration.

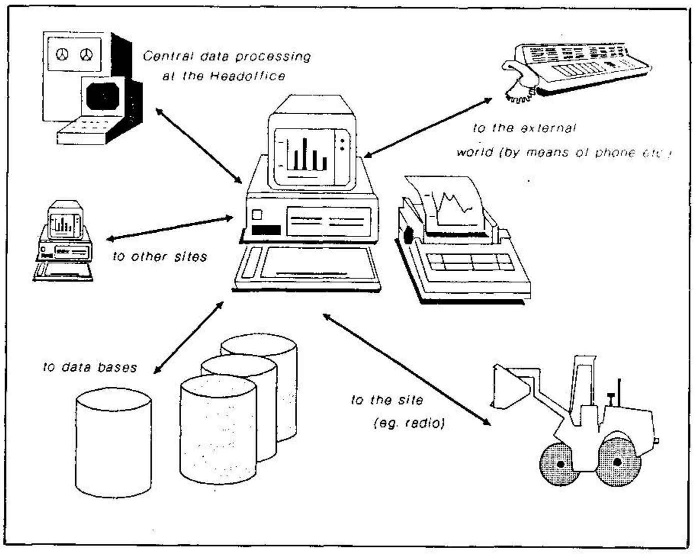


*Fig.1 PC as central element*

### 1.1 Analysis of the site manager's job

The tasks of the site manager are among the most varied (and most satisfying) that arise during a construction process. We examined the following aspects which we got from our own experience, from books or from discussions with other people:

- *Which tasks have to be settled by the site manager and how can these tasks be classified?*
- *What information is used at what time by those engaged in the building process, how accurate and in what manner has it to be?*
- *For which tasks and how far can we already offer computerised solutions?*

Detailed listings were made of the analysis of the site managers tasks and the flow of information. They cover a big part of all possible aspects. Hereby we found out that a splitting of the tasks into rhythme, importance and expenditure is useful. It is easy to find a measurement for the rhythme and the expenditure. The importance, however, is always in relation with urgence and needs (not to forget the site manager's personal affinity that makes him select one job out of a big heep of unfinished works).

The flow of information was split into necessary and useful information and into emitter and receiver of information.

Emanating from these investigations we consider a computer aid as being useful where:

- *data processing and/or calculations have to be done,*
- *the site manager has to write, to report or to draw a diagram,*
- *solutions have to be found and proved,*
- *plans and results must be analysed, compared and estimated.*

## 1.2 The basic idea

We ought to provide the site manager with a computerised assistant with advisory functions. (*This "domain assistant" will in the future eventually together with an "office assistant" and a "communication assistant" form the only interface between user and computer [1]*). This assistant serves him, in relation to his daily routine jobs as an multipurpose aid to manage his tasks.

We imagine the adviser to be a consultatively useable instrument which opens the way (based on knowledge) to how to deal with the most important problems of the site manager and which offers the necessary modules (as far as implemented) for their treatment. The adviser should help to supervise all incoming and outgoing information, the costs and to a certain point the technical problems of a site and it should indicate the manager arising problems as early as possible. Moreover the system should support the manager in time management and as far as necessary remind him the treatment and fulfilling of jobs.

It should be possible to connect some desired applications with the basic module in such a way that they can be called and treated with a standardized user interface. That means that the adviser himself calls external cooperators which can deal with calculating or data intensive tasks (*programs*) or which are able to advise him (*expertsystems*) if he likes to base his work on expert's experience.

In order that such an expert system will be attractive for the site manager, a strong stimulus for the daily use has to be created. One argument will be that the daily administration jobs can be done computer-aided to make them

*-faster / more efficient,*
*-more comprehensive / more systematical,*
*-qualitatively better.*

Furthermore all the information proceeded by the adviser will be of easier access in case of further use and treatment.

The function of the basic module is to direct the access with all other modules, perhaps to choose an appropriate data bank or knowledge domain and to maintain the dialogue with the user of the system. Therefore it can be called a meta-expert-system.

