

Stéphane HANROT
Paul QUINTRAND
Jacques ZOLLER (GAMSAU)
Eugène CHOURAQUI
Philippe DUGERDIL
Philippe FRANÇOIS
Michel RICARD (GRTC)
Christian GIRAUD (IIRIAM)

Le projet TECTON :
Un système expert
de CAO
intégrant le savoir
architectural

Résumé :

L'objet de ce projet est de réaliser une maquette informatique intégrant une partie du "savoir-faire" architectural. Il repose principalement sur la conviction que ce "savoir-faire" est partiellement contenu dans le vocabulaire de l'architecture et dans les documents graphiques produits par les architectes. Ce vocabulaire et ces documents graphiques sont le résultat d'une codification qui s'est enrichie au fil des siècles.

Ce projet en est à ses débuts. En conséquence, on se limitera à présenter les bases théoriques et les hypothèses sur lesquelles est fondé le projet, l'état de la recherche et les résultats espérés. En particulier sont abordés les problèmes de représentation des connaissances, le recours aux langages orientés objets et la modélisation géométrique en CAO.

I - INTRODUCTION.

L'analyse des systèmes de CAO (ou désignés comme tels) en architecture montre que leur utilisation ne se fait que dans les phases finales du projet, lorsque le projet est "conçu". Les systèmes matériels et logiciels apparaissent pour les plus performants, comme des machines à instrumenter les projets, c'est-à-dire des outils de mise au point et d'achèvement des projets, ou encore plus simplement, comme machines à dessiner.

Un des problèmes majeurs auxquels se trouvent aujourd'hui confrontés les producteurs de logiciels est d'avoir à concevoir un outil permettant la modélisation d'objets incomplètement décrits et des problèmes "mal définis" caractérisant la conception architecturale.

Les méthodes de programmation classiques sont inadaptées à ce type de problème ; en revanche, les recherches en intelligence artificielle se sont données pour objectifs d'apporter des éléments de réponses pouvant contribuer à leur résolution.

L'approche "intelligence artificielle" appliquée à notre domaine, l'architecture, ne doit pas être abordée en terme classique de "conception" de "système expert" qui se caractérise par l'objectif essentiel de représenter et de manipuler des modes de raisonnement ; il ne s'agit pas de faire une machine à résoudre les problèmes architecturaux dont on connaît la complexité, mais plutôt de créer un outil intégrant une part du savoir architectural relatif à la description et à la manipulation des objets architecturaux en cours de conception, tout en respectant les stratégies du concepteur. Il s'agit aussi de restituer en situation informatique tout le processus de transfert d'information d'un état de problème à l'autre lors des étapes de simulation/communication.

II - MODELISATION DU SAVOIR ARCHITECTURAL

Le savoir architectural est riche. Comment, sans l'appauvrir dégager des régularités qui permettent à la fois de structurer l'implémentation informatique des connaissances, et offrir aux utilisateurs un outil proche de leur mode de résolution des problèmes architecturaux.

Pour ce faire il a été recherché, au travers des filtres que nous offrent le vocabulaire architectural et les représentations graphiques mis en oeuvre dans le projet, une classification des objets architecturaux et des relations qu'ils entretiennent.

Ceci a conduit à reconnaître des "niveaux de définition" de l'objet architectural et un système de relation particulier : le "calage".

2.1. Niveaux de définition.

On en spécifie quatre. Ils sont établis d'après une correspondance entre les échelles de la représentation consacrées par l'usage (1/500 - 1/200 - 1/100 - 1/50 - 1/20) et les classes d'objets susceptibles d'être vus à ces échelles. Chaque niveau constitue une couche d'un arbre de décomposition de l'objet architectural considéré. Certaines décompositions ne sont pas gérées par ces couches, aussi on accepte qu'un objet d'un niveau donné puisse être partitionné :

(: exemple { mur (tête, faces, corps, partie courante ...))
 n_2 n_2

Ces partitions permettent de préciser la définition de l'objet. Par ailleurs, ces niveaux de définition renvoient à des champs de problèmes pertinents sur l'objet architectural.

2.2. Les Relations.

Les objets sont liés entre eux selon des relations diverses. On présente ici la relation de calage qui met en jeu :

- les attributs morphologiques des objets (limites, dimensions, parties)
- les attributs architectoniques (axes, nus, alignements, aplomb)
- des êtres qui n'appartiennent pas aux objets : les êtres architectoniques (trames, axe de symétrie...voir en § 3) utilisés dans le projet architectural comme outil de coordination de l'espace tridimensionnel.

