

MÉMOIRE

présenté au

LABORATOIRE D'ANALYSE ET D'ARCHITECTURE DES SYSTÈMES DU CNRS

en vue de l'obtention du titre :

**HABILITATION A DIRIGER DES RECHERCHES
DE L'UNIVERSITÉ PAUL SABATIER DE TOULOUSE**

Spécialité : Systèmes Informatiques

par

Thierry VILLEMUR

**MODÈLES ET SERVICES LOGICIELS
POUR LE TRAVAIL COLLABORATIF**

Soutenu(e) devant le jury :

Brigitte Pradin

Président

Michel Diaz

Directeur de Recherche

Jean-Marc Geib

Hervé Guyennet

Francis Lepage

} Rapporteurs

Abdelmadjid Bouabdallah

Examinateur

Remerciements

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) du Centre National de la Recherche Scientifique (CNRS), dirigé durant cette période par Monsieur J.C. Laprie et Monsieur M. Ghallab. Je tiens à les remercier pour m'avoir accueilli en tant qu'enseignant chercheur dans ce laboratoire.

Je remercie l'ensemble des responsables successifs du groupe Outils Logiciels pour la Communication (OLC), messieurs M. Diaz, J.P. Courtiat et F. Vernadat, qui, par leur animation scientifique, contribuent à maintenir un cadre de recherche performant.

De façon plus particulière, j'adresse mes vifs remerciements à Monsieur M. Diaz, sous-directeur du LAAS-CNRS, qui, en plus de toutes ses responsabilités, a accepté de superviser la rédaction de ce mémoire. Que son aide dans la synthèse de mes travaux et ses conseils pertinents puissent trouver ici la marque de ma sympathie la plus sincère.

Je suis très reconnaissant à :

Monsieur Abdelmadjid Bouabdallah, Professeur à l'Université de Technologie de Compiègne,

Monsieur Michel Diaz, Directeur de Recherches au CNRS,

Monsieur Jean-Marc Geib, Professeur à l'Université des Sciences et Technologies de Lille,

Monsieur Hervé Guyennet, Professeur à l'Université de Besançon,

Monsieur Francis Lepage, Professeur à l'Université de Nancy,

Madame Brigitte Pradin, Professeur à l'Université Paul Sabatier Toulouse III,

pour l'honneur qu'ils me font en participant à ce jury d'habilitation, et plus particulièrement à Messieurs Geib, Guyennet et Lepage qui ont accepté la charge de Rapporteur.

J'adresse un grand merci à tous mes collègues du groupe Outils Logiciels pour la Communication que j'ai côtoyés durant ces années. Les discussions fructueuses et chaleureuses menées avec les différents membres de l'équipe contribuent à fournir un cadre de travail convivial, enrichissant, et efficace. Mes remerciements s'adressent aussi à V. Baudin, du service Informatique et Instrumentation, pour sa participation constante à de nombreux projets de recherche communs.

Finalement, je tiens à étendre mes remerciements à l'ensemble des personnes qu'une activité de recherche riche et soutenue, avec des contacts au niveau national, européen et international, m'a permis de découvrir, de rencontrer et pour certaines d'en faire des amis.

Table des matières

Partie 1 - Coursus suivi

1. Curriculum Vitae	3
Coordonnées.....	3
Etat civil	3
Profession.....	3
Formation	3
2. Activités d’enseignement	4
Enseignements créés	4
DUT GTR première année	4
DUT GTR deuxième année.....	4
Licence Pro RMS	4
Enseignements dispensés	5
DUT GTR première année	5
DUT GTR deuxième année.....	5
Licence Pro RMS	5
Responsabilités pédagogiques.....	5
3. Publications.....	6
4. Encadrement de stages/DEAs/thèses	15
Encadrement terminé.....	15
Encadrement en cours	16
5. Animation scientifique et rayonnement	17
Animation.....	17
Conférences.....	17
Comités scientifiques	17
6. Relations avec le monde industriel	18
Projets auxquels j’ai participé	18
Projets avec responsabilité administrative/scientifique.....	18
Valorisation scientifique	19

Partie 2 - Synthèse des travaux
--

Introduction	23
Démarche de recherche	23
Plan de la synthèse des travaux	25
Chapitre 1. Cadre de nos travaux.....	27
1. Travail collaboratif. Définitions	27
2. Travail collaboratif. Collecticiel asynchrone	28
2.1 Messageries avancées.....	28
2.2 Systèmes basés sur le Web.....	31
3. Travail collaboratif. Collecticiel synchrone	33
3.1 Outils pour des communications informelles.....	34
3.2 Outils de partage d'application	41
3.3 Outils de partage de documents	44
3.4 Plates-formes de collecticiel synchrone	46
3.5 Bilan	47
4. Modèles de session.....	47
4.1 Intermezzo.....	48
4.2 CoAct	49
4.3 Espaces partagés.....	51
4.4 Coordination de groupe.....	51
4.5 Synthèse des modèles de session	52
5. Services de communication pour la gestion de session.....	53
5.1 Tango 2.....	55
5.2 Habanero 3.0	55
5.3 DOE 2000.....	56
5.4 DISCIPLÉ	57
5.5 Synthèse des systèmes gestionnaires de session	57
Chapitre 2. Modélisation de sessions collaboratives	59
1. Introduction	59
1.1 Principaux éléments de session.....	59
1.2 Objectifs de la modélisation.....	60
2. Diagrammes de coordination	60
2.1 But du modèle	60
2.2 Modèle.....	61
2.3 Dynamique des sessions.....	64
2.4 Avantages du modèle	70
2.5 Positionnement	71
3. Contributions pour des services de communication orientés modèle	72
3.1 Implantation des diagrammes de coordination.....	72
3.2 Visualisateur de session	73

4. Contributions à la modélisation de scénarios pour la téléformation	74
4.1 Projet TOPASE – Formation de pilotes et d’agents de maintenance.....	75
4.2 Projet Lab@Future – Laboratoire scolaire distribué.....	76
4.3 Scénario d’apprentissage collaboratif	83
5. Synthèse	85
Chapitre 3. Gestion de sessions collaboratives	87
1. Introduction	87
1.1 Cadre de conception des services et des protocoles de communication	87
1.2 Objectifs	88
2. Gestion de session	89
2.1 Service de gestion de session	89
2.2 Protocole	92
2.3 Réalisation.....	93
2.4 Positionnement.....	93
3. Contribution à la gestion du travail collaboratif.....	95
3.1 Langage et méthode	95
3.2 Architecture d’implantation	96
3.3 Prototype	98
3.4 Bilan	100
4. Contribution méthodologique	101
4.1 Motivations	101
4.2 Méthodologie bi modèle.....	102
4.3 Technologie middleware	109
4.4 Implantation bi modèle pour le service V2 de gestion de session	110
4.5 Bilan	111
5. Contribution architecturale.....	112
5.1 Motivations	112
5.2 Architecture détaillée	113
5.3 Implantation	117
5.4 Bilan	119
6. Synthèse	120
Conclusion et perspectives	121
1. Conclusion.....	121
2. Perspectives à moyen terme	124
2.1 Modèles	124
2.2 Services de coordination	124
2.3 Travail collaboratif.....	132
3. Perspectives générales.....	132
3.1 Thématique.....	132
3.2 Contexte et positionnement.....	133
3.3 Objectif scientifique	134

3.4 Méthodes et approches	135
Références	139

Partie 1

Cursus suivi

1. Curriculum Vitae

Coordonnées

Thierry Villemur
2 rue de Jolimont
31500 Toulouse
Tel : 05 61 61 22 86
Email : villemur@laas.fr

Etat civil

Né le 21 octobre 1968 à Auch (Gers)
Marié
Nationalité française

Profession

Maître de Conférences
Département Génie des Télécommunications et Réseaux
IUT B de Blagnac

Groupe de recherche Outils Logiciels pour la communication (OLC)
LAAS-CNRS

Formation

Baccalauréat C (juin 86)
DEUGS A, Université Paul Sabatier (juin 88)
Licence/Maîtrise d'Informatique, Université Paul Sabatier (septembre 90)
DEA d'Informatique, Université Paul Sabatier (septembre 91)
Thèse d'Informatique, Université Paul Sabatier (janvier 95)

2. Activités d'enseignement

Enseignements créés

	Cours	TDs	TPs
DUT GTR	64 heures	72 heures	220 heures
Licence Pro. RMS	8 heures		8 heures

DUT GTR première année

Mise en place des enseignements de première année lors de la création du département Génie Télécommunications et Réseaux (GTR) de l'IUT (septembre 95 – juin 96).

Cours/TDs/TPs créés dans les disciplines :

- Algorithmique et programmation C/JAVA (20 h cours, 36 h TDs, 72 h TPs)
- Assembleur (72 h TPs)
- Bureautique (24 h TPs)
- Réseau couches basses (20 h Cours)

DUT GTR deuxième année

Mise en place de nouveaux enseignements lors du démarrage de la deuxième année du département Génie Télécommunications et Réseaux de l'IUT (septembre 96 – juin 97).

Cours/TDs/TPs créés dans les disciplines :

- Ingénierie des protocoles (8 h cours, 12 h TDs)
- Réseau couches hautes (8 h cours, 20 h TDs)
- Programmation système (8 h cours, 16 h TDs, 24 h TPs)
- Configuration réseau (24 h TPs)

Licence Pro RMS

Nouveaux enseignements mis en place pour la Licence Professionnelle Réseaux Mobiles et Sécurité (RMS) (septembre 2002 – juin 2003).

Cours et TPs créés dans la discipline :

- Middleware (8 h cours, 8 h TPs)

Enseignements dispensés

	Cours	TDs	TPs
DUT GTR	44 heures	20 heures	144 heures
Licence Pro. RMS	8 heures		8 heures

DUT GTR première année

Algorithmique et programmation C/JAVA (20 h cours, 72 h TPs)

Bureautique (24 h TPs)

DUT GTR deuxième année

Ingénierie des protocoles (8 h cours, 12 h TDs)

Réseau couches hautes (8 h cours, 8 h TDs)

Programmation système (8 h cours, 2*24 h TPs)

Licence Pro RMS

Middleware (8 h cours, 8 h TPs)

Responsabilités pédagogiques

Responsable du suivi des étudiants diplômés DUT GTR. Le travail consiste à suivre les nouveaux diplômés du DUT pour connaître leur devenir après la formation.

Membre de la commission de spécialistes 61-63 de l'Université de Toulouse II

Titulaire de 1995 à 1999

Suppléant de 2004 à 2006

3. Publications

Ouvrages	Chapitres	Revue	Conférences
2	2	6	35

Rapports de contrat	Rapports internes
18	17

Ouvrages (contribution)

K.DRIRA, J.M.MOLINA-ESPINOSA, O.NABUCO, L.M.RODRIGUEZ PERALTA, T.VILLEMUR

Product data and workflow management

Rapport LAAS N°01599

Cooperative environments for distributed systems engineering, Lecture Notes in Computer Science 2236, Springer, N°ISBN 3-540-43083-0, 2001, pp.107-151

V.BAUDIN, K.DRIRA, T.VILLEMUR, S.TAZI

A model-driven approach for synchronous dynamic collaborative e-learning

Rapport LAAS N°02265

E-Education applications: human factors and innovative approaches, Ed. C.Ghaoui, Information Science Publishing, ISBN 1-59140-292-1, 2004, Chapter III, pp.44-65

Ouvrages (éditeur)

K.DRIRA, A.MARTELLI, T.VILLEMUR

Cooperative environments for distributed systems engineering

Rapport LAAS N°01598

Lecture Notes in Computer Science 2236, Springer, 279 p., N°ISBN 3-540-43083-0, 2001

J.P.COURTIAT, C.DAVARAKIS, T.VILLEMUR

IFIP TC3 Technology Enhanced Learning workshop (Tel'04). IFIP World Computer Congress

Rapport LAAS N°04703

Springer, N°ISBN 0-3872-4046-2, 2004, 188p.