To provide the system with all the necessary information about the proceedings on the site, a regular (if possible daily) consultation should take place.

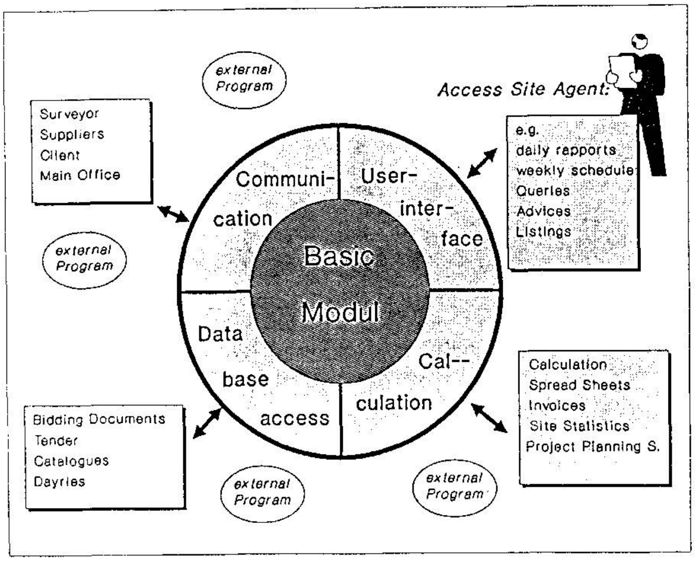


*Fig. 2 The site manager's adviser*

We imagine the following daily routine consultation in the sense of a "*management report*"

- *The goal of the "meeting" is to prepare a daily report for the own overhead business management or perhaps for the customer, too.*
- *In the course of the meeting the actual situation of the site is "discussed". The system has got a dynamic access to the "history" of the site and, if wished to the "history" of other sites of the firm (wich information refers to the "history" has to be decided later).*
- *The system asks all information from the site manager which is necessary to write down a day report. If possible the information is derivated from the "history" (e.g. "Is the weather as it was yesterday?").*
- *If a certain information is not at hand (e.g. the foreman's daily report) the system keeps it pending and asks for it in one of the following sessions. A subsequent input should be possible with a shortened procedure e.g. by a quantity surveyor.*
- *Additionally detailed working plans for the next 2 - 4 weeks should be produced in the sense of a "rolling planning".*
- *Because the system should have at hand the bid, the project plans as well as the actual capacity information and progress reports, it is possible to advice the site manager in his planning work.*
- *finally the system can present and file the results of a session in any desired way or form.*
- *It is possible that during a session check backs to other systems and persons can be arranged.*
- *After each session, a list with all the actually pending jobs of the site manager can be printed out.*

## 2. EXPERIENCES WITH ES

### 2.1 First Steps

At our Institute we developed (partly in cooperation with the University of Innsbruck) four expert-sytems, using the shell *XIplus*.

The first one concerned the conversion of a part of the SIA-Norm 118 *(consequences resulting from change orders caused by the client)*. In the whole we needed 70 rules, thereof 34 rules served the control of the lapse of the consultation. The system was equipped with help-functions and explanations, and during the following attempt to bring it into use among our students we experienced that the already tested system broke down at the moment when all the helps and explanations had been called (memory overflow).

Three further expertsystems were built up at the University of Innsbruck. They help with the preparation of building sites that means the planning of stationary equipment. The following topics were worked on:

- crane disposition, *the system was split into 16 modules with 158 rules, thereof 58 rules concerned the consultation control.*
- accommodation of the crew, *the system was split into 5 modules with 124 rules, thereof 12 rules concerned the consultation control.*
- Dimensioning of the concret mixing plant, *the system was split into 6 modules with 87 rules, thereof 19 rules concerned the consultation control.*

Three main points came up during this works:

