REVUE FRANÇAISE D'AUTOMATIQUE, D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

A. DAVID
C. ROUCAIROL

Un algorithme d'aide à la conception en architecture

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 7, n° V3 (1973), p. 69-81.

<http://www.numdam.org/item?id=RO_1973__7_3_69_0>

R.A.I.R.O.
(7e année, septembre 1973, V-3, p. 69 à 81)

UN ALGORITHME D'AIDE A LA CONCEPTION EN ARCHITECTURE

par A. DAVID et C. ROUCAIROL (1)

Résumé. — *Un projet architectural est défini par l'ensemble des exigences qu'il doit satisfaire.*

La méthode proposée établit un recouvrement particulier par des cliques maximales (représentant des problèmes architecturaux simples) du graphe défini par les exigences et leurs liaisons; puis, elle permet de chercher une recomposition hiérarchique, qui indique à l'architecte l'ordre suivant lequel il peut aborder les différents problèmes et combiner leurs solutions.

INTRODUCTION

L'importance croissante des programmes architecturaux (urbanisme, industrie, santé publique...) amène les architectes à résoudre des problèmes d'aménagement de plus en plus complexes.

La démarche traditionnelle du concepteur, qui consiste à faire, au départ, des choix engageant l'ensemble du projet sans toutefois avoir une information complète, se révèle inadaptée pour résoudre de tels problèmes. En effet, les décisions les plus importantes sont prises à un moment où l'architecte n'est pas capable de les envisager dans l'ensemble de leurs causes et de leurs effets. En détaillant son analyse, il est souvent amené à contredire ses premiers choix, ce qui l'oblige à reprendre intégralement ou en partie l'étude du projet, et ceci peut-être plusieurs fois. Aussi, dès 1964, Christopher Alexander, dans son livre « Note sur la synthèse de la forme » [1], propose une méthode permettant d'aborder le problème de la conception architecturale d'une façon plus rationnelle. Il préconise une démarche où le concepteur commence son étude au niveau des problèmes les plus particuliers et, remontant progressivement par synthèses successives, arbitre les conflits qui se créent, ceci jusqu'à la synthèse finale.

(1) Université de Paris VI, Institut de Programmation.

A partir de ces idées, nous avons élaboré avec MM. Duplantier, Lablaude et Noviant, architectes, une méthode d'aide à la conception qui peut être décomposée ainsi :

(1) Définition du projet (Détermination de la liste des exigences).

(2) Détermination de la matrice d'interaction.

(3) Traitement informatique.

(4) Mise en forme architecturale.

1. DETERMINATION DE LA LISTE DES EXIGENCES

Le projet architectural est défini par un ensemble d'*exigences* auxquelles la solution architecturale doit répondre pour que la forme soit parfaitement adaptée à la fonction (université, hôpital, usine...) et à son contexte (géographique, économique, politique...). Les exigences seront de nature technique, mais aussi de nature économique, biologique, humaine, sociologique... Ainsi, pour en dresser la liste, le concepteur peut faire appel à différents consultants : économistes, organismes de statistique, usagers (par le biais d'enquêtes), psychologues, sociologues...

EXEMPLE : Un problème artificiel « la conception d'une chambre de malade et son intégration dans un service hospitalier » [8] a été défini par la liste des 51 exigences suivantes dont nous ne donnons que quelques extraits :

1. Le malade ne doit pas être gêné par les bruits provenant de l'extérieur de la chambre.

(...)

10. Les matériaux constituants ne doivent pas présenter de dangers physiques ou chimiques.

11. La lumière naturelle doit être suffisante et dosée de façon à ne pas gêner le malade.

12. La lumière artificielle doit être suffisante et dosée de façon à ne pas gêner le malade.

(...)

16. Le malade doit pouvoir avoir une bonne perception du monde extérieur.

17. Le malade doit pouvoir préserver son intimité.

18. Les éléments constitutifs de la chambre doivent avoir un aspect agréable.

19. Le personnel soignant, hôtelier et les visiteurs doivent pouvoir entrer et sortir facilement et sans gêner le malade.

(...)

27. Le malade doit pouvoir agir ou influer sur les données réglables de son environnement.

28. La chambre et ses équipements doivent pouvoir s'adapter aux habitudes et au mode de vie du malade.

31. Le personnel doit avoir à sa disposition tout le matériel nécessaire aux soins du malade.

(...)