Doctorat

T.VILLEMUR

Conception de services et de protocoles pour la gestion de groupes coopératifs

Rapport LAAS N°95002

Doctorat, Université Paul Sabatier, Toulouse, 3 Janvier 1995, N°1952, 166p., Jury : G.JUANOLE, Président
Examineurs : E.HORLAIT, M.TREHEL, M.DIAZ, P.AZEMA, B.BOUCARE, JJ.MERCIER, JJ.PANSIOT, M.RAYNAL

Reuves Scientifiques

M.DIAZ, T.VILLEMUR

Membership services and protocols for cooperative frameworks of processes
 Rapport LAAS N°93002
 Special Issue of Computer Communications on Group Communications, Vol.16, N°9, pp.548-556,
 Septembre 1993

M.DIAZ, T.VILLEMUR

Formation of private conversation subgroups in a cooperative group of processes
 Rapport LAAS N°94289
 Journal of the Brazilian Computer Society, Vol.1, N°1, pp.46-58, Juillet 1994

T.VILLEMUR, M.DIAZ, F.VERNADAT

Validated design of dynamic membership services and protocols for cooperative groups
 Rapport LAAS N°95085
 Annales des Télécommunications, N°11-12, pp.859-873, Nov-Déc 1995

T.VILLEMUR, V.BAUDIN, S.OWEZARSKI, M.DIAZ

Multimedia tools supporting the work of distributed synchronous cooperative groups
 Rapport LAAS N°98220
 Cluster Computing, Vol.2, N°1, pp.61-74, 1999

K.DRIRA, T.VILLEMUR, V.BAUDIN, M.DIAZ

A design methodology applied to distance learning support software
 Rapport LAAS N°00297
 Interactive Learning Environments, Vol.9, N°1, pp.51-78, Avril 2001

T.VILLEMUR, K.DRIRA, V.BAUDIN, M.DIAZ

Services, methodologies and platforms for cooperative environments. The LAAS-CNRS experience
 Rapport LAAS N°02028
 Invited paper. Revue Studia Informatica Universalis, Vol.2, N°HS.2, pp.213-234, 2003

Manifestations avec actes

M.DIAZ, T.VILLEMUR

Communications dans des groupes structurés et dynamiques d'agents
 Rapport LAAS N°94282
 6èmes Rencontres Francophones du Parallélisme (RenPar'6), Lyon (France), 7-10 Juin 1994 , 325p.,
 (Résumé)

T.VILLEMUR, M.DIAZ, F.VERNADAT, P.AZEMA

Verification of services and protocols for dynamic membership to cooperative groups
 Rapport LAAS N°94283
 Workshop on Computer-Supported Cooperative Work, Petri Nets and Related Formalisms, Saragosse
 (Espagne), 21 Juin 1994 , pp.73-92

F.VERNADAT, T.VILLEMUR, P.GRADIT, M.DIAZ

Distributed management of dynamic membership of agents in cooperative group
 Rapport LAAS N°95176
 European Research Seminar on Advances in Distributed Systems, L'Alpe d'Huez (France), 3-7 Avril 1995 ,
 pp.236-241

T.VILLEMUR, M.DIAZ, F.VERNADAT

Une approche pour la conception validée de services et de protocoles coopératifs

Rapport LAAS N°95359

Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'95), Rennes (France), 10-12 Mai 1995 , 16p.

M.DIAZ, T.VILLEMUR, F.VERNADAT

Formal design of cooperative systems

Rapport LAAS N°95295

15th International Symposium on Protocol Specification, Testing and Verification (PSTV'95), Varsovie (Pologne), 13-16 Juin 1995 , pp.341-345

P.OWEZARSKI, T.VILLEMUR, M.DIAZ

Conception d'un système de visioconférence coopératif

Rapport LAAS N°96134

RenPar'8, Bordeaux (France), 20-24 Mai 1996, pp.25-28

P.OWEZARSKI, T.VILLEMUR, M.DIAZ

Conception et implémentation d'un système de visioconférence coopératif à N intervenants

Rapport LAAS N°96133

Journées de Recherche sur le Contrôle Réparti dans les Applications Coopératives (CRAC'96), Paris (France), 30-31 Mai 1996, pp.87-92

V.THOMAS-BAUDIN, M.DIAZ, P.OWEZARSKI, T.VILLEMUR

Design and realization of a synchronous cooperative shared electronic board

Rapport LAAS N°96442

Advanced Technology Workshop 1996 Europe, Toulouse (France), 8-10 Juillet 1996, 14p.

M.DIAZ, F.VERNADAT, T.VILLEMUR

Spécification et réalisation formelles de systèmes coopératifs

Rapport LAAS N°96406

Conférence invitée. Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'96), Rabat (Maroc), 14-17 Octobre 1996, pp.359-375

T.VILLEMUR, V.BAUDIN, S.OWEZARSKI, M.DIAZ

An integrated platform for cooperative teleteaching

Rapport LAAS N°98185

5th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'98), Oslo (Norvège), 8-11 Septembre 1998

Lecture Notes in Computer Science 1483, Eds. T.Plagemann, V.Goebel, 1998, Springer, ISBN 3-540-64955-7, pp.59-70

T.VILLEMUR, P.OWEZARSKI, M.DIAZ

N-TSVS: a videoconference tool for generic cooperative groups

Rapport LAAS N°98186

1998 MultiMedia Modelling (MMM'98), Lausanne (Suisse), 12-15 Octobre 1998, pp.102-111

T.VILLEMUR, M.DIAZ

A collaborative membership service and protocol for structured groups

Rapport LAAS N°99118

International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas (USA), 28 Juin - 1er Juillet 1999, Vol.IV, pp.2115-2121

T.VILLEMUR, K.DRIRA, M.DIAZ

Design of a group membership service on top of a distributed JAVA actor platform

Rapport LAAS N°99117

7th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'99), Cape Town (Afrique du Sud), 20-22 Décembre 1999, pp.119-124

T.VILLEMUR, K.DRIRA*A methodology for the design of collaborative services*

Rapport LAAS N°00076

Conference on Software: Theory and Practice, Beijing (Chine), 21-25 Août 2000, pp.592-599

K.DRIRA, T.VILLEMUR, V.BAUDIN, M.DIAZ*A multi-paradigm layered architecture for synchronous distance learning*

Rapport LAAS N°00126

26th EUROMICRO Conference. Workshop on Multimedia and Telecommunications, Maastricht (Pays-Bas), 5-7 Septembre 2000, Vol.2, pp.158-165

T.VILLEMUR, K.DRIRA*Proposition d'une méthodologie pour la conception de services collaboratifs*

Rapport LAAS N°00147

8ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'2000), Toulouse (France), 17-20 Octobre 2000

*Ingénierie des protocoles. Qualité de service, multimédia et mobilité, Hermes, ISBN 2-7462-0177-1, 2000, pp.429-444***G.J.HOYOS RIVERA, J.P.COURTIAT, T.VILLEMUR***A design framework for collaborative browsing*

Rapport LAAS N°01104

IEEE 10th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'2001), Cambridge (USA), 20-22 Juin 2001, pp.362-367

K.DRIRA, M.DIAZ, T.VILLEMUR, M.JMAIEL, A.E.M.BEN HAMADOU, A.HADJ KACEM*Cooperative systems for information sharing and exchange*

Rapport LAAS N°01442

IEEE 10th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'2001), Cambridge (USA), 20-22 Juin 2001, pp.313-314, (Résumé)

L.M.RODRIGUEZ PERALTA, T.VILLEMUR, K.DRIRA*An XML on-line session model based on graphs for synchronous cooperative groups*

Rapport LAAS N°01101

2001 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2001), Las Vegas (USA), 25-28 Juin 2001, pp.1257-1263

J.M.MOLINA-ESPINOSA, K.DRIRA, T.VILLEMUR*The responsibility management system for collaborative meetings scheduling in the distributed system engineering project*

Rapport LAAS N°02234

IEEE International Workshop on Knowledge Media Networking (KMN'2002), Kyoto (Japon), 10-12 Juillet 2002, pp.114-119

V.BAUDIN, K.DRIRA, T.VILLEMUR, M.DIAZ*Supporting distributed experts in e-meetings for synchronous collaboration*

Rapport LAAS N°02293

2002 IEEE International Conference on Systems Man and Cybernetics (SMC'02), Hammamet (Tunisie), 6-9 Octobre 2002, 6p.

T.VILLEMUR*An actor-based design architecture for distributed synchronizable applications*

Rapport LAAS N°02071

10th International Conference on Software, Telecommunications & Computer Networks (SoftCOM'2002), Dubrovnik (Croatie), 8-11 Octobre 2002, pp.361-365

V.BAUDIN, J.P.COURTIAT, R.L.GOMES, G.J.HOYOS RIVERA, T.VILLEMUR

An e-learning collaborative platform for laboratory education

Rapport LAAS N°03055

4th International Conference on Information Technology Based Higher Education and Training (ITHET'03), Marrakech (Maroc), 7-9 Juillet 2003, pp.200-205

T.VILLEMUR, L.M.RODRIGUEZ PERALTA, K.DRIRA

Conception d'un service de gestion de session orienté modèle pour des groupes collaboratifs synchrones

Rapport LAAS N°03244

10ème Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'2003), Paris (France), 7-10 Octobre 2003, pp.33-49

J.P.COURTIAT, C.DAVARAKIS, M.FAUST, H.KAUFMANN, D.MWANZA, A.TOTTER, T.VILLEMUR

Evaluating lab@Future, a collaborative e-learning laboratory experiments platform

Rapport LAAS N°04245

European Distance and E-Learning Network (EDEN'2004), Budapest (Hongrie), 16-19 Juin 2004, 6p.

E.HAMMAMI, T.VILLEMUR, K.DRIRA

An on-line component deployment system for dynamic collaborative sessions

Rapport LAAS N°04610

IEEE International Symposium and School on Advanced Distributed Systems (ISSADS'2005), Guadalajara (Mexique), 24-28 Janvier 2005, 17p.

E.HAMMAMI, T.VILLEMUR

Déploiement adaptatif des composants dans les sessions collaboratives

Rapport LAAS N°05285

5ème Colloque International sur les Nouvelles Technologies de la Répartition (NOTERE'2005), Gatineau (Canada), 29 Août 1er Septembre 2005, pp.31-40

Manifestations avec actes à diffusion limitée

T.VILLEMUR, M.DIAZ

Services et protocoles pour la gestion d'apartés dans des groupes coopératifs

Rapport LAAS N°97171

Colloque National de Recherche Universitaire : exemple des IUT (CNRIUT'97). Modélisation & Simulation en : Mathématiques, Statistiques, Informatique, Physique, Toulouse (France), 14-16 Mai 1997, 12p.

V.BAUDIN, S.OWEZARSKI, J.L.CAMES, T.VILLEMUR, P.OWEZARSKI, M.DIAZ, J.F.SCHMIDT

Conception d'un environnement de télé-formation synchrone. Projet TOPASE

Rapport LAAS N°98245

1st Conférence Scientifique sur les Nouvelles Technologies de l'Information et de la Communication dans les Formations d'Ingénieurs et dans l'Industrie (NTICF'98), Rouen (France), 18-20 Novembre 1998, pp.53-64

L.M.RODRIGUEZ PERALTA, T.VILLEMUR, K.DRIRA

An XML-based session model for synchronous cooperative groups

Rapport LAAS N°00516

Workshop on Software Architectures for Cooperative Systems (CSCW'2000), Philadelphie (USA), 1er Décembre 2000, 4p.