- *The more narrow and specific the domain was the more interesting it became for the enduser. That was the result of the fact that special cases could be included and a real knowledge transfer was possible. Contradictionary we found out that the application became boring with a too narrow domain. Who is going to buy a shell just to find out if he should lodge his workers in a hotel, in mobil containers or in temporary barracks.*
- *The work involved to get a reasonable course of the consultation is sometimes enormous with rule-based shells. It became evident that it is of high importance to structure the rules and to know exactly the derivation mechanism of the interference machine.*
- *The subdivision of the applications into modules was necessary because the active memory of small machines (PC) is very restricted. The resulting difficulty consists in planning these modules as far as possible as homogenious domains because the data transfer between the modules is only possible in one direction and a backtracking over the module limits is not possible. Research efforts are made to push forward this modularity by means of a "blackboard" architecture [2], as to our knowledge no commercial shell supports these possibilities.*

Our experiences with these minor applications can be resumed as follows: the developing of an expert system, even of very small applications, needs analysis. It is important that the dialog control is thought over very carefully because everything that appears on the screen and the time when it appears can be directed only indirectly through the interference machine. This is contradictionary to the procedural (algorhythmic) approach.

One of our experience was also, that the whole backtracking mechanisme (WHY...?) which is considered a central element of expert systems, was only a good debugging-aid for the developer. Within these small systems it was rarely used by the enduser or only during the first run due to curiosity.

As a consequence from these experiences we looked for a "big" system to develop our advisory system for the building site manager. For further student works at the university of Innsbruck we are going to exchange the elderly XIplus against another PC-shell. We wish to get more functions and more capacity as well as a licence-free runtime module which can be tested by a running building site without beeing forced to buy additional copies of the shell.

For the development of the advisory system we got the shell TWAICE from Nixdorf running in UNIX and which is not (yet) available on a PC-platform.

## 2.2 The Shell TWAICE

TWAICE has, compared to other shells, some peculiarities which I like to mention hereunder:

- *The shell is based completely on PROLOG. In some way it can even be considered as a programgenerator to write PROLOG-programs. The advantage hereof is that the interference mechanisme is very fast and powerful (with even the possibility to ask WHY a fact was NOT concluded). PROLOG has, however, always some traps ready for the programmer.*
- *TWAICE is a rulebased shell wich is supported by an object-hierarchy. The rules refer to attributes of the objects which can adopt one or more values even with different probabilities, too. The objects (and the attributes) of TWAICE have got a predefined number of slots which can be filled with certain information (e.g. default values for an attribute or the indication whether it is permitted to ask for the value of an attribute or not) or which can be used to build up the rule base (e.g. EXIST: does the object exist?).*
- *The flow of a session can be controlled by an object hierarchy, rules needn't be written therefore. The inference mechanisme is described in figure no 3.*
- *Under some conditions predictated by the developer (e.g. rule-directed) TWAICE can create several instances of the objects that are defined in the kowledgebase (e.g. an engine has got 4 pistons). All of them get the same slot values and the same rulebase is used to evaluate their attribute values.*

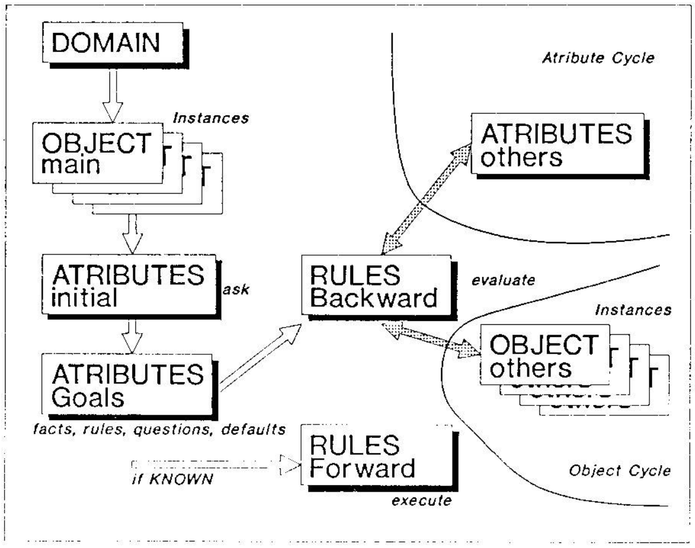


*Fig.3 The Inference- Engine of TWAICE*

- *TWAICE offers a lot of primitives (PROLOG- predicates), besides C-primitives can be called. The system therefore is very powerful but it requires for some knowledge at least what concerns PROLOG.*

## 2.3. Experiences with the Shell

First of all I transferred the XIplus-Model "SIA 118" and discovered that I needed less rules, but on the other hand the user interface was a lot more complex and difficult to program than I was used to from the PC. Everyone who is used to colours, graphics and mice to interact with his PC will have a lot of difficulties.