Le calage est un système référentiel tel qu'il existe un "référéncié" lié par calage à "un référent". Les relations de calages sont en nombre restreint ("au nu de", "à l'alignement de", "à l'aplomb de", "au règne de", "à l'arasé de", "à niveau").

Ces relations permettent de positionner les objets entre eux dans l'espace du projet comme dans l'espace du chantier.

III - REPRESENTATION SYMBOLIQUE DU SAVOIR ARCHITECTURAL.

3.1. Description sommaire du modèle.

La représentation des connaissances est fondée sur le modèle orienté objet, à travers un langage en cours de développement autour de PROLOG. Les classes et les instances sont décrites à l'aide d'un ensemble de champs et les champs à l'aide d'un ensemble d'aspects précisant les propriétés et les valeurs des champs dans lesquels ils interviennent. Certains de ces champs et de ces aspects sont définis a priori dans le langage, les autres dépendant du domaine d'application.

Les champs prédéfinis sont :

SORTE-DE : permet de relier une classe fille à sa ou ses classes mères. Il traduit une relation d'inclusion. Ce champ à valeurs référentielles contient donc une liste de noms de classes.

EST-UN : permet de relier une instance à une classe. Il traduit une relation d'appartenance. C'est un champ à valeur référentielle qui ne peut contenir qu'une seule valeur. Par la définition d'un champ différent pour cette relation, nous avons levé l'ambiguïté existant souvent dans les langages orientés objet entre la représentation de l'appartenance et celle de l'inclusion.

PARTIE-DE : traduit la relation entre un objet complexe et l'ensemble de ses parties. C'est un champ à valeur référentielle qui permet de relier une classe à une ou plusieurs autres classes ou une instance de classe à une ou plusieurs autres instances de classes. Ce champ, particulier à notre modèle, se trouve dans un objet représentant une partie d'un ou plusieurs objets complexes et contient le nom de ce ou ces objets

complexes. Ce champ permet d'introduire un mécanisme d'héritage sélectif : un objet représentant une partie d'objet complexe ne peut hériter que d'un sous-ensemble des champs de l'objet complexe. On donnera donc explicitement au niveau du champ PARTIE-DE la liste des champs que l'on peut hériter.

Les aspects statiques sont :

VALEUR : contient une valeur terminale ou une liste de valeurs terminales. Une valeur terminale est une chaîne de caractères alphanumériques.

REFERENCE : contient une valeur référentielle ou une liste de valeurs référentielles. Une valeur référentielle est un nom de classe ou d'instance de classe. On introduit cet aspect pour lever l'ambiguïté existante souvent entre valeur référentielle et valeur terminale.

DEFAULT : introduit une valeur terminale par défaut. Cet aspect permet de mettre en oeuvre un type particulier de raisonnement par défaut.

DOM-VALEUR : contient une liste de valeurs terminales constituant le domaine d'instanciation d'un champ à valeur terminale.

DOM-REFERENCE : contient une liste de valeurs référentielles constituant le domaine d'instanciation d'un champ à valeurs référentielles. La distinction entre ces deux domaines permet également de lever l'ambiguïté sur le type de leur contenu.

HERITAGE : contient la liste des champs qui peuvent être hérités sélectivement par un objet. Il est toujours associé au champ PARTIE-DE.

Les aspects dynamiques sont :

SI-AJOUT : contient la procédure (démon) qui sera déclenchée lors de l'ajout d'une valeur dans le champ.

SI-SUPPRIME : contient la procédure qui sera déclenchée lors de la suppression d'une valeur dans un champ.

SI-BESOIN : contient la procédure permettant de calculer la valeur associée à un champ lors d'une demande d'information.

CONTROLE : contient la procédure qui permet de contrôler la cohérence entre une valeur terminale instanciée pour un champ et son domaine.

3.2. Les mécanismes d'héritage.

a) l'héritage vertical

Il est mis en oeuvre à l'aide des champs SORTE-DE et EST-UN. L'héritage vertical est défini entre une classe fille et une classe mère tel que la classe fille ait accès à tous les champs de la classe mère. Comme le champ "sorte-de" peut contenir plusieurs noms, on a un héritage multiple.

Le processus d'héritage vertical multiple est le suivant: si une valeur est requise pour un champ au niveau d'une instance, on examine tous ses champs. Si le champ cherché n'y est pas présent, on examine toutes les classes que l'on peut atteindre par héritage vertical selon une stratégie "en largeur d'abord", et on ramène la liste des valeurs trouvées selon chacun des chemins d'héritage.

b) L'héritage sélectif

Il est mis en oeuvre à l'aide du champ PARTIE-DE. Il consiste à définir au niveau du lien "partie-de" d'une classe la liste des champs qui peuvent être hérités par une classe "partie". Si l'on a deux classes reliées par une relation PARTIE-DE et que l'on crée une instance de chacune d'elles, le champ PARTIE-DE sera également présent au niveau des instances. Donc l'héritage sélectif agit au niveau des instances et non plus au niveau des classes comme pour l'héritage vertical, et permet d'hériter d'une valeur instanciée. Lors d'une recherche de valeur pour un champ, absent, d'une instance, on vérifie si il y a un champ "partie-de" dans l'instance. Si oui on va consulter, au niveau des classes, la liste des champs que l'on peut hériter sélectivement. Si le champ cherché y est mentionné, on va lire le champ cherché dans l'instance composée correspondante.

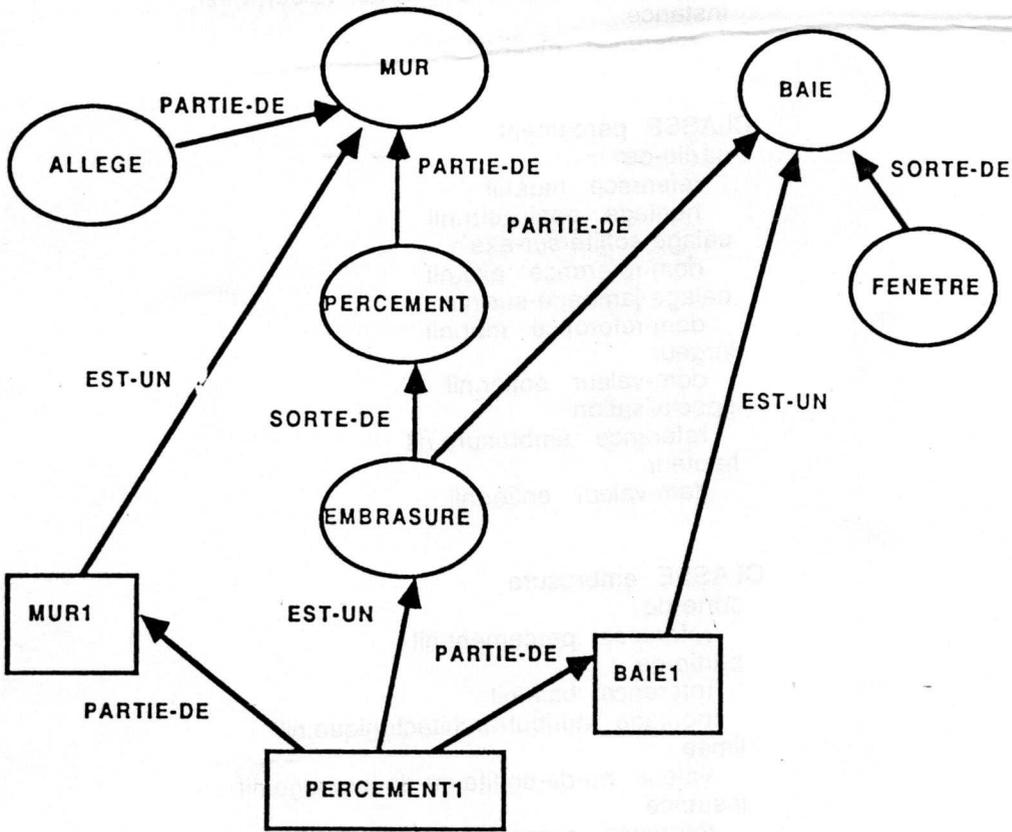
c) Composition des héritages

Il est possible que l'on puisse simultanément hériter d'un champ verticalement et sélectivement. Les principes de résolution des conflits sont les suivants :

- L'héritage sélectif est prioritaire sur l'héritage vertical, car il permet d'hériter une valeur instanciée pour un champ.
- L'héritage sélectif inhibe l'héritage vertical depuis l'instance de départ.
- Si l'héritage sélectif ne conduit pas à l'obtention d'une valeur instanciée, l'héritage vertical est déclenché à partir de la dernière instance explorée par héritage sélectif.

3.3. Exemple de base de connaissance décrite dans les termes du modèle proposé.

a) Représentation graphique des liens entre classes et instances



b) Description des objets de la base avec leurs champs et aspects.

BASE DE DONNEES DES CLASSES

CLASSE mur
 longueur
 dom-valeur entier.nil
 epaisseur
 dom-valeur entier.nil
 partie-allege
 dom-reference allege.nil
 partie-percement
 dom-reference percement.nil
 calage-axe-sur-axe
 dom-reference axe.nil
 calage-tete-sur-nu
 dom-reference mur.nil

hauteur

dom-valeur entier.nil

controle a.nil -> verifier-domaine(a).nil

si-besoin a.nil -> demander-valeur(a).nil

instance

reference mur1.nil

CLASSE percement

partie-de

reference mur.nil

heritage epaisseur.nil

calage-soffite-sur-axe

dom-reference axe.nil

calage-jambage-sur-nu

dom-reference mur.nil

largeur

dom-valeur entier.nil

specialisation

reference embrasure.nil

hauteur

dom-valeur entier.nil

CLASSE embrasure

sorte-de

reference percement.nil

partie-de

reference baie.nil

heritage attribut-architectonique.nil

limite

valeur nu-de-soffite.nu-de-jambage.nil

instance

reference percement1.nil

CLASSE baie

attribut-architectonique

valeur axe-vertical.axe-horizontal.nil

partie-embrasure

dom-reference embrasure.nil

specialisation

reference fenetre.nil

instance

reference baie1.nil

CLASSE fenetre

sorte-de

reference baie.nil

caracteristiques

valeur etanche.nil

partie-fermeture
dom-reference fermeture.nil

BASE DE DONNEES DES INSTANCES

INSTANCE mur1
est-un
 reference mur.nil
partie-percement
 reference percement1.nil
epaisseur
 valeur 2.nil
longueur
 valeur 10.nil
calage-axe-sur-axe
 reference axe-de-mur2.nil

INSTANCE percement1
largeur
 valeur 1.nil
hauteur
 valeur 2.nil
est-un
 reference embrasure.nil
partie-de
 reference mur1.baie1.nil
calage-soffite-sur-axe
 reference axe-de-structure1.nil

INSTANCE baie1
est-un
 reference baie.nil
partie-embrasure
 reference percement1.nil

IV - REPRESENTATION GRAPHIQUE DU SAVOIR ARCHITECTURAL

4.1. Présentation du niveau graphique de représentation.

Les systèmes de CAO sont en général articulés autour d'un modèle géométrique, l'information non graphique étant, dans son intégralité, attachée à ce modèle.

Par contre dans un système de CIAO la géométrie n'est qu'une des connaissances du système et servira tout à la fois à la manipulation des formes géométriques, à la visualisation des objets sur les terminaux et à l'interaction avec le modèle conceptuel pour décrire ou modifier, à partir du niveau graphique, la connaissance qui le concerne.

Il faudra donc un modèle géométrique suffisamment souple permettant tout aussi bien de représenter un objet incomplètement défini (l'esquisse) qu'un objet terminal dans toute sa complexité (voir § 2.1).

Trois niveaux de modélisation seront nécessaires : un niveau purement conceptuel, un niveau formel d'expression des contraintes et un niveau de représentation graphique, solution du précédent en terme de modèle tridimensionnel.

La description des modèles envisagés a été particulièrement étudiée pour le niveau formel d'expression des contraintes ; deux approches ont été examinées successivement, l'approche analytique et l'approche orientée objet.

4.2. Approche analytique.

Dans l'approche analytique l'idée sous-jacente est de trouver un formalisme de modélisation géométrique dans lequel on puisse exprimer les contraintes sur les objets sous la forme de systèmes d'équations linéaires, ceci afin de pouvoir réaliser une implémentation efficace en PROLOG.

En conséquence l'approche déclarative développée par C. Van Wyk dans le langage IDEAL a tout son intérêt. Ce langage permet de dessiner dans le plan complexe des formes élémentaires spécifiées par des "boîtes" au sens couvert par cette notion dans ce langage. Toute boîte contient deux parties : l'une déclarative dans laquelle des équations linéaires sur des nombres complexes spécifient les points remarquables de la boîte ainsi que les contraintes qui les lient ; l'autre partie, procédurale, constituée par les instructions permettant de dessiner la forme.

Ainsi en définissant la boîte MUR on instancie un mur particulier en ajoutant aux équations de cette boîte un nombre suffisant de contraintes afin que toutes les variables soient liées. De même l'expression de différents cas de calage de 2 murs l'un par rapport à l'autre s'exprime facilement et met en jeu uniquement des contraintes linéaires ou quasi linéaires ordonnées.

L'effort a été porté sur l'extension de ce modèle à un espace à trois dimensions où l'on ne dispose plus de la puissance algébrique des complexes, en particulier pour l'expression des relations angulaires. L'expression des relations de calage fait alors intervenir des contraintes de distances de point à un axe ou à un plan et les équations qui en découlent ne sont plus linéaires.

4.3. Approche objet.

L'approche objet de la modélisation géométrique, cohérente avec la structure mise en place au niveau conceptuel (voir § 3), offre de plus l'intérêt de ne manipuler que des objets aux valeurs toujours instanciées, ne serait-ce que par défaut.

En revanche la modification des paramètres d'un objet entraîne des opérations plus complexes. Le langage utilisé pour modéliser une telle approche doit posséder un ensemble de fonctionnalités permettant de traiter les contraintes entre objets (un extension de ce type a été proposée par A. BORNING autour de SMALLTALK : THINGLAB).

4.4. Evaluation des approches.

Le choix entre ces deux approches est fonction des critères suivants :

- Aptitude à modéliser un objet non complètement défini
- Possibilité d'identifier les objets ou parties d'objets dans un contexte (niveau de conception)
- Aptitude à satisfaire les contraintes existantes, lors de la modification ou de l'ajout de nouveaux objets
- Cohérence du modèle aux différents niveaux de définition

4.5. Champ retenu pour la validation.

Une partie restreinte de l'univers des objets architecturaux utilisés par l'architecte en phase de conception a été retenu pour tester la performance de ces modèles vis à vis des contraintes portant sur les objets.

Le champ de travail de cette phase préparatoire se limite à l'étude des murs du point de vue de leur modélisation géométrique.

- Attributs morphologiques (dimensions, nature de forme, limites)
- Attributs architectoniques (les axes et les nus pris en comptes lors du calage).

L'association langage objet/contraintes est un outil particulièrement intéressant pour la manipulation de la complexité inhérente au projet architectural. L'utilisateur définit toutes les relations et laisse au système la planification de la résolution de l'intégralité des contraintes liant les différents objets.

L'extension graphique de PROLOG permettra de traduire les différents niveaux de modélisation graphique dans cette approche objet et de restituer la composante graphique comme l'une des connaissances prises en compte dans un système de CAO.

CONCLUSION

L'intérêt de cette approche est qu'elle permet de prendre en compte le savoir architectural dans sa dimension complexe, avec les finesses qu'exige le projet architectural et sans préjuger des méthodes de conception - d'un point de vue informatique ce travail conduit à la mise au point d'un modèle de représentation des connaissances orienté objet autour de PROLOG permettant de saisir toute la complexité du savoir architectural.

BIBLIOGRAPHIE.

Le vocabulaire de l'architecture. Ministère des affaires culturelles, Imprimerie Nationale, Paris, 1972.

CHOURAQUI E., DUGERDIL Ph. - Application des langages orientés objet à la CAO de l'architecture. Actes des journées AFCET sur les Langages orientés objet, Bigre, n° 48, janvier 1986.

DUGERDIL Ph. - Une méthodologie orientée objet pour la représentation des connaissances en CAO, Mémoire de DEA en Informatique et mathématique, Faculté de Luminy, Univ. d'Aix-Marseille II. 1985.

FREGIER M. - Etude de la représentation des composantes morphologiques des données du projet architectural dans un contexte de CAO. GAMSAU/Plan construction. 1984.

GIRAUD Ch. - The PHP representation. CAD. 1984.

QUINTRAND P. - Identification et description des composants architecturaux et constructifs manipulés dans le projet d'architecture GAMSAU/BETEREM/Plan construction. 1985.

QUINTRAND P. et al. - La CAO en architecture. Paris Hermès Publishing. 1985.

Nom des auteurs : Stéphane HANROT, Paul QUINTRAND, Jacques ZOLLER
(GAMSAU)
Eugène CHOURAQUI, Philippe DUGERDIL, Philippe
FRANCOIS, Michel RICARD (GRTC)
Christian GIRAUD (IIRIAM)

Adresse : GAMSAU/EAM - 70, Route Léon Lachamp 13288 MARSEILLE CEDEX
09 - Téléphone : 91.41.11.85
GRTC/CNRS - 31, Chemin Joseph Aiguier 13402 MARSEILLE CEDEX
09 - Téléphone : 91.22.40.00
IIRIAM - 2, Rue Henry Barbusse 13241 MARSEILLE CEDEX 01 -
Téléphone : 91.91.76.59