V.BAUDIN, K.DRIRA, T.VILLEMUR, S.TAZI

Une approche synchrone pour une télé-expertise distribuée

Rapport LAAS N°00557

1ères Journées Francophones des Modèles Formels de l'Interaction (MFI'01), Toulouse (France), 21-23 Mai 2001, Vol.III, pp.363-377

L.M.RODRIGUEZ PERALTA, T.VILLEMUR*Un modèle de session XML basé sur des graphes pour des groupes coopératifs synchrones*

Rapport LAAS N°02006

5èmes Journées Doctorales Informatique et Réseaux (JDIR'2002), Toulouse (France), 4-6 Mars 2002, pp.161-170

L.M.RODRIGUEZ PERALTA, T.VILLEMUR, K.DRIRA, J.M.MOLINA-ESPINOSA*Managing dependencies in dynamic collaborations using coordination diagrams*

Rapport LAAS N°02527

6th International Conference on Principles of Distributed Systems (OPODIS'02), Reims (France), 11-13 Décembre 2002, pp.29-42

E.HAMMAMI, T.VILLEMUR, K.DRIRA*Service de déploiement dynamique de composants coopératifs dans un environnement pair-à-pair*

Rapport LAAS N°03450

4èmes Journées Scientifiques des Jeunes Chercheurs en Génie Electrique et Informatique (GEI'2004), Monastir (Tunisie), 15-17 Mars 2004, 8p.

L.M.RODRIGUEZ PERALTA, T.VILLEMUR, K.DRIRA*A model-driven and web-enabled coordination approach for managing synchronous collaborative sessions*

Rapport LAAS N°04525

5a Conferência da Associação Portuguesa de Sistemas de Informação (CAPSI), Lisbonne (Portugal), 3-5 Novembre 2004, 10p.

Rapports de Contrat**M.DIAZ, T.VILLEMUR***Présentation et classification des applications coopératives*

Rapport LAAS N°93164

Contrat CNET FT N°92.1B.178. Lot 3, Avril 1993, 25p.

M.DIAZ, T.VILLEMUR*Elements de synthèse sur la coopération et sur le travail coopératif*

Rapport LAAS N°93165

Contrat CNET FT N°92.1B.178. Lot 3, Avril 1993, 17p.

M.DIAZ, T.VILLEMUR*Services et protocoles d'adhésion à des groupes coopératifs dynamiques*

Rapport LAAS N°93274

Contrat CNET FT N°92.1B.178. Lot 3, Juillet 1993, 48p.

V.THOMAS-BAUDIN, G.CICCHELERO, M.DIAZ, T.VILLEMUR, F.BAUDIN, J.F.SCHMIDT*Intégration des techniques de synchronisation et de coopération dans l'application de téléenseignement aérofornation VACBI*

Rapport LAAS N°93348

Contrat CNET FT N°92.1B.178. Lot 4, Juin 1993, 40p.

M.DIAZ, P.OWEZARSKI, T.VILLEMUR*Une définition logique de la coopération basée sur le partage des données*

Rapport LAAS N°94008

Contrat CNET FT N°92.1B.178. Projet CESAME. Lot 3, Janvier 1994, 19p.

M.DIAZ, T.VILLEMUR

Formation d'apartés dans des groupes coopératifs d'agents
Rapport LAAS N°94172
Contrat CNET FT N°92.1B.178. Projet CESAME. Lot 3, Avril 1994 , 20p.

L.ROBERT, D.GAUSSENS, T.VILLEMUR

Etat de l'art sur le télé-enseignement
Rapport LAAS N°97226
Projet TOPASE N° 3464, Juin 1997, 42p.

T.VILLEMUR

La coopération dans TOPASE
Rapport LAAS N°97228
Projet TOPASE N° 3464, Juin 1997, 30p.

T.VILLEMUR

Services et protocoles pour gérer des groupes coopératifs
Rapport LAAS N°97229
Projet TOPASE N° 3464, Juin 1997, 28p.

T.VILLEMUR, P.OWEZARSKI, M.DIAZ

Conception et implantation d'une application de vidéoconférence basée sur des groupes coopératifs génériques et structurés
Rapport LAAS N°98108
Convention TOPASE N° 96-2-93-0291, 10 Avril 1998, 24p.

T.VILLEMUR, K.DRIRA

Nouveaux services méthodologies et environnements pour le travail coopératif distribué
Rapport LAAS N°99152
Programme Telecom du CNRS, Projet TL 97028, Mars 1999, 11p.

C.CARDONNEL, T.VILLEMUR

Projet CAMERA. Gestion des échanges : caractérisation des échanges et des protocoles
Rapport LAAS N°99492
Projet CAMERA. Convention N°98.2.93.0326, Novembre 1999, 15p.

C.CARDONNEL, T.VILLEMUR

Projet CAMERA. Gestion des échanges : spécifications et maquetage
Rapport LAAS N°99493
Projet CAMERA. Convention N°98.2.93.0326, Novembre 1999, 17p.

C.CARDONNEL, T.VILLEMUR

Projet CAMERA. Gestion des échanges : protocoles de synchronisation de copies d'applications distribuées
Rapport LAAS N°00069
Projet CAMERA. Convention N°98.2.93.0326, Février 2000, 19p.

T.VILLEMUR, V.BAUDIN, K.DRIRA, S.TAZI

Télé-expertise distribuée
Rapport LAAS N°02040
Projet Région TEDI N° 99008957, Février 2002, 19p.

V.BAUDIN, J.P.COURTIAT, R.L.GOMES, G.J.HOYOS RIVERA, T.VILLEMUR

Lab@Future architecture specification
Rapport LAAS N°03313
Lab@Future, Project IST-2001-34204, Juillet 2003, 84p.

V.BAUDIN, J.P.COURTIAT, R.L.GOMES, G.J.HOYOS RIVERA, T.VILLEMUR
Lab@future development prototype
 Rapport LAAS N°04351
 Lab@Future, Project IST-2001-34204, Juin 2004, 57p.

V.BAUDIN, J.P.COURTIAT, T.VILLEMUR
Lab@Future evaluation prototype
 Rapport LAAS N°04686
 Lab@Future, Project IST-2001-34204, Décembre 2004, 37p.

Rapports LAAS

M.DIAZ, T.VILLEMUR, F.VERNADAT, P.AZEMA
Verification of services and protocols for dynamic membership in cooperative groups
 Rapport LAAS N°94039, Janvier 1994, 7p.

P.OWEZARSKI, T.VILLEMUR, M.DIAZ
Conception et implémentation d'un système de visioconférence coopératif
 Rapport LAAS N°96401, Octobre 1996, 35p.

T.VILLEMUR, P.OWEZARSKI, M.DIAZ
Design and implementation of a videoconferencing application based on generic structured cooperative groups
 Rapport LAAS N°98221, Mai 1998, 33p.

T.VILLEMUR, V.BAUDIN, S.OWEZARSKI, M.DIAZ
TOPASE : une plate-forme intégrée support de téléenseignement synchrone
 Rapport LAAS N°98377, Septembre 1998, 18p.

T.VILLEMUR, V.BAUDIN, S.OWEZARSKI, M.DIAZ
A cooperative group model and its application to a multimedia virtual meeting platform
 Rapport LAAS N°98380, Septembre 1998, 33p.

K.DRIRA, T.VILLEMUR, V.BAUDIN, M.DIAZ
A multi-paradigm layered architecture for distance learning using the Web
 Rapport LAAS N°99496, Novembre 1999, 30p.

L.M.RODRIGUEZ PERALTA, T.VILLEMUR, K.DRIRA
An XML architecture for a synchronous session model
 Rapport LAAS N°00506, Novembre 2000, 23p.

T.VILLEMUR
Formal design of a distributed input multiplexer
 Rapport LAAS N°01043, Février 2001, 15p.

T.VILLEMUR, V.BAUDIN, K.DRIRA, M.DIAZ, X.GUANGYOU, S.YUANCHUN
A comparative framework for synchronous distance learning environments
 Rapport LAAS N°01096, Mars 2001, 4p.

E.CHEN, T.VILLEMUR
Co-view. A collaborative Web Browser based on peer-to-peer network
 Rapport LAAS N°01465, Septembre 2001, 50p.

L.M.RODRIGUEZ PERALTA, T.VILLEMUR, K.DRIRA

Design of a coordinated session management service
Rapport LAAS N°01501, Novembre 2001, 15p.

T.VILLEMUR, L.M.RODRIGUEZ PERALTA, K.DRIRA

Conception d'un service de gestion de session coordonnée
Rapport LAAS N°02042, Février 2002, 12p.

L.M.RODRIGUEZ PERALTA, T.VILLEMUR, K.DRIRA, J.M.MOLINA-ESPINOSA

Managing dependencies in dynamic collaborations using coordination diagrams
Rapport LAAS N°02158, Mars 2002, 10p.

J.M.MOLINA-ESPINOSA, K.DRIRA, T.VILLEMUR

The responsibility management system for collaborative meetings scheduling in the distributed system engineering project
Rapport LAAS N°02356, Septembre 2002, 11p.

E.HAMMAMI, T.VILLEMUR, K.DRIRA

Déploiement en ligne de composants pour des sessions collaboratives dynamiques
Rapport LAAS N°03523, Décembre 2003, 16p.

T.VILLEMUR, K.DRIRA

CSCW experience for distributed system engineering
Rapport LAAS N°04357, Juin 2004, 18p.

S.MOTA GONZALEZ , B.FONTAN, T.VILLEMUR, P.DE SAQUI SANNES, J.P.COURTIAT

UML-based modeling and formal verification of authentication protocols
Rapport LAAS N°05247, Mai 2005, 13p.

4. Encadrement de stages/DEAs/thèses

J'ai encadré et co-encadré depuis la fin de ma thèse jusqu'à la période actuelle :

Stage de licence	Stages de maîtrise	Stages d'ingénieurs	DEAs	Thèse soutenue	Thèses en cours
1	3	4	4	1	2

Encadrement terminé

Stages Ingénieur ENSEEIHT. Stéphane NICOLAS et Najla TRIKI (avril 1995 – juin 1995)

Etude et amélioration de services et de protocoles pour la gestion de groupes coopératifs

Stage Ingénieur ENAC. Roland HAMON (février 1996 – juin 1996)

Interface graphique pour visioconférence synchronisée à N utilisateurs

Stage Ingénieur ENSEEIHT et DEA. Lanto RANDRIAMIHARISOA (juillet 1997 – juin 1998 et juin 1998 – septembre 1998)

Intégration de services de gestion de sous-groupes (apartés) à l'intérieur d'une visioconférence structurée

Stage deuxième année IUP. Cédric CARDONNEL (mai 1998 – août 1998)

Conception et réalisation d'une interface graphique pour la gestion de groupes coopératifs

Stage troisième année IUP. Cédric CARDONNEL (avril 1999 – août 1999)

Etude et maquetage d'un environnement de télé-apprentissage

Stage Maîtrise Université de Bucarest, Roumanie. Mirela IONESCU (août 1999 – décembre 1999)

Intégration d'un service de gestion de groupes dans la plate-forme de coordination CDK

Stage DEA. Cédric CARDONNEL (septembre 99-juin 2000)

Outils de partage d'applications pour du téléenseignement synchrone

Doctorat. Laura RODRIGUEZ-PERALTA (octobre 99-avril 2003)

Etude et Proposition de nouveaux services de coopération-coordination pour des applications multimédias

Stage DEA co-encadré avec K. Drira. Adil HBABI (septembre 2000 – juin 2001)

Etude et adaptation de l'environnement Coordination Development toolKit au support CORBA Orbacus 4

Stage Ingénieur université de Tsinghua Chine. Enyi CHEN (mai 2001 – septembre 2001)

Co-view. A collaborative Web Browser based on peer-to-peer network

Stage Ingénieur INSA co-encadré avec V. Baudin. Carim KHOYRATTY (mars 2002 – juin 2002)

Conception d'un outil de synchronisation d'applications distribuées multimédia

Stage Ingénieur INSAT et DEA. Emir HAMMAMI (septembre 2002 – juin 2003)

Service Web pour le déploiement automatique de composants dans des sessions collaboratives

Stage 4ème année Ingénieur INSA. Lucie BOULPICANTE (juillet 2005 – août 2005)

Standards et techniques pour la visualisation de graphes dynamiques

Encadrement en cours

L'encadrement en cours comprend deux thèses.