Another problem was the PROLOG specific treatment of arithmetic expressions or the syntax of procedures (which are necessary even for simple, non-automatic generated questions). These (beginner) problems are not described in the manuals, but a short training with PROLOG and the study of the examples were sufficient to help solving most problems.

But because some more PROLOG-specialities (e.g. the special treatment of the negation NOT) were integrated in TWAICE, the full function of the shell can only be used with some knowledge in PROLOG. PROLOG-Experts, however, prefer a direct implementation of their expert systems in the pure language.

Because the above mentioned possibility of instantations coincides very well with the concept of a meta expertsystem that is newly started every day, I pushed forward quickly into this field of the shell.

To test the capacity I made it run three models of different size in an endless loop until I got a range overflow or until the time to get an answer was too long. Figure 4 shows the results.

Looking backward, we can say that the number of facts limited the models, whereas a fact can be, for example, a generated object or a traced attribute. Even if it were possible to upgrade the system's hardware, it would not be possible with an information input of over 100 facts daily to treat a building site as instances of several individual days.

TWAICE Capacity Test

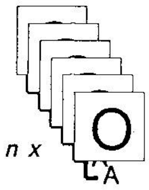
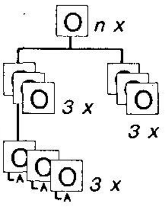
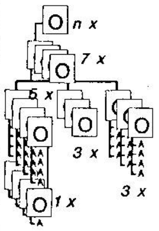
| | TEST 1 | TEST 2 | TEST 3 |
|---|---|---|---|
| MODEL |  |  |  |
| No generated | 1445 | 116 | 6 |
| Time for 1st | ca. 1s | 19s | 300s |
| Time for last | 18s | 2700s | unknown |
| Break by: | Overflow | time out | time out |

*Fig. 4 TWAICE Capacity test*

A next trial would be to write the facts that are no longer used into a database and to read them in again when necessary. Although this system would be possible, we would resign many possibilities of the expertsystem attempt. The reading in into the expertsystem environment has to be selective according to predifined critirias.

Within our Institute, we dispose of some experience in handling PC database systems, but due to the fact that the shell is based on an UNIX environment, the database had to be based onto this platform as well. Therefore I decided to resign on further tests with the TWAICE solution.

The question was then, if we could continue with a declarative attempt for the basic system.

## 2.4 Declarative or procedural?

One of the advantage of expert systems is that the base of knowledge can be completely unstructured, what means that we talk of a declarative environment. Each knowlegde element has got access (through the interference mechanisme) to each other knowledge element at any time of the session. That implements a big workspace with reserves to store new information (in the case of TWAICE the new instances).

Opposite to this we get the procedural or algorhythmic approach. Here the system developer has to know exactly what has to be treated at what time and where, if necessary, the parameters can be called and stored.

The advantages of the declarative programming are most convincing where, by means of small programming elements, a high variety of results can be achieved.

A well known example is the PROLOG-predicate:

**Member (element, list)**

which can be used as a test procedure, access or generating procedure.

However, if an enduser works with an expert system the goal of the consultation is already defined at the beginning of the session, a lot of systems are explicitly capable to attend one goal only, e.g. to discover the fault of an engine or to find the ideal combination of machines for a special use.

For our construction manager's advisory system we decided to resign on shells as a master system and we try to master the data flow in a more traditional way. In the very future we shall try to come to a solution for this problem by means of an object-orientated attempt.