34. Le malade doit pouvoir être transporté rapidement et facilement de sa chambre vers les autres services (et vice versa).

35. Le personnel doit pouvoir venir facilement donner des soins au malade.

(...)

45. La configuration des lieux et des équipements doit respecter en tout point la législation hospitalière française.

46. La configuration des lieux, des équipements et des matériaux doit permettre un entretien économique (rentabilité d'entretien).

47. Les éléments constituants doivent avoir une durabilité en rapport avec leur prix d'investissement (rentabilité d'usure ou d'obsolescence).

48. Le prix d'investissement ne doit pas dépasser les disponibilités et, si possible, leur être inférieur (rentabilité d'investissement).

(...)

50. L'organisation des lieux et des équipements doit permettre que le travail du personnel soit efficace et rapide (rentabilité de fonctionnement).

51. La configuration des lieux et des équipements doit autoriser toutes possibilités d'évolution et de modifications ultérieures.

2. DETERMINATION DE LA MATRICE D'INTERACTION

Il est peu probable qu'à ce niveau l'architecte trouve une solution qui soit un compromis satisfaisant entre les différentes exigences. En effet, il est généralement impossible de répondre à chacune des exigences indépendamment de toutes les autres : beaucoup d'entre elles se verraient apporter des solutions contradictoires. Aussi, pour surmonter les conflits qui peuvent apparaître, cherche-t-on les liaisons ou *interactions* entre tous les couples d'exigences.

En fait, deux types d'interaction ont été définis : la *coaction* et l'*interdépendance.*

Les exigences se traduisent par un certain nombre de contraintes sur l'ensemble des organes de construction (on appelle « organe » tout élément constitutif d'un bâtiment : structure porteuse, murs extérieurs,...).

Deux exigences sont coactives si elles contraignent le même organe.

EXEMPLE : les exigences 19 (entrées et sorties faciles) et 34 (transport rapide du malade) vont influer toutes deux sur le choix des ouvertures intérieures (portes).

Deux exigences sont interdépendantes quand, indépendamment de toute notion d'organe, il est impossible de donner une réponse à l'une sans avoir envisagé une réponse à l'autre.

EXEMPLE : les exigences 11 (lumière naturelle) et 12 (lumière artificielle) sont interdépendantes; en effet, si l'on désire une intensité lumineuse correcte dans la chambre du malade, il va falloir doser la part provenant de l'éclairage artificiel et celle provenant de l'éclairage naturel.

La coaction n'excluant pas l'interdépendance, l'architecte est amené à déterminer deux matrices différentes : la matrice de coaction et la matrice d'interdépendance.

2.1. Détermination de la matrice de coaction Mca

Le concepteur établit la liste-type de tous les organes pouvant composer un bâtiment.

Il traduit dans une échelle de valeurs convenable l'action de chacune des exigences sur chacune des classes d'organes.

A partir du tableau exigences-organes ainsi obtenu, il peut représenter les coactions entre les différentes exigences. En effet, deux exigences étant d'autant plus coactives qu'elles agissent de la même façon sur les mêmes classes d'organes, l'évaluation de leurs ressemblances (en utilisant la distance du χ^2 par exemple) permet d'obtenir la matrice de coaction Mca. Une forte coaction entre deux exigences est représentée par une forte valeur de l'élément correspondant dans la matrice Mca.

2.2. Détermination de la matrice d'interdépendance Mi

L'architecte étudie chacun des couples d'exigences et évalue leurs interdépendances suivant une échelle de valeurs préalablement choisie.

2.3. Détermination de la matrice d'interaction M

Les matrices Mca et Mi sont éventuellement « normalisées » (leurs éléments ont des valeurs comprises entre 0 et 1).

La matrice M s'obtient en faisant une combinaison linéaire convexe des éléments des deux matrices Mca et Mi :

$$M(i,j) = \alpha\ \mathrm{Mca}(i,j) + (1 - \alpha)\ \mathrm{Mi}(i,j) \qquad 0 \leqslant \alpha \leqslant 1$$

Le coefficient α permet à l'architecte de favoriser soit la coaction, soit l'interdépendance, dans la détermination des interactions.

L'importance d'une interaction est représentée par la valeur correspondante dans M.

On définit un *seuil d'intérêt* $s(0 \leqslant s \leqslant 1)$; il permet de transformer la matrice M en matrice binaire :

Si $M(i,j) < s$ alors $M(i,j) = 0$, sinon $M(i,j) = 1$

En faisant varier la valeur du seuil s, on peut jouer sur le taux de remplissage de M.