Doctorat en cours co-encadré avec M. Diaz.

Emir HAMMAMI (octobre 2003 – octobre 2006)

Services Web pour la cohérence automatique des composants de sessions collaboratives dynamiques complexes

Doctorat en cours co-encadré avec M. Diaz.

Sara MOTA GONZALEZ (octobre 2004 – octobre 2007)

Protocoles pour des communications de groupes sécurisés dans des sessions multi utilisateurs

5. Animation scientifique et rayonnement

Collaboration	Animation de groupe de travail	Participation à des comités scientifiques	Co-organisation de conférences
1	1	4	2

Animation

Gestion de la collaboration entre le LAAS et l'Université de Tsinghua (Chine) dans le cadre du programme PRA SI99-01 : "Services CORBA pour la coordination des activités distribuées". Cette collaboration bilatérale a duré deux ans, d'octobre 1999 à septembre 2001. Elle a permis la visite d'un chercheur du LAAS à l'université de Tsinghua, la visite de deux chercheurs de cette université pendant quinze jours au LAAS, et l'accueil en stage d'un élève ingénieur chinois pendant cinq mois.

Co-animation (avec J.P. Jessel et L. Charon), de janvier 2003 à décembre 2003, du groupe de travail "Interactions Hommes Systèmes" dans le cadre de la fédération FERIA qui regroupe les laboratoires IRIT, LAAS et ONERA.

Conférences

Co-organisateur (avec M. Diaz et K. Drira) d'une session spéciale sur la collaboration dans les applications réseaux multimédia et les collecticiels, dans la conférence "EUROMICRO Workshop on Multimedia and Telecommunication", Maastricht, Hollande, 4-7 septembre 2000.

Co-organisation (avec J.P. Courtiat et V. Baudin) de l'atelier Tel'04 (Technology Enhanced Learning) qui s'est déroulé le 22 août 2004 dans le cadre de la conférence IFIP World Computer Congress.

Co-organisation (avec P. de Saqui-Sannes et K. Drira) de la conférence NOTERE'06 (Nouvelles Technologies de la Répartition) qui s'est déroulée du 6 au 9 juin 2006.

Comités scientifiques

Membre du comité de programme des conférences ISADS'2000, ISADS'2002 et ISADS'2005 (International Symposium on Advanced Distributed Systems).

Membre du comité scientifique de la revue Interactive Technology and Smart Education, depuis février 2004.

6. Relations avec le monde industriel

Projets auxquels j'ai participé

TOPASE (juin 1996 – décembre 1997) projet "Autoroutes de l'Information" sur la Téléformation Avancée pour l'Aéronautique et l'Espace

Ce Projet a été une collaboration AIRBUS - ISSAT - LAAS - SYNELEC - MMS/ARAMIHS - IRISA - ENSICA sur les Systèmes Distribués d'Enseignement Professionnel Multimédia. Le maître d'œuvre était AIRBUS. TOPASE a étudié la distribution de l'enseignement des pilotes et des agents de maintenance des avions AIRBUS, maintenant délivré de façon classique, sur PC dans une classe habituelle. Un concept de classe virtuelle a été proposé, l'instructeur étant dans un centre AIRBUS (Toulouse, Miami, Pékin) et les apprenants dans les villes des compagnies aériennes. Nous avons participé au développement et à l'évaluation d'une plate-forme prototype qui met en œuvre le concept de classe virtuelle au travers de réseaux informatiques et qui supporte les interactions des participants distribués.

DSE (janvier 2000 – janvier 2002), avec Alenia, EADS, IABG, SILOGIC, D3, LIP6

DSE, du programme IST de la Communauté Européenne, a eu pour objectif de réaliser un support logiciel pour l'ingénierie distribuée dans le domaine de l'espace. Il a inclus la définition des deux niveaux, réseau et applicatif, puis la conception et la réalisation de la couche middleware nécessaire pour coordonner les activités distribuées. Dans ce cadre, nous avons participé à la spécification et à la réalisation du logiciel support, analysé les processus de conception, proposé une méthodologie de modélisation et d'implémentation, et développé les services de gestion des groupes et des sessions.

Projets avec responsabilité administrative/scientifique

"Nouveaux services et nouvelles méthodologies pour le travail coopératif " projet Appel d'Offre Télécommunications 97 du CNRS (septembre 1997 – septembre 1999)

Ce projet a impliqué deux laboratoires de recherche, d'un côté le LAAS-CNRS et l'équipe ICARE de l'IUT B Blagnac. Le maître d'œuvre était le LAAS-CNRS. Il s'agissait de traiter les questions liées au problème générique de la coopération et de la coordination dans les systèmes coopératifs distribués communicants. On a proposé d'un côté, puis mis en œuvre une technique de conception et d'implantation haut niveau. D'un autre côté, on a cherché à définir et à implanter des protocoles de gestion et de coordination des groupes de travail.

J'avais en charge la responsabilité administrative, financière (30 kEuros), et scientifique de ce projet.

CAMERA (décembre 1998 - décembre 2000) projet "Autoroutes de l'Information" sur le Télé-apprentissage et la Télé-assistance en milieu professionnel ou opérationnel.

Ce projet a bénéficié d'un financement du Ministère de l'Industrie, et a impliqué trois laboratoires de recherche (LAAS/IRIT/CESBIO) et cinq partenaires industriels (MATRA MS&I, CLARIS, A6, STAR, REALIX). Le maître d'œuvre était MATRA-MS&I. Le but principal du projet a été l'étude, la proposition et l'intégration d'une plate-forme logicielle didactique permettant

le télé-apprentissage et la télé-assistance au-dessus de moyens de communication banalisés de type RNIS, Internet, avec quelques extensions vers des réseaux à haut débit.

J'avais en charge la responsabilité administrative, financière (60 kEuros pour le LAAS) et scientifique du LAAS pour ce projet.

TEDI (septembre 2000 – septembre 2001), projet région Midi-Pyrénées sur la Télé Expertise Distribuée, impliquant le LAAS, l'ENSICA, le LIHS, les sociétés CLAIRIS et SILOGIC.

Le but original de ce projet, dont le maître d'œuvre était le LAAS, a été de concevoir une plate-forme logicielle multimédia composée d'outils de collaboration synchrone qui mettent en relation des experts et des formateurs. Les activités des experts et des formateurs se déroulaient de façon intégrée, tant au niveau des communications que de la mise en œuvre de leurs méthodes de travail personnelles que de leur pédagogie. Ceci a permis de généraliser notre approche au domaine plus vaste de la télé expertise.

J'ai eu en charge la responsabilité scientifique de l'ensemble du projet.

Lab@Future (mai 2002 – avril 2005), avec Systema Informatics, Thales, Parallel Graphics, Université de Brème, Université de Vienne, CSV, Université d'Helsinki et ETH Zurich.

Ce projet européen IST (www.labfuture.net), dont le maître d'œuvre était Systema Informatics, a eu pour but de développer une plate-forme collaborative pour expérimenter des théories pédagogiques (constructivisme social, théorie de l'activité et apprentissage expansif) dans un contexte de réalité virtuelle et augmentée, avec des utilisateurs mobiles. Notre contribution a porté sur la réalisation d'une plate-forme de communication et de collaboration qui assure les échanges entre utilisateurs distribués géographiquement, et qui gère et coordonne les sessions d'utilisateurs.

J'ai eu en charge, en plus de ma participation scientifique, la partie administrative et financière (180 kEuros pour le LAAS) du partenaire LAAS pour ce projet.

Valorisation scientifique

L'environnement prototype logiciel PLATINE (PLAte forme de Télé-Ingénierie coopérative synchrone distribuée), développé dans le cadre des activités de recherche, a entraîné en 2001 un dépôt de dossier de valorisation auprès du CNRS. Le code du prototype PLATINE sera diffusé sous licence logiciel libre.

Je suis associé à cette démarche en tant que co-auteur du prototype.

Partie 2

Synthèse des travaux

Introduction

Le travail collaboratif est un domaine applicatif assez récent (son apparition date des années 90) qui correspond à l'utilisation de systèmes informatiques pour supporter le travail de groupes distribués d'utilisateurs connectés en réseau. Il s'agit d'un domaine multidisciplinaire qui, en plus des domaines informatiques, bénéficie de l'apport des sciences sociales (théorie du travail et des organisations) et des sciences cognitives (planification, gestion de tâches de groupe).

L'approche que nous allons suivre se propose d'étudier le travail collaboratif sous l'angle de l'informatique répartie : notre démarche fait partie du domaine « conception des systèmes distribués informatiques », adaptée aux particularités des systèmes collaboratifs. Nous nous proposons de travailler dans ce domaine en terme de méthodologies et d'environnements support qui faciliteront et assisteront la mise au point, l'activation et la gestion de plateformes composées d'outils collecticiels multimédias. De façon plus précise, nous nous intéressons à la conception de services de communication orientés modèle pour le domaine du travail collaboratif.

Démarche de recherche

Pour guider et illustrer notre démarche de recherche, nous avons défini un canevas qui se compose de trois classes : la classe *Modèles*, la classe *Services de Coordination* et la classe *Travail Collaboratif*. Chacune de ces classes donne le cadre dans lequel les résultats proposés s'inscrivent. Un lien entre deux classes donne la contribution que la classe source apporte à la classe destination. La figure 0.1 synthétise l'ensemble de notre démarche de recherche.

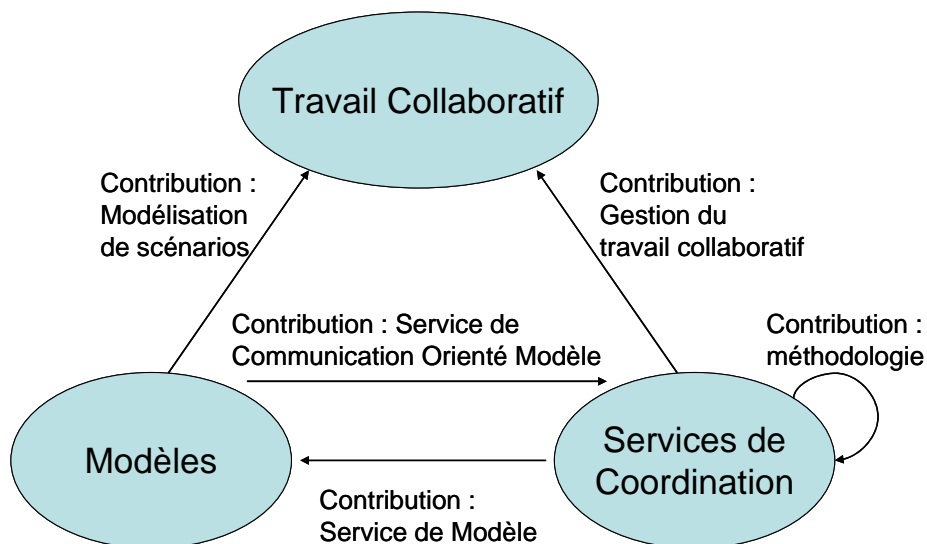


Figure 0.1. Canevas de notre projet de recherche

La classe *Modèles* définit une des caractéristiques fondamentales de notre démarche. L'intérêt premier de modéliser un système en le transcrivant dans un formalisme mathématique est de lui associer certaines propriétés que l'on pourra vérifier formellement grâce au formalisme retenu. Ces propriétés dépendent bien entendu du type de formalisme utilisé ainsi que de son pouvoir d'expression. Dans le cadre de nos études, comme première contribution, le modèle retenu a servi à définir plusieurs scénarios de sessions de travail

collaboratif. Un autre intérêt important des modèles est de pouvoir être utilisés et instanciés par des services de coordination. Cette deuxième contribution garantit la généricité de ces services et, dans un contexte d'ingénierie des systèmes, participe à la définition et à la conception de services de coordination programmables par modèle.

