20A

7th International Conference on Systems Research, Informatics and Cybernetics
(August 15-21, 1994, Baden-Baden, Germany)

PRECONFERENCE

PROCEEDINGS

ADVANCES IN COMPUTER-BASED BUILDING DESIGN SYSTEMS

(Focus Symposium, August 16th)

Focus Symposium Chair:

Professor Jens Pohl
Director, CAD Research Center
Design Institute
College of Architecture and Environmental Design
California Polytechnic State University
San Luis Obispo, California, USA

Sponsored by:

The International Institute for Advanced Studies
in Systems Research and Cybernetics
and
Society for Applied Systems Research

Professor George E. Lasker
Chairman

(ISBN: 0-921836-20-1)

Preface

Commercial CAD software has evolved very gradually over the past 25 years from an initial singular focus on drawing automation. Even today, in the majority of architectural offices CAD stations are still employed as largely stand-alone units to serve the specific purpose of drawing production and manipulation. Conceived in this limited application realm, CAD-drawing packages have been implemented as large programs that typically encompass all of their functionality within a single process.

The first significant departure from this centralized approach came with the realization that the value of drawings can be greatly increased if the objects that they contain are directly linked to the information that they represent (i.e., functions, relationships to other objects, materials, cost, ownership, availability, etc,). While this led initially to the addition of attribute files (e.g., as symbol libraries) and spreadsheets within the same CAD program, it was soon recognized that the need for a great deal of data could not be accommodated efficiently in the internal data structure of the CAD package. In any case, this data structure was designed to represent objects geometrically rather than semantically.

In the late 1970s and early 1980s, search for a more suitable data management facility focussed attention on the relational database management system. The provision of two-way linkages between relational databases and drawing and solid modelling packages has not been entirely successful in addressing these needs. Unfortunately, the internal structure of the CAD program did not lend itself well to external connections. Since the CAD package had been designed as a large self-contained program it was difficult to find convenient data entry and exit points without significant consequences to the operational flow of the package. As a result, even today most of these CAD-database linkages are time consuming, cumbersome, and functionally constrained.

Attempts to integrate the geometric representations of the CAD program with the data structures employed by the external databases were only partially successful. Attempts to store the geometry of the drawing together with the non-geometric attributes in the relational database failed. Relational databases are optimized to process large numbers of short transactions and are therefore unable to accommodate the much longer graphic transactions required in the CAD environment, within reasonable response time limits. The alternative solution, to store the geometry separately from the attributes and link the two with embedded pointers, has imposed another layer of complexity on an already contrived connection.

Nevertheless, the addition of virtually unlimited amounts of attribute information to drawing objects has provided an opportunity to incorporate decision making assistance in the CAD environment. However, for several reasons progress in this promising area has been disappointingly slow. First, the geometric descriptions created by current CAD systems are essentially limited to points, lines, faces and shapes. A description that is useful for reasoning purposes should represent the geometry of the artifact in terms of geometric objects that have meaning in the real world in which the artifact will be constructed and used. In other words, for a CAD system to support the design activity it must contain high level knowledge about the evolving design solution and its context.

Second, it has been difficult to link reasoning agents to CAD programs that are designed for single-tasking environments and are highly demanding of the available resources in their execution environment. Accordingly, initial commercial attempts in the middle and late 1980s were limited to the incorporation of single expert systems in mostly the in-house CAD systems of large engineering, aerospace, and construction firms. Although such reasoning agents have been successful their usefulness has been largely restricted to sub-problems, such as staircase construction, welding, lighting, and thermal analysis.

Third, when these early explorations led to the conclusion that effective design decision support will require, at the very least, multiple agents interacting within an integrated CAD environment, no suitable cooperative model was found to be readily available. Only a few appeared to exist, mostly in university research units, as engines for theoretical research endeavors.

During the early 1990s, researchers have increasingly focussed on the design and implementation of distributed, cooperative architectures that allow multiple human designers and computer-based design assistants (i.e., agents) to collaborate. It is therefore not surprising that the proceedings of this focus symposium on Computer-Based Building Design Systems should include a majority of papers dealing with various aspects of cooperative computer-integrated design environments. This trend is reflected particularly in Sections 1 and 2, where authors address both theoretical and philosophical questions relating to design knowledge and user-computer interaction, and describe various approaches for the implementation of cooperative design systems.

