

CRAC'96

30-31/05/96 - Paris (Univ. Pierre et Marie Curie)

Marie-Chantal PONS,

soumis le vendredi 3 mai 1996 aux

Journées de Recherche sur le

Contrôle Réparti dans les Applications Coopératives

Plan de l'article :

Un langage de description d'applications coopératives

Marie-Chantal PONS

Site d'Etudes et de Recherches en TELÉcommunications, Direction des Constructions Navales,
Avenue de la Tour Royale, 83800 Toulon Naval - mcp@newton.ctsn.dga.fr - 94.16.24.23

Bernard MERIALDO

Institut EURECOM, 2229 route des crêtes, 06904 Sophia-Antipolis

Résumé : Cet article présente un modèle générique pour décrire des applications multimédia collaboratives, par exemple, les applications de téléconférences, les éditeurs multi-utilisateurs, le téléenseignement, les médiaspaces, ou le "work-flow" - c'est à dire la circulation automatique de documents. Le modèle suppose que ces applications prennent place dans un même environnement capable de supporter différentes phases de façon cohérente, en autorisant une description interne des configurations, de leurs successions et de leurs liens. L'article décrit brièvement quelques systèmes existants. Le modèle est ensuite exposé d'un point de vue statique à travers les différents objets. Puis, l'aspect dynamique est précisé par le déroulement des sessions. La dernière section aborde le langage de description de session qui permet à l'utilisateur de configurer facilement son application.

1. Introduction

Les applications multimédia collaboratives se diversifient de plus en plus. Des logiciels pour le travail de groupe aux vidéoconférences en passant par les messageries, tous s'appuient sur une structure de communication, de partage de ressources, voire même de collaboration. Certaines applications comme l'édition collaborative nécessitent l'adjonction du partage de documents. De nombreux outils proposés permettent de renforcer l'aspect structurel de l'application et donc d'en améliorer l'efficacité¹. Par exemple, la gestion de l'ordre du jour avec des horaires, et des sujets à respecter assistent le président de séance lors d'une réunion.

Un système efficace doit offrir une structure de base à laquelle peuvent se rattacher divers outils afin d'obtenir l'application désirée. Ce système de gestion d'applications multimédia coopératives peut s'appuyer sur un langage de description afin que tout utilisateur programme une session d'une manière facile et rapide. Le formalisme devra être pour cela explicite et concis. L'utilisation d'un langage de description n'est pas indispensable. Une autre solution est de proposer diverses applications préconfigurées. L'utilisateur choisit alors parmi ces applications. Afin d'éviter ces restrictions, un langage de programmation sera étudié.

L'aspect intéressant est la souplesse offerte par un langage. Le besoin de personnalisation s'est fait ressentir ces derniers temps. Différents protocoles ont été mis en place. Dans les protocoles technologiques comme dans les systèmes de conférences [ELL91], le contrôle de flux ou "floor control" distribue la parole à tour de rôle entre les participants. La structure imposée assiste l'utilisateur peu expérimenté mais peut être aussi ressentie comme restrictive par un utilisateur averti. Un autre type de protocoles est le protocole social. Les participants conviennent eux-mêmes de règles de conduite (qui peut communiquer avec qui, qui peut dérouler telle fenêtre...) Dans l'éditeur multi-utilisateur temps réel *GROVE* (*GRoup Outline Viewing Editor*), les participants sont libres; ils ont rapidement développé leur propre protocole afin de favoriser la collaboration. Ce même système dans un autre environnement aurait pu se révéler inefficace et inéquitable. Ces deux exemples démontrent qu'un collectif doit être de plus en plus adaptatif. C'est à dire que l'utilisateur doit pouvoir se forger une **application adaptée à ses besoins et à son environnement**. Un langage de programmation est un moyen d'y parvenir facilement.

Cet article présente un modèle générique pour des applications multimédia collaboratives. Il débute par une brève présentation de systèmes à l'origine du concept. Il expose ensuite le modèle d'un point de vue statique. Puis, le fonctionnement dynamique est traité afin de mieux comprendre le déroulement des sessions. Après avoir présenté l'architecture du modèle, l'article conclut sur le langage de description de session. Ce travail correspond à des recherches en cours. Nous définissons actuellement des scénarios d'expérimentation afin d'améliorer et de valider le langage.

