

Modélisation Graphique et Sémantique d'un Environnement Coopératif de Conception

M. Djoudi¹, A. Zidani²

¹Département d'Informatique
Université de Poitiers, France

²Institut d'Informatique
Université de Batna

Résumé

Nous proposons dans cet article, une approche basée sur la modélisation des connaissances d'un environnement de conception, une démarche s'appuyant sur une représentation objet et réseaux sémantiques qui devrait constituer un apport méthodologique important pour les concepteurs.

Cette démarche repose sur trois étapes, la modélisation de l'environnement à l'aide d'outils graphiques, la traduction du modèle dans un environnement objet et la description des primitives d'interaction s'appliquant sur les connaissances du système.

La méthode proposée a pour but de permettre à un utilisateur de décrire son environnement le plus naturellement possible, par des commandes de haut niveau d'abstraction. Le travail constitue un premier pas vers le développement d'un système coopératif qui permet à plusieurs utilisateurs de construire et d'interagir avec l'environnement [1].

Mots clés

Modélisation graphique, Conception d'objets, Systèmes coopératifs, Interface graphique, Réseaux sémantiques, Base de connaissances.

1. Introduction

Malgré les progrès technologiques, avec des processeurs de plus en plus rapides, une capacité mémoire en perpétuel augmentation, la conception d'objets à l'aide de techniques classiques pose un certain nombre de problèmes :

- l'impact de la modification d'une partie de l'objet est difficilement réalisable, sans un recalcul de l'ensemble de l'objet.
- les objets ne disposent pas suffisamment de sémantique, le partage des structures de données par des fonctionnalités différentes n'est pas évidente. En particulier, la dynamique de l'environnement et l'interaction sont difficilement réalisables en l'absence d'une sémantique forte des objets.
- de façon naturelle, la description de l'environnement par les objets est plus stable que celle réalisée par les traitements, particulièrement dans un système coopératif.

Partons de ces constats, nous proposons une méthode basée sur les connaissances de l'environnement (objets et les relations et contraintes associées). Ceci dans le souci de maîtriser la cohérence de l'ensemble des informations participant à la description de l'environnement.

La modélisation proposée repose donc sur une approche objet avec la prise en compte de types de nœuds et de relations des réseaux sémantiques [2].

2. Objectifs de la Démarche

La démarche a pour objectif de permettre à chaque utilisateur de décrire son environnement le plus naturellement possible, par des commandes de haut niveau d'abstraction.

L'utilisateur exprime ainsi l'image mentale de l'environnement à modéliser, et la détermination des valeurs numériques précises est à la charge de l'application. Les commandes lui permettent d'exprimer la forme, la texture des objets de l'environnement ainsi que les positions et les relations entre objets.

Chaque commande sera ensuite interprétée par un grand nombre d'actions élémentaires par le modèleur géométrique en vue de sa visualisation [3].

Il est nécessaire de disposer de commandes de haut niveau avec le traitement des connaissances floues telle que : plus grand, à côté de, ...

La démarche proposée s'appuie sur les étapes suivantes :

- la conception et la réalisation de l'interface graphique incluant les possibilités d'interaction par des commandes de haut niveau sur l'environnement ;
- la conception de l'environnement prenant en compte les contraintes relatives aux objets ;
- le développement d'outils de visualisation de l'environnement basé sur les techniques connues de la modélisation géométrique.

Le système est conçu à l'aide des techniques de conception par objets qui offrent de nombreux avantages : le concept de classe, l'héritage et la modularité. Cette approche permet la réutilisabilité et la facilité d'intégration des différentes parties du système. Chacun des composants (objet, lien, événement) du système est donc représenté par une classe. La modélisation par objet permet de concevoir des objets quelconques et leurs affecter des attributs (par exemple : nom, taille, texture, etc.) et des propriétés (par exemple grandir, éclaircir, etc.).

3. Architecture Logicielle

La plate-forme proposée permet la simulation d'un environnement, avec la possibilité d'interaction par des commandes évoluées et la manipulation d'informations symboliques spatiales [3].

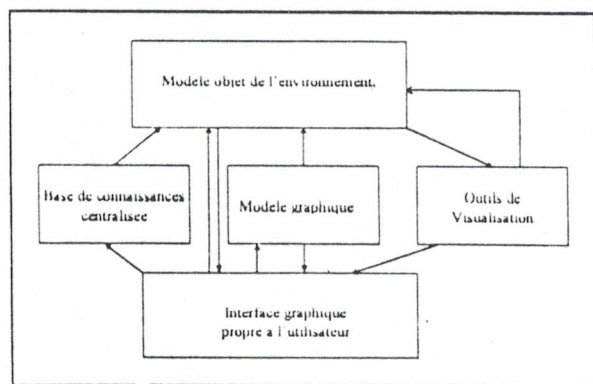


Fig. 1. Architecture simplifiée du système.

Le système est conçu pour permettre :

- la définition d'objets simples ou composés et leur rattachement à une hiérarchie de concepts;

- l'interaction avec des objets de l'environnement par l'intermédiaire de l'interface propre à chaque utilisateur;
- la représentation de l'image mentale de l'environnement sous forme d'un graphe conceptuel;
- la visualisation des objets de l'environnement selon différentes vues et par plusieurs utilisateurs.

4. Structure de la Base de Connaissances

L'objet constitue un cadre dans lequel sont rassemblées les caractéristiques (attributs) de l'objet modèle de l'environnement. Chaque attribut est défini librement par l'utilisateur lors de la description du modèle (par exemple pour l'objet chaise, un attribut peut être le nombre de pieds). L'objet peut ainsi regrouper un grand nombre d'éléments de description divers relatifs à un objet.

Chaque objet est décrit par un ensemble d'attributs et de propriétés. Les objets ayant des attributs communs et qui partagent les mêmes propriétés sont regroupés dans une même classe.

Un objet peut être simple ou composé, un objet simple est décrit par ses attributs qui décrivent par exemple la forme géométrique ou la texture. Un objet composé est constitué d'un ensemble de composants. Chacun des composants est un objet avec sa position, et des liens avec les autres composants. Chaque objet simple ou composé hérite les propriétés de sa classe.

Par ailleurs, les objets héritent des attributs et des méthodes d'une classe générique. Un objet de la classe générique a comme propriétés : la position, orientation, centre de gravité, visibilité, liens, ainsi que le nom de la classe propre de l'objet [4].

5. La Modélisation Graphique

Le modèle de structuration des connaissances retenues est inspiré des travaux sur les réseaux sémantiques et sur le concept d'objet. Ce modèle permet d'établir une représentation de l'environnement à partir d'un graphe exprimant des relations entre différents éléments [2].

Dans le processus de construction d'un environnement de bureau, il est possible de reconnaître un objet *table* ayant différentes caractéristiques (*nb-pieds, forme, couleur, ...*)

Les opérations *déplacer la table, poser un livre sur la table, ...* n'affectent pas la relation *table* avec ses attributs.

Les traitements consistent d'une part à parcourir les relations, d'autre part à affecter, modifier, ou détruire les valeurs des éléments du modèle.

Pour faciliter la phase de conception, on utilise une représentation graphique à structure de graphe où les

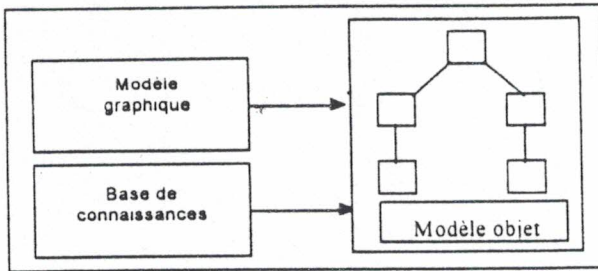


Fig. 2. Processus de traduction en modèle objet

Les fonctions de visualisation sont décrites à l'aide d'un autre modèleur. Elles doivent cependant accéder aux objets déjà définis.

Les fonctions seront intégrées dans l'architecture issu du traducteur : ajout de méthodes sur des classes issues d'un modèle, nouvelles classes d'objets et méthodes associées.

Les règles de traduction dans le modèle objets sont :

- Un concept correspond à une classe dans le modèle objet. Les propriétés agrégées à un concept correspondent aux variables d'instances de la classe associé au concept.
- La relation de généralisation entre deux concepts se traduit par l'héritage dans le modèle objets.
- Chaque contrainte est traduite en tant que méthode de classe d'une classe générique unique appelée *contrainte*. Une classe générique correspond à une classe qui n'a jamais d'instances. Elle n'existe pas directement dans le monde réel, seules ses sous classes possèdent des instances. Cette classe possède uniquement des méthodes de classe implémentant le code à exécuter pour vérifier chacune des contraintes.
- Chaque agrégation est traduite par la mise en place d'une dépendance entre les deux classes concernées.

6. L'Interface Graphique

L'interface graphique doit disposer d'un ensemble de moyens d'entrée, telle que des menus et des boutons qui servent à activer les commandes, des sélecteurs de fichiers, un sélecteur de texture, un sélecteur de direction, etc. Elle propose, de plus, plusieurs vues de l'environnement sous différents angles (voir figure 3) ainsi que des boîtes de dialogues pour communiquer avec les utilisateurs.

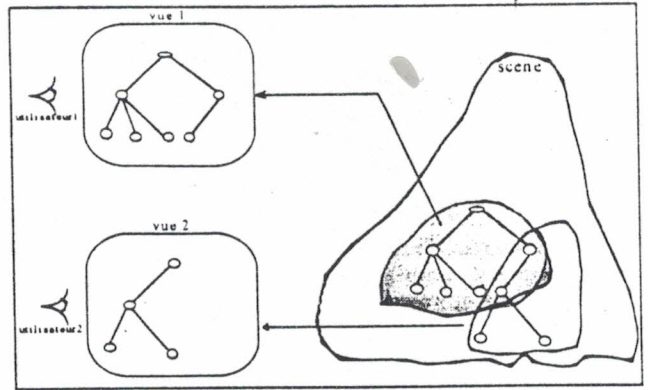


Fig. 3. Schéma de deux utilisateurs ayant deux vues adjacentes.

L'interface graphique permet l'accès à tous les objets de l'environnement par l'intermédiaire de commandes de haut niveau d'abstraction et des attributs flous. Le résultat étant la visualisation de l'environnement demandé.

Il s'agit de la description des objets de l'environnement et les liens qui les unissent. Un environnement est vue comme un objet composé, constitué de plusieurs objets reliés entre eux par des liaisons définissant ainsi une structure hiérarchique.

Des liens dynamiques se créent au moment de l'entrée en contact de deux objets (ex : *poser une bouteille sur une table*). Ils permettent l'interaction de l'utilisateur avec l'environnement de façon réaliste et cohérente.

On pourra imposer des contraintes sur les relations entre les objets, de sorte que le mouvement de la table implique celui de la bouteille. Ultérieurement, on pensera à introduire les lois physiques dans les liens pour assurer une interaction plus réaliste. La bouteille tomber si la table est mal prise en mouvement [5].

7. Construction Coopérative de l'Environnement

Les possibilités d'interaction au sein d'un groupe d'utilisateurs permettent aux individus qui le composent de corréler leurs comportements de manière adaptative pour faire face à des situations diverses. Cette corrélation fonctionnelle implique l'existence d'une cause commune : le problème à résoudre. La conception coopérative est un processus cyclique, qui s'arrête lors le but est atteint et qui passe par les étapes suivantes :

- Décomposition du problème initial en un ensemble de tâche;
- Distribution des tâches sur les membres du groupe;
- Résolution individuelle des tâches;
- Intégration des solutions partielles;

Cette façon de voir le problème permet une grande flexibilité, car aucune solution n'est imposée; les modifications spatio-temporelles de l'environnement entraînent au retour des modifications du comportement du groupe qui peut ainsi s'adapter au nouvel environnement. Elle permet aussi une plus grande robustesse, car l'erreur individuelle n'est pas exclue et devient au contraire porteuse de solutions nouvelles pour le groupe en lui permettant de se sortir de configuration pièges. Le mécanisme de rétroaction permettent la coordination des comportements individuels.

Par ailleurs, et pour une meilleure coordination, un utilisateur doit pouvoir "visualiser" son état par rapport à la tâche à réaliser par le groupe, ce qui lui permet de prévoir ses actions prochaines. Cela permet également d'informer les utilisateurs de leur état et des tâches en attente. Le couplage entre l'état du groupe et celle de l'environnement s'exprime explicitement pour un usager par une règle du type :

SI état des individus voisins

ET état de l'environnement

ALORS action à prendre

7. Conclusion

Nous avons présenté dans cet article le formalisme basé sur les objets et les réseaux sémantiques pour la modélisation d'un environnement de conception. Notre souci du départ était de structurer l'univers de l'application, rassembler l'ensemble des éléments de connaissances relatifs à un concept, le couplage des aspects déclaratifs et procéduraux et la hiérarchisation des connaissances et des raisonnements.

Notre objectif à terme est de permettre à deux ou plusieurs collaborateurs travailler virtuellement sur le même environnement et pouvoir ainsi y effectuer un réel travail coopératif de conception. L'interaction coopérative intervient lors de la construction de l'environnement et durant la phase de simulation [1, 6, 7].

L'autre but envisagé est de décrire l'environnement en langage naturel et de permettre une interaction multimodale combinant geste et reconnaissance de la parole pour sa construction. Les mouvements de l'objet pourront se faire ainsi par les commandes de l'interface graphique, par un énoncé en langage naturel et/ou en combinant langage naturel parlé et geste de désignation [8].

Références

- [1] Brahim Belattar, Zidani Abdelmadjid, Mahieddine Djoudi, Vers un environnement de simulation coopératif, MOSIM'97, Conférence Francophone de Modélisation et Simulation, Rouen, 5-6 juin 1997.

- [2] R. Courdier, D. Herin-Aim, R. Galeia, Une démarche et un modèle de conception à base d'objets et de réseaux sémantiques, TSI, Vol 12, N°3, 1993, pages 285-318
- [3] D. Plemenos, Contribution à l'étude et au développement des techniques de modélisation, génération et visualisation de scènes. Le projet Multiformes. Thèse de doctorat d'état, Université de Nantes, Novembre 1991.
- [4] J. M. Vlissides and M. A. Linton, Applying Object-Oriented Design to structured Graphics, Proceedings of the USENIX C++ Conference. Denver, Colorado, October 1988.
- [5] G. Teege, HieraStates : Flexible Interaction with Objects. Institut Fur Informatik, Technische Universität München, TUM 19441, Dezember 1994.
- [6] Mark Roseman & Saul Greenberg, Building Real Time Groupware with Groupkit, a Groupware Toolkit. Rapport interne, Univ. Calgary Alberta, Canada. 1995.
- [7] Q. Wang, Networked Virtual Reality, Master of Science, University of Alberta, 1994.
- [8] N. Vigouroux, N. Gaildrat, R. Caubet et G. Perennou, Une architecture d'interface pour l'interprétation multimodale : application à un modèleur déclaratif de scènes. 4^{ème} journées sur l'ingénierie des interfaces homme-machine, Paris, 1992.