It is my believe that much of this current work in cooperative, user-computer partnerships will form the basis of a new generation of decision-support systems that will allow computer-based design environments to contribute intelligently to the design activity.

Jens Pohl June 24th, 1994

7th International Conference on Systems Research, Informatics and Cybernetics

August 15-21, 1994, Baden-Baden, Germany

SPECIAL FOCUS SYMPOSIUM

Computer-Based Building Design Systems

August 16th, 1994

SYMPOSIUM PROGRAM

TUESDAY, August 16th

8:00 to 8:20 **Authors' Meeting - all authors are requested to attend meeting in the Kongresshaus.**

8:30 to 10:15 **SESSION 1: Cognitive Aspects of Computer-Based Design**

'Epistemic and Cognitive Frames for Knowledge-Based Design'
T.Oksala, Dept.of Architecture, Helsinki University of Technology, Helsinki, Finland.

'CAD Used as Interface for a DSS'
T.Calosci, Dipartimento di Processi e Metodi della Produzione Edilizia, Universita degli Studi di Firenze, Italy.

'Intelligent Active Assistance for Knowledge-Based Environmental Design'
D.Shaw, T.Napier, D.Yang and R.Stottler, US Army CERL and Dept. of Computer Science, University of Illinois and SHAI, Artificial Intelligence Consulting, USA.

'Interfacing CAD Representations and a Knowledge-Based System'
J.Bermudez and M.Haley, Graduate School of Architecture, University of Utah, USA.

10:30 to 12:45 **SESSION 2: Models of Integration and Cooperation**

'May Preconditions of Architectural Projects be Regarded as Cases?: Development of a Case-Based Tool'
V.Bottelli, P.Drago and C.Fogh, Dipartmento di Progettazione Programmazione e Produzione Edilizia, Politecnico di Milano, Milano, Italy.

'A Model for Collaboration Among Distributed Computer-Supported Human Designers'
T.Khedro, CIFE, Stanford University, Stanford, USA.

'DDS - Distributed Design System: A Paradigm for Modeling Conceptual Architectural Design'
I.Svetel, IMS Institute, Belgrade, Yugoslavia.

'Cooperative Distributive Construction Problem Solving'
B.Jones and M.Riley, University of Brighton and University of Southampton, Sussex and Hampshire, England.

'An Object Agent Approach to Integrated Cooperative Decision Making'
K.Pohl, CAD Research Center, Cal Poly, San Luis Obispo, USA.

14:15 to 16:00 SESSION 3: Models of Assistance and Computation

'HeGeL-II: Heuristics and Optimization Based Search in Early Design'
O.Akin and R.Sen, Dept.of Architecture, Carnegie Mellon University, Pittsburgh, USA.

SESSION 4: Education and Practical Realities

'A Computer-Based Teaching Tool for Traditional Chinese Wood Construction'
A.Li and J.Tsou, Dept.of Architecture, Chinese University of Hong Kong

'A Knowledge-Based Design Model for Designing Row-Houses'
G.Cagdas, Faculty of Architecture, Istanbul Technical University, Istanbul, Turkey.

'Computer Aided Projects: Quality Assurance Procedure in Building Design'
M.Esposito, Dipartmento di Processi e Metodi della Produzione Edilizia, Universita di Firenze, Italy.

Table of Contents

SECTION 1: Cognitive Aspects of Computer-Based Design

'Epistemic and Cognitive Frames for Knowledge-Based Design'
T.Oksala, Dept.of Architecture, Helsinki University of Technology, Helsinki, Finland.

'CAD Used as Interface for a DSS'
T.Calosci, Dipartimento di Processi e Metodi della Produzione Edilizia, Universita degli Studi di Firenze, Italy.

'Human Activities in Knowledge-Based Architectural Design Systems'
S.El-Attar, Dept.of Architecture, Al-Azhar University, Cairo, Egypt, and CAD Research Center, Cal Poly, San Luis Obispo, USA.