2. Un aperçu de l'existant

De nombreux systèmes tels que la *Touring Machine* de Bellcore [ARA93], le *Mediaspace* de Xerox PARC [OLS91], *RAVE* à Cambridge [BUX90] ou encore *CAVECAT* de l'Université de Toronto [MAN91], ont développé des environnements de communication multimédia afin que des personnes puissent collaborer sans contraintes de distances géographi-

¹. De nombreux travaux ont démontré que l'apport d'une structure entraîne de meilleurs résultats (plus de décisions prises, gain de temps, meilleure satisfaction des utilisateurs). On peut citer à titre d'exemples, les méthodes Delphi et NGT [HIL78].

ques. Le problème du contrôle d'accès a souvent été rencontré dans ces expériences. Par exemple, le système Touring Machine offre des sessions privées (accessibles sur invitation) et des sessions publiques (ouvertes à tous). Notre modèle propose différents modes que nous verrons par la suite. Le langage de programmation de Bellcore, Rendezvous, assiste le développement d'applications multi-utilisateurs [HIL93].

Des instituts de normalisation se sont également intéressés au contrôle de sessions avec le protocole Generic Conference Control (GCC) qui gère la création, la terminaison de conférences, les participants dont le rôle de meneur prévoit des conférences chronométrées [ITU93]. Toujours dans le cadre de téléconférences multimédia, les recommandations T.120 de l'ITU (International Telecommunication Union) portent sur la gestion de la communication entre sites distants [ITU95].

A partir d'une taxonomie de différents types de collaborations, [RAJ95] propose un modèle formel pour des applications multimédia collaboratives. Selon une démarche similaire, les concepts du modèle de description de sessions coopératives obtenus ont servi de support pour développer le langage de description de sessions coopératives.

La conception d'environnements de Travail Coopératifs Assistés par Ordinateur (TCAO) est également étudiée à travers la plate-forme Co-Learn [CRO95] [HOO95].

3. Le modèle en statique

3.1 Introduction

L'objectif du modèle est de décrire des activités aussi variées que participer à une réunion administrative pour voter des budgets, suivre une formation sur un nouveau logiciel, faire circuler une note de service ou jeter un coup d'oeil chez un collègue afin de savoir s'il est disponible. Le modèle présenté par la suite envisage donc une multitude d'applications par le biais d'un langage de description de session. Il gère la communication, le partage de ressources et la collaboration des individus.

3.2 Exemple

Voyons à travers un exemple les principaux concepts du modèle. Monsieur X avait planifié une réunion professionnelle d'avancement de projet. Malheureusement, un contretemps dans sa société ne lui a pas permis de se déplacer jusqu'à l'endroit de la réunion. Il va alors participer à la réunion grâce à un système de téléconférence. Il s'inscrit comme utilisateur au plus précisément comme acteur, auprès du système en indiquant ses coordonnées (carte de visite, localisation de son poste de travail), son équipement multimédia (caméra, moniteur vidéo, microphone ...) et les flux de données rattachés. Il demande à rejoindre en retard une session de type réunion à laquelle il était invité. Si les retardataires sont admis, il prendra alors le rôle de participant ou d'observateur dans la réunion. Avec le premier rôle, il peut participer activement à la conversation. Dès son entrée dans la réunion, il reçoit les documents que les autres personnes ont reçus en début de réunion (dossier du projet étudié) ainsi qu'un résumé de ce qui a été dit (cela n'est possible que dans le cas d'une rédaction temps réel d'un compte-rendu). Il accède sur son écran à l'ordre du jour afin de se situer dans la chronologie des sujets abordés.

3.3 Les concepts

D'une façon plus générale, les principaux concepts entrant en jeu dans la description d'une session sont les suivants :

- acteur : cela représente un utilisateur identifié par sa carte de visite. L'accès est éventuellement contrôlé par un mot de passe.

- activité : l'acteur va participer à une activité. Cela peut être une réunion professionnelle structurée, une connexion spontanée pour demander un conseil, un cours par vidéoconférence ou encore la circulation automatique d'un document auprès de ses collègues - ou plus couramment "workflow".

- session : les activités sont classées dans des sessions typées afin de faciliter la gestion d'activités répétitives avec les mêmes personnes et les mêmes objectifs, ainsi que le séquençement de tâches. Par exemple, un enseignement est prévu comprenant une préparation, des cours suivis de travaux dirigés puis d'une séance de contrôle. Les acteurs élèves et professeurs ne seront enregistrés qu'une seule fois. Le même document support de cours sera disponible pendant la série d'activités.