### 2.5 What is problematic with shells?

Shells are offered today in great variety, in all price categories (the expensive ones are called development environments) and in all comfort degrees. All together have one thing in common: the developer needn't worry about the inference process, he "only" should build up the knowledge base.

But all the same, the knowledge engineer has to think about the user interface and the content of the application, should his application ever come into use. Besides, he has to know his shell very well, LISP-based shells behave differently as PROLOG-based or C-based do.

The hundreds of expert systems which are often mentioned today, e.g. DuPont [3], are often small applications which are comparable with decision tables, they often cannot be maintained and they do not make special demands on in- and output.

On the other hand, the shells are ideal aids for novice or, referring to computer science, untrained developers (in our case the students) to get useful solutions with a relatively small expenditure. Seen from this side, they represent for the logic programming what spread sheets represent for calculating tasks and minor database applications: An aid that brings the computer capacity nearer to the enduser, but not a working instrument for the computer engineer to build up bigger applications.

## 3. FURTHER PROCEEDING

### 3.1 Object orientated approach

After having come to the conclusion that an expert systems approach, as well as any other software project, predicts a careful analysis and especially a modelling of the tasks and the data flow, I split the building site model into the following three models: cost model, resource model and administration model.

I like to continue with the definition of these models in an object-orientated environment as independent objects and to write down and test the therefore necessary procedures. At the moment of the redaction of this paper (May 1989), these jobs had just started (with SMALLTALK V), they are encouraging even though difficulties with the data bank capacity rise again, but they appear now in a familiar (PC) environment. I hope to be able to enter more deeply into the further proceeding of the job at the time of the congress.

### 3.2 Cost model

The cost model has to treat the material and the immaterial cashflow of the building site (figur 4). The essential requirement of each consideration of the building site as a cost and profit center is that the site manager is always clear about his cost and efficiancy.

The output is described in the tender and bidding documents and provided with costs in the calculation. Throughout a continuous consideration of the forthcoming costs, a relation can be established between the real output and the SHOULD-costs.

The derivation can be explained as a fault in the calculation, as a justified or unjustified falling-off in output compared with the assumptions or as a rise in price. (Herefore the use of an expert system would be nice).

This area of the construction-site management is rather well known, we worked on it for some time [4].

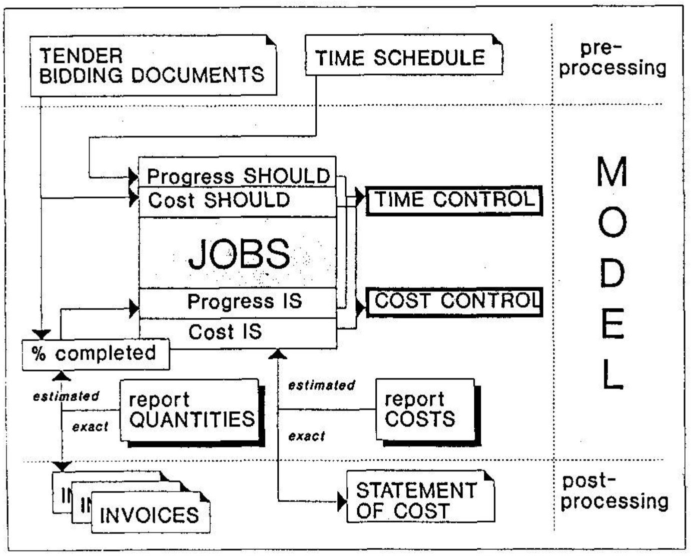


*Fig. 5 The Cost Model*

Difficulties rose only with the estimation of the actual progress and the therewith connected outstanding, additional expenditure as well as with the different presentation of the internal cost calculation and representation.

Therefore the definition of the *jobs*, as an integrated and finished output unit shall be integrated [5]. Jobs, macro-processes or, as a students called them once with an appropriate definition "building sites within the building site", have the properties that they are much more handy for any controlling approach as the traditionally used costruction labour keys (same job at different building parts) or pure physic block- or storey-splitting.

Jobs cannot only be used in the disposition but also in the weekly planning, the surveying and the estimating of the output.

