What software for instruction in architecture: today features and needs for the future.

Andrea Scianna[1]

Abstract.

In the last years the computer technology has evolved very fast; new tendencies were also exploded in computer programming, such as object oriented programming, but the world of Computer Science still records other news as the introduction in the market of new graphic environments or operating systems such as Windows 95.

As results of this evolution all the applications currently used in CAAD teaching and for practice, were heavily updated; so they have acquired new power for the immediate use but some other features, interesting for CAAD instruction, were lost.

Today, the creation inside some CAD programs of those little applications that are one of the highest moment of "learning by doing" method, starts to require more knowledge of some computer science techniques and of the operating systems too.

Some questions arise from the drawn situation:

- *are the features of various today packages used in computer aided architecture instruction well suited for this purposes ?*
- *doesn't requires today's CAAD software technology a strong job both for teachers and students in the brief time available for computer use learning in the schools of architecture?*
- *what are the features of the various kinds of software, used for CAAD instruction, the teaching community must require to software producers?*

The article, making a survey on different kinds of packages used for CAAD instruction, also on the basis of personal experiences of the author at the Faculty of Engineering in the University of Palermo, examines the features they are provided and the needs they might be provided.

1. Introduction

Computer science evolves very fast both for hardware and for software technology. This evolution could make available more powerful and easy to use instruments, but not always it happens; often first implementation of new hardware or software technologies are difficult to use so users reject them and producers cut off them from their products or not improve them any longer. This fact is also due to a lack in the culture of users that don't immediately see the utility coming from a technological innovation that also could be advantageous. And rejection of technological changes is more frequently as the change concerns a level apparently far from tasks carried on by people (teachers, students, professionals) but connected with it. People often don't understand the implication of a technology with their job.

Education could have an important role to solve this problem: future designers should be accustomed to accept technological changes in a dynamic way avoiding pre-established stiff mental schemes and getting them ready to accept technological changes .

[1] I.E.R.En. CNR - via Cardinale Rampolla, 8/d - 90142 Palermo, Italy ; e-mail: scian@mbox.unipa.it

Computer sciences technologies are employed at three different levels:

- hardware level;
- base software level;
- application software level.

Some operations necessary to perform some tasks for a specific job, such as CAAD, are possible only if base technology exists. So if we want to carry on something that our application software can't execute, we must see if the operating system (a level down in respect with application software) permit it us. This is a good reason for which we should interact in computer technology development (as users, teachers or researchers) at different levels asking to producers instruments well suited for our activity. But what are instruments used both in CAAD education and in professional practice ?

In the first part of this paper, attempting to point out some necessary requirements for programs to be used for CAAD, design stages of a project are considered and for each of them are listed different kind of software and the possible type of use. Successively, for each kind of software are highlighted yesterday's and today's main features and also the foreseeable evolution.

If need may be different according to the kind of software, also on the base of my previous experience I think that a thing must bear in mind: generally people tend to use in practice, for a long time, the same software used at school or at the university. I noted also a certain fear when it is necessary to change the software, even if used only for a short time, with a new one more powerful. Since programs are first used to learn at school and then for practice, I think that in the development of an applications both practical and educational needs must be taken in account by software firms and this is not always evident.

So in another section are listed some general features I think should be present in programs used for CAAD both in education and in practice too.

Then at the end of this paper some considerations are made about teachers involved with CAAD, and their interaction with computer science evolution.

2. Programs for building design stages

In this section, for each design stage, are analysed computer programs should be used within.

1.1 Briefing stage

In this stage, once the site is chosen, are performed all those actions aimed to:

- user's needs definition;
- analyse the feasibility of the project;
- characterise both environmental and technological system needs;
- define technological system features;
- define cost parameters of the project to realise and to verify compatibility of possible technical solution with environmental and user requirements.

In this stages are very useful word-processors and spreadsheets; laws and regulations collections organised as hypertext are very useful in this phase.

Site features should also be known accessing to a GIS systems installed in public land registry offices also through Internet.