A cette matrice binaire symétrique, on associe un graphe $G = \langle X, U \rangle$ où l'ensemble des sommets X est l'ensemble des exigences, et où l'ensemble des arêtes U est l'ensemble des interactions retenues.

3. ALGORITHME

Lorsque l'architecte essaie de satisfaire une exigence, il doit envisager toutes les exigences auxquelles elle est liée de façon à arbitrer les conflits éventuels. Il a donc intérêt à traiter ensemble plusieurs exigences quand elles sont très liées entre elles, et tout particulièrement les sous-ensembles d'exigences quand celles-ci sont nombreuses et toutes en interaction deux à deux. Ces considérations nous ont conduits à chercher les *cliques maximales* du graphe G.

Définitions

1. Les cliques d'un graphe correspondent aux sous-matrices carrées pleines de sa matrice associée (ce sont des sous-graphes complets).

C est une clique si et seulement si

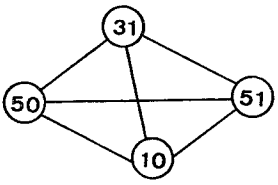
$$i \in C \Rightarrow \prod_{j \in C} M(i, j) = 1$$

2. Toute partie non vide d'une clique étant une clique, une clique qui n'est partie propre d'aucune autre clique est dite maximale;

C est une clique maximale si et seulement si

$$i \in C \Leftrightarrow \prod_{j \in C} M(i, j) = 1$$

EXEMPLE : la clique maximale n° 122 est composée des exigences 31, 50, 51, 10 :



L'ensemble de toutes les cliques maximales fournit un recouvrement du graphe G.

Toute clique est contenue dans au moins une clique maximale : des exigences peuvent donc faire partie de plusieurs cliques différentes. Lors de la synthèse architecturale, elles seront donc étudiées plusieurs fois, avec des contextes différents. L'exigence 28, par exemple, appartiendra certainement à des cliques où se poseront les problèmes de choix du mobilier, d'un type de fenêtre ou de façade. De plus, on peut penser qu'une exigence aussi abstraite que l'exigence 45 (respect de la législation hospitalière française) apparaîtra dans de nombreuses cliques. L'introduction d'exigences de cette nature dans la définition du projet architectural, contrôlées instinctivement dans une démarche traditionnelle, permet au concepteur d'éviter certains oublis et erreurs. La possibilité pour une exigence d'appartenir à plusieurs cliques maximales est un des intérêts de cette méthode qui diffère ainsi, essentiellement, d'autre méthodes d'aide à la conception utilisant soit des programmes classiques d'analyse des données [11], soit des programmes effectuant des dichotomies sur un graphe [4].

3.1. Recherche des cliques maximales

Un algorithme simple nous permet d'engendrer toutes les cliques sans omission ni répétition, l'ordre de génération des cliques maximales dépendant de l'ordre dans lequel on considère les exigences.

3.1.1. Classement des exigences

Les sommets peuvent être ordonnés par degrés croissants; la notion de *degré généralisé* [9] permet d'obtenir un classement plus intéressant.

On construit une suite de vecteurs $V_0, V_1, ..., V_m, ...$:

1. V_0 est un vecteur dont toutes les coordonnées valent 1.
2. $V_m = \mathrm{M}\, V_{m-1}$ où M est la matrice d'interaction associée au graphe G.
3. On classe les sommets par ordre croissant de coordonnées de V_m. Si à partir d'un certain rang, l'ordre induit sur les sommets reste stable, quand on passe de V_m à V_{m+1}, on s'arrête, sinon on va en 2.

Les premiers sommets sont « les sommets les moins liés aux sommets les moins liés », les derniers sont « les sommets les plus liés aux sommets les plus liés ». Une exigence, telle que l'exigence 48 (rentabilité d'investissement), se retrouvera évidemment en fin de liste, alors que les exigences 14 (accès facile aux sanitaires) et 16 (bonne perception du monde extérieur), très spécifiques, seront en tête.

3.1.2. Sélection des cliques maximales

Le nombre de cliques maximales d'un graphe est en général très important (on en a engendré 226 sur un graphe possédant 51 sommets et 355 arêtes);

au moment de la mise en forme, l'architecture devra toutes les analyser. Or, l'intérêt des réponses varie suivant la spécificité des cliques. Aussi, pour faciliter sa tâche, on ne retient une clique maximale que si l'une au moins de ses arêtes n'appartient à aucune des cliques déjà engendrées et sélectionnées.

