

INTERACTIVE TOOL AIDING TO OPTIMISE THE BUILDING ENVELOPE DURING THE SKETCH DESIGN

Jean-Marie HAUGLUSTAINE and Sleiman AZAR
 LEMA – University of Liège
 Chemin des Chevreuils 1, Bât. B52/3
 B – 4000 LIÈGE 1 (Belgium)
 Phone: +32 4 366 94 83 – Fax: +32 4 366 95 62
 E-mail: jmhauglustaine@ulg.ac.be

ABSTRACT

The PC software AMCE proposes building actors a Multiple Criteria Decision Aid procedure (MCDA) to optimise the building envelope about cost and energy performances, during the sketch design.

Two main modules are linked, where the user interactively goes:

- ◆ a first one manages parameters describing the project requirements;
- ◆ a graphic pen-based module allows to draw the sketch; it calculates geometric parameter values and put them back in the first module.

Permanently informed on foreseeable performances, the user can any time search for optimal scenario giving best satisfaction to one actor or most preferred compromise to the set of actors.

INTRODUCTION

Using MCDA methodologies, the approach is an attempt to cope with the difficult problem of improving the dialog between several actors involved in the design of a building envelope. The MCDA framework can help *the sketch design*, the earliest and crucial step, in the whole building envelope design. The main purpose of this paper is to present the MCDA framework which can help the sketch design, the earliest step, besides being a crucial one, in the whole building envelope design.

STAGES OF THE SKETCH DESIGN

We can consider three stages that lead actors to the sketch design (see Figure 1). The first is a definition and negotiation stage between several private and public actors, where they express their requirements and preferences, based on several criteria, parameters, regulations and constraints. In the second stage, the actors build several scenarios by defining parameters and assessing their values, with some imprecision and tolerance. The architect or some expert system then computes performances on the selected criteria. This second stage is called the feasibility stage, because it results in two questions:

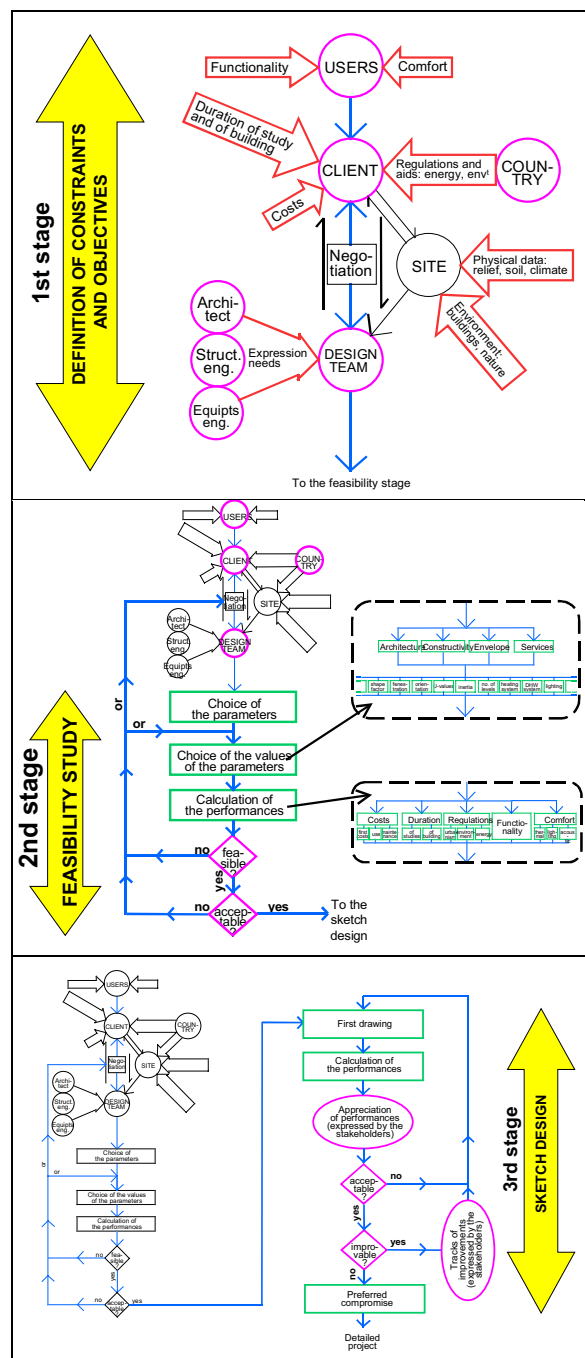


Figure 1: The preliminary stages of the design process

Is the scenario feasible? Is the scenario acceptable to the stakeholders?