Stage	a) Briefing	b) Schematic Design	c) Detail Design
Program types			
1) Word Processor	reports on general features	reports on design features	reports on building or components features, specifications, construction activities
2) Spreadsheet	perfunctory costs estimate	rough project cost estimate	generation and pricing of bill of quantities costs estimate
3) Hypertext		looking technical information in laws or regulations collections on CDs	
4) Drafting		layout parametric sketches	
5) Cad		general dimensions and layout definition	drawings of buildings parts and components
6) Gis	site information assumption		
7) Quantity Material and Costs Estimate			generation and pricing of bill of quantities costs estimate
8) Thermal Analysis		calculation to establish the best shape in respect with orientation and site environmental features	final thermal behaviour evaluation plants design
9) Acoustic Analysis		checking of design features in comparison with site features rooms acoustical behaviour evaluation	acoustic verification of building components final rooms acoustical response evaluation
10) Visual Analysis		checking of lighting features also in comparison with site features rooms illuminance evaluation	final daylight and glare index evaluation rooms light sources definition
11) Structural Analysis			structure elements verification

Table A- Use of programs of the CAAD environment in the different project stages

In this stage are usually drawn up text documents. Spreadsheet tables should be easily embedded into project's documents and automatically updated once a data is changed on the table.

1.1 Schematic design stage

In this stage are defined:

- the building environmental system;
- all the buildings parts are also defined giving them dimensions and exactly establishing relations between them;
- all construction systems;
- schematic design of technological system;
- rough project costs.

In this stage, compatibility between technological features and environmental (indoor and outdoor) requirements (lighting, acoustical and thermal) must be checked.

In this phase are produced all the documents necessary to obtain approvals and loans too.

To draw up schematic design are particularly useful word processors, spreadsheets and cad programs; the last two ones aren't usually used in a productive way with full use of data exchange mechanism with other programs of CAAD environment.

It should be extremely useful to have programs for parametric design in which any label or parameter is linked by-directionally with a cell of spreadsheet table. This feature is absolutely necessary to define dimensions of the building and his parts taking in account limitations or constraints concerning volume, costs, room areas etc. 3D CAD programs with rendering functions let the designer to best refine the envelope shape and virtual reality instruments with multimedia features should let him to live sensations as he were living in the building.

1.2 Detail design stage

In this stage the designer exactly defines geometrical and morphological features of rooms and of building parts and components (the envelope, the structure, plants, equipment). Are also performed calculations on components behaviour; this calculations require programs based on accurate algorithms; automatic generations of details must be performed by every program concerning building element design (walls, plants, structure).

In this stage details, bills of quantities, work prices and detailed cost analysis should be automatically generated from drawings of schematic design stage and material databases.

Final two and three dimensional representation should be realised; accurate thermal, acoustic and lighting analysis are performed and reports should be automatically drawn up.

3. Software features for instruction and professional practice

In the previous chapter was proposed a list of software used in the different stages of design development. In this chapter, for each of previous mentioned program I will try to briefly trace the evolution of programs used for CAAD: both yesterday and today features will be listed together with foreseeable evolution and future needs.

1.1 Word-processors

W*ord processors* should be used in all stages of the project for writing out all the documents necessary to show its features (stages form ***a)*** to ***c)*** of table 1).

About general features of this programs it is possible to say that:

- in the *past*: it was not possible to insert any image (created with drafting programs) or table (coming from spreadsheet programs) that were inserted manually in the document after it was printed separately from them;
- *today*: it is possible to insert drawings and tables in the document through embedding functions of today's programs. Embedding functions must be present at operating system level so word processors and other programs must activate them. In most word processors is present a built-in language useful for example to draw and manage forms to be used recursively.
- in the *future*: object linking and embedding functions of today operating systems will be enhanced to have more power in the document editing.

1.2 Spreadsheets

Spreadsheets should be useful to perform different types of calculation: the designer should use little functions generally realised by itself for choosing the right design solution, taking in account site feature, orientation of the building, envelope shape et cetera. By this programs costs evaluation at the different stages of the project can also be executed. But today this programs are generally not used by most people as a design instruments but as a data presentation instruments, performing very poor calculations with them.

In my experience I observed that integrated packages including word-processor, spreadsheet and database modules are not well suited for CAAD because they are poor of functions and do not contain any macro language that it is necessary to write user functions and to exchange data with other programs. Students or professional users require advanced data functions and macro language when the degree of confidence with instruments growths and they learn to use spreadsheet as a real design instrument.

About general features of this programs we can say:

- in the *past*: it was not possible to insert data tables into any document and was not possible to establish any kind of dynamic link to data of other programs such as CAD programs;
- *today*: it is possible to insert tables directly into documents so when data are changed in the spreadsheet program, the data in the document can be automatically updated; spreadsheets contain built-in object-oriented language usually basic-like.
- in the *future*: data embedding will probably be enhanced together with capability to write programs with built in languages to best manage data.

Data exchange functions with drafting or cad programs should be improved.