3.1.3. Algorithme

La famille $\mathcal{C}_{max}$ des cliques maximales sélectionnées peut être engendrée par l'algorithme suivant :

$C_i = \langle X_i, U_i \rangle$ on note la ième clique maximale engendrée et retenue; $V_i = \bigcup_{j \leqslant i} U_j$

l'ensemble des arêtes des i premières cliques maximales C_i).

1. $V_0 = \varnothing$, $i := 1$;

La première clique, qu'on appelle $C = \langle X_c, U_c \rangle$, est engendrée par le premier sommet, qu'on appelle s : $X_c = \{ s \}$.

2. C est une clique; si elle est maximale *et* si $U_c \not\subseteq V_{i-1}$,
alors $C_i = C$
$V_i = V_{i-1} \cup U_i$
$i := i + 1$
aller en 4,
sinon

3. On considère l'ensemble $S = \{ s' \in X \mid s' > s$ et $X_c \cup \{ s' \}$ engendre une clique $\}$; s'il n'est pas vide, on prend son plus petit élément qu'on appelle s, on l'adjoint à X_c et on retourne en 2.

4. Si X_c ne se réduit pas au dernier sommet, alors on appelle s son plus grand élément, on ôte s à X_c et on retourne en 3,
sinon

5. $\mathcal{C}_{max}$ est l'ensemble des cliques C_i.

Cet algorithme est combinatoire, mais contrairement à d'autres algorithmes plus rapides [2] et [6], il engendre les cliques les unes après les autres, permettant ainsi d'introduire facilement un critère de sélection et de ne ranger en mémoire que les cliques maximales retenues.

3.2. Classification hiérarchique

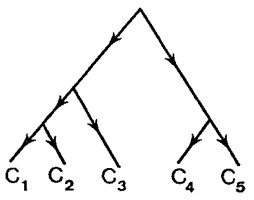
L'ensemble des cliques maximales est un recouvrement du graphe G : les problèmes architecturaux qu'elles représentent ne sont pas indépendants. Afin de guider le concepteur dans ses différentes synthèses on répartit les cliques maximales en un certain nombre de classes à l'intérieur desquelles elles se ressemblent plus que celles appartenant à des classes différentes. Il s'agit donc de construire une *hiérarchie H* sur l'ensemble $\mathcal{C}_{max}$.

Définition [3].

Soient $\mathcal{C}_{max}$ et H un ensemble de parties de $\mathcal{C}_{max}$; on dit que H est une hiérarchie sur $\mathcal{C}_{max}$ si :

1. $\mathcal{C}_{max} \in H$;
2. $\forall\, C_i \in \mathcal{C}_{max}$, on a $\{ C_i \} \in H$;
3. Quels que soient h et h', éléments de H, si $h \cap h' \neq \varnothing$ alors on a soit $h \subset h'$, soit $h' \subset h$.

Le couple ($\mathcal{C}_{max}$, H) peut être représenté par une arborescence A dont les nœuds symbolisent les différentes parties appartenant à H; ainsi l'arborescence A (fig. 1) correspond au couple ($\mathcal{C}_{max}$, H) :



$\mathcal{C}_{max} = \{C_1, C_2, C_3, C_4, C_5\}$, $H = \{h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9\}$ où
$h_1 = \{C_1\}$, $h_2 = \{C_2\}$, $h_3 = \{C_3\}$,
$h_4 = \{C_4\}$, $h_5 = \{C_5\}$, $h_6 = \{C_1, C_2\}$
$h_7 = \{C_4, C_5\}$, $h_8 = \{C_1, C_2, C_3\}$,
$h_9 = \{C_1, C_2, C_3, C_4, C_5\}$.

Figure 1
Arborescence A

3.2.1. Construction de la hiérarchie

Définitions :

1. Soit un graphe $G' = \langle X', U' \rangle$, l'indice σ associé à G' est défini par :

$$\sigma(G') = \frac{|U'|}{|X'|\,(|X'| - 1)}$$

Cet indice varie entre 0 et 1 et prend la valeur 1 si et seulement si G' est complet.

2. Soient $\mathcal{C}_1$ et $\mathcal{C}_2$ deux parties de $\mathcal{C}_{max}$. On peut leur associer deux sous-graphes de G : $\mathcal{C}_1 = \langle X_1, U_1 \rangle$, $\mathcal{C}_2 = \langle X_2, U_2 \rangle$.