- rôle : nous venons de voir dans l'exemple, deux rôles pour une session de téléenseignement : élève et professeur. Cela illustre bien que ces deux personnes n'ont pas les mêmes droits. Dans le scénario, l'acteur choisissait le rôle de observateur ou de participant. Faisons un petit zoom sur un rôle important. Dans une session de réunion, le maître de conférence qui peut avoir différents degrés d'intervention², dirige le débat, contribue au bon déroulement de la rencontre, prend quelquefois les décisions. Le système a pour objectifs de l'aider dans ces tâches. La notion de rôle permet d'attribuer ou non la parole à un acteur, de lui laisser accès à certains documents ...

- poste : il regroupe l'équipement et les flux de données rattachés pour que l'acteur sur ce poste soit connecté avec les

². En effet, le maître de conférence est plus ou moins directif selon le type de réunion (réunion d'information, d'échanges, pour résoudre un problème, ou de négociation ...) et les personnes présentes. On trouve alors les termes d'animateur, de meneur, ou de président de séance selon le cas [NOY89].

autres participants.

- équipement : aux postes, seront rattachés différents appareils physiques ou logiciels utilisés pendant l'activité.

- document : au cours de réunions, de séances d'enseignement, ou de "workflow", les utilisateurs travaillent ensemble sur des documents. Ce concept est donc indispensable pour gérer la distribution, le contrôle d'accès, la circulation automatique et partage de documents.

Une hiérarchie entre les objets permet un héritage de propriétés. Supposons une session de type téléenseignement définie avec Monsieur Y professeur, comme responsable qui débutera et terminera l'application. Les activités rattachées à cette session prendront par défaut ces caractéristiques. Cela évite la répétition de configurations et ainsi facilite la phase d'initialisation.

Figure 1 : Les objets*

* Note : pour davantage de clareté, sur la figure, les liens entre les objets sont indiqués entre les classes. En réalité, ces liens se font entre les objets instances.

La figure 1 ci-dessus présente les objets dans le cas d'une session de téléenseignement. L'acteur bruno a le rôle de professeur dans ces activités. Il utilise le poste dont le nom est machine1 et dispose d'un document support de cours durant toute la session. Chaque instance d'un objet possède une configuration. Dans l'exemple, l'objet classe de type session comporte son nom : classe, son type : téléenseignement, réunion, connexion spontanée, messagerie vidéo, ou workflow, la clé : ici c'est le mot de passe 'bonjour', et aussi l'accès à la session, les modes de démarrage et de fin, le type d'annonce.

4. Le fonctionnement dynamique

4.1 Le déroulement

Deux phases sont à considérer. Une première phase correspond à l'initialisation de l'application. L'utilisateur va configurer le type des activités, les acteurs, les différents modes d'accès, l'agenda, les rôles ... Cela doit être le plus simple et rapide possible. La seconde correspond au déroulement des activités. Le système gère les prises de parole, l'ordre du jour, les apartés, la messagerie entre les utilisateurs ... Pour cela, des requêtes sont émises afin d'activer, terminer mettre à jour les différents objets. On peut citer les requêtes de démarrage/fin d'une activité, de l'attribution de la parole, d'un rôle ou l'arrivée d'un nouvel acteur.

Une requête reçue ne sera exécutée que sous certaines conditions. Le commencement d'une activité par exemple, sera acceptée selon le mode de démarrage prévu. Cela peut être automatique à une heure donnée, par le premier connecté ou un responsable. Afin de connaître ses droits, un acteur prend des sous-rôles appelés états : attente si l'activité n'a pas encore démarré, actif s'il a la parole, fermé s'il est absent ... Ainsi, le système sait si l'acteur demandeur est autorisé à émettre la requête.

Les personnes prévues pour la session ont un contrôle d'accès différent (accès privé) des autres (accès public). La personne désirant rejoindre une activité consulte la liste des activités programmées. Celles en mode invisible n'apparaissent pas, celles en mode fermé sont interdites d'accès, et pour celles en mode semi, l'autorisation sera délivrée par le responsable. L'accès n'est pas contrôlé en modes libre, et ouvert, dans le second cas, le responsable est prévenu d'un nouvel arrivant.. Ces modes s'appliquent également pour les apartés et la messagerie entre les acteurs.

Différentes configurations sont disponibles. Par exemple, une personne parle pendant que les autres écoutent. Cela correspond à une activité de présentation d'un projet avec des transparents pendant une réunion ou bien une séance de cours devant des élèves. Une autre configuration est la discussion pendant laquelle les acteurs peuvent interagir librement. Un exemple est l'activité de "brainstorming" qui correspond à une phase d'une réunion pendant laquelle les individus exposent leurs idées sans aucune restriction.