Les *Services de Coordination* que nous nous proposons de définir et de concevoir dans la classe de même nom, sont des entités logicielles qui contrôlent et supervisent les différents utilisateurs qui travaillent en groupe de façon distribuée. Dans notre cadre de travail, comme première contribution, les services de coordination gèrent les différents éléments requis (utilisateurs, documents, informations) pour la réalisation de sessions cohérentes de travail collaboratif. En deuxième contribution, nous nous intéressons à différentes méthodologies et architectures pour décrire ces services, en privilégiant leurs approches de conception formelles. Du point de vue architectural, en troisième contribution, nous proposons la définition d'un service de modèle qui ne gère que les éléments du modèle indépendamment de leur signification.

La classe *Travail Collaboratif*, sous son aspect informatique répartie, représente finalement notre domaine d'application. Cependant, du fait de son aspect multidisciplinaire originel et de sa relation directe avec l'utilisateur final, le lien avec les autres domaines - notamment les Interfaces Homme Machine et les Sciences Sociales - doit être conservé. Ceci est nécessaire pour quantifier l'« utilisabilité » et l'apport supplémentaire de ces environnements collecticiels pour le travail de groupe. Ce lien a notamment été maintenu au travers de plusieurs projets contractuels qui ont permis la mise en place d'évaluations utilisateur sur les plans techniques et applicatifs.

Le canevas de la figure 0.1 peut s'instancier par les principaux résultats que nous avons obtenus. Chaque classe est instanciée par le résultat central de chacune d'entre elle (figure 0.2). Les flèches donnent les contributions de chacun de ces résultats, vers une autre partie qui est utilisatrice de ces résultats.

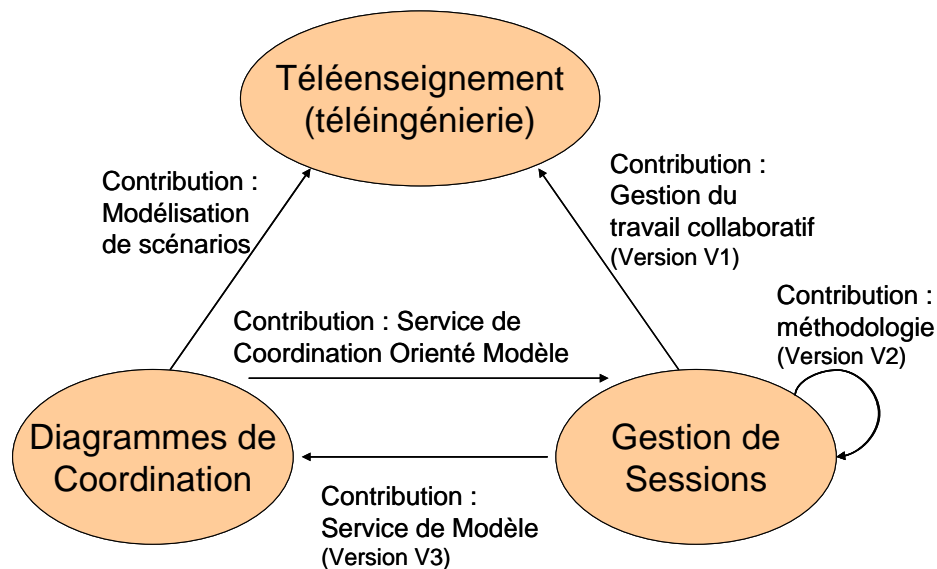


Figure 0.2. Résultats obtenus

Chacun des principaux résultats sera détaillé par la suite. Le résultat *Diagramme de Coordination*, avec ses contributions en terme de service de coordination orienté modèle et de modélisations de scénarios, et son application au *Téléenseignement* et à la *Téléingénierie*,

seront développés dans le chapitre 2. Les résultats en terme de *Gestion de Sessions*, avec leurs contributions respectives, et leur application au *Téléenseignement*, seront approfondis dans le chapitre 3. Le résultat *Téléenseignement/Téléingénierie* désigne les domaines particuliers d'application de nos travaux, domaines liés aux cadres contractuels.

Plan de la synthèse des travaux

Le plan du mémoire reprend les points isolés dans notre démarche de recherche.

Le premier chapitre situe le contexte de nos travaux. De ce fait, il développe le canevas de notre projet de recherche (figure 0.1). Bien entendu, nous n'avons pas la prétention d'une description exhaustive : nous orienterons et focaliserons nos descriptions vers les résultats obtenus, décrits dans les chapitres suivants.

Pour la classe *Travail Collaboratif*, nous définissons tout d'abord les principaux termes et mots-clés qui font partie du travail collaboratif (Collaboration, Sessions, Interactions synchrones/asynchrones, Collecticiel). Par la suite, nous procédons à une classification des principaux outils collecticiels selon leur domaine d'interaction asynchrone ou synchrone, et nous analysons leur évolution au cours de ces dix dernières années, sur les plans fonctionnels, architecturaux et technologiques.

Pour la classe *Modèles*, nous présentons, classons et analysons un ensemble de modèles significatifs qui ont été utilisés pour représenter des sessions.

Pour la classe *Services de Coordination*, nous nous focalisons sur les plates-formes qui gèrent des sessions : nous présenterons une analyse des principaux gestionnaires de sessions existants que nous comparons et classifions.

Cette partie préambule illustre les besoins en services de gestion de session de travail collaboratif et de coordination des outils collecticiels.

Le deuxième chapitre présente nos résultats en terme de modélisation de sessions. Il décrit l'approche orientée modèle pour représenter des sessions collaboratives et présente les trois éléments à co-gérer (utilisateurs, outils, données) pour ces sessions. Par la suite, il détaille l'instance *Diagrammes de Coordination* et décrit les deux contributions vers les instances *Gestion de Sessions* et *Téléenseignement/Téléingénierie* de la figure 0.2.

Pour l'instance *Diagrammes de Coordination*, nous décrivons le modèle de diagrammes de coordination, qui s'appuie sur des graphes étiquetés dynamiques, et que nous avons retenus pour modéliser les sessions collaboratives et pour lier entre eux leurs éléments.

Le lien vers l'instance *Gestion de Sessions* se fait par l'implantation du modèle, implantation qui est utilisée par un service de gestion de session orienté modèle. Cette réalisation en XML facilite l'affichage de l'état de la session au travers de n'importe quel navigateur XML ; elle renforce de ce fait la prise de conscience du groupe par les utilisateurs. Finalement, cette implantation remplit le but principal recherché par notre contribution : obtenir un modèle programmable et utilisable de façon automatique par une couche logicielle de gestion de session.

Le lien vers l'instance *Téléenseignement/Téléingénierie* se fait par la modélisation de scénarios (figure 0.2). La facilité de représentation de l'état de la session et de son évolution dans le temps est illustrée par la spécification de plusieurs scénarios provenant du domaine du

téléenseignement : formation de pilotes d'avion et d'agents de maintenance, puis accès à des laboratoires virtuels distribués pour des élèves.

Le troisième chapitre traite de la gestion de sessions collaboratives. Il est centré sur la conception d'un service de gestion de session programmable par modèle. Ce service gère l'évolution du groupe dans le temps et garantit que l'état courant de la session, donné par le modèle, correspond bien à l'état réel des utilisateurs connectés en session. Ce service est le résultat de l'instance *Gestion de Sessions* de la figure 0.2.

Notre première contribution vers l'instance *Téléenseignement/Téléingénierie* porte sur un point de vue conception. Elle met en avant, lors de la réalisation de ce service, le grand intérêt des techniques de description formelles en terme de qualité de réalisation et de possibilité d'implantation automatique. Cette implantation automatique a été utilisée dans un prototype support de scénarios de téléenseignement collaboratif.

Notre deuxième contribution qui reboucle sur l'instance *Gestion de Sessions*, porte sur un point de vue méthodologique. Une nouvelle conception du service de gestion de session se fait selon une approche bi modèle, l'un pour gérer l'aspect structurel du système, l'autre pour l'aspect comportemental de ses composants élémentaires. Cette nouvelle implantation se fait dans un environnement JAVA/acteur.

Notre troisième contribution vers l'instance *Diagrammes de Coordination* considère l'aspect architectural. Pour notre dernière version prototype, nous proposons un nouveau découpage en introduisant une couche de modèle, générique, utilisable par des services de plus haut niveau, dans notre cas un service de gestion de session. Ceci augmente la modularité et la réutilisabilité des éléments de notre système.

Chapitre 1. Cadre de nos travaux

Ce premier chapitre situe le contexte de nos travaux, le travail collaboratif, selon le canevas du projet de recherche composé des trois classes : *Travail Collaboratif*, *Modèles*, *Services de Coordination* précédemment établi. Il illustre les besoins en terme de modèles de session, de services de gestion de session de travail collaboratif et de coordination des outils collecticiels.

Ce chapitre se compose des parties suivantes : La partie 1 définit les principaux termes et mots-clés qui font partie du travail collaboratif. Les parties 2 et 3 procèdent à une classification des principaux outils collecticiels respectivement selon les domaines d'interaction asynchrone, puis synchrone, tout en analysant leur évolution au cours de ces dix dernières années, sur les plans fonctionnels, architecturaux et technologiques. Les trois parties 1, 2 et 3 représentent la classe *Travail Collaboratif* de notre canevas de recherche. Dans la partie 4, qui représente la classe *Modèles* de notre canevas, un ensemble de modèles significatifs qui a été utilisé pour représenter des sessions sont présentés, classés et analysés. La partie 5, associée à la classe *Services de Coordination* de notre canevas de recherche, répertorie, analyse et classe les principaux gestionnaires de sessions que nous avons rencontrés.

1. Travail collaboratif. Définitions

Le travail collaboratif [LONC03], appelé à l'origine travail coopératif (ou Computer Supported Cooperative Work), correspond à l'utilisation d'ordinateurs en réseau pour le travail de groupes d'utilisateurs. Pour clarifier nos propos, deux citations très générales, données par Ellis et Kraemer en 91 et 88, définissent bien le domaine des systèmes collaboratifs [ELLI91], [KRAE88] :

Système à base d'ordinateurs qui supporte des groupes de personnes réalisant en commun une tâche ou un but et qui fournit une interface pour accéder à un environnement commun

Système informatique qui facilite la résolution de problèmes par un ensemble de décideurs travaillant en groupe

Apparu dans les années 90, le travail collaboratif est issu de quatre grands domaines :

- Sciences sociales (sociologie, théorie des organisations), prenant en compte l'organisation des personnes, leurs apports, l'efficacité d'un groupe
- Intelligence artificielle distribuée, sciences cognitives pour l'interprétation de la sémantique des informations, la planification, l'aide à la réalisation de tâches en commun
- Interfaces homme machine, pour la conception d'interfaces multiutilisateurs
- Informatique répartie, systèmes distribués et réseaux pour les stockages, transferts et échanges d'information

Le travail collaboratif et les activités de groupe sont organisés en sessions. Une session, unité essentielle du travail en groupe, se compose d'un ensemble d'utilisateurs qui partagent des intérêts communs [DOMM99]. Ces utilisateurs se connectent de divers emplacements et manipulent des données, partagées à l'aide des outils support de la collaboration [DRIR01].