Posons $\mathcal{C}_1 \cup \mathcal{C}_2 = \langle X_1 \cup X_2, U_1 \cup U_2 \rangle$; c'est un sous-graphe de G.

Une formule de proximité entre parties $\mathcal{C}_1$ et $\mathcal{C}_2$ de $\mathcal{C}_{max}$ est définie par :

$$d(\mathcal{C}_1, \mathcal{C}_2) = \sigma(\mathcal{C}_1 \cup \mathcal{C}_2)$$

$d(\mathcal{C}_1, \mathcal{C}_2)$ varie entre 0 et 1; en particulier si $C_i \in \mathcal{C}_{max}$ alors $d(C_i, C_i) = 1$ et $0 \leqslant d(\mathcal{C}_{max}, \mathcal{C}_{max})$. Pratiquement, elle s'apparente à un indice de similitude, c'est-à-dire que si $\mathcal{C}_1$ et $\mathcal{C}_2$ sont deux parties de $\mathcal{C}_{max}$ telles que $\mathcal{C}_1 \supset \mathcal{C}_2$ alors $d(\mathcal{C}_1, \mathcal{C}_2) < d(\mathcal{C}_1, \mathcal{C}_2)$.

Construire une hiérarchie sur $\mathcal{C}_{max}$ revient à construire une suite de forêts $A_0, A_1, \ldots$ par la relation de récurrence suivante :

Som A_i est l'ensemble des racines des arborescences de la forêt A_i.

(1) Som $A_0 = \{\{ C_i \} \mid C_i \in \mathcal{C}_{max} \}$.

(2) Soit $a_h = s_h \cup s'_h$ tel que

$$d(s_h, s'_h) = \text{Sup}\,(d(s, s') \mid s, s' \in \text{Som}\, A_{h-1} ; s \neq s') ;$$

alors Som A_h = Som $A_{h-1} \cup \{ a_h \} - \{ s_h, s'_h \}$, c'est-à-dire que A_h s'obtient à partir de A_{h-1} en plaçant le sommet $a_h = s_h \cup s'_h$ au-dessus de s_h et s'_h.

Dans l'exemple précédemment cité, la hiérarchie construite sur l'ensemble des 137 cliques maximales sélectionnées peut être représentée par l'arborescence de la figure 2.

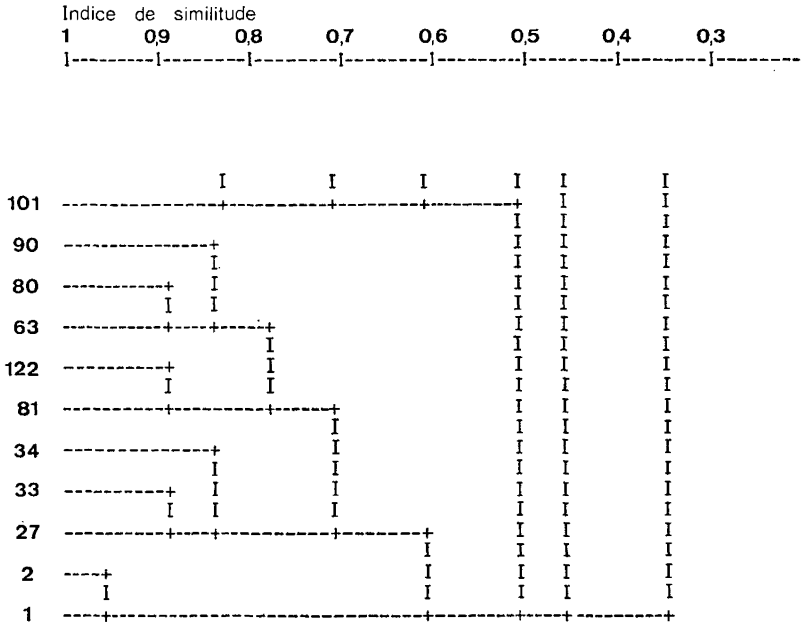


Figure 2
Exemple d'édition de l'arborescence

4. MISE EN FORME ARCHITECTURALE

Le concepteur étudie d'abord toutes les cliques, qu'il traite comme des problèmes architecturaux particuliers. Pour chacune d'elles, il propose, aidé au besoin par des spécialistes, une ou plusieurs solutions techniques, sous forme de croquis rapides, de références, d'annotations...