Loops are introduced in the process when negative responses are given to these questions, until entering the third stage, the actual sketch design, becomes possible. After a first drawing of the project, performance calculations can be refined. The stakeholders then judge these performances as acceptable or not, and improvable or not.

Loops occur in the case of non-acceptability or improbability:

- ◆ each actor is able to search for the optimal set of strategic parameters, which corresponding scenario gives the most satisfactory performances according to the criteria he considers as relevant;
- ◆ all the actors may try to find the preferred compromise, defined by the values of strategic parameters which scenario is giving those best performances that equally satisfy all the actors together.

MULTIPLE ACTORS FEATURE OF THE BUILDING DESIGN PROCESS

The design process of a building is typically interactive (Conan, 1989), requiring the participation and mandatory satisfaction of several actors, each facing multiple criteria decisions:

- ◆ The *client* initiates the whole operation; he may be the property developer, the owner, but also a future occupant or a future user... He is increasingly involved in the design process and in the search for design fitness to match his requirements. Furthermore, the client's legal responsibility continuously increases, as does his responsibility in the building act. He's well personally responsible for his building's energy performance, the environmental protection, etc.
Faced with these new duties, the client requests increased help from the design team. The project should allow the client to assume his new responsibilities, while reaching his objectives of cost, building period duration, comfort and functionality. The client is considered, however, as having no technical capacity, so the software should ask soft questions, using familiar vocabulary.
- ◆ The *project author* is a generic term that more often designates the architect, but also the consulting engineers (for structure, equipment...), the quantity surveyor, the project manager, etc. The project author's mission is so heavy, he often neglects matters he considers as non-essential, e.g. the energy aspects. Though a great deal of software is available to compute energy performance, it is, unfortunately, rarely used by architects, in practice. Software should help project authors in their task, while maintaining

their particular objectives, e.g., their expressed needs and their wish for increased notoriety.

- ◆ The *public authorities*, representing the country or the region, are authors of constraining lawful and normative standards (thermal, urbanistic or environmental regulations, etc.); some public policies are translated into constraining levels of energy performances.

Finally, we see that the goal of the MCDA process and the linked software is to help each actor express his particular needs, either in the form of constraints or multiple objectives as well as to build an interactive tool, capable of dialoguing with each actor in his own language, while providing useful performance indicators.

Every architectural project is progressively defined, necessitating frequent interrogations to the actors and making the corresponding adjustments. The software should thus be able to allow for these frequent data adjustments.

When adjustments essential to the project viability are discovered too late in the design process or during the construction itself, they lead to additional costs, very often out of proportion when compared to their actual importance.

A first feasibility check of the project is important, in order to ensure that at least one solution exists. The feasibility study enriches the negotiation and/or incites the actors to negotiate the objectives and the associated means again, when the solution set is empty. If a feasibility study is not mandatory, i.e., when the solution set is not empty, it may save time through a parametric definition of the project, which describes the relevant intervals of the main parameters and, especially, their robustness, i.e. performance stability to parameter variations.

While remaining very imprecise in the feasibility stage, project data may be sufficient to define a "scenario" – see (Roy & Bouyssou, 1993) and (Maystre & Bollinger, 1999) – that characterises the architectural choices and allows evaluating resulting performances.

AMCE SOFTWARE: AN INTERACTIVE TOOL

AMCE is an acronym for "Aid to the Multiple criteria Conception of the building Envelope" (in French: Aide Multicritère à la Conception de l'Enveloppe de bâtiment). The software is developed with the programming language Allegro© Common Lisp (Franz Inc.). It runs on a common PC, with the following minimal characteristics: a 120 MHz Pentium microprocessor, Windows 95, 98 or NT 4.0 operating system, 15 Mb hard disk capacity, 32 Mb RAM storage capacity (64 Mb recommended) and a SVGA graphic card with a palette of 64.000 colours.

The usual starting point in energy software is the drawing of the sketch design itself: see (Crawley et

al., 1998 and Rivard et al., 1995). This approach lacks a first parametric phase, which allows defining a set of feasible solutions, among them, the preferred choice.

In our proposed design procedure, each actor chooses – and manages – the values of the main parameters, in order to satisfy selected criteria, based on individual preferences and requirements.

AMCE has been built to facilitate an interactive introduction of technical and preference data by each actor, based on his individual needs and capacity, and to compute performance criteria.

Selected criteria: project performances

The software evaluates performances concerning energy aspects for heating, air-conditioning (HVAC), domestic hot water (DHW) and artificial lighting, as well as construction and use costs of the project:

- ◆ four regulation performances: K-level (as required in Belgium), Be heating energy needs (in Wallonia), G/Gref, B/Bref, C/Cref (in France);
- ◆ yearly energy consumption expressed in physical units (kWh), for heating and domestic hot water, air-conditioning and artificial lighting;
- ◆ associated energy consumption costs;
- ◆ project construction cost, VAT and fees excluded.