Une session peut être synchrone ou asynchrone.

Dans une Session Synchrone, les membres du groupe sont présents en même temps pour réaliser le travail collaboratif. Ils dialoguent de façon interactive et interagissent en temps-réel sur les outils et les données manipulées dans la session. Un exemple de ce type de session, est une session de téléconférence.

Dans une Session Asynchrone, la co-présence des différents membres du groupe n'est pas nécessaire. De ce fait, ne pouvant réaliser de dialogue interactif, les membres d'une session asynchrone ne communiquent que via des médiums asynchrones tel que l'email.

La séparation entre Session Asynchrone et Session Synchrone, introduite à l'origine à cause des caractéristiques techniques des réseaux sous-jacents, est bien entendu fortement liée au type de travail de groupe réalisé. Certains travaux collaboratifs peuvent cependant se dérouler dans des modes asynchrones, ou synchrones. Par exemple, la production d'un document au travers d'un éditeur collaboratif peut se faire suivant les deux modes. L'élaboration des parties du document peut se faire de façon asynchrone. Cependant, des phases de coordination synchrones peuvent être nécessaires pour faciliter la production d'un document unique et cohérent.

Par la suite, nous avons gardé cette séparation asynchrone/synchrone comme base de notre classification, pour plus de clarté.

Le collecticiel (ou groupware) [KARS94] désigne tous les produits logiciels, outils, services ou plates-formes, conçus pour des groupes d'utilisateurs.

2. Travail collaboratif. Collecticiel asynchrone

Les systèmes asynchrones ont été les premiers développés, à partir des années 90. Nous les avons classés en deux catégories :

- Les messageries
- Les systèmes basés sur le Web

2.1 Messageries avancées

Ces systèmes offrent des fonctionnalités supplémentaires par rapport aux messageries classiques, avec la prise en compte de groupes d'utilisateurs, de leur organisation et de leurs caractéristiques respectives. Les premiers systèmes ont été très utilisés dans les domaines d'assistance des procédures de bureautique ou de contrôle du déroulement d'opérations (workflow) [AALS02]. Par la suite, ces messageries ont été couplées avec des bases de données classiques, puis elles ont évolué en des messageries multimédias complexes. Pour illustrer cette évolution, nous citons trois réalisations : un projet de recherche, un outil courant et un projet qui a débouché sur un prototype avancé.

2.1.1 GGroup Activity Environment

Le projet de recherche GGroup Activity Environment (GRACE) [BENF91], [BENF92] s'est déroulé sur deux ans pour développer des communications asynchrones de groupe dans un environnement Open System Interconnection (OSI). Il a permis de développer un modèle conceptuel de groupes asynchrones dont les membres partagent des informations, et de définir un service de communications de groupe pouvant être utilisés par un vaste ensemble d'applications collaboratives asynchrones. Ce service s'appuie sur la messagerie électronique X.400 [ISO10021]. Les informations manipulées sont structurées de façon hiérarchique, et sont regroupées en domaines qui permettent de relier les membres des groupes aux différentes données manipulées. Les objets du système sont nommés au moyen du répertoire X.500 [ISO9594].

Ces travaux ont permis la réalisation de deux applications prototype. La première est un service d'aide (Help Desk) pour un département informatisé. Les personnes connectées au système émettent des demandes lorsqu'elles rencontrent des problèmes. Le système trie et aiguille les questions suivant les compétences des gens. Les personnes ayant un rôle de spécialiste reçoivent directement ces demandes, et peuvent formuler les réponses. La deuxième application proposée est une conférence asynchrone entre groupes d'utilisateurs, équivalente à un tableau de bulletins. Chaque domaine forme un forum portant sur un sujet précis, le contrôle des informations échangées étant fait par un membre modérateur. Les personnes rejoignent ou quittent les domaines définis, suivant leurs centres d'intérêt.

2.1.2 Lotus Notes

Lotus Notes [LOTU03], [CHIV03] est un système collecticiel basé sur Internet pour la collaboration asynchrone entre groupes de travail. Sa première version a été réalisée en 1982. Lotus Notes fonctionne avec un type unique de base de donnée qui permet la combinaison de données structurées (relationnelles) et non structurées (documents, textes...). En liant les communications par courrier électronique à ce type de base de données, Lotus Notes permet la création de canaux de communication et de partage de documents qui peuvent être dédiés à certaines fonctions particulières ou à certains groupes de travail (groupe des développeurs, groupe marketing...), à l'intérieur d'une organisation.

Lotus Notes se compose de :

- une plate-forme de discussion avec une messagerie et des groupes de discussion,
- un générateur de base de donnée qui gère l'organisation des fichiers et des données,
- un créateur et indexeur de documents qui leur attribue différentes propriétés et différents niveaux d'accès.

Ce découpage facilite l'organisation, le partage, le stockage et les modifications qui peuvent être réalisées par plusieurs utilisateurs.

Les documents non structurés des utilisateurs de Lotus Notes sont inclus dans la base de données qui leur associe des champs utilisables pour la recherche et l'indexation. Un lien étroit est maintenu entre les messages échangés et les documents de la base de données. Les documents sont classés dans des sections organisées de façon hiérarchique. De plus, Lotus Notes autorise et maintient la création de fils de discussion, en ordonnant les messages requête et réponses, et en les attachant au documents de référence de la discussion. Toute recherche renvoie le dernier état de la discussion, et garantit que les documents de toute discussion en cours, potentiellement modifiés, sont mis à jour avant d'y accéder.

Les dernières versions de Lotus Notes s'interfacent avec les navigateurs classiques.

Il existe d'autres environnements concurrents, mais qui offrent des fonctionnalités similaires. Les plus répandus sont :

- Microsoft Exchange [EXCH03]
- Novell GroupWise [GROU03]

2.1.3 Design Implementation and operation of a distributed ANnotation Environment

L'application DIANE [DIAN98], [DRIR01], issue du projet européen Design Implementation and operation of a distributed ANnotation Environment (juillet 1996 - septembre 1998), est un système d'annotations multimédia distribuées capable de manipuler des données informatiques, des liens hypertextes, des sons, des images, des captures d'écran, et des mouvements de souris. Les annotations créées sur n'importe quelle application, sont stockées dans des espaces partagés qui autorisent leur recherche et leur présentation.

L'architecture de DIANE, distribuée, se compose de terminaux utilisateurs et de serveurs d'annotations, qui sont utilisés pour stocker et pour accéder aux documents annotés. DIANE inclut un système de transport de données qui autorise l'accès aux serveurs d'annotations soit de façon interactive (temps réel), soit de façon asynchrone (par message expédié, puis stocké dans une messagerie), suivant le type de réseau utilisé. DIANE permet ainsi l'utilisation de réseaux rapides tels que ATM ou plus conventionnels comme le réseau téléphonique commuté. La présentation des documents multimédias se fait en synchronisant les flux de données qui les composent.

Le développement de DIANE est fait en JAVA. Le prototype a été évalué grâce à des essais dans plusieurs domaines d'utilisation, en particulier :

La télé-médecine: Les médecins pathologistes du département des pathologies de l'Hôpital Général de Mandesa et de l'Hôpital Universitaire du Vall d'Ebron à Barcelone (Espagne) ont utilisé DIANE pour se consulter les uns les autres en cas de diagnostic difficile et pour préparer des sessions de téléenseignement sur des sujets spécifiques.

Le télé-apprentissage: Le laboratoire European Centre for Parallel Computing at Vienna (VCPC) est un centre spécialisé dans les outils et techniques de programmation parallèle. Il a produit avec DIANE des supports d'apprentissage sous la forme d'unités de cours.

2.1.4 Evolution des messageries avancées

La figure 1.1 résume l'évolution des systèmes de messagerie collaboratives. Les messages textuels sont manipulés au travers de fonctions avancées par les systèmes des années 80. Dans les années 90, les systèmes de messagerie collaboratives incluent des bases de données qui associent les messages à des contextes, maintenant ainsi l'ensemble des conversations liées à un sujet. L'arrivée du multimédia dans les années 95 enrichit et transforme les contenus textuels. Finalement, l'évolution la plus proche de nous concerne l'intégration des systèmes de messagerie collaboratives aux navigateurs Web, ces derniers servant d'interface utilisateur universel pour l'accès au courrier électronique.

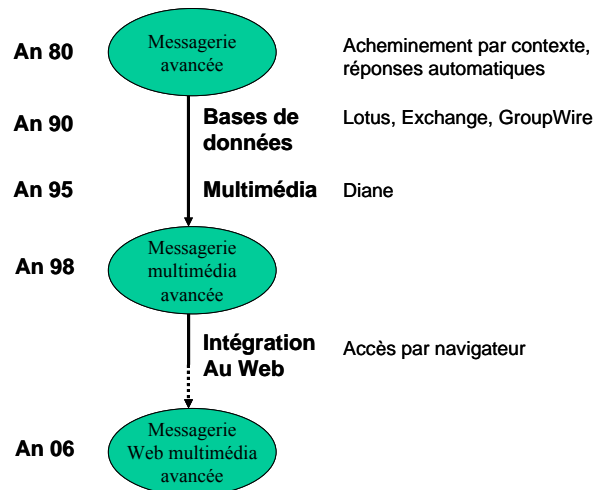


Figure 1.1. Evolution des systèmes de messagerie collaboratives

2.2 Systèmes basés sur le Web

Le Web a été créé à l'origine dans les années 90 pour offrir une interface unique pour l'accès à des données diverses, hétérogènes et distribuées. Cette interface unique, appelée navigateur, communique avec des serveurs Web et utilisant le protocole Hyper Text Transfer Protocol (HTTP). Au fur et à mesure de l'évolution technologique du Web, les navigateurs ont été amenés à accéder et à traiter de nouvelles données dont les valeurs changent dans le temps (données dynamiques), qui évoluent de façon interactive (formulaires), et qui possèdent des contraintes temporelles (flux multimédias). La dernière évolution du Web concerne l'intégration de l'accès à des applications, les navigateurs devenant l'interface distante de ces applications. On parle souvent dans ce cas de portails Web.

D'après la définition du grand dictionnaire terminologique [TERM04], un portail est un « Site Web dont la page d'accueil propose, en plus d'un moteur de recherche, des hyperliens avec une foule d'informations, et des services applicatifs utiles et attrayants ». Comme exemple de service, on peut trouver un serveur de courrier électronique gratuit, une sélection de moteurs de recherche, des actualités, la météo, les cotes de la Bourse, des raccourcis pour les achats en ligne.

« L'objectif des portails est, du point de vue de leurs créateurs, d'attirer et de fidéliser les internautes au point de devenir leur porte d'entrée dans Internet, c'est-à-dire la page de démarrage du plus grand nombre d'entre eux, ce qui peut entraîner des revenus importants, notamment en publicité et en vente de liens.

Il existe des portails généralistes et des portails thématiques, dont certains sont personnalisables. Yahoo.com, Excite.com, France.com et Canada.com sont des exemples de portails. »

La dimension collaborative a été ajoutée au Web dans les années 95, avec notamment l'outil Basic Support for Collaborative Work (BSCW) [BSCW04]. Cet outil généraliste, en plus de l'accès aux documents stockés sur un serveur, définit des groupes d'utilisateurs avec des droits d'accès. Ces droits utilisateurs structurent l'espace des documents et définissent quelles informations peuvent être accédées par les utilisateurs. D'autres outils sont plus ciblés vers des domaines spécifiques, comme le téléenseignement.

2.2.1 WebCT

La plate-forme WebCT [WEBC03] [PORT03] est un système basé sur Internet qui permet le développement, la gestion, puis la diffusion asynchrone de modules de formation. Le coeur de WebCT s'appuie sur un serveur Web.