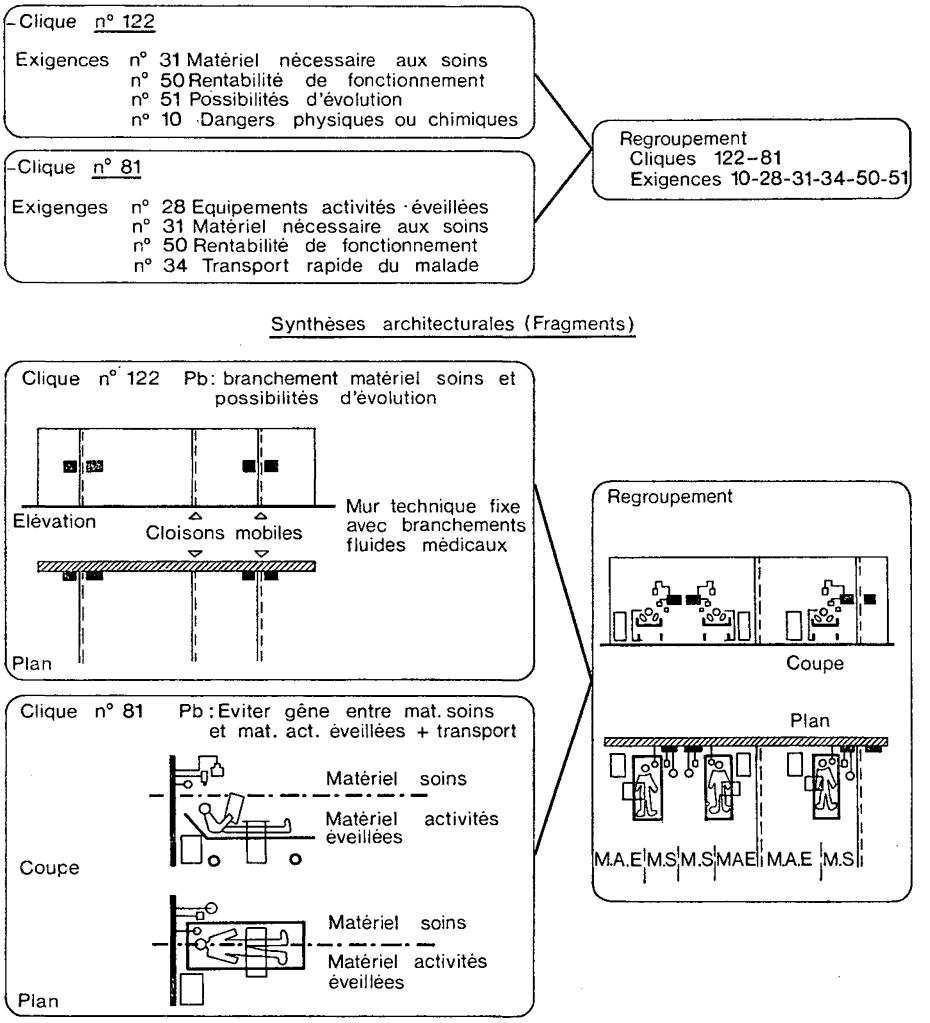


Figure 3

Exemple de synthèse réalisé par l'architecte

Puis, conformément à l'ordre fournit par la classification hiérarchique, il cherche à faire coïncider ces différentes synthèses partielles, en arbitrant les conflits susceptibles d'apparaître. Ainsi, progressivement, il arrive à la synthèse finale : la mise en forme architecturale du projet étudié (fig. 3).

5. LE PROGRAMME RECOMPH

Le programme RECOMPH est écrit en FORTRAN IV et a été testé sur CDC 3600 et IBM 370-165.

A partir des données suivantes :

— le tableau organes-exigences,

— la matrice d'interdépendance,

— le coefficient α et le seuil s,

il fournit à l'utilisateur (sorties sur imprimante) :

— une image des données (texte des exigences, matrice d'interaction M);

— la liste des cliques maximales sélectionnées,

— le diagramme de recomposition hiérarchique avec, pour chaque regroupement, les noms des parties réunies, les noms des différentes exigences concernées,

— le dessin de l'arborescence A associée à la hiérarchie (fig. 2).

Le programme a été utilisé par plusieurs architectes. En particulier :

— pour l'étude de programmation hospitalière,

— lors de la conception des pôles d'équipements d'une unité touristique UPA 3 de la côte Aquitaine (étude réalisée à la SERTI).