'Intelligent Active Assistance for Knowledge-Based Environmental Design'
D.Shaw, T.Napier, D.Yang and R.Stottler, US Army CERL and Dept.of Computer Science, University of Illinois, and SHAI, Artificial Intelligence Consulting, USA.

'Interfacing CAD Representations and a Knowledge-Based System'
J.Bermudez and M.Haley, Graduate School of Architecture, University of Utah, USA.

Conclusion

The prototype refinement model, ... aper, is based on the shape grammar formalism and t ... wledge for a chosen prototype. The typological, topol ... wledge of the English row-houses has been defined and ... ng the design product have been set. The knowledge-ba ... in the context of this shape grammar and of the knowle ... algorithm formulating the routine design process is in t ... oduct, which has been generated at the end of the implen ... possible solutions from the design space for the row-houses.

On the other hand, this knowledge-based design system simulates the human thinking in a routine design process. For this reason, it can be used for architectural design education in the context of computer-aided design. It can help the students to grasp the architectural design process better. And it is also significant that it exposes the reasoning involved in generating alternative design solutions.

References

Brown,F.E. and J.P.Steadman (1991a); The morphology of British housing: an empirical basis for policy and research. Part 1: Functional and dimensional characteristics; Environment and Planning B, Vol.18 (pp. 277-299).

Brown,F.E. and J.P.Steadman (1991b); The morphology of British housing: an empirical basis for policy and research. Part 2: topological characteristics; Environment and Planning B, Vol.18 (pp. 385-415).

Coyne,R.D., Rosenman,M.A., Radford,A.D., Balachandran,M. and J.S. Gero (1990); Knowledge-based Design Systems; Addison-Wesley.

Gero,J.S. and M.A.Rosenman (1989); A Conceptual Framework for Knowledge-Based Design Research at Sydney University's Design Computing Unit; Proc. 4th Int. Conf. on Artificial in Design (ed. J.S.Gero), Springer-Verlag (pp. 363-382).

Rosenman,M.A., Gero,J.S. and R.E.Oxman (1992); What is in a Case: The Use of Case Bases, Knowledge Bases and Databases in Design; CAAD Futures'91, (ed.G.N.Schmitt), Vieweg, Wiesbaden (pp. 285-300).

Smith,L. (1986); Investigation Old Building; British Council Press.

Steadman,P., Brown,F. and P.Rickaby (1991); Studies in the morphology of the English building stock; Environment and Planning B, Vol.18 (pp. 85-98).

Stiny,G. (1980); Introduction to shape and shape grammar; Environment and Planning B, Vol.7 (pp. 343-351).

Taş,G.D. (1993); A shape Grammar Model for Designing Row Houses; Istanbul Technical University, Inst. of Applied Science, Ms. Thesis.

'Computer Aided-Projects' Quality Assurance Procedure in Building-Design

Proposal for an Application Experiment

M.A. Esposito
Dipartimento di Processi e Metodi della Produzione Edilizia
Università di Firenze, Italia
E-Mail: EPU@CESIT1.UNIFI.IT

Abstract

The objective of the Proposal for an Application Experiment is aimed at the definition and implementation of an IT (Information Technology) procedure for Quality Management in Building-Design according to the European standard EN-29001.
The procedure should be based on the concept of building performance design and aimed to define a decision support system for the optimization of the design process. Furthermore the procedure should provide a full building-design process documentation as required by Quality Assurance standards and should allow the design team the possibility to obtain the certification.
The application experiment develops an optimised design procedure for professionals (i.e. architects or engineers small companies) based on a widely-circulated software environment in the building sector.
The application should be exploited using debugged 3rd generation tools and 4G tools as an extension of a standard CAD (Computer Aided Design) environment running on PC (Personal Computer) integrated with EDMS (Engineering Document Management Systems).
The application experiment would permit to obtain a dynamic management and documentation procedure during the design process.
Software engineering practice in small companies should accord with Software Quality Assurance criteria.

Keywords

Quality Management, Computer Aided Design, Engineering Document Management Systems, software engineering.