Actor's preferences

The architect, owner, tenant, property developer, etc., anyone intervening in the building process, can participate in its elaboration. Each actor defines a “scenario” summarising his desiderata and requirements. With the button “Actor’s preferences”, he specifies, for each criterion (Figures 2 and 3):

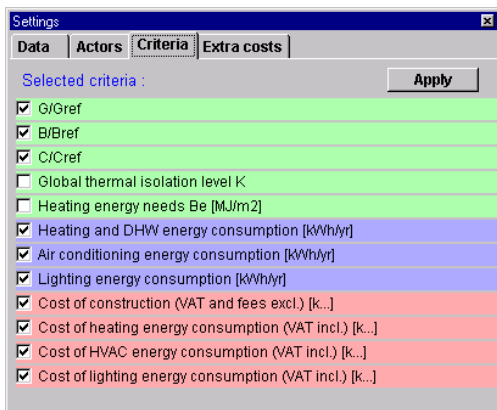


Figure 2: Actor’s preferences for criteria

- ◆ the maximum admissible critical value;
- ◆ an associated preference function, using the Promethee typology true (≡ usual) criterion, quasi-criterion, pseudo-criterion, etc. (Brans et al., 1986);
- ◆ weights he gives to the criteria: corresponding to the Promethee multicriteria approach (Schärlig, 1996).

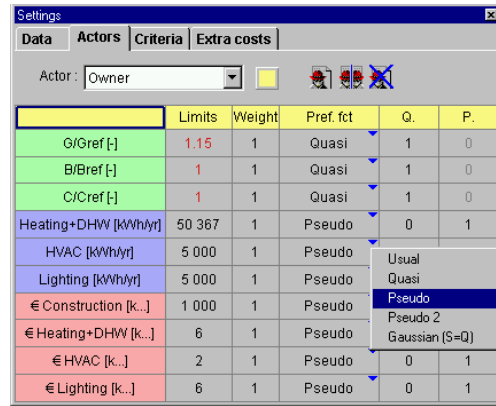


Figure 3: Choice of relevant performance criteria

A weight is like a coefficient of the importance the actor attaches to each criterion. Each user allocates any null or positive weight to each criterion: the criterion receiving the highest weight is the most important one, while the criterion with the smallest weight is the less important one.

The software so gives each criterion the relative weight it has in proportion with the sum of all the weights accorded to the criteria set. When an actor gives a null weight to a criterion, he expresses that he’s completely indifferent to this criterion he considers not interesting at all. Weight allocating is a personal subjective matter depending on the value scale each actor feels on performance criteria.

Actor's scenarios and questionnaires

An actor’s scenario is a set of values the actor gives to the parameters he can understand and control. The introduced parameters are just necessary and sufficient to the preliminary performance estimates. At this stage of the project, prior to any drawing of the draft, the geometrical parameters replace the geometrical variables, which will be defined later.

The collection of parameter preferred values characterising the project uses a questionnaire that is organised by items, i.e. thermal comfort, fenestration, solar aptitude, thermal characteristics of the envelope, equipment sophistication, etc.

All answers are previously filled in with default values, so that the user questioning is mostly facilitated: the user may modify them as he chooses. In this way, the parameters necessary for other evaluations (corresponding to criteria considered as irrelevant) are not to be introduced.

In order to fit the user’s technical capacity, the questionnaire is organised on two levels:

- ◆ a semantic level, fitting the client’s technical capacity; it helps the client specify his demands and requirements;
- ◆ a more technical level is proposed to the project author, who defines the preliminary characteristics of the project by answering no more than 60 questions.

For example, a screen is proposed to the client, asking him for the shape aspects he can define (upper part of Figure 4). The same question, but on a more technical level, is proposed to the project author (lower part of Figure 4).

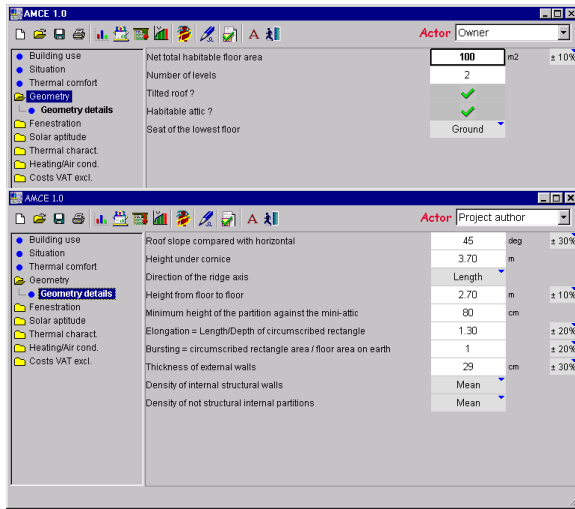


Figure 4: Questionnaire proposed to the client (Geometry) and to the project author (Geometry: details)

Of course many parameters (and associated questions) may depend on other ones. For example, when specifying that the roof is a flat roof, the user has not to answer questions about attics or roof edge: these questions are suppressed from the questionnaire and are not asked. But, if a positive – and not nil – angle is still chosen for the roof slope, these questions appear again and must be asked if default values are not in accordance with actor's wishes.