4.2 L'architecture du système

Le système est composé de différents modules (Figure 2). Une couche de présentation réalise l'interface entre l'utilisateur et le système. Elle saisit les requêtes et les transmet au gestionnaire d'applications. Celui-ci consulte la configuration de l'application en cours ainsi qu'une base de règles afin d'accepter ou de rejeter la requête. Cette configuration indique les autorisations : démarrage de l'activité, aparté, noms, mots de passe, états des acteurs, liens entre les acteurs, rôles, postes, activités ...

Figure 2 : L'architecture

Un composant, le gestionnaire de ressources, contrôle le partage de ressources. Il effectue diverses opérations sur les équipements physiques. En effet, il réserve, alloue, libère les canaux d'échanges de données multimédia entre les caméras, microphones, magnétoscopes et différents autres appareils. Des modèles de configurations offrent à l'utilisateur une assistance pour initialiser son application.

5. Le formalisme

5.1 Un langage à objets

Les différents concepts du modèle sont intuitivement représentables en pratique par des objets. De plus, comme nous allons le voir, chaque objet est constitué d'une partie valeurs et d'une partie méthodes. Les bénéfices de la technologie objet apportent alors différentes propriétés au langage. Les objets indépendants obtenus offrent une certaine modularité qui les rend facilement modifiables. Comme une instance hérite de la classe mère, une activité de télésurveillance hérite de la configuration (différents modes) de la session mère. A cette notion d'héritage s'associe celle de réutilisabilité. En effet, une même classe servira pour générer plusieurs instances.

La figure 3 ci-dessous présente la description de l'objet 'activité'. L'application s'inscrit dans le cadre d'une session de téléenseignement non présentée comprenant la description des autres objets tels que la session et les acteurs.

[Activité

[Config

```
-Session      : cours_informatique
-Acteur       : Bernard, Anne, Eric, Fred, Patrick, Sophie, Paul, Pierre, Claire
Nom           : classe2
Enbref       : séance d'exercices de Mr Mérialdo, option multimédia à Eurecom
Type         : td
Resp         : Bernard
Deb          : 220196,10h00
Modedeb      : resp
Fin          : 220196,12h00
Modedefin    : resp
Accespub: ouvert
Accespriv: ferme
Sortie       : semi
Retard       : 10
Periode      : hebdo
Aparte       : ferme
Message      : ouvert
```

Config]

[Comportement

```
Contraintes: eqpt.label(camera), activite(classe1) etat(achevee)
Etats       : init, attente, active, achevee
ActionsEm: terminer acteur, lier acteur poste
ActionsRec: creer acteur, terminer acteur, lier acteur role, separer acteur role, demarrer pause, terminer pause
```

Comportement]

Activite]

Figure 3 : Exemple de syntaxe : déclaration d'un objet activité de type téléenseignement

5.2 La partie configuration

Le langage permet à l'utilisateur d'indiquer la configuration d'une application à venir; c'est à dire les différents participants ainsi que leur(s) rôle(s), le type d'activité, la date et l'heure prévues ... cela revient à initialiser les paramètres de chaque objet. C'est la première partie du langage appelée Config, qui bien que mise à jour durant la session, est statique.

L'activité est du type travaux dirigés (td), donc prévoit des interactions entre les élèves et le professeur. Différents mo-

des sont définis dans la configuration. Pour les modes de démarrage et de terminaison dans l'exemple, le début de l'activité est prévu le 22 janvier 1996 à 11 h, mais elle commencera lorsque le responsable le demandera. D'autres modes sont disponibles comme le démarrage si tous les participants prévus seront présents, ou au premier connecté. Le mode d'accès à l'activité signifie ici, que seules les personnes prévues sont autorisées à participer à l'activité. Le retard admis est de 10 minutes. La périodicité de l'activité, les autorisations pour les apartés et la messagerie entre les participants sont également indiquées.

Dans la configuration, les liens statiques sont indiqués (signe '-' en début de ligne). Cela correspond aux prévisions : activités dans session, poste pour acteur, rôles pour acteur ... Ensuite, durant la phase de déroulement, les liens réels entre les objets, appelés liens dynamiques, seront établis suite à des actions de l'utilisateur.

5.3 La partie comportement

La seconde partie du langage correspond au comportement dynamique du système. La déclaration comprend les contraintes, les états et les différentes requêtes disponibles associés à l'objet. Les contraintes permettent de formuler des restrictions. Elles sont exprimées comme suit : `type_objet.nom_champ(valeur_champ)`

Par défaut, le champ est le nom de l'objet. `eqpt.label(camera)` signifie que pour accéder à l'activité, les acteurs doivent posséder une caméra. La seconde contrainte est liée au séquençement des activités. L'activité `classe2` ne peut pas démarrer si l'activité `classe1` n'a pas eu lieu.