Current status of organization and software engineering practise in small AE companies

The current status of software engineering practise in small Architect/Engineering companies, assumed as case study can be described as the typical customization activity for a CAD application versus the building-design activity. The software processing provides shaping and dimensional control, drafting and data base implementation.
The typical functions generally adopted are:
A-storing and retrieving the evolving design information in a structured manner through system-layering and data base co-ordination;
B-representing and presenting by 3D visualization the design information with printouts at an appropriate stage of the design's evolution for the purposes of assessment and client review;
Today's mission of the CAD application in these types of small companies, especially in Italy, is to produce more rapid projects draws (scheme and detail design), to permit fast modifications and 3D views and to relate draws with the specifying phase documents.

In spite of this fact the method statement is going to be introduced to the companies' operative procedure because of EN-29001 standard and the directive 92/50/CE and, in Italy, in relation to the L.109/94 on public bids.

Design Process Scheme

The first step of the experiment proposal is the analysis of the building-design working processes adopted in the company and the assessment of the software engineering current status.
Main Authors (Cornick, 1991) represent the Building-Design as a process divided in four phases:

1- the briefing phase
2- the scheme-design phase
3- the detail-design phase
4- the specifying phase

The design process model can be described as an analysis-synthesis-evaluation recurring process (characterised by feed-back loops), in which the outputs of one phase may affect the inputs of the next phase as well as its own phase.
Each phase contains a set of typical tasks (operable by people and elements such as methods, environment, equipment, etc.). A set of typical tasks for the first phase is:

1.1- Building requirements list
1.2- Location problems and site study
1.3- Cost and time definition

Generally the CAD application in the company supports only some of these types of tasks -i.e. there is not a direct file import procedure from the GIS (Geographical Information System) environment where territory and environmental spatial analysis should be conducted in most cases-.
The second phase is generally developed in five steps:

2.1- Alternative solutions
2.2- Location draws
2.3- Space use and functions
2.4- Dimensional final definition of the building
2.4- Space materials study
2.5- Validated scheme design

In this phase a company uses the software support for each task, but, for example, there is not a dynamic link (by an automated design test procedure) between the technical specification of the building requirements (see 1.1) and the validated scheme design (2.5) which represents the output of the 2nd phase.
The third phase (detail phase) consists of the following tasks:

3.1- Structure study
3.2- Envelope study
3.3- Alternative solutions
3.4- Services study
3.5- Fittings study
3.6- Technologies study
3.7- Detail drawing set

Generally small companies find out very difficult and time consuming to gather, from the different information sources (company, client, other consultants, suppliers, contractors), all the information necessary to the company consultants and quality responsible to apply the quality tests to the building project.
This one is therefore the phase where the absence of continuity in the software processing is more evident: architects and engineers often are working in separate layers (and not by networked work-stations) without the possibility to cross the design control in a dynamic way. I.e. the task 3.2 it is generally developed before 3.1 for aesthetic needs, and many times the

structure such as the technological hypothesis must be modified a posteriori. We can observe the same regarding others technical studies and related building quality design tests.
Detail draws are lastly a collection of data, often conducted in a very traditional way and aimed to implement different sub-sets of separate technical layers. These first three phases of the building design are recurring processes.
The specifying phase is integrated with the detail designing phase even it has its specific tasks and output. During this phase the output of phase 3 has to be translated in the subsequent tendering and construction phases. Small companies generally adopt seven tasks:

4.1- Constructable draws set
4.2- Contractual requirements
4.3- Temporary works requirements (1)
4.4- Technical specifications (2)
4.5- Design test methods (2)
4.6- On site and laboratory test methods (2)
4.7- Estimate of costs

Many companies manage the constructable drawing and the estimate of costs by software, but, in the future it would be necessary to link all the other Quality Management tasks to them.
In particular one of the main objectives of the application experiment is to define a dynamic procedure to obtain validated draws and instruction for construction in relation to a pre-defined project requirements.
To this aim the integration of EDMS standard functions in the experiment purpose is foreseen.

Generally the status of software engineering practise in small companies may be described through the following elements:

<u>-Organizational issues:</u>
There is a designated Project Manager (SPM) for the software development project who naturally reports to a Business Project Manager (BPM) more for the Companies small dimension than because of formal procedure.
A Software Quality Assurance (SQA) function for the software development process is actually missing.
The availability of non-software resources (i.e. data from different sources, for specialists and consultants information for the project implementation, etc.) which are critical to the success of the project are managed with an appropriate planning phase but are actually, more oriented towards the product functionality than to the SQA.