For strategic parameters, each actor specifies the accepted interval in which the parameter could vary. He so expresses his level of uncertainty and/or tolerance.

Five different levels of “tolerance” intervals are proposed to the actor, for his chosen value v : level 1: $v \pm 50\% v$; level 2: $v \pm 30\% v$; level 3: $v \pm 15\% v$; level 4: $v \pm 5\% v$; level 5: no tolerance.

The level 5 corresponds to a parameter that is prescribed by conditions outside the project and so the design team cannot change. Sometimes, one or both limits of his chosen interval is/are technically prohibited. In this case, the software introduces the largest technically feasible values within the preferred interval. This tolerance interval mechanism is crucial for further performance sensitivity analysis, and for the improvement step. If an actor enlarges intervals so that they are just limited by technical intervals, he may increase the set of admissible scenarios, e.g. his feasible decision space. Each time he reduces his intervals, his decision space is reduced accordingly as well as the degree of improvement he may hope for in the next stage.

For each parameter, the right click of the mouse provides on request the following help by a more

explicit definition of the parameter or values encountered in previous projects: the cultural approach of the project author is given here, consisting in a database of projects already executed by him, with associated parameter values.

Displaying and specifying scenario performances

A specific window (see Figure 5) lists the energy and cost performances obtained by all the scenarios defined by the actors (in bold: the actor currently using the software). Note that any actor can try more than one scenario.

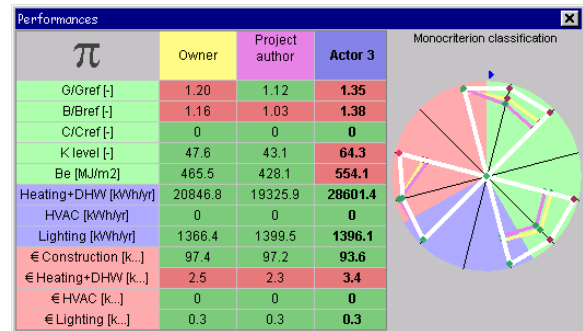


Figure 5: Performance window

In order to clearly display the satisfaction reached on the criteria specified by each actor, the background colour of each result cell is displayed: green when the performance is satisfactory for the actor and red, when it is not.

Figure 6 presents a Promethee total pre-order of the scenarios: the owner's scenario is globally the best one, while the Actor 3's one is less good and the project author's scenario is the worse.

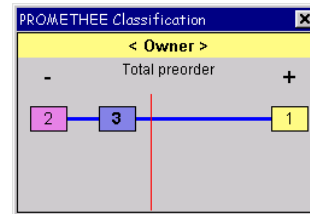


Figure 6: Promethee total pre-order of the scenarios

Sensitivity analysis

The software calculates the performance variation for the criteria considered by the actors, according to the parameter variations, thus guiding the project author towards an improved solution. The sensitivity of a chosen performance to the variation of one or two parameters can easily be obtained, so.

First, AMCE produces the σ performances for each chosen tolerance interval of the parameters and for each actor. For example, Figure 7 displays, for each criterion, the performance obtained with the interval of values (level 1: $v \pm 50\% v$) of the net total habitable floor area around its central value (100 m²). The construction cost is 38,2 k€ for 100 m²; it increases 43,4 % for a 50 % increase in area, reaching 54,7 k€, while a decrease of 50 m²

produces, *ceteris paribus*, a fall of 41 % for a new amount of 22,5 k€. On this exemplary display, we observe that some values are presented in a red cell: f.i., making variations of 50 % of floor area becomes a prohibited scenario for the criteria B/Bref and still remains unsatisfactory relating to yearly “Heating + DHW” energy cost, of course.