## 3.3 Resource Model

The second important task which is the building site manager's duty is the disposition of the available machines and people to the jobs that have to be done on site. The bidding documents give us the points, but now, the spent hours and the real stand of output have to be considered in the weekly disposition. The model provided for looks as in figure 6.

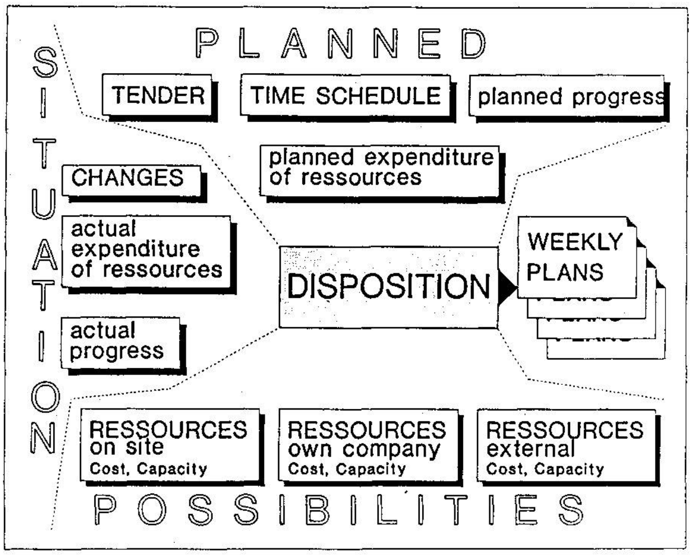

PLANNED
SITUATION
TENDER
TIME SCHEDULE
planned progress
CHANGES
planned expenditure of ressources
actual expenditure of ressources
DISPOSITION
WEEKLY PLANS
actual progress
RESSOURCES on site Cost, Capacity
RESSOURCES own company Cost, Capacity
RESSOURCES external Cost, Capacity
POSSIBILITIES

*Fig. 6 The Resource Model*

### 3.4 Administrator Model

With the administrator model we are going to spare the most of the site manager's time and therewith creat the biggest profit. We intend to concentrate all the outgoing information of the site in a central model and we shall read all the incoming information into this model whenever feasible. The working out of this model is postponed for the moment because it has to submit the two main models and because it is able to do so. (figure 7).

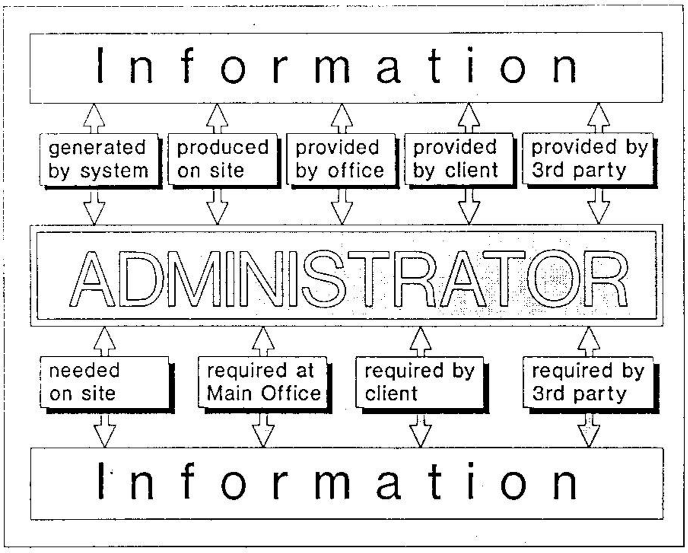


*Fig. 7 The Administrator Model*

**REFERENCES**

[1] GMD, Computer als Assistenten, Journal Führung und Organisation, June 1987

[2] HAYES-ROTH B., A Blackboard Architecture for Control, Artificial Intelligence, 26/1985

[3] various authors, Application Corner, PC AI, 1987-1989

[4] STRADAL O., Short Term Management on Building Sites, IABSE Journal, February 1981

[5] LESSMANN H., PC-Einsatz für die Baustellenkontrolle,IBETH Publikation, April 1987