<u>-Standards and procedures:</u>
There isn't any formal procedure of structured software, implementation methods, only the data analysis phase is well defined: ERA (Entities Relationship Analysis) methodologies are in fact common practise.
Furthermore being the majority of programming activities regulated by subcontracting agreements the major stages of the software development process are periodically reviewed and monitored.
The customization of the commercial package is often preferred because of the small investment.

<u>-Metrics:</u>
Metrics to evaluate the activities efficiency and to reach higher levels of optimization of the software production process are generally not used.

---

1 according with the standard 92/57/CEE
2 new procedures in conformity with the EN-29001 standard.

-Process Control:
It is often quite good but not formalized because of companies' size and is mainly due to the sub-contractors control necessity.

-Tools and Technology:
Third generation languages plus macro languages associated with CAD environments are more often used. Traditional 3rd Generation Languages debuggers are still widely circulating. Innovative packages for work group and EDMS environments and CASE (Computer Aided Software Engineering) tools are very rare known and adopted.

In conclusion, we can observe that at the moment a real software life cycle is not defined within the small AE companies organizations and the actual trend is to limit the IT support to traditional software areas such as CAD.
The paper purpose is to join the necessity of a quality assurance system with the re-engineering of work-processes supported by a more effective IT application.

Required Status of the organization and software engineering practise upon completion of the experiment

The conceptual model assumed for the application experiment is illustrated in Fig.1.
The model shows an effective quality management support according to the standard EN-29001 in building-design services- i.e. in the detail phase the working process is represented as a chain of sub-phases characterised by feed-back loops from the subsequent task to the preceding task. Computer controlled feed-back loops are needed for maintaining the consistency in meeting project requirements.
The IT system (and for instance the information generated) are elements of the task set. As explained in the QM literature (Cornick, 1991) any deficiency of these elements can have two main causes:

1)a lack of skills or knowledge
2)a lack of appropriate procedure as a result of non-quality related professional practices.

The experiment proposal is aimed to avoid these deficiencies providing both work as well as continuous environment control for the project.
A great emphasis must be focused on the system documentation capacity (see 4.16 of EN-29001 standard) to release any project decisions to the relative input-outputs data/information during quality audits or final tests.
The general model of the experiment is described in the Fig.3 within the phased workplan.
The Fig.2 shows the structure of the system that should be adopted for the application experiment.
The software engineering activity will be focused on developing an automated link between the related building-project steps. Quality technical and standard information within the related building-project steps. Quality performance evaluations, related design tests and scientific know-how are necessary to the building quality management and should be integrated in the computer aided design procedure, defining a set of rules (declarative data bases and procedural files) managed by automated work flow tools.
The experiment is aimed to work with Information Technology industry standards. Rather than build a system upon a set of proprietary tools (such as a proprietary graphic system or a proprietary data base management system) the proposal plays, in an integrated way, with today's industry standard CAD and DBMS software. The graphic front-end would use a very widely circulated graphics environment; as well as the DB back-end link which should be designed to work with an integrated SQL database.
The main changes required in applying the experiment are shown in Fig.3 and are explained below.

-Organizational issues:

Software Quality Assurance (SQA) function for the software development process inside the Application Experiment introduction.
-The software manager training on in-house software project procedures should be planned. Particularly the training is aimed to improve the awareness and capacity on companies' QA system.