1.3 Drafting programs

Drafting programs (generally for 2D drawing) should be useful, in the schematic design stage to look for the optimal design solution, taking in account constraints imposed by site nature, building orientation, envelope shape, regulations et cetera. By using both these programs and spreadsheets programs too, making use of data linking feature between the two types of programs, all cost calculations at different level of definition of the project could also be performed. Today's drafting programs are normally used for drawing but data exchange functions with spreadsheet, when this feature exists, are not generally used by low or medium level users.

About general features of this programs we can say:

- in the *past*: drawings or sketches were not inserted through a software mechanism in any document and no data were transmitted dynamically to spreadsheet programs to evaluate performances and or design costs;
- *today*: it is possible to insert drawings directly into documents so when it is required, drawings can be edited by clicking them in the document; dynamic data exchange feature with other programs is present only in few programs;
- in the *future*: data embedding will probably be enhanced together with capability to write programs with built-in languages to best manage data.

1.4 CAD programs

CAD programs, in addition to drafting functions, also contain functions for solid modelling, dimension driven design, database linking, high quality rendering. Nevertheless CAD programs are generally used, by more people, as drafting programs; actually advanced functions for dimension driven design and database linking are rarely used. Probably what happen is due to two main reasons:

- the use of these functions is today not always easy;
- people aren't yet get ready to use CAD programs as design instruments.

So the advanced or full use of CAD programs is actually possible only in those architectural firms where are people experienced and specifically dealing with computer applications; these are usually big firms.

About general features of this programs we can say:

- in the *past*: it was not possible to generate dimension driven drawings or rendering and animation;
- *today*: it is possible to generate rendering and walk-through animation; in general purpose CAD programs, dimension driven functions well suited for architectural design are very limited; normally it is possible only 2D dimension driven design; functions for dynamic data exchange with spreadsheet and database programs aren't easy to activate; only few users, as for drafting programs, make advanced use of CAD programs; besides very few are users that also write small applications in any languages, built into CAD programs,
- in the *future*: data embedding will probably be enhanced together with capability to write applications by built-in languages to best manage data; it is foreseeable that functions for dynamic data exchange, very useful for dimension driven design and parameters based cost evaluation will not be enhanced and yet these functions should be improved. Another kind of functions I think will be enhanced are those ones for parametric design and dimension driven design. Today the most advanced programs for parametric design are used for mechanical design or industrial design. By this programs, elements in the design are related to one another; relationships are defined giving sufficient information about size, location, extent and/or shape of the elements in the model so this last can be recalculated to fit new constraints. All this must be performed with variational geometry or procedural-relational method. Today's CAAD applications are based on an oldest method; different modules exist into the CAD program each permitting generation of building components; but once a components is drawn its geometry cannot be interactively recalculated modifying a parameter while CAD programs based on dimension driven or variational geometry permit all this.

1.5 Hypertext and multimedia software

Professionals generally make use of hypertext applications to easy consult collections of laws or regulations for the various field of interest. Generally they do not need to built an hypertext so do not need hypertext authoring software.

Multimedia presentation, in practice, are created to show a project to contractors or clients; this job also involves the use of CAD and/or advanced rendering programs. In education multimedia applications are very useful to show to the student the relations and the weight of visual component and acoustical component in the project, giving the student a way to live same sensations as those ones lived if in presence of the real building (once realised).

In education hypertext and multimedia software are very useful for Computer Aided Instruction.

- in the *past*: multimedia programs didn't exist; a project was presented only by hand made drawings and sketches; some CAD programs contained functions to realise movies, but it was impossible to link them sound, rapidly as today, with non professional instruments;
- *today*: multimedia software are powerful and also hardware advances in Pcs' technology make possible to realise multimedia presentations with low cost instruments. Some CAD or advanced rendering programs contain functions to export movies in standard file formats. So movies can be inserted easily in a presentation, or it is possible to mix them;
- in the *future*: the capability to run multimedia presentation on every machine not depending from the operating system or processor type will be possible. Today this it is already possible with some Internet browsers; this feature will probably improved.

1.6 Virtual web applications

In the last year many associations, research institutes and universities are publishing, on the Web, their software written in Java language; this fact permit to people to launch applications on remote servers and to perform calculations interesting for their job, so knowledge developed by the subjects that published the programs can be shared by all.

Generally limited to professional living in the same nation, through the virtual web library it is possible to consult technical information, regulations or laws.

Collaborative computing based on very fast network as ISDN, require today a great effort from base software producers to improve data sharing mechanisms. Base software applications for collaborative design are today very few and at an experimental stage.