Le mode auteur, accessible à distance, assiste le concepteur et lui permet d'organiser le contenu de ses cours en pages et en modules. Le contenu de ces modules peut être réalisé par d'autres outils de présentation, comme Macromedia Dreamweaver, Microsoft FrontPage ou Microsoft Powerpoint. WebCT peut récupérer tout document converti au format HTML.

Les cours réalisés peuvent contenir :

- Tout document de type texte, image, vidéo et audio
- Des pages d'aide comme des index pour des recherches, des glossaires
- Des hyperliens vers d'autres documents ou sites externes

Les utilisateurs distants n'ont pas besoin d'installer de logiciel spécifique pour accéder aux cours : ils utilisent un navigateur standard. Les étudiants enregistrés à certains cours délivrés par WebCT communiquent avec leur instructeur en utilisant la messagerie électronique, des groupes de discussion ou un dialogue textuel.

L'évaluation des étudiants se fait au travers de Questionnaires à Choix Multiples et de Devoirs.

Les enseignants peuvent aussi recevoir des données qui mesurent l'impact de leur cours.

WebCT contient en dernier lieu des fonctions d'administration intégrées, pour le suivi et la gestion des apprenants. L'outil de syllabus décrit la structure de l'enseignement délivré. Un calendrier définit les dates de suivi des cours ainsi que les dates d'examen et de rendu des devoirs. En dernier lieu des outils de gestion de notes et de suivi de l'avancée des étudiants sont aussi disponibles.

WebCT est l'outil le plus utilisé dans le monde (des centaines d'établissement et d'entreprises réparties sur plus de 80 pays) pour la gestion et la diffusion de cours. Nous citerons notamment le programme d'étude numérique [MCCL00] de l'université de Liverpool, qui est l'un des premiers et l'un des plus avancés en Europe. Tous les supports de cours des étudiants sont numérisés dans le format HTML et accessibles depuis chez eux. L'Université de Toulouse 1 [CYBE03] a aussi mis en place un enseignement juridique pour une licence en droit, appelée Cyberlicence, qui s'appuie sur WebCT. Finalement, la formation continue à l'INSA de Toulouse [INSA03] se sert de WebCT comme support de formation.

Quelques environnements importants avec des fonctionnalités similaires sont :

- Learning space [LEAR03]
- Scolastance [SCOL03]

2.2.2 Evolution des systèmes basés sur le Web

La figure 1.2 synthétise l'évolution des systèmes de Web collaboratif. Les premiers systèmes, généraux, introduisent à partir de 1995 une dimension de groupe sous forme de droits d'accès aux documents Web. Par la suite, ces systèmes se spécialisent selon certains domaines d'application. Le téléenseignement est l'un des plus caractéristiques, mais d'autres systèmes se développent, dédiés à la gestion de documents techniques ou bien de données produit. Ces systèmes évoluent naturellement vers des portails Web qui intègrent d'autres

applications, pas forcément collecticiel, mais qui sont très liées au domaine d'application choisi. Par exemple, dans le cas du téléenseignement, des applicatifs de suivi des étudiants, ainsi que de gestion des salles, sont couplés au système de Web collaboratif et sont accessibles au travers d'une même interface portail Web.

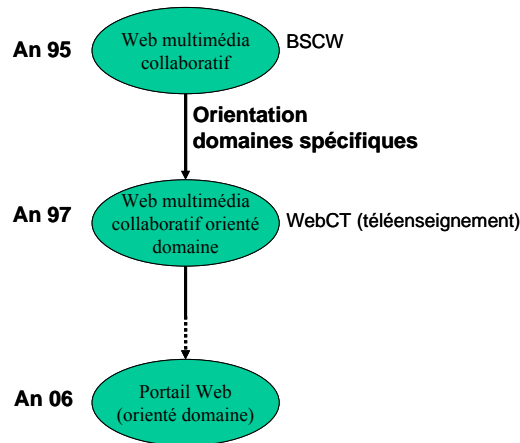


Figure 1.2. Evolution des systèmes de Web collaboratif

3. Travail collaboratif. Collecticiel synchrone

Les plates-formes de collecticiel synchrone [KARS94] autorisent des échanges interactifs (temps réel) entre les membres distribués géographiquement, en co-présence virtuelle. Ils supportent des interactions synchrones entre utilisateurs.

Les premiers systèmes collecticiels synchrones ne manipulent que du texte. Par la suite, l'énorme progrès de l'informatique multimédia et l'avènement de réseaux à haut débit depuis les années 90 ont fortement dynamisé le collecticiel pour le travail collaboratif synchrone. Le transport de flux vidéo multimédias est désormais possible entre groupes d'utilisateurs distribués géographiquement, ce qui a entraîné l'apparition de nouvelles plates-formes multimédia support de travail de groupes d'utilisateurs en co-présence virtuelle [KARS94].

Au début, les plates-formes de collecticiel synchrone comprennent un ensemble d'outils peu couplés entre eux qui supportent les fonctionnalités élémentaires d'échange d'information pour assurer le bon déroulement du travail associé au groupe :

- (i) communications informelles entre utilisateurs
- (ii) partage d'applications et d'espaces de travail
- (iii) partage de documents

Chacune des fonctionnalités précédentes est illustrée par une sélection d'outils significatifs de collecticiel synchrone.

Par la suite, les plates-formes de collecticiel synchrone mettent en œuvre un ensemble de services qui coordonnent les outils collecticiels entre eux, et qui gèrent les utilisateurs et les ressources dont ils ont besoin pour accomplir leur travail de groupe. Ces services, souvent regroupés sous le terme "coordination", forment la "glue logicielle" des outils collecticiels.

3.1 Outils pour des communications informelles

Les outils de communication informelle supportent les dialogues réalisés au sein du groupe synchrone. Deux classes ont été isolées pour ces communications, selon le type de média utilisé pour dans les échanges :

- l'Internet Chat Relay, qui manipule des informations textuelles
- les vidéoconférences, qui échangent des flux audio et vidéo

3.1.1 Internet Chat Relay (IRC)

"Internet Relay Chat" (IRC), appelé couramment "Chat", date de 1988 [IRC88]. Ce service n'utilise que le texte pour faire des conversations de groupe en temps réel, avec des personnes connectées à internet dans des "espaces" virtuels publics ("room"). Une fois connecté à un espace virtuel identifié, les messages des participants de cet espace apparaissent à l'écran les uns à la suite des autres. Chaque participant est identifié par un surnom. IRC est un protocole qui fonctionne selon le modèle client/serveur. Il faut simplement se brancher à un serveur, choisir un canal de discussion, puis un espace de discussion, avec un logiciel client. Les logiciels client IRC les plus utilisés sont Visual IRC (VIRC) [VISU03] et mIRC [MIRC03].

L'IRC est encore actuellement très utilisé lors de communications de groupes synchrones comme solution de secours, du fait de sa simplicité, de sa robustesse et surtout de la très faible bande passante nécessaire pour dialoguer.

3.1.2 Vidéoconférence sur Internet

Les vidéoconférences ont été les premiers outils à traiter les données multimédias sur des ordinateurs. Cette intégration au monde informatique s'est faite, à notre avis, grâce à l'évolution des trois axes suivants :

- La technologie matérielle utilisée pour l'acquisition et la capture des données multimédias
- L'évolution des réseaux, non seulement en terme d'augmentation de débit mais aussi en terme de services nouveaux (garanties temporelles et diffusion multipoint)
- La proposition de normes de codage communément adoptées et implantées par les outils

Les paragraphes suivants illustrent ces trois axes, au travers de réalisations significatives sélectionnées.

a) Evolution matérielle

Les premières cartes de capture, de numérisation et d'affichage pour la vidéoconférence ont été disponibles dans les années 90. Cette évolution matérielle s'est faite selon trois grandes étapes.

Les premières cartes font des acquisitions en mode analogique, aux formats PAL, SECAM ou NTSC. Un exemple de ce type de matériel est les cartes Parallax [PARA03] que nous avons manipulées lors de nos premiers projets multimédias, notamment les projets CESAME et TOPASE. Les étapes de numérisation, de compression, puis d'affichage s'effectuent grâce aux processeurs dédiés de la carte, ce qui a pour principale avantage de ne pas alourdir la charge du processeur de la machine. Les données numériques obtenues, codées selon le principe de l'algorithme Motion JPEG, sont stockées dans un format propriétaire. Leurs trois grands inconvénients : (1) leur coût (de 10 kEuro à 150 kEuro) ; (2) la disponibilité que sur

station de travail ; (3) des formats de données propriétaires, ont limité leur diffusion. Leur production a cessé fin 1998.

Les années 95 voient l'apparition de nouveaux types de cartes d'acquisition et de numérisation, telles que les cartes Osprey de la société Viewcast [VIEW03]. Elles possèdent des entrées analogiques comme précédemment, mais aussi une entrée numérique de type DV. Leur coût raisonnable (de 165 Euros à 900 Euros), leur adaptation à différents types de matériel (PCs et stations de travail) et l'utilisation de formats de données standard H.261, favorisent leur diffusion. Leur inconvénient, relatif car vite compensé par la constante évolution de la puissance de traitement des processeurs, est d'utiliser le processeur de la machine hôte pour réaliser les traitements de compression.

La dernière évolution consiste en la technologie directement numérique de type Webcam [LOGI03], [CAMB03]. L'image, numérisée par un capteur qui saisit directement les images sous forme de matrice de pixels, est acheminée sur la machine hôte (un PC classique) au travers d'un bus numérique tel que l'USB. La compression et la mise en format standard sont faites par le processeur de la machine. Le prix des Webcam, très bas (moins de 50 Euros), et leur facilité d'installation sur une connectique externe, expliquent leur popularité d'utilisation. Elles supplantent les cartes d'acquisition et sont les périphériques les plus répandus pour la vidéoconférence de bureau.

b) Services réseau

Deux types de services réseau sont très importants pour l'utilisation de logiciels de vidéoconférence. Le premier type considère l'aspect temporel des données multimédia échangées. Le deuxième type étend les communications à des groupes multipoint, pour permettre des communications interactives entre groupes synchrones.

Services avec garanties temporelles

RTP (Realtime Transport Protocol) [RFC1889] et son protocole de contrôle RTCP (Realtime Transport Control Protocol) permettent respectivement de transporter et de contrôler des flots de données qui ont des contraintes temporelles.

Chaque flux RTP est décomposé en un ensemble de paquets émis périodiquement. Chaque paquet est estampillé et contient une date d'horloge d'émission. Le protocole RTP introduit un retard supérieur à la gigue estimée du réseau. La restitution du flux est réalisée à la réception, en utilisant les informations d'horloge de chaque paquet : le récepteur restitue les paquets en respectant leur synchronisation temporelle et il supprime les paquets reçus en retard. Chaque flux multimédia est codé de façon séparée : ainsi, un flux RTP audio est toujours séparé d'un flux RTP vidéo.

Le protocole RTCP associe toujours un flux de contrôle à un flux de donnée. Le but de ce flux est de contrôler la qualité du flux monomédia reçu, en obtenant chez un émetteur un retour sur la qualité, en terme de taux de perte de paquets et de paquets arrivés en retard. Ce retour permet de modifier ou d'adapter la qualité du flux si nécessaire. La quantité de trafic RTCP se limite à 5% du trafic multimédia global.

Les protocoles RTP et RTCP utilisent le protocole UDP, et les flux transmis sont identifiés par deux ports consécutifs : le port pair pour le flux RTP, le port impair pour RTCP.

Diffusion

La diffusion permet d'envoyer en une seule fois une même information à destination d'un ensemble d'utilisateurs.