SENSITIVITY			
σ	Net total habitable floor area		
	50 [-50%]	100	150 [+50%]
G/Gref	1.0 [+2.7%]	1.0	1.0 [-1.8%]
B/Bref	1.1 [+3.2%]	1.0	1.0 [-2.5%]
K level	50.72 [+6.7%]	47.52	44.91 [-5.5%]
Be	494.38 [+22.0%]	405.38	372.60 [-8.1%]
Heating+DHW	9103.9 [-41.1%]	15454.9	21660.4 [+40.2%]
HVAC	0 [+0%]	0	0 [+0%]
€ Construction	22.5 [-41.0%]	38.2	54.7 [+43.4%]
€ Heating+DHW	1.2 [-37.4%]	1.9	2.6 [+36.5%]
€ HVAC	0 [+0%]	0	0 [+0%]

Figure 7: Window showing performance sensitivity to the variation of net total habitable floor area

Figure 8 (3-D graph) and Figure 9 (2-D graph) give the variation of the performance “Energy consumption (in kWh) for heating and domestic hot water”, to both continuous and technically acceptable variations of the net total habitable floor area and of the roof slope together; all the other parameters are remaining unchanged at their decided central values. When the user clicks anywhere in the 2-D graph (Figure 9), he can obtain the corresponding values of floor area and roof slope, with the performance obtained by this pair of values, all other scenario parameters remaining unchanged.

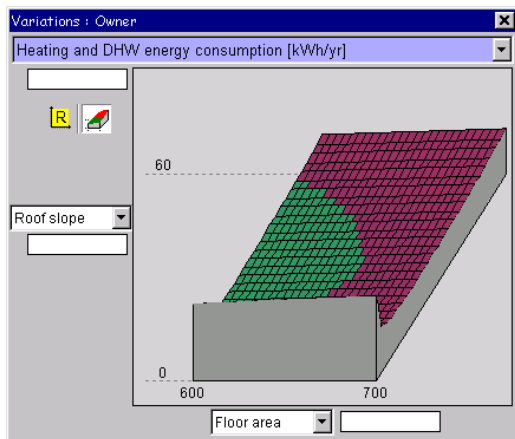


Figure 8: Some performance sensitivity to the variations of the floor area and the roof slope (3-D graph)

At the end of the feasibility study, still before any drawing has been made, two possibilities exist:

- Actors know the certain existence of at least one solution, as defined by the satisfactory parameter scenario. In this favourable case, they also know the sensitivity of the performances to the parameters. This information can be useful to arbitrate the divergent choices or possibly to improve design.

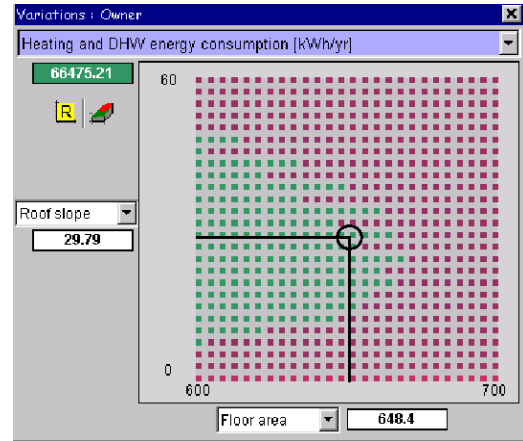


Figure 9: Some performance sensitivity (2-D graph)

- Or they are not able to find a scenario leading to acceptable performances. In this case, the sensitivity analyses largely document the negotiation that must follow evidence of an empty set of solutions. Negotiation then attempts to relax the most sensible constraints.

Therefore, this first stage goes beyond a simple feasibility study, which would only show whether or not a solution to the architectural problem exists. Defining a set of solutions, this parametric approach also locates a scenario space where the preferred feasible choice certainly exists. The combination of parameter intervals of validity/tolerance indeed defines a solution space, which respects the criteria considered as relevant. Definition of this solution space is essential for the strategy of future project improvement.

OPTIMISATION PROCEDURE

With an optimisation procedure using a genetic algorithm, the software generates a random population of individuals. Each individual is so a combination of possible values of the strategic parameters among the 8 billion possible combinations resulting from the authorised values of the 13 (up to now) strategic parameters within varying interval the actor has previously chosen. Each “individual” is a vector of 13 position representing the strategic parameters.

The optimisation process is organised as follows:

- A first population of individuals (which size is chosen by user) is randomly generated;
- The fitness f_x of each individual x is calculated:

$$f_x = \frac{1}{d}$$

where d is like a “weighted distance” to optimal performance in the 13 dimension space of solutions:

$$d = \sqrt{\sum_{i=1}^n \left(\frac{P_i}{C_i} \cdot W_i \right)^2}$$

with:

- P_i = performance obtained relating to criterion i (all performances are to be minimised);
- C_i = maximal authorised (or wished) value related to criterion i ;
- W_i = weight allocated to P_i ;
- n = number of criteria.
- ◆ Statistical evaluations: maximum, minimum and mean fitnesses are calculated; if the maximum fitness is larger than the absolute best – obtained up to now – individual’s fitness, the individual corresponding to the maximum fitness is saved as the absolute best one; otherwise the previous absolute best individual is kept;
- ◆ Stop conditions: the optimisation process is stopped if maximum number of generations is reached or if the absolute best fitness increase over several generations is too small, so that the convergence is considered as obtained;
- ◆ Population scaling: the population fitnesses are scaled in order to avoid any premature convergence, i.e. any local optimum;
- ◆ Reproduction loop:
 - Selection procedure: fitnesses are distributed on a pie and the Russian roulette randomly selects a fitness, with a larger chance to come across a high fitness; the corresponding individual (parent) is selected and a second one in the same way;
 - Crossover operator is applied to the selected pair of parents, giving children individuals;
 - Mutation operator is also applied to children individuals resulting from crossover;
- ◆ A new population is so obtained, of new individuals. The process returns to step 2 (fitness evaluation).

The software so finds the best combination of strategic parameter values (Figure 10) that, in fact, generates the most efficient scenario whose performances are the best satisfactory in accordance with the actor’s preferences.

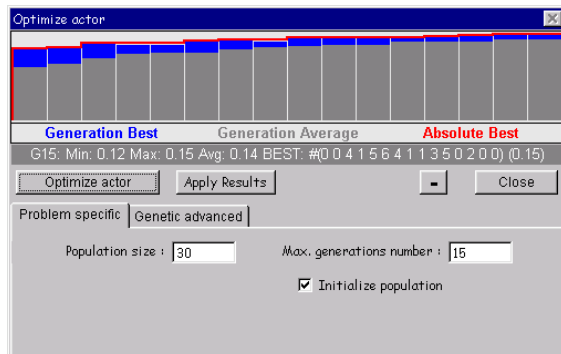


Figure 10: One-actor optimisation

Monoactor optimisation procedure

In the monoactor optimisation, the actor’s satisfaction is the convergence rule of the genetic algorithm.

On Figure 5, the Actor 3’s scenario gave worst performances when compared to the other ones. After the optimisation, the new Actor 3’s performances (displayed on Figure 11) have been largely improved.

Not only G/Gref and B/Bref are now acceptable, but all this scenario performances are largely better than those obtained with the other scenarios, even for the “Heating + DHW” energy cost, remaining beyond the admissible value previously defined by Actor 3.

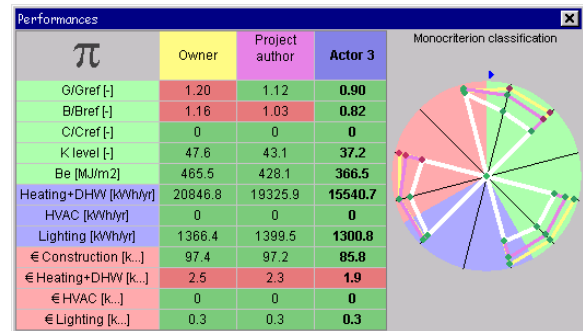


Figure 11: Optimised performances of Actor 3’s scenario

Multiple actors optimisation procedure

All the actors concerned by the project may find their most preferred compromise, by using the multiple actors optimisation procedure. This optimisation routine is similar to the monoactor one, except of its convergence rule. In this case, the software searches for the scenario giving, in the same time:

- ◆ the best possible performances;
- ◆ the most equal satisfaction for all actors: when a project obtains good performances but gives actors too much different satisfactions, this project further life is threatened by the unsatisfied actor(s).

Before the procedure, a checking routine displays the parameter(s) conflicting with actors, i.e. the parameter(s) whose actors’ authorised intervals have no intersection (Figure 12). When it happens, one can say that the scenarios are not comparable, so that any preferred compromise does not exist. The conflict parameters must so be first negotiated between actors, before any optimisation.

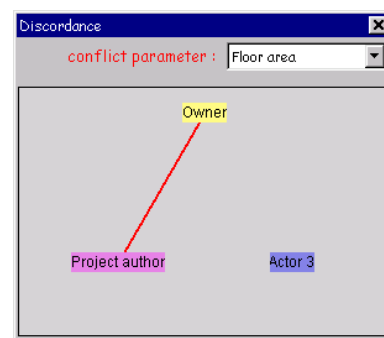


Figure 12: Display of conflict parameters (here the floor area) between the actors

In this way, the individual fitness f_x is differently defined than in the monoactor procedure:

$$f_x = \frac{1}{d_{\max}} \cdot \frac{1}{(d_{\max} - d_{\min})}$$

where d_{\max} is the maximum value (the worst performance) encountered in d_{aj} values related to actor a_j and d_{\min} the minimum d_{aj} value. The fitness optimisation has two parallel objectives, corresponding to the two terms in the f_x expression:

- ◆ maximising the performances is similar as minimising their maximum weighted distance (d_{\max}). Because no performance is nil and at least one mandatory regulation performance always remains, the denominator d_{\max} may approach to 0, but cannot be nil.
- ◆ equalising the actors' satisfactions is the same as minimising the difference between d_{\max} and d_{\min} values. The actors' satisfactions are equal when $d_{\max} = d_{\min}$ and the corresponding individual gives the parameter values of the preferred compromise.