1.7 Daylight, thermal and acoustic behaviour evaluation programs

These programs are very useful both in schematic design phase and in detail design phase. But it is necessary, in the two different stages (schematic and detail design), to use different kind of programs.

In the schematic design stage simple calculations must be performed on general characters of buildings, or on theirs' parts. For example could be necessary to establish the right span of a balcony to best protect some parts of facade taking in account illuminance conditions of rooms facing to it.

Programs to use during schematic design stage should be very user friendly. They should require a reduced number of data for contour conditions definition and for the object to design. They

should be provided of good interactive computing features to permit a cyclical refinement process of a design solution.

In the schematic design phase it is necessary to check the acoustical, thermal and visual design features with site features and general regulations constraints. This type of calculations can also be performed using home-made programs written in a simple language like Basic or in the built-in language of spreadsheet programs.

In the final phase of schematic design stage and in detail design stage are performed more accurate calculations about behaviour of building components in respect with environmental and inhabitants constraints. This programs also permit the automatic generation of details, bill of materials and design reports. Usually these are commercial programs since the creation of such complex programs might require a lot of work from people of an architectural firm.

1.8 Visualisation and other rendering programs.

In the last years we have seen the integration of rendering complex algorithms into CAD programs.

For the future, is foreseeable the application to CAD programs of algorithms and procedures for generation of complex solid models (typical of computer graphics applications for cartoons and tv programs). Pcs' CAD software with very powerful solid modelling functions are today very few and the generations of 3D complex solid models not always is simple.

Today's rendering programs and especially those ones in the future should contain VRML modules (VRML plug-ins) permitting them to open 3D scenes realised other diffused programs in the market and mix scenes with object drawn with themselves.

4. Software features for CAAD teaching

Speaking about the features of each program, used in CAAD teaching or in practice could require many pages and a lot of time. But there are many feature the are common to all kind of software and I think must be present in each good software on the market, particularly in programs of the CAAD environment.

These features are:

- *user friendly interface*; all functions should be activated with icon buttons; it could be better if icon were standardised; all menu labels relating to general commands (such as open file, close, import, export, cut, paste etc.) should be always in the same position for all kind of software and for programs of same kind; all this avoid the user feel lost when he try to word with unknown software;
- *cross platform compatibility*; this feature can guarantee continuity in the use of software as at school as in professional practice when the designer decides itself what kind of computer to purchase;
- *macro language presence*; macro language must always be present as it is very important to adapt software instruments to particular needs of the user, enhancing productivity in repetitive tasks; but macro language must be very simple: to write macros it should not necessary the user know the interaction with operating system; for example the user should define and successively call input boxes without know anything about API of the operating

system; complex languages as C or C++ must be reserved to professional programmers as they require much computer science knowledge (for example C programming requires a lot of care in variable definition or in pointers' use, and a lot of work to port application across platforms). I think both Autolisp (that is poor of functions) and Visual basic-like languages are well suited for instruction or every day macros writing in practice (unfortunately Autodesk cut out Visual basic support from the latest version of AutoCad).

- *command line presence*; command line is necessary because interaction through command line get the user accustomed to familiarise with command names and sequences he can successively put into macros;
- *command sequence recording* permits the user to write applications in a very easy way; it is possible, in a little time, to transform a command sequence in a macro, once sequence itself is written on the disk only changing input data values with variables' names;
- *full DDE support*; data dynamic exchange support is very important in those operation requiring links between objects of different programs (tables' cells, graphic entities and data field); regarding the link between CAD objects and spreadsheets' cells it should be possible to set links not only between properties of graphical object and spreadsheets' cells but also between data structures (set of attributes of blocks, list or arrays) and spreadsheets' cells;
- *full OLE support*; this feature is the only that software houses really improve since it is important in all office automation tasks; it is very important also for workgroup and collaborative design.
- *contextual help and suggestion box presence*; it is necessary to guide users especially when they learn to use instruments for the first time trying software commands;
- *extended network support*; this function is very important for workgroup activities and should be extended also for collaborative design;

5. Teachers and teaching

From the overview drawn in previous sections as concern the great number of instruments and the complexity and dimension of knowledge necessary to use them, it get out how difficult is today CAAD teaching. And, if at the CAD start times to know the only software in the market was enough, today it is no longer possible. If today a teacher want to be a reference point for students, he should know features of a great number of programs and all possible techniques of use them in the design process.