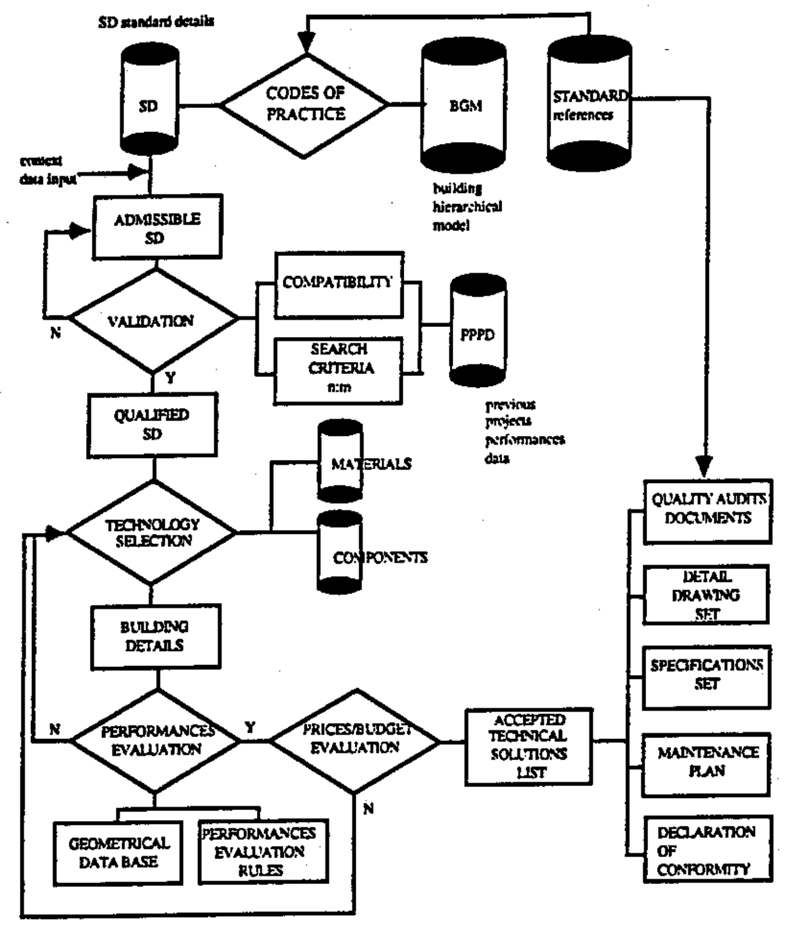


FIG.1- Example of a procedure in building-design details phase

In the application experiment the availability of non-software resources should be guaranteed by the definition of specific and formal procedures.

-Standards and procedures:
Companies must adopt a formal procedure of structured software and implementation methods.

-Metrics:
Companies must adopt metrics to evaluate the efficiency of software customization activity.

-Process Control:
The process control must be formalized to conform the SQA requirements and to reach a higher level of optimization of the production process.

-Tools and Technology:
The introduction of a relational database environment, to create the design-database should be added with 4GL and CASE inside the experiment.
The prototyping methods introduction is considered very useful in supporting the final software interface definition considering the Building-Design software complexity.

The software formal documentation introduced through the defined quality level will offer the possibility to plan the maintenance at the stated level of quality and manage the entire software life-cycle even if limited to package software customization.
Standards and procedures formal definition will support this process while avoiding the excess of "paper production" in accordance with the Companies size and organization, and would introduce electronic workflow practice.