DRAWING THE SKETCH DESIGN

The third stage of the decision process will allow a sketch design, but, at any time, the drawing of the first draft could be realised by the project author, using the EsQUIsE module, developed in LEMA.

EsQUIsE software (Leclercq, 1999) is an experimental computer-based prototype interface for capturing and interpreting the architect's sketch, by locating its architectural concepts: border line, functional space and topology.

The aim of this prototype is to compose a spatial semantic representation of the architectural project, in order to feed diverse computer architectural design evaluation routines and to serve as a tool with interface that complies to the designer's working technique. The EsQUIsE pen-based module performs the capture and the synthesis of the lines drawn on the digital tablet. These lines are drawn in black (opaque walls), in blue (glazed walls) or in magenta (comments and furniture): see Figure 13.

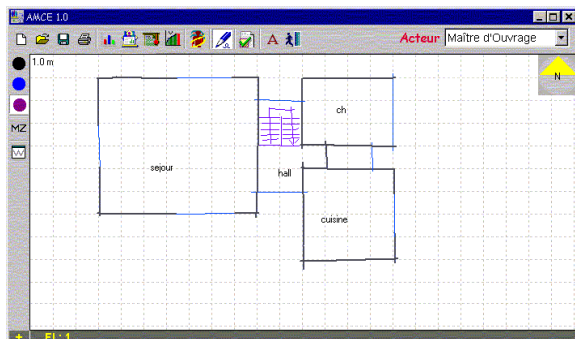


Figure 13: Original sketch (ground floor) drawn on a digital tablet or with the mouse

By studying the contacts between the synthesised lines, EsQUIsE materialises the spaces to be occupied (Figure 14). Several procedures then

deduce the topological relationships of the described architectural project (see Figure 15).

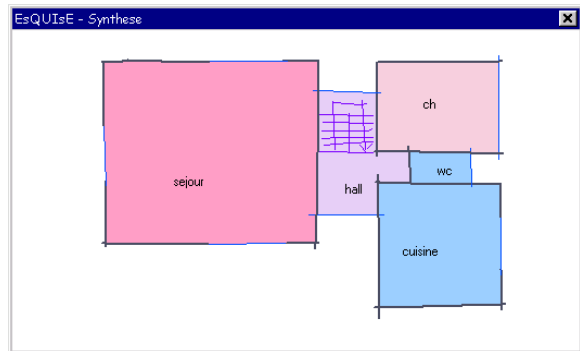


Figure 14: Synthesised sketch of the ground floor

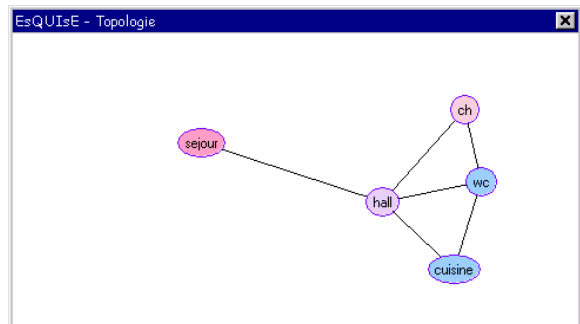


Figure 15: Spaces topology of the ground floor

The project author names the functional spaces. On this basis, the programme fixes their characteristics necessary for the evaluations.

For example, the default comfort temperature assigned to each occupied space is fixed according to its function as described in the captions: 18°C for the kitchen, 24°C for the bathroom, etc.; the user may of course change it if he prefers another value.

The advantage of using a man-machine interface based on the semantic analysis of an architectural sketch is that one is spared the fastidious measuring work of the architect's blueprint.

Instead of the two or three days usually required to measure and encode, energy and cost performances are supplied directly after the drawing of the last line of the sketch. On the other hand, it avoids any wrong manual input of numerical values.

This drawn support constitutes a preliminary basis for discussion with the client, whose understanding requires a graphical expression, that gives him the means of checking how well the project corresponds to his desiderata, including implicit ones (non-verbalised). This first response is already rich, particularly with regard to evaluations.