Proposé par S. Deering dans les années 90, IP multicast est le premier protocole de diffusion à être disponible au dessus d'Internet [RFC1112]. Ce protocole définit des groupes ouverts d'utilisateurs, identifiés par une adresse IP spécifique et un numéro de port. Le service de diffusion mis en place est conforme aux caractéristiques d'IP, à savoir non fiable, sans garantie d'ordre, ni de délivrance. Le succès de ce protocole vient de son rapide déploiement, à partir de 92 dans le réseau expérimental Mbone (multicast bone), déployé au niveau mondial en utilisant une technique de tunneling (fabrication d'un réseau de diffusion statique au dessus de l'Internet mondial). Ce déploiement expérimental a permis de mettre au point et de tester des outils de vidéoconférence multiutilisateurs pour des téléseminaires et des réunions virtuelles.

Actuellement, du fait du succès du Mbone, de nouveaux protocoles de diffusion, notamment Protocol Independent Multicast Sparse Mode (PIM-SM) et Multiprotocol Extensions for Border Gateway 4 (MBGP), ont été intégrés dans les routeurs d'Internet, ce qui permet de supporter un mode de diffusion natif, de même niveau que les transmissions de paquets IP.

Outils du Mbone

Les logiciels "Vic" et "Rat" (figure 1.3) sont les outils de vidéoconférence les plus utilisés pour des vidéo et audioconférences au dessus du Mbone. Ils sont particulièrement illustratifs des possibilités et des limites du Mbone.

Vic (VIdeoConferencing) [McCA97] et Rat (Robust Audio Tool) [RAT03] sont deux logiciels de vidéo/audio conférence qui font partie d'une suite d'outils de collecticiel synchrone, développée par le "Network Research Group" du laboratoire "Lawrence Berkeley National Laboratory" en collaboration avec l'Université de Berkeley en Californie. Les premières versions furent développées par Steven McCanne, dans le cadre de sa thèse en 1996.

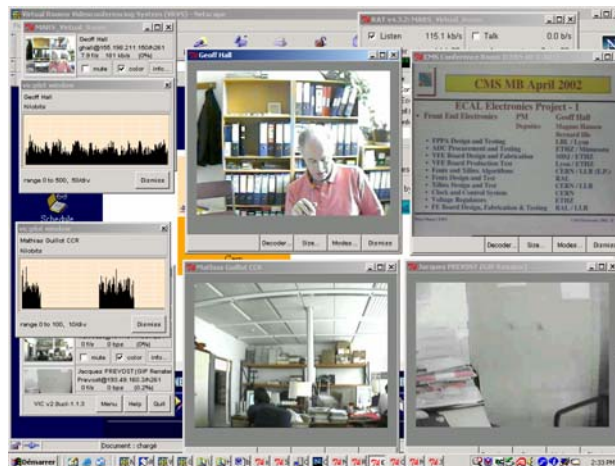


Figure 1.3. Utilisation de Vic et de Rat

Le codage et la compression des trames vidéo, originellement basés sur la recommandation H.261 [H261], ont évolué pour être compatibles avec la norme H.263 [H263] qui est la référence et la norme la plus utilisée en terme de codecs vidéo. Ces deux normes manipulent le plus souvent des images de 352x288 pixels, ou format CIF (Common InterMediate Format). Elles font partie du cadre de normalisation H.323 [H323], cadre le plus utilisé en terme de vidéoconférence.

Les formats de compression audio les plus couramment supportés sont G.711 PCM [G711], G.726 ADPCM [G726] et GSM [GSM03]. La norme G.711 correspond au standard international pour transmettre la voix téléphonique sur un canal numérique au débit de 64 kbits/sec. G.726, plus performante, requiert des canaux au débit de 16 à 40 kbits/sec, suivant la qualité audio souhaitée. La norme GSM [TUB03], qui provient de travaux de l'ETSI (European Telecommunications Standards Institute), est un codec adapté au téléphone mobile. Elle offre un débit de 13 kbits/sec.

La synchronisation des images, ainsi que la synchronisation audio se font en utilisant le protocole Real-time Transport Protocol (RTP) [RFC 1889].

"Vic" et "Rat" peuvent initialement fonctionner en mode point à point en utilisant le protocole de Transport UDP. Cependant, le grand atout de ces outils est de supporter des communications multipoint, au dessus du protocole IP multicast [RFC 1112] [TUY02]. Il combine ainsi les avantages de la synchronisation multimédia fournie par le protocole RTP avec celui d'une vraie diffusion multipoint.

Ces deux outils sont très utilisés par la communauté réseau pour leurs nouveaux tests et développements, mais aussi par plusieurs communautés d'utilisateurs (physiciens [VRVS03], académiques [ARIS03]...). Ils ont été portés sur de nombreuses plates-formes et systèmes d'exploitation Unix et Windows. Ils sont donc utilisés à titre expérimental, ce qui garantit leur évolution et maintenance, mais aussi en mode opérationnel pour des téléconférences et des téléseminaires [ARIS03].

c) Norme et architecture pour la vidéoconférence : H.323

La norme H.323 [H.323], [SCHL03], [PREV03], [VINC02] est un ensemble de standards et de recommandations pour les conférences multimédias sur des réseaux de paquets commutés, ces réseaux pouvant être de type intranet ou internet. H.323 est définie pour harmoniser la voix sur IP. Elle fait partie de la série H.32x qui traite de la vidéoconférence au travers différents réseaux, depuis des réseaux téléphoniques bas débit, de type ISDN, LAN ou Internet. Le trafic échangé peut également être géré et contrôlé.

Composants

Cette norme comprend quatre composants majeurs : les terminaux, les passerelles, les gatekeeper et les unités multipoint.

Les terminaux H.323 sont les terminaux clients finals des utilisateurs. Ils fournissent des communications bidirectionnelles vers d'autres équipements terminaux. La communication peut se faire par :

- (a) un canal audio seulement,
- (b) deux canaux audio et donnée,
- (c) deux canaux audio et vidéo,
- (d) trois canaux audio, vidéo et donnée.

Ces terminaux peuvent être matériels ou logiciels.

Les passerelles (gateways) sont un point terminal H.323. Elles fournissent des communications bidirectionnelles entre des terminaux qui appartiennent à des réseaux constitués de piles de protocole différentes, par exemple entre un réseau IP et le RTC ou RNIS. Elles assurent, entre autres, le codage et le décodage de la voix, la mise en paquets, la suppression de l'écho.

Le gatekeeper H.323 est un élément optionnel pour définir et gérer des zones de conférence H.323. Lorsqu'un gatekeeper est activé dans un réseau local, tous les clients H.323 de ce réseau doivent l'utiliser. Un gatekeeper contrôle la qualité opérationnelle du réseau en réalisant les fonctions suivantes pour les composants de type terminaux, passerelles et unités multipoint déclarés dans sa zone.

- Contrôle d'admission : il autorise seulement l'accès aux composants déjà enregistrés et reconnus, limitant de ce fait le nombre de terminaux en cours de communication.
- Contrôle de bande passante : lors du contrôle d'admission, le gatekeeper vérifie que le réseau possède encore suffisamment de bande passante avant d'accepter l'appel.
- Translation d'adresse : le gatekeeper traduit les identifiants des terminaux, proches de numéros de téléphone, et leurs alias, en adresses reconnaissables par le réseau, par exemple des adresses IP.

Finalement, on peut considérer un gatekeeper comme une sorte de "routeur" applicatif H.323.

L'Unité de Contrôle Multipoint (MCU) supporte des conférences entre trois utilisateurs ou plus. Cette unité se compose d'un Contrôleur Multipoint (MC) obligatoire et d'un Processeur Multipoint (MP) facultatif. Le contrôleur MC se charge des négociations entre tous les terminaux pour déterminer les capacités de traitement audio et vidéo communes aux différents terminaux. Il contrôle les ressources de la conférence et détermine, si nécessaire, quels flux audio ou vidéo peuvent être émis en mode diffusion. Le processeur MP intervient lorsque le réseau n'offre que des communications point à point. Il centralise les flux audio et vidéo provenant des terminaux, les traite, les filtre, puis les rediffuse vers chacun d'entre eux. Ce mode centralisé simule une diffusion réelle.

Remarque : Bien que la norme H.323 prévoit un ensemble de communications allant du mode centralisé au décentralisé, la plupart des implantations actuelles ne supporte que le mode centralisé. Cependant, des initiatives apparaissent pour faire converger H.323 et la diffusion IP multicast, telle celle du logiciel Microsoft Exchange 2000 Conferencing Server [CONF03].

Normes

La figure 1.4 présente la pile des principaux protocoles qui composent la norme H.323.

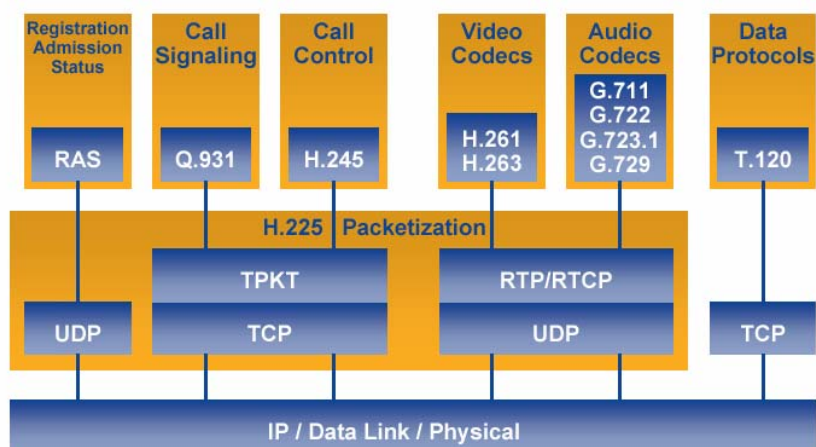


Figure 1.4. Principaux protocoles de la norme H.323

Protocoles de contrôle

La norme H.225 s'occupe de la signalisation des appels et contrôle la mise en paquets des flux de données multimédias.

La norme Registration Admission Status (RAS) gère la découverte des gatekeeper, l'enregistrement auprès d'un gatekeeper, la résolution de noms entre les alias H.323 et les adresses IP. Elle s'occupe aussi du contrôle d'admission et du contrôle de la bande passante.

Le protocole de signalisation d'appel Q.931 est identique à celui utilisé pour des canaux ISDN.

La norme H.245 pour le contrôle d'appel se charge principalement de la gestion des médias et des flux de données.

Standards pour l'échange de données de conférences. Norme T.120

La norme T.120 regroupe un ensemble de spécifications pour standardiser les échanges de données entre plusieurs types d'applications collaboratives. Nous citerons brièvement les principales :

La norme T.126 spécifie le format des données échangées par une application de type tableau blanc. Elle gère l'espace multiutilisateur fourni par ce type d'application.

La norme T.127 précise le format d'échange des fichiers et des images partagées par un tableau blanc.

Le standard T.128 autorise le partage d'une application entre plusieurs utilisateurs. Un groupe d'utilisateurs peut ainsi partager et interagir sur une même instance d'application.

La norme T.134 décrit le dialogue textuel multiutilisateur ("chat").

Protocoles d'échanges multimédias

Les protocoles RTP et RTCP, décrits précédemment, sont utilisés pour les échanges multimédias, numérisés au travers des codecs H.261 et H.263 pour la vidéo, et des codecs G.711 à G.729 pour l'audio [JAVV04].

Vidéoconférences H.323

Les produits de vidéoconférence les plus répandus qui respectent le format H.323 sont :

- Microsoft Netmeeting [NETM03]
- Microsoft Messenger [MESS04]
- Cu See Me [CUSE03] de la société First Virtual Communications [FVC03]

d) Tendances d'évolution des outils de vidéoconférence

La figure 1.5 synthétise les grandes étapes qui marquent le développement des systèmes de vidéoconférence, selon les trois axes précédemment identifiés :

- La technologie matérielle
- Les services réseau
- La normalisation des codages