Obviously teachers should choose the most powerful programs for CAAD teaching, verifying the presence of those functions useful both for education and for professional practice. Programs should permit to student or professional to learn it easy during their job (learning by doing), also by a good help or step by step guide. And yet the teacher must show to the students all possible ways to use software instruments, trying to generalise concepts so they can easily solve problems relevant to the use of whatever program purchased by the firm, where they will carry on their job after graduate or post-graduate.

Teaching techniques and courses organisation must be periodically revised, more frequently as in the past, to get them more adapt to the evolution of computer science.

Since today, a great number of kind of computer programs exists, it is impossible to cover them within a single CAAD course. Perhaps CAAD teaching, in addition to a start course, should be distributed in the different years of curricula and perhaps in every design course. And also if

every course of the design curricula should cover a stage of project process, in each of them could be taught CAAD techniques relating to the specific design stage. So teaching in every course should be carried on by a team of teachers.

Also according to concepts I expressed before, teacher must be mean for transmission of technological culture not only constituted by set of procedures and instruments, but of dynamic learning process, as of generation both of new interaction's contexts with instruments and of new languages.

They can get the students accustomed to carry on their job by new brain-strategies, flexibility, creativeness, self-learning ability during everyday job; these all are things absolutely necessary to be in step with technological evolution.

6. Conclusions: what to do for the future?

When we talk about the relationships between hardware or software instrument and the user, we arrive at the conclusion that our actions or way to carry on a task cannot be influenced by instruments; we should operate in a more general way as possible. But really the opposite things happens; we can carry on our task if we have the right instrument and if it is easy to use too; this also happen because we are accustomed in a particular way. For example if we have a software instrument that permit us to work in 3D we try to use it in 2D and perhaps it seems us too difficult to use; so as we are not accustomed to think in 2D or we think 3D and operate in 2D and then we collect 2D data to show in 3D what we thought. We are accustomed to see plan, facades and sections of a building (2D representations) but really building is a 3D object with it's faces (the facades and the roof).

Then I think it is necessary, in a moment in which computer science evolution is proposing very revolutionary and powerful instruments to remain in step with such evolution trying to influence this process as actors.

In the CAAD field are arriving new loads, in terms of instruments yesterday used for tv movies or films, graphical publishing or mechanics.

As teachers or researchers we must filter features of these instruments to sustain the development of those ideas that are really useful for CAAD. Generally happen that a teacher, when an instruments appear on the market, study it to discover what it is possible to do with and perhaps if it is possible to do a thing he thought some years before; but also when he discover what he thought is possible a new version of programs is on the market, not containing the interesting function: for example in the latest versions of Microsoft Windows DDE mechanism, that make possible to establish the connection of data of two different programs (between properties of graphical objects and alphanumeric data), was not improved and perhaps in the next versions it will cut out. This kind of mechanism is very useful evaluate building costs on the basis of general dimensions or general thermal behaviour through the envelope data. Improving DDE mechanism should be possible to link a set of data (structured as a frame) of a table with graphical objects.

Then taking part to the evolution process it is absolutely necessary contributing if possible to creation of instruments containing few really useful functions.

Some proposals could be the institution of groups of interest for establishing feature of CAAD instruments or the activation with the help of software firms of "usability labs" inside

universities to compare existing instruments with real designers needs and to propose very useful features.

7. References

Bragstad Jerry, ”I’m a Drafter, Not a Programmer”, CAD++ Newsletter, January 1994.

Glennie W. L., ”*Europe ‘94: A Visitor’s Report on the state of CAAD in Education*”, in ”THE VIRTUAL STUDIO”, Proceedings of the 12th European Conference on Education in Computer Aided Architectural Design, T.W. Maver and J. Petric Editors, ECAADE, University of Strathclyde, Glasgow 1994.

Folini Franco, ”*Progettazione e sistemi Cad parametrici*”, Pixel n. 5, 1993.

Franklin Carl, *”On VB4, OLE, and the Future”*, Internet WWW article at http://www.apexsc.com/vb/lib1.html.

Joseph C., Wood D.,”*Back to drawing board*”, Personal Computer World, April 1994.

Mitchell W.J., ”COMPUTER-AIDED ARCHITECTURAL DESIGN”, Van Nostrand Reinhold, 1977.

Petric J., Maver T., *”A classification of multimedia applications in architecture”*, in the Multimedia and Architectural Disciplines, Proceedings of the 13th European Conference on Education in Computer Aided Architectural Design in Europe, B. Colajanni and G. Pellitteri Editors, Dipartimento di Progetto e Costruzione Edilizia, Università di Palermo, Palermo 1995.