The application tested in the late stages of EsQUIsE is a classical calculation module MZS (for MultiZone Stationary) also previously developed in LEMA: it computes the multizone evaluation of the building energy needs, taking internal and solar gains into account.

After the drawing of the draft by the project author, the user can go back to the parameter module, where geometrical data previously defined are replaced by the ones generated by the drawn sketch (Figure 16). Other thermal or economical data of the questionnaire are unchanged.

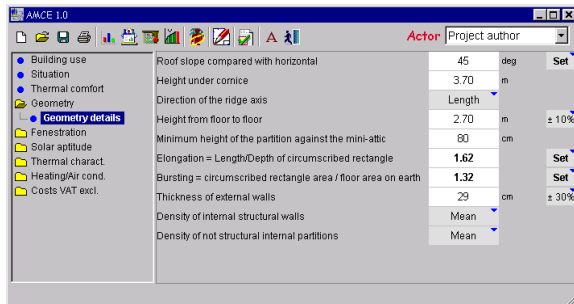


Figure 16: Geometrical data replaced by the ones resulting from the sketch drawn with the EsQUiSE module. The same evaluation routines check project performances of the drawn project, that are displayed in the performance window:

- ♦ either the performances are reached;
- ♦ or the performances are not reached. Any actor could try to satisfy each requested performance by using the parameter sensitivity analyses or the optimisation procedure.

In the case of a conflict between reached and desired performances, a negotiation would be initiated, in order to refine objectives and parameter values.

The project author presents the draft to the building owner, who is now able to react and intervene on parameters and desired performance values.

Improvement (and/or optimisation) of the draft, by its iterative modification and/or by the drawing of new drafts, will take the procedure back to previous stage.

CONTINUATION OF THE PROJECT DESIGN

The result of the feasibility stage is a sketch which parameters – accepted by the several actors – give the best performances. It circumscribes the most efficient choices related to the energy and cost performances of the future building.

It now constitutes the project basis to be used in the continuation of the design process during the detailed design, that falls under topics currently developed and commonly used in architecture.

CONCLUSIONS

With respect to usual architectural practice, our new methodology tracks provide as advantages:

- ♦ they give enhanced help to the client within an uneasy technical context, where his responsibility is increasing, especially regarding environmental regulations;
- ♦ they alleviate the work load of the project author and increase energy and cost performances of the project;

- ♦ they supply a multiple criteria decision aid for elaboration and negotiation of a preferred compromise between the several actors.

In conclusion, the final product will be used to interest more architects and help them in building energy performance, both a present and future ecological concern.

ACKNOWLEDGEMENT

The present Physical Doctorate thesis is undertaken thanks to a research contract concluded by Électricité De France (EDF-Direction des Études et Recherches), with the LEMA (Dir.: Prof. A. DUPAGNE).

It comes with the scope of the French participation to the Annex 32 of the International Energy Agency, which research topic was the Integral Building Envelope Performance Assessment.

BIBLIOGRAPHY

- BRANS J.P., VINCKE Ph., MARESCAL B. (1986), *How to Select and how to Rank Projects: the Promethee method*, in European Journal of Operational Research, 24 (2), 228-238.
- CONAN M. (1989), *Méthode de conception pragmatique en architecture*, Ministère de l'équipement et du logement : Plan, construction et architecture, Paris, France, 112 pp.
- CRAWLEY D.B., LAWRIE L.K., PEDERSEN C.O., LIESEN R.J., FISCHER D.E., STRAND R.K., TAYLOR R.D., WINKELMAN F.C., BUHL W.F., ERDEN A.E., HUANG Y.J. (1998), *Developing a New-Generation Building Simulation Tool in the United States*, Thermal Performance of The Exterior Envelopes of Buildings VII Conference, Clearwater Beach, USA, 699-708.
- LECLERCQ P. (1999), *Interpretative Tool For Architectural Sketches*, International Roundtable Conference *Visual and Spatial Reasoning in Design: Computational and Cognitive Approaches*, MIT, Cambridge, USA.
- MAYSTRE L.-Y., BOLLINGER D. (1999), *Aide à la négociation multicritère - Pratique et conseils*, Presses Polytechniques et Universitaires Romandes, Collection Gérer l'environnement.
- RIVARD H., BÉDARD C., FAZIO P., HA K.H. (1995), *Functional Analysis of the Preliminary Building Envelope Design Process*, in Building and Environment, Vol. 30, n°3.
- ROY B. and BOUYSSOU D. (1993), *Aide Multicritère à la Décision: Méthodes et Cas*, Economica, Paris.
- SCHÄRLIG A. (1996), *Pratiquer Électre et Prométhée*, Presses polytechniques et universitaires romandes, Collection "Diriger l'entreprise", Vol. 11, 173 pp.