6. CONCLUSION

L'emploi d'une telle méthode ne supprime, ni ne remplace le travail du concepteur; bien au contraire, elle exige de lui une plus grande rigueur intellectuelle et un approfondissement plus poussé des questions, en contre-partie desquelles, elle fournit un support de réflexion, une structuration des décisions (fig. 4).

Mais la responsabilité du choix incombe toujours au concepteur : la solution envisagée par l'architecte X n'est pas nécessairement celle qu'aurait imaginée l'architecte Y avec les mêmes aides. Cette souplesse convient parfaitement en ce domaine où l'art du constructeur doit encore se manifester et où l'éthique personnelle peut continuer à jouer un rôle.

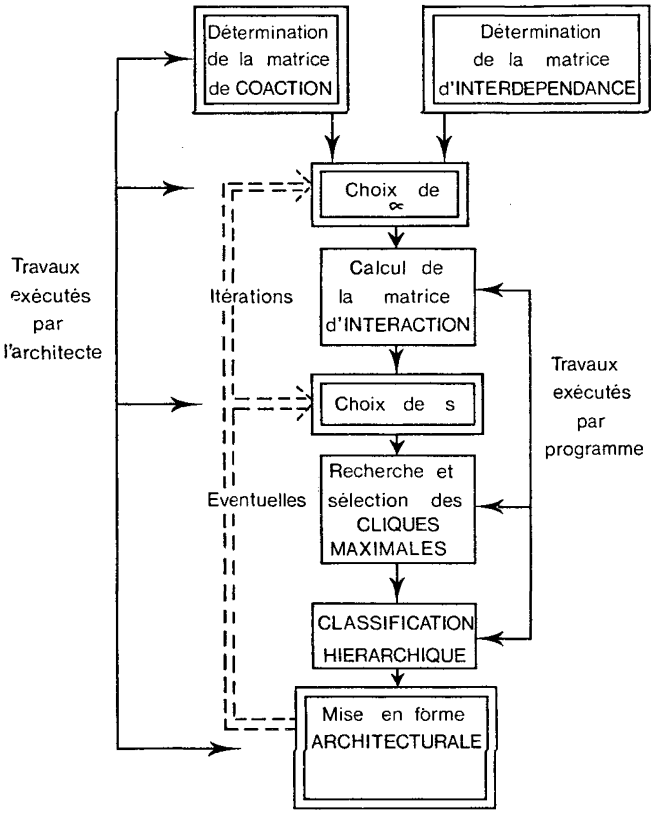


Figure 4

Les travaux exécutés par programme s'insèrent dans l'ensemble du travail de l'architecte

REFERENCES

[1] C. ALEXANDER, *Notes sur la synthèse de la forme*, Dunod, 1964.

[2] J. G. AUGUSTON et J. MINKER, *An analysis of some graph theorical cluster techniques*, J. ACM 17, 4 (oct. 1970).

[3] J. P. BENZECRI, *Leçons sur l'analyse des données*, Laboratoire de statistique de l'Université Paris VI, 1969.

[4] A. BERNHOLTZ et E. BIERSTONE, *Computer augmented design*, Cahier CSTB n° 99, mai 1969.

[5] C. BERGE, *Graphes et hypergraphes*, Dunod, 1970.

[6] D. CORNEIL et G. D. MULLIGAN, *Corrections to Bierstone's algorithm for generating cliques*, J. ACM 19, 2 (april 1972).

[7] A. DAVID et C. ROUCAIROL, Aide à la conception en architecture. Actes du colloque « Analyse des données et applications à l'architecture et à l'urbanisme », Institut de l'Environnement, mai 1972.

[8] H. M. DUPLANTIER, P. A. LABLAUDE et P. NOVIANT, *Recherche d'une logique de conception applicable au domaine architectural*, Thèse de diplôme d'architecte, ENS Beaux-Arts, juillet 1971.

[9] J. L. LAURIERE, *Sur la coloration de certains hypergraphes*, Thèse de 3e cycle, Université de Paris VI, juin 1971.

[10] J. C. LERMAN, *Les bases de la classification automatique*, Gauthier-Villars, Collection Programmation, 1970.

[11] J. P. MAROY, *Une utilisation de l'analyse des données dans la conception architecturale*, Actes des journées « Informatique et conception en architecture », IRIA, 1971.