Les états aident à décider de l'acceptation ou du refus d'une requête. Des règles sont associées aux requêtes afin de déterminer les conditions d'acceptation ainsi que les actions suite à une réponse positive. Par exemple, l'utilisateur peut demander le démarrage de l'activité. Le système vérifie d'abord les conditions de faisabilité. C'est à dire dans ce cas, il s'assure de l'existence d'une telle activité et de son état (cela évite de lancer une application inconnue ou déjà en cours). Ensuite, les conditions de permission sont consultées. La configuration indique le mode de démarrage de l'activité. Si la mise en route de l'activité est réalisée par le responsable, le système recherche si l'acteur à l'origine de la requête possède ce rôle.

Deux types de requêtes sont envisagés. Une première série correspond à la phase d'initialisation. Elle comprend les requêtes pour créer, modifier, supprimer et lier des objets. Par exemple, pour créer une session la commande est : `new_session(nom, type, config, contraintes, requêtes en émission, requêtes en réception)`

Cela entraîne la création de l'objet session avec les différents paramètres indiqués. Les autres requêtes sont liées au déroulement de l'activité : arrivée d'un acteur, prise d'un rôle ... Dans le langage, les requêtes agissant sur chaque objet sont indiquées. Lorsque l'utilisateur souhaite démarrer la session, la couche présentation envoie un message au gestionnaire d'applications.

6. Conclusion

Le modèle présenté permet de décrire les diverses interactions de l'utilisateur. Le langage en cours de développement doit permettre, par sa simplicité, de programmer facilement les différents types de situation de collaboration. Nous sommes en train de réaliser un démonstrateur qui présentera le déroulement de différents scénarios, par exemple, une session de téléenseignement avec le rôle de professeur qui dirige les élèves, ou la simulation d'une réunion avec les différentes phases, la gestion des interventions et de l'ordre du jour.

7. Références

[ARA93]M. Arango, L. Bahler ... Bellcore Information Networking Research Laboratory, The Touring Machine System. Communications of the ACM, 36, 1, 68-77, 01/93.

[BUX90]Bill Buxton, Tom Moran, EuroPARC's Integrated Interactive Intermedia facility (IIIF) : early experiences. Multi-User Interfaces and Applications, IFIP, 11-33, 90.

[CRO95]Pascal Croisy, Collecticiel temps réel et apprentissage coopératif : des aspects sociaux et pédagogiques jusqu'au modèle multi-agent de l'interface de groupe. Thèse de l'Université des Sciences et Technologies de Lille, 02/95.

[ELL91]Clarence Ellis, Simon Gibbs, Gail Rein, Groupware : Some Issues and Experiences. Communications of the ACM, 34, 1, 38-58, 01/91.

[HIL93]Ralph D. Hill, Tom Brinck, John F. Patterson, Steven L. Rohall, Wayne T. Wilner, The rendezvous Language and Architecture. Communications of the ACM, 36, 1, 62-67, 01/93.

[HIL78]Starr Roxanne Hiltz, Murray Turoff, Structured Communications. The Network Nation : Human Communication via Computer. Addison-Wesley Publishing Company, 287-337, 1978.

[HOO95]Frédéric HOOGSTOEL, Une Approche Organisationnelle du Travail Coopératif Assisté par Ordinateur : Application au projet Co-Learn. Thèse de l'Université des Sciences et Technologies de Lille, 11/95.

[ITU93]ITU-Telecommunications Standardization Sector - Study Group 8. Draft recommendation T.124 T.GCC, Geneva, 16-25/11/93.

[ITU95]ITU-Telecommunications Standardization Sector - Study Group 8. Initial Draft of T.120, BT Laboratories, John Boucher, 6/10/95.

[MAN91]M. M. Mantei, R. M. Baecker, A. J. Sellen, W. A. S. Buxton et T. Milligan, Experiences in the use of a media space. In ACM SIGCHI Conference on Human Factors in Computing Systems, ACM Press, 203-208, New Orleans, 1991.

[NOY89]Didier Noyé, Réunionite, guide de survie. Editions INSEP, 1989.

[OLS91]Margrethe H. Olson and Sara A. Bly, The Portland experience: A report on a distributed research group. International Journal of Man Machine Studies, 34(2), 211-228, 1991.

[RAJ95]Sreeranga Rajan, P. Venkat Rangan and Harrick M. Vin, A formal basis for structured multimedia collaborations. In Int. Conf. on Multimedia Computing and Systems, IEEE Computer Society Press, 194-201, Washington, 15-18/05/1995.