

# vers un environnement de simulation coopératif

Belattar Brahim\* , Zidani abdelmadjid\* , Djoudi Mahieddine \*\*

(\*) Institut d'Informatique, Université de Batna, Rte Boukhrouf Med El Hadi, 05000 Batna -Algérie  
Tél. (+213) - 4- 85- 00 - 36 Fax. (+213) - 4- 85- 83 - 75

(\*\*) Laboratoire d'Informatique de l'Université de Poitiers, UFR Sciences Fondamentales et Appliquées,  
40 Avenue du Recteur Pineau , 86022 Poitiers Cedex - France

---

**Résumé :** *L'objectif de notre article est d'analyser les possibilités d'intégration des concepts du Travail Coopératif (T.C.A.O.) dans le domaine de la simulation. Un bilan des recherches qui prédominent ces dernières années dans le domaine de la simulation est fait en premier. Ceci nous permet de mettre en évidence l'originalité de notre travail. Les concepts de base du travail coopératif sont introduits. Nous montrons l'adéquation de ces concepts avec la méthode de conduite d'un projet de simulation. Nous proposons des idées directrices pour la conception d'un environnement de simulation basé sur la coopération (ESC) en mettant en évidence les niveaux où la coopération peut intervenir. Les intérêts pratiques d'un tel environnement seront mis en évidence sur un exemple.*

**Mots clés :** *Simulation, Coopération, T.C.A.O., Réseaux informatiques, Interface H/M multi-utilisateurs.*

**Abstract :** *This paper analyze the possibilities of integrating Computer Support Cooperative Work (CSCW) concepts in simulation environments. A review of important researches ongoing in simulation area are made. Motivations of our researches are justified and originality of our approach is highlighted. We propose a basic ideas to design a new generation of simulation environments based essentially on cooperative work. Practical benefits of such environments especially in the model construction phase are pointed out.*

**Keywords :** *Simulation, Cooperation, CSCW, Computer Networks, Multi-users Man/Machine Interface.*

---

## 1. Introduction

Les travaux de recherches et de développement qui ont été consacrés durant ces dernières années au Logiciel de Simulation (L.S.) sont très impressionnants. Ceci a été motivé d'une part, par une clientèle plus exigeante en matière de qualité du logiciel et d'autre part, par une concurrence très serrée sur le marché de ce type de logiciel et pour lequel soulignons-le, le secteur industriel représente une part très importante sinon la plus importante.

Par ailleurs, les L.S. actuellement disponibles sont à notre connaissance, tous mono-utilisateur même ceux intégrant des techniques informatiques telles que : I.A., A.O.O, Systèmes Experts, Parallélisme, Distribution, etc.. Ceci est dû en grande partie au fait que ces logiciels sont développés avec l'objectif principal de satisfaire le plus grand nombre de clients possible afin d'en tirer le maximum de profit. Cette politique de développement a conduit à une multitude de produits (L.S.) plus sophistiqués les uns que les autres notamment les versions sur P.C. ou station de travail. Certaines sociétés de développement vont même jusqu'à proposer toute une gamme de produits dont chacun est orienté vers un domaine d'application précis (FMS, AGV, etc.). C'est le cas par exemple de la société Pritsker & Associates qui commercialise SLAM II, TESS, MAP/1, FACTOR/AIM, SLAMSYSTEM, pour ne citer

que ceux là.

Aujourd'hui, la tendance est plus que jamais à la décentralisation de l'informatique : c'est l'ère des stations de travail, des micro-ordinateurs, des portables et des ordinateurs reliés par un réseau informatique. La solution qui a consisté par le passé à investir des sommes importantes dans le développement d'applications typiquement centralisées et mono-utilisateur risque d'être écartée dans un très proche avenir. La communauté scientifique a été sensible à cette tendance et beaucoup d'équipes de recherches s'investissent déjà dans un domaine nouveau en informatique : le Travail Coopératif Assisté par Ordinateur (T.C.A.O.) ou CSCW (Computer-supported cooperative work). Certaines applications qui par le passé étaient proposées sous forme d'applications mono-utilisateur (Editeur de texte, Outil de C.A.O.) existent aujourd'hui sous forme d'application à l'usage d'un groupe (Groupware Application). La prise en compte de la notion de groupe dans la conception d'une application a été grandement favorisée par les possibilités de connexion via un réseau informatique d'un certain nombre d'individus devant coopérer pour la réalisation d'un projet commun.

Notre objectif est donc l'analyse des possibilités d'intégration des concepts du travail coopératif dans un environnement de simulation. Nous nous intéresserons dans la suite de notre article, uniquement aux modèles de simulation à événements discrets et à leur utilisation dans le domaine des systèmes de production

## **2. Bilan des recherches en cours**

Les recherches qui ont prédominé ces dix dernières années dans le domaine de la simulation concernent essentiellement l'utilisation des techniques d'intelligence artificielle, de programmation orientée objet et de systèmes informatiques distribués et dont nous allons synthétiser les orientations principales.

### **2.1 I.A. et simulation**

L'utilisation des techniques d'I.A., notamment les systèmes experts dans le domaine de la simulation s'est faite selon deux approches. La première a consisté à adjoindre à un L.S. classique des composantes intelligentes dont chacune a pour but soit d'automatiser une des phases du cycle de vie du modèle de simulation (ex: génération automatique du modèle), soit de guider l'utilisateur dans la réalisation de celle-ci (ex: Analyse des résultats, validation du modèle, etc.). Une composante intelligente n'est autre qu'un système expert utilisant une ou plusieurs bases de connaissances et opérant par couplage avec le simulateur. Ce couplage peut se faire de différentes façons dont les principales sont présentées dans [O'KE 86]. L'avantage de cette approche est sa facilité de mise en oeuvre par contre son inconvénient réside dans le fait que l'utilisateur doit effectuer manuellement toutes les autres étapes qui ne sont pas prises en charge par la composante intelligente. Nous pouvons citer par exemple les travaux de [SAR 86].