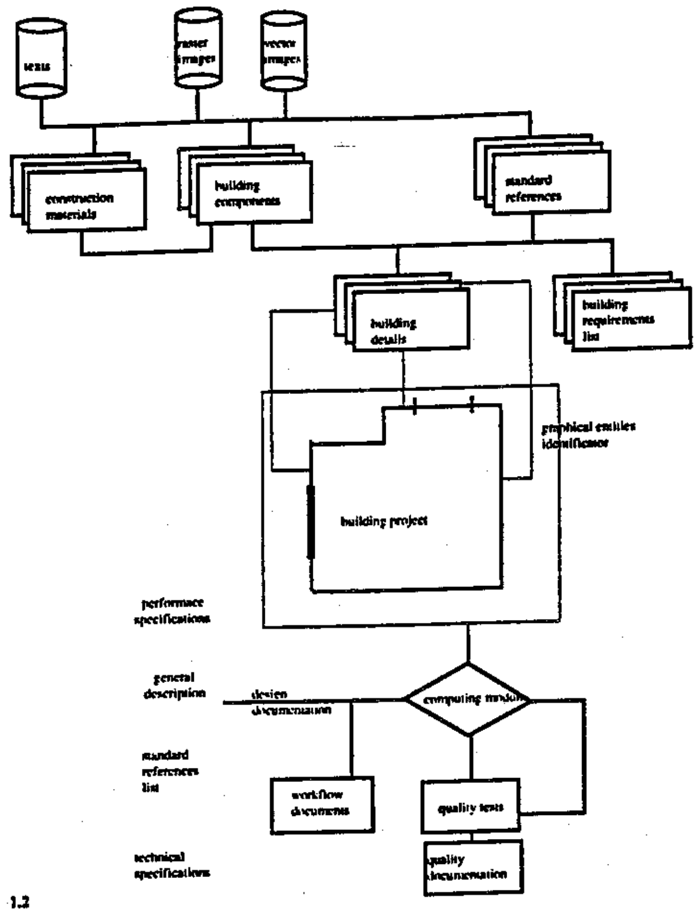
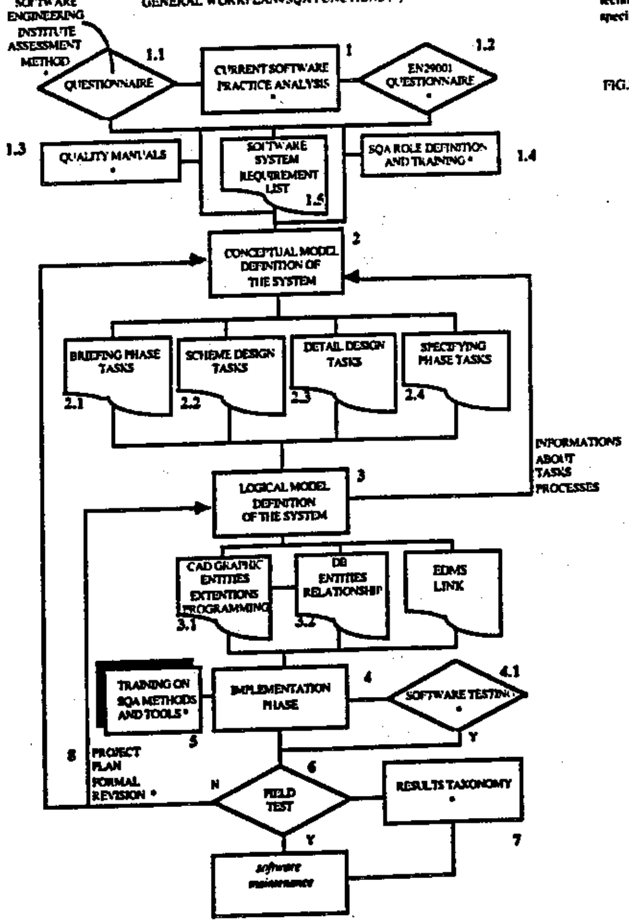


FIG.2-example of the experiment contents

Fig.3-EXPERIMENT
GENERAL WORKPLAN+SQA FUNCTIONS (*)



Anticipated benefits

After the experiment's application the software engineering in small companies should support more efficiently the Quality Management procedure of the building-design process.
This fact will create the possibility to certify the software and help the whole company Quality System implementation in conformity to the standard EN-29001.

In particular an entire set of planned and systematic actions are necessary to offer an adequate quality assurance of the software and of the building-design services organization in a company and should be formally managed and documented by the system.
In fact most of the corporate Quality

System documentation necessary for building-Design should be an output of the software system itself.
In such way a company should be able to demonstrate, where requested, the quality documentation as a software performance, and with the IT support the "paperless office" would rapidly reduce the generation of the paper mountain typical of a bureaucratic approach to quality systems. The application of the QS would become a real tool in cutting time and costs to support efficiency.

References

CORNICK Tim, *Quality Management for Building Design*, Butterworth Architecture Management Guides, Butterworth-Heinemann, Guildford, 1991, 218 pp, ISBN O-7506-1225-8.
UNI-EN 29001, 1988
OICE (Associazione delle Organizzazioni di Ingegneria e di Consulenza Tecnico-Economica), *Linee guida per l'applicazione della norma UNI-EN 29001 sui sistemi qualità ai servizi di ingegneria*, new edition, Rom 1993.
QUASER, *Come costituire un sistema qualità certificabile*, GLCT-1, Milano, Marzo 1994.
CE, *ESSI (European System and Software Initiative) Guide for Proposers*, DG XIII, Information Technologies and Industries, and Telecommunications, 1993.