La deuxième approche a consisté à intégrer dans un même environnement toutes les phases du cycle de vie du modèle qui peuvent alors être vues comme entièrement à la charge de cet environnement. Par exemple, le travail de modélisation se résume souvent à une simple description via une interface graphique du système à simuler. La traduction de cette description en un programme exprimé dans un langage de simulation cible est transparente à l'utilisateur. Souvent, ce dernier n'a même pas accès au code source du programme obtenu. Plus récemment, une approche d'intégration dans un même environnement des concepts de l'A.O.O., des systèmes experts et des modèles de simulation à événements discrets a été adoptée par [Zei 96]. L'un des points intéressants de cette approche est la possibilité de définir un système expert comme une composante du modèle de simulation lui-même.

### **2.2 A.O.O. et simulation**

L'approche orientée objet a été utilisée de façon intensive ces dernières années dans le domaine de la

simulation. Les principaux travaux se sont surtout intéressés à son utilisation dans la modélisation et la simulation des systèmes à événements discrets qui se prêtent bien à une modélisation par “ *objets* ”.

Parmi les travaux s'inscrivant dans cette perspective, nous pouvons citer l'environnement IMDE (Integrated Model Development Environment) [Llo 94] qui offre une interface graphique pour la construction d'un modèle et génère en sortie du code source en MODSIM II ou C++. Le langage MODSIM II est un langage de simulation purement orienté objet dont le projet de développement a été initié par l'US Army. IMDE peut être utilisé dans divers domaines d'application parmi lesquels les réseaux de communication, des systèmes de production ou des systèmes logistiques. C'est le seul projet de recherches qui à notre connaissance prend en compte la notion de groupe et d'interface multi-utilisateurs. La persistance des objets d'un modèle de simulation est assurée grâce à l'utilisation d'une base de données orientée objet à laquelle ont accès les utilisateurs et qui leur offre la possibilité de *coopérer* pour construire un modèle de simulation.

Une autre approche qui ne manque pas aussi d'intérêts a consisté à intégrer les concepts de l'A.O.O. et ceux des systèmes à règles de production dans un même environnement. Le système Extool [Ata 94] est un exemple d'environnement de simulation basé sur cette approche. A chaque objet du modèle de simulation, on peut associer une base de règles de production qui seront utilisées à des fins de prise de décision au cours de la simulation. L'intérêt pratique de cette approche est la possibilité de doter les objets du modèle d'un mécanisme de raisonnement qui leur permet au fur et à mesure de l'évolution de la simulation de participer au processus de prise de décision sur l'état courant du système simulé.

### 2.3 systèmes distribués et simulation

Un système informatique distribué est en général caractérisé par un ensemble de processus s'exécutant sur des ordinateurs connectés par un réseau de communication et par le biais duquel ces processus s'échangent de l'information. Les techniques de modélisation et d'implémentation qui ont été utilisées dans ce domaine ont été largement reprises dans le domaine de la simulation en vue de réduire principalement les temps d'exécution d'une simulation.

Un modèle de simulation distribué peut donc être vu comme un ensemble de processus logiques (PL) s'exécutant de manière concurrente et modélisant chacun un composant physique du système réel modélisé ou processus physique (PP)[Cha 94]. Les interactions entre les PP sont modélisées par l'envoi de messages datés entre les PL correspondants.

Une autre approche de modélisation utilisant conjointement les concepts de l'A.O.O. et de l'informatique distribuée permet de voir un modèle de simulation comme un ensemble d'objets distribués sur un ensemble d'ordinateurs reliés par un réseau et communiquant par envoi de messages. Chaque objet sert dans ce cas à modéliser un composant du système réel. L'exécution d'un tel modèle de simulation consiste à invoquer les méthodes associées aux objets selon une séquence basée sur le temps. Du fait que les objets ne sont pas liés dans un même programme exécutable, ceux-ci invoquent les méthodes par des envois de messages explicites.

Parmi les projets les plus importants que nous avons recensés dans le domaine de la simulation distribuée nous avons retenu celui du développement d'un environnement de simulation distribuée et interactif ou D.I.S. (Distributed Interactive Simulation) initié par le département Américain de la défense (DoD). L'objectif principal du projet est l'interconnexion de différents simulateurs appartenant au DoD (simulateur de chars, d'avions, de détecteurs, etc.) en vue de simuler différents scénarios de combats [Wie 94]. L'intégration des modèles environnementaux est aussi possible et permet de prendre en compte une certaine réalité dans laquelle évoluent les différentes entités. Il faut souligner que le protocole d'interconnexion qui a été défini dans le cadre de ce projet a été adopté comme un standard par IEEE.

### 2.4 impact de la recherche sur le L.S.

Il y a juste trois ans, [Mac 94] avait mené une étude en vue d'établir une liste de critères dont souhaitaient disposer les utilisateurs au sein du L.S. Les résultats de cette étude sont à notre avis très significatifs vu la notoriété des sociétés qui avaient participé à cette étude (IBM, Pritsker & Associates, CACI, Xerox, Aerospace, etc.) et dont certaines ont une grande expérience dans le développement du L.S. et d'autres comme utilisatrices de ce type de logiciel. Les dix critères qui ont été retenus par cette étude et classés par ordre de priorité sont les suivants :

Ordre	Critère
1	Interface H/M conviviale
2	Possibilité de disposer d'une B.D. pour organiser les données
3	Outil de mise au point (Debugueur)
4	Interaction Via une Souris
5	Section d'assistance dans la documentation du L.S.
6	Sauvegarde des modèles de simulation
7	Sauvegarde des Résultats
8	Entrée des données et des commandes par clavier
9	Librairie de modules réutilisables
10	Affichage graphique des données et Résultats Statistiques

**Figure 1 : Liste des critères souhaités par les utilisateurs (d'après [Mac 94])**

Ces critères sont surtout destinés aux équipes de développement qui en veillant à leur intégration dans le L.S. garantirait son succès auprès d'un large éventail d'utilisateurs. L'idéal serait donc de disposer d'un L.S. qui offrirait toutes ces possibilités à la fois. Cependant, il faut noter que dans la pratique, la prise en compte de trop de critères augmente les coûts de production du logiciel et compliquent sa réalisation. C'est pour cela qu'en général, les équipes de développement se contentent de satisfaire certains critères au détriment de certains autres. Néanmoins, le constat aujourd'hui est que parmi les dix critères retenus par l'étude, seuls certains critères (ex : 2 , 3 et 9 ) ne sont pas encore largement implémentés par les L.S alors que les autres si.

Cependant si nous essayons de comparer les résultats de cette étude avec ceux obtenus en 1979 par Tuncer I. Oren et Bernard P. Zeigler [Ore 79] nous constatons qu'à peu de choses près les critères souhaités par les utilisateurs ont un lien direct avec les lacunes présentes dans les L.S. et mises en évidence par ces deux chercheurs. La question qui vient donc directement à l'esprit est celle de savoir si nous devons nous contenter aujourd'hui de continuer à s'investir dans des recherches avec le seul objectif de remédier à des lacunes mises en évidence voilà 15 ans et de répondre ainsi aux besoins des utilisateurs ; ou de repenser plutôt l'architecture du L.S. en prenant en compte les possibilités offertes par les technologies actuelles en vue de rendre possible le travail en groupe qui est prépondérant dans tout projet de simulation comme nous le verrons plus loin et qui n'est pas encore pris en compte par les L.S.

Nous sommes convaincus que c'est la deuxième solution qu'il faudrait expérimenter car si le but essentiel des équipes de développement était de produire des L.S. suffisamment conviviaux pour populariser la simulation, nous estimons que cet objectif est largement atteint aujourd'hui. Maintenant, les nouvelles tendances de l'informatique nous poussent à adopter de nouvelles approches de sorte à répondre aux exigences d'une nouvelle génération d'utilisateurs finaux (End Users) plus aguerris et mieux préparés.

### 3. Simulation et travail coopératif

#### 3.1 notion de groupe dans une simulation

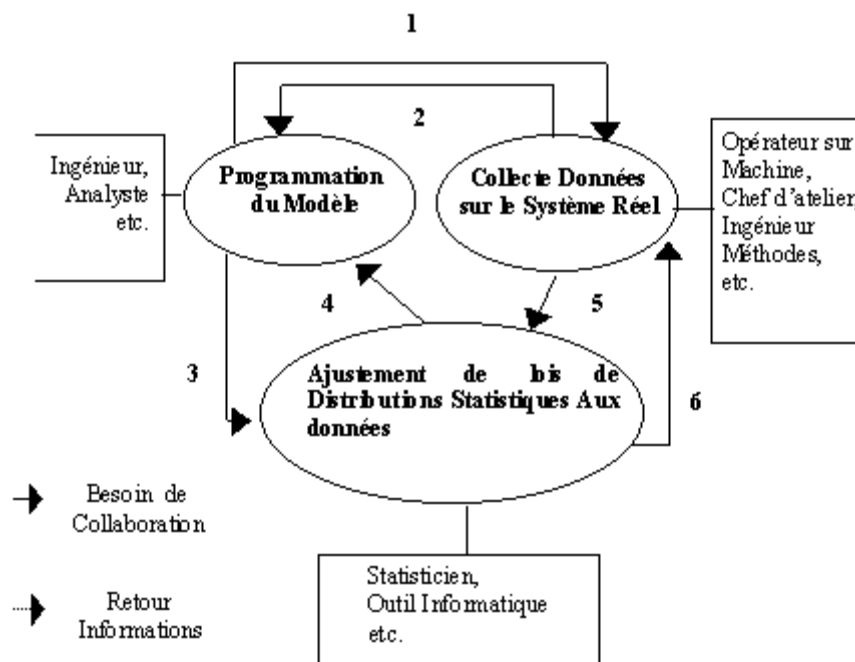
D'un point de vue externe, on a souvent tendance à voire en l'ingénieur ou l'analyste le seul acteur d'un projet de simulation surtout si celui-ci maîtrise bien le logiciel de simulation qui lui sert de support pour la programmation du modèle. Or, chacun sait que sur le plan pratique, beaucoup de personnes seront sollicitées pour alimenter le projet en données ou en expertises afin qu'il aboutisse (statisticien, opérateur sur machine, chef d'atelier, ingénieur méthodes, programmeur, client de l'étude, etc.). De plus, il est communément reconnu que pour pouvoir utiliser correctement et intelligemment des méthodes de simulation, il faut disposer de connaissances plus ou moins solides dans des domaines variés (Probabilités et Statistiques, Modélisation, Programmation, etc.). Malheureusement, il est souvent rare pour ne pas dire impossible de trouver au sein de l'entreprise un cadre possédant à lui seul toutes les connaissances requises pour pouvoir mener à terme un projet de simulation sans avoir recours aux connaissances d'un certain nombre de collaborateurs internes ou externes à son entreprise. Tout projet de simulation implique donc de près ou de loin un groupe de personnes (structuré ou non en équipe) chacune jouant un plusieurs rôles dans le projet. Par conséquent, un projet de simulation quelle que soit son envergure, ne peut et ne pourra jamais être considéré comme l'oeuvre d'une personne isolée et ce même si les attributions de rôle dans le projet ne sont pas faites de manière explicite.

Les L.S. actuellement proposés (même sous forme d'environnements intégrés) offrent une interface H/M mono-utilisateur qui ne favorise pas le travail de groupe. Ceci veut dire que les phases du cycle de vie du modèle de simulation nécessitant le concours de plusieurs personnes, sont réalisées sans le support du L.S. Il en est de même si on se concentre sur une tâche précise du projet telle que la construction du modèle ou l'analyse des résultats de la simulation avec les possibilités de coopération qu'elles induisent et qu'aucun L.S. ne supporte.

### 3.2 Niveaux de coopération dans une simulation

Un projet de simulation peut être vue comme la réalisation d'un ensemble de tâches chacune d'elles nécessitant des compétences particulières et dont certaines ne peuvent avoir lieu que si d'autres ont été achevées. Par exemple, l'analyse des résultats ne peut se faire qu'après avoir programmé et exécuté le modèle de simulation. De même que la programmation du modèle (codage) nécessitera des données sur le système réel modélisé qui serviront après transformation, de fixer les paramètres de ce programme. La réalisation de l'ensemble de ces tâches par un groupe ne peut donc se faire que sur la base d'un travail basé sur la coordination des tâches et sur la collaboration et la coopération entre les membres.

La figure 2 montre les interactions entre quelques tâches faisant partie de tout projet de simulation et les besoins en collaboration qu'elles induisent dans la pratique. L'acteur chargé de la tâche de programmation du modèle de simulation exprime le besoin de disposer des données caractérisant le système réel afin de compléter le modèle (Flèche N° 1). Si les données sollicitées sont disponibles, elles seront fournies à la partie qui en a fait la demande. Dans le cas contraire, l'acteur chargé de la tâche de collecte des données déclenche les procédés adéquats pour les rassembler puis les mettre à la disposition de cette même partie (Flèche N° 2). Une fois les données disponibles, l'acteur chargé de la tâche de programmation du modèle exprime le besoin d'ajuster ces données par des lois de distributions statistiques pour pouvoir les intégrer dans le modèle (Flèche 3). L'acteur chargé de la tâche d'ajustement des données par des lois de distributions statistiques reçoit donc des données brutes et produit au retour des distributions statistiques avec pour chacune les paramètres qui la caractérisent en tenant compte des possibilités offertes par le L.S. servant à la programmation du modèle (Flèche 4). Il peut à tout moment solliciter la collaboration de l'acteur chargé de la collecte des données sur le système réel surtout pour le cas de données mal formulées ou incomplètes (Flèches 5 et 6).



**Figure 2 : Exemples de collaboration entre tâches**

## 4. Concepts de base du travail coopératif

Une activité basée sur la coopération repose essentiellement sur l'articulation d'actions individuelles et d'actions collectives au sein d'un groupe et visant à réaliser des buts communs qui ne correspondent pas nécessairement aux buts individuels. Une *coopération* implique une division des responsabilités entre les acteurs pour l'exécution d'une tâche et la volonté d'adhérer aux buts de l'autre. Elle doit donc être distinguée d'une *collaboration* qui repose sur une action conjointe des acteurs et l'entretien d'une compréhension mutuelle de la tâche à accomplir mais chaque acteur ayant ses propres objectifs [Bre 92]. La conception et la réalisation d'une application basée sur le travail coopératif posent encore de nombreux problèmes. Ceci est dû en grande partie à un manque de méthodes d'analyse de l'activité coopérative et de formalismes permettant de la décrire [JAM 94]. Dans ce type d'applications, il devient nécessaire que des outils soient mis à la disposition des membres du groupe pour leur permettre de structurer leur travail ainsi que le fonctionnement de leur groupe de façon à aider véritablement le travail coopératif. Ceci ne peut être atteint que grâce à un environnement informatique plus riche que dans le cas d'une application centralisée et mono-utilisateur du fait qu'il doit supporter plusieurs canaux de communication simultanément. Techniquement, cet environnement repose en général sur un réseau Audio/Vidéo piloté par des moyens informatiques dans lequel chaque utilisateur dispose d'une station de travail, d'un moniteur vidéo, d'une caméra, de hauts parleurs et d'un microphone. A l'aide de logiciels adaptés, tout utilisateur peut ainsi établir une connexion avec n'importe quel autre utilisateur du groupe [Sal 93].

## 5. Vers un environnement de simulation basé sur la coopération (ESC)

### 5.1 Objectifs visés par un ESC

Comme toute application de groupe, l'objectif d'un ESC sera avant tout de permettre à un groupe d'utilisateurs de réaliser une ou plusieurs tâches d'un projet de simulation (construction du modèle,

programmation, analyse des résultats, etc.) au moyen d'un système informatique qui leur garantit la coopération. Un tel environnement repose donc avant tout sur un système de communication Homme-Machine-Homme pour lequel des choix de conception doivent être faits à différents niveaux et que nous n'aborderons pas ici (voir [Sal 94] pour plus de détails sur ce sujet).

Ainsi, le succès d'un ESC passe par une bonne connaissance de l'organisation de l'équipe (ou groupe), de l'organisation des tâches à accomplir et des moyens qui seront mis à la disposition de l'équipe. Une équipe bien organisée facilite la communication et le partage de l'information entre les différents membres conduisant ainsi à une coordination des tâches plus efficace. Parmi les facteurs à prendre en compte lors de la conception nous avons :

1) Type de structure de contrôle : il s'agit de décider entre une structure de contrôle hiérarchique ou bien plate dans laquelle contrairement à la structure hiérarchique, les membres ont le même pouvoir de décision. Ainsi, on pourra spécifier clairement qui doit rendre compte à qui et quelles décisions doivent être prises et à quel niveau de la hiérarchie.

2) structure de la communication : Il s'agit de définir les rôles de chaque membre et de spécifier qui peut communiquer avec qui et à quel sujet.

3) support de la communication : il s'agit d'identifier les outils à utiliser pour assurer la communication entre les membres. La diversité des supports peut aller d'un simple courrier électronique (E-mail) jusqu'au canal audio.

4) Organisation des tâches : un projet de simulation est par nature un ensemble de tâches pour lesquelles il faudra identifier des responsabilités et déterminer les interactions qui peuvent exister entre elles et leurs effets sur les relations entre les responsables.

## 5.2 organisation des tâches dans un ESC

La structure de contrôle la plus adéquate pour notre ESC est une structure hiérarchique à deux niveaux dans laquelle un utilisateur privilégié joue le rôle de chef de projet et les autres utilisateurs se verront attribuer des rôles par le chef de projet selon leurs compétences. Pour cela, il est important de préciser la granularité de l'information que les membres peuvent manipuler et qui sera une caractéristique de notre ESC. Si on se place au niveau de la tâche de construction d'un modèle de simulation, nous pouvons prendre le modèle entier comme granule, ou considérer que le modèle peut être composé à partir de sous-modèles qui peuvent servir alors de granules. Une troisième possibilité qui n'a pas d'intérêt pratique apparent est celle de considérer comme granule le composant élémentaire du modèle (pièce, machine, file d'attente, etc.).

La technique d'attribution de rôles que nous avons retenu est largement inspirée de celle d'octroi et de révocation de privilèges utilisée dans le domaine des bases de données et qui s'exprime selon le schéma : <GRANT *liste\_de\_privilèges* ON *li\_d'objets* TO *liste\_utilisateurs*>. Nous proposons d'enrichir ce schéma par une clause WITH qui permet de spécifier la liste des utilisateurs avec lesquels il est possible de coopérer pour la réalisation d'une tâche. La figure suivante donne quelques exemples d'utilisation de cette technique dans le cadre de notre ESC.

<b>GRANT</b> <i>Créer, Modifier</i> <b>ON</b> <i>Atelier3.mod</i> <b>TO</b> <i>Jean</i> <b>WITH</b> <i>Robert, Jacques</i> (a)	<b>GRANT</b> <i>Créer, Modifier</i> <b>ON</b> <i>Atelier2.mod</i> <b>TO</b> <i>Paul</i> <b>WITH</b> <i>NONE</i> (b)
<b>GRANT</b> <i>Voir-ET-Question</i> <b>ON</b> <i>Atelier2.mod</i> <b>TO</b> <i>Pascal, Robert</i> <b>WITH</b> <i>ALL</i> (c)	<b>GRANT</b> <i>Créer, Modifier</i> <b>ON</b> <i>Atelier1.mod</i> <b>TO</b> <i>René</i> <b>WITH</b> <i>NONE</i> (d)

**Figure 3 : Technique d'attribution de rôles adoptée par l'ESC**

Le schéma (a) donne à Jean les privilèges de Créer et Modifier le modèle de simulation ayant comme nom Atelier3.mod en pouvant coopérer avec Robert et Jacques. Le schéma (b) donne à Paul les privilèges de créer et Modifier tout seul le modèle de nom Atelier2.mod (pas de possibilité de coopération). Le schéma (c) donne à Pascal et Robert les privilèges de voir et questionner tous les autres membres du groupe travaillant sur le modèle Atelier2.mod (parmi lesquels Paul). Le schéma (d) donne à René le privilège de créer et modifier le modèle Atelier1.mod seul.

On voit qu'avec notre solution, il devient facile de définir tous les scénarios de coopération imaginables. Pour retirer un privilège ou une liste de privilèges à un membre, il suffit de lancer une commande de révocation (REVOKE) ayant le même schéma que GRANT mais sans clause WITH. Nous remarquons aussi qu'il est possible d'attribuer à un utilisateur plusieurs rôles s'il est jugé qualifié pour travailler sur plusieurs tâches différentes.

L'octroi et la révocation de privilèges est la responsabilité du chef de projet qui possède quant à lui tous les privilèges. Cette approche est donc en pleine adéquation avec la structure de contrôle hiérarchique que nous avons retenue pour l'ESC

### 5.3 Choix du type d'interface pour un ESC

Les interfaces H/M (IHM) à base de fenêtres exploitant la métaphore de bureau se sont imposés avec les environnements tels que X-Windows sur station de travail et MicroSoft Windows sur PC. Actuellement, de nombreux travaux sont en cours en vue de définir de nouvelles métaphores qui permettront de construire des IHM plus performantes orientées surtout vers les applications de groupe [Chal 94]. La métaphore de bureau (2D) a tendance à être substituée par la métaphore de pièce ou Room (3D) pour élargir l'espace de travail et ainsi augmenter la densité de l'information. Une autre approche qui s'éloigne radicalement de l'utilisation bureautique de l'informatique est celle utilisant la réalité virtuelle et avec laquelle l'écran est remplacé par un casque et de nouveaux outils d'interaction entre l'utilisateur et la machine sont introduits (ex: le Gant).

Compte tenu des critères de la figure 1 (§ 2.4), nous pensons que dans le cas d'un ESC, une IHM basée sur la métaphore de bureau qui est aujourd'hui rentrée dans les moeurs, est largement suffisante.

La prise en compte de la configuration matérielle dont dispose chaque membre permet à l'ESC d'associer une interface graphique (IG) propre à chaque membre. Par exemple, un membre n'ayant pas de matériel pour exploiter le son, ne verra pas au niveau de son interface les icônes permettant d'activer les outils de traitement du son. De même que la prise en compte des privilèges permet d'activer ou de désactiver certaines icônes de l'IG correspondant aux activités associées à des tâches pour lesquelles un membre n'a pas reçu de privilèges. Par exemple, un membre qui n'a pas eu le privilège d'exécuter une activité de la tâche de construction d'un modèle de simulation, ne verra pas au niveau de son IG les icônes représentant les outils de construction d'un modèle de simulation (Editeur graphique de modèles,



etc.).

Grâce à la technique d'octroi de privilèges que nous adoptons, il devient facile d'enrichir la structure <GRANT ... ON ... TO .... WITH ....> avec une nouvelle clause USING <Liste\_de\_Matériels> afin de spécifier la configuration matérielle que peut utiliser un membre durant un travail coopératif impliquant d'autres membres spécifiés dans la clause WITH de la même structure.

#### **5. 4 Choix d'une politique d'interaction avec l'ESC**

Dans les applications de groupe en général, il est nécessaire d'identifier les modes de coopération selon lesquels les membres pourraient interagir avec l'application et communiquer ainsi entre eux. Un ESC doit être capable d'assurer trois types de communications qui sont :

1- une communication H/M avec les applications classiques utilisées localement par chaque membre (traitement de texte, tableur, etc.). Parmi ces applications figure nécessairement le logiciel de simulation servant à la programmation des modèles et à leur exécution surtout dans le cas des utilisateurs ayant des rôles dans les tâches de construction du modèle et de programmation.

2- une communication H/M qui met en jeu la coopération et dont le but essentiel est la manipulation coordonnée des objets(modèles, sous-modèles) partagés par les membres du groupe et ce dans un vrai mode multi-utilisateurs.

3- une communication H/H qui met en jeu l'échange de messages (textuels, graphiques, sonores) soit en temps réel soit en temps différé (E-mail, téléphone, Télécopie, etc.)

Concernant la communication H/M de type 2, celle-ci peut se faire selon différents modes de coopération. Si on se limite à la tâche de construction d'un modèle de simulation et en tenant compte de la structure de contrôle hiérarchique retenue et de la technique d'octroi et de révocation de privilèges que nous avons proposée, il devient possible d'identifier les modes de coopération suivants qui sont inspirés de [Gro 94] :

*1- Travailler séparément* : c'est le mode normal qui n'offre aucune possibilité de coopération à l'intérieur d'une tâche donnée. Chaque utilisateur travaille séparément mais il n'est pas exclu qu'un utilisateur puisse utiliser un travail fait par un autre (modèle, sous-modèle, etc.)

*2- Versions alternatives* : plusieurs membres peuvent développer différentes versions d'un même modèle (ou sous-modèle). Le choix de la version à retenir sera à la charge de la structure de contrôle (i.e. chef de projet).

*3- A tour de rôle* : un seul membre peut modifier à un instant donné un modèle ou un sous-modèle de simulation. La différence avec le mode (1) est qu'ici un membre peut obtenir le droit de travailler sur un modèle ou un sous-modèle sans attendre que l'utilisateur qui était affecté à ce travail ne termine.

*4- Échange dynamique* : deux membres peuvent s'échanger dynamiquement un modèle ou une partie d'un modèle transférant ainsi le droit d'accès d'un membre vers un autre.

*5- Travail synchrone* : au niveau de la coopération, c'est le mode le plus intéressant mais c'est aussi le plus difficile à mettre en oeuvre. Les modifications faites par un membre sont immédiatement rendues visibles à tout les membres en coopération avec lui (principe du WYSIWIS).

L'intégration de tous ces modes au sein de l'ESC n'est pas chose aisée car il faudra nécessairement gérer plusieurs fenêtres d'interactions qui engendrent des problèmes de gestion de la concurrence et de synchronisation, ce qui laisse deviner la complexité de réalisation d'un tel environnement. Néanmoins, nous sommes rassuré sur ce plan car des boîtes à outils pour le développement d'applications de groupe

commencent à être diffusées (ex: GroupKit [Ros 93]), ce qui ne peut que nous encourager à poursuivre sur cette voie.

## 5.5 exploitation du son dans un ESC

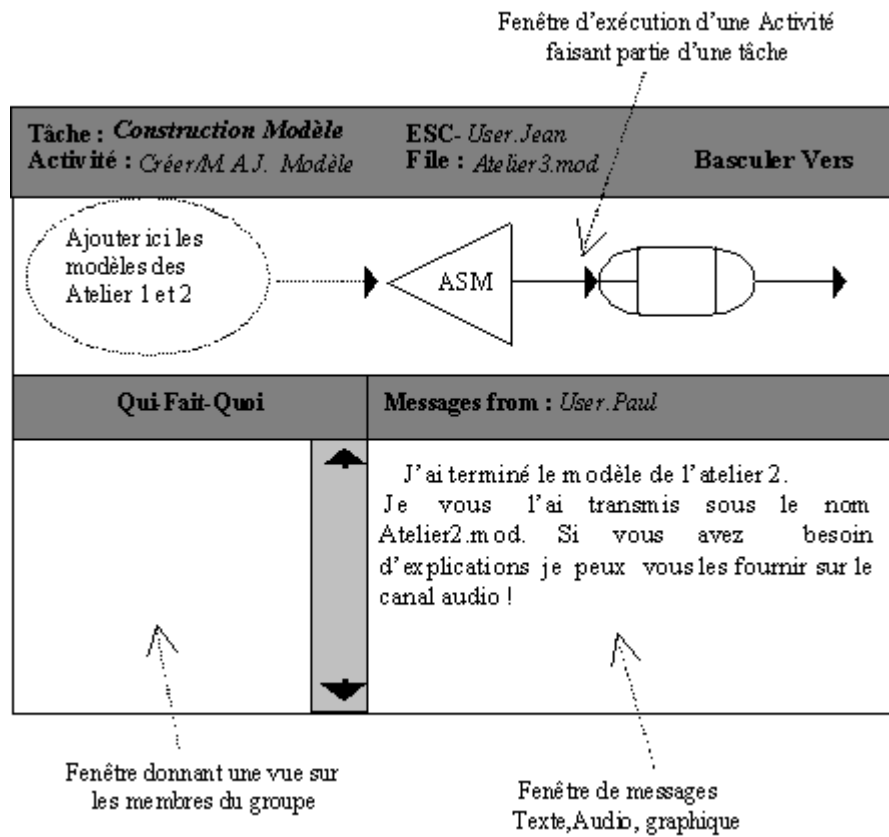
L'exploitation du canal audio dans les interfaces H/M fait l'objet de recherches intensives. D'un point de vue technique, on peut remarquer que la plupart des constructeurs livrent maintenant du matériel et du logiciel pour le son [tro 93]. Dans le cadre de notre ESC, le son peut être exploité à différentes fins. Les tâches au niveau desquelles le son peut réellement aider la coopération sont :

- Construction du modèle de simulation : le son peut être utilisé pour faire des annotations vocales sur le modèle de simulation ou sur certaines parties jugées non encore finalisées. Ces annotations peuvent servir à un usage personnel ou être destinées à d'autres membres du groupe ayant à travailler sur le même modèle. L'intérêt pratique de cette solution par rapport à une annotation à base de texte est que le son est plus rapidement assimilé par le destinataire et en terme de production par l'auteur. La présence d'une annotation vocale sur un modèle de simulation peut être simplement signalée par une icône représentant "*une oreille*" par exemple ; la restitution du message chez le destinataire se fera en cliquant sur cette icône.

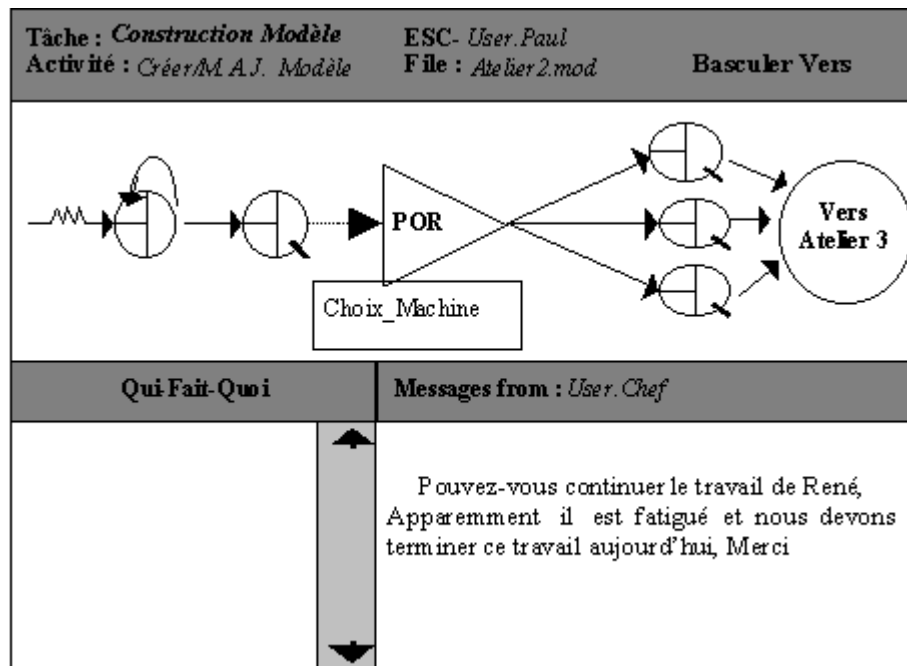
- Analyse des résultats de la simulation : cette tâche produit généralement un document (rapport) dans lequel sont consignés les résultats de la simulation et leur analyse. L'enrichissement de ce document par des annotations vocales permettrait d'adjoindre tous les détails techniques concernant l'analyse. L'intérêt de cette solution découle du fait que dans l'ESC, plusieurs membres peuvent réaliser séparément la même tâche. L'analyse faite par un membre peut servir à un autre membre comme support de confrontation. Ainsi, une analyse très documentée (annotée vocalement) facilitera l'entretien d'une compréhension mutuelle entre les membres. Il faut souligner que dans le cas où les membres concernés sont éloignés géographiquement, la communication entre ces membres peut se limiter à un simple transfert de document grâce à un support de communication tel que E-mail.

## 6 Mise en oeuvre de l'ESC

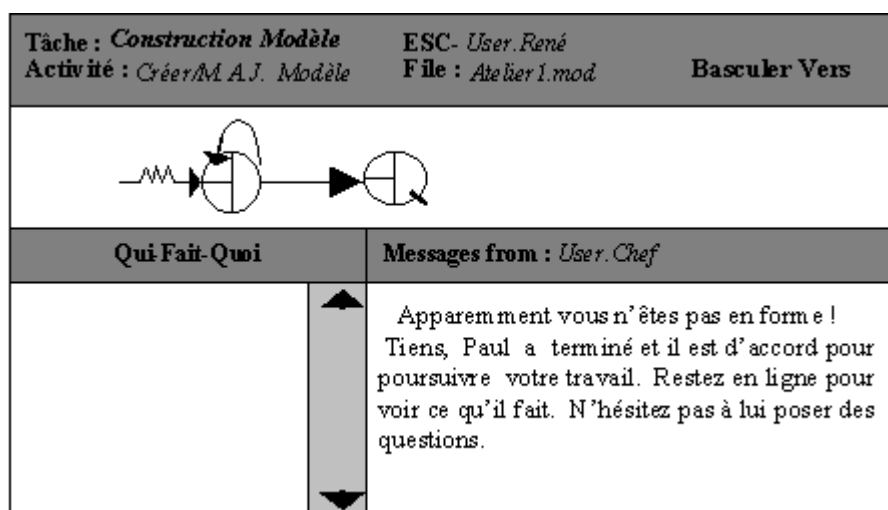
Considérons un cas simple de système à modéliser composé de trois ateliers. Les ateliers 1 et 2 disposent d'un certain nombre de machines réalisant des opérations sur des pièces indépendamment l'un de l'autre. L'atelier 3 reçoit les pièces en provenance des ateliers 1 et 2 et les assemble puis les met dans des cartons par lot de 10 et les achemine vers un stock. Le chef de projet décide d'affecter l'utilisateur Jean à la modélisation de l'atelier 3, Paul à celle de l'atelier 2 et René à celle de l'atelier 1. Il s'agit pour nous de visualiser l'allure des écrans de ces trois utilisateurs afin de montrer ce que nous entendons par un ESC.



**Figure 4 : Allure de l'écran de l'utilisateur Jean**



**Figure 5 : Allure de l'écran de l'utilisateur Paul**



**Figure 6 : Allure de l'écran de l'utilisateur René**

## 7. Conclusion

Nous espérons que les idées que nous avons présentées dans cet article motiveront d'autres équipes à s'investir dans ce domaine de recherches encore balbutiant. Nous nous sommes intéressé ici uniquement à l'analyse des possibilités d'utilisation des concepts du travail coopératif dans le domaine de la conduite d'un projet de simulation. C'est une approche qui est certes très prometteuse mais qui nécessite encore des approfondissements. Cependant les idées que nous avons proposées peuvent être largement exploitées dans un autre créneau tel que l'enseignement coopératif de la simulation. Dans ce cas, on pourrait facilement substituer dans la structure de contrôle que nous avons retenue pour l'ESC, le chef de projet par le professeur et les autres membres du groupe par les apprenants. C'est principalement sur ce créneau que nous travaillons actuellement.

Nous disposons pour cela d'un environnement matériel se composant d'un ensemble de PC 486 DX2 dont certains sont équipés de matériel traitant le son (carte, microphone, etc.), le tout connecté sur un réseau local. L'environnement logiciel que nous expérimentons actuellement pour le prototypage d'un ESC appliqué à l'enseignement coopératif est à base du système d'exploitation LINUX (Distribué par la société Walnut Creek CD-ROM, 4041 Pike Lane, Ste D-904 Concord CA 94520) qui offre une panoplie d'outils de développement (C, C++, X-Windows, TCP/IP, etc.) lui permettant de rivaliser avec ceux offerts par les stations UNIX.

## Bibliographie

[Ata 94] Homa Atabakhsh, Albert W. Chan, "Object Oriented Rule Based Manufacturing Enterprise Simulation" in Proceedings of the O.O. Simulation Conference, 1994, Vol. 26 N°2, p. 9-14

[Bre 92] P. Brezillon, "Explication et Coopération" Rapport de Recherches N° 92/31, Laboratoire Laforia, Institut Blaise Pascal, 1992, p. 1-16

[Cha 94] W. Chang, L.P. Jones, "Message Oriented Discrete Event Simulation" Simulation Vol. 63, N°2, Aout 1994, p. 96-104

[Chai 94] C. Chaillou, L. Wibaux, L. Barme, "Une Expérience d'interface 3D exploitant la Notion de Pièce" Actes IHM'94, p. 191-196

[Der 93] A. C. Deryck, C. Chaillou, L. Dumesges, "La Troisième Dimension dans les Interfaces de

Collecticiels'' Actes IHM'93, p. 81-86

[Gro 94] K. Gronbaek & Al., "Systems : A Dexter-Based Architecture" Communication of The ACM, Vol. 37, N°2, Février 1994, p. 65-74

[Jam 94] F. Jambon, J. Coutaz, "Contrôle Aérien et Liaison de Données : Vers un Plus Grand Partage de l'Information" Actes IHM'94, p. 109-114

[Llo 94] Capt. Bradley A. Lloyd, P.K. Clark, "Object Oriented Simulation With IMDE" in Proceedings of the O.O. Simulation Conference, 1994, Vol. 26 N°2, p. 131-136

[Mac 94] G. T. Mackulak, J.K. Cochran, P. A. Savory, "Ascertaining Important Features for Industrial Simulation Environments" Simulation Vol. 63, N°4, Octobre 1994, p. 211-221

[Ore 79] Tuncer I. Oren, Bernard P. Zeigler, "Concepts for Advanced Simulation Methodologies" Simulation Vol. 32, N° 3, Mars 1979, p. 69-82

[O'ke 86] R. O'keefe, "Simulation and Expert Systems : A Taxonomy and Some Examples" Simulation Vol. 46, N° 1, Janvier 1986, p. 10-16

[Rol 93] J. P. Rolland, C. Vial, B. David, "L'enseignement Assisté par Ordinateur Coopératif : Une Démarche d'Etude" Actes IHM'93, p. 87-94

[Ros 93] M. Roseman, S. Yitbarek, S. Greenberg, "GroupKit Reference Manual : A Guide to its Architecture, Interprocess Communications and Programs" University of Calgary (Octobre 1993).

[Sal 93] D. Salber, J. Coutaz, "Groupe+Vidéo+Audio = MédiaSpace : Etat de l' Art et Revue de Problèmes" Actes IHM'93, p. 67-72

[Sal 94] D. Salber, J. Coutaz, "Taxinomie des mécanismes de Connexion pour la Communication Homme-Machine-Homme" Actes IHM'94, p. 150-156

[Sar 86] R. G. Sargent, "An Exploration of Possibilities for Expert Aid in Model Validation " Modelling & Simulation Methodology in the AI Era (edited by M.S. Elzas, T. I. Oren, B. P. Zeigler), p. 279-297

[Tro 93] C. Tronche, "Utilisation du Canal Audio Dans Le Collecticiel " Actes IHM'93, p. 101-107

[Wie 94] F. Wieland, H. Heckathorn, "An Object Oriented Technologie For Integrating Environmental Effects and Distributed Simulations" in Proceedings of the O.O. Simulation Conference, 1994, Vol. 26 N°2, p. 43-48

[Zei 96] Bernard P. Zeigler, Tae H. Cho, Jerzy W. Rozenblit, "A knowledge based simulation Environment for Hierarchical Flexible Manufacturing" IEEE Transactions on Systems, Man, And Cybernetics, Part A : Systems and Humans , Vol. 26, N°1, Janvier 1996 p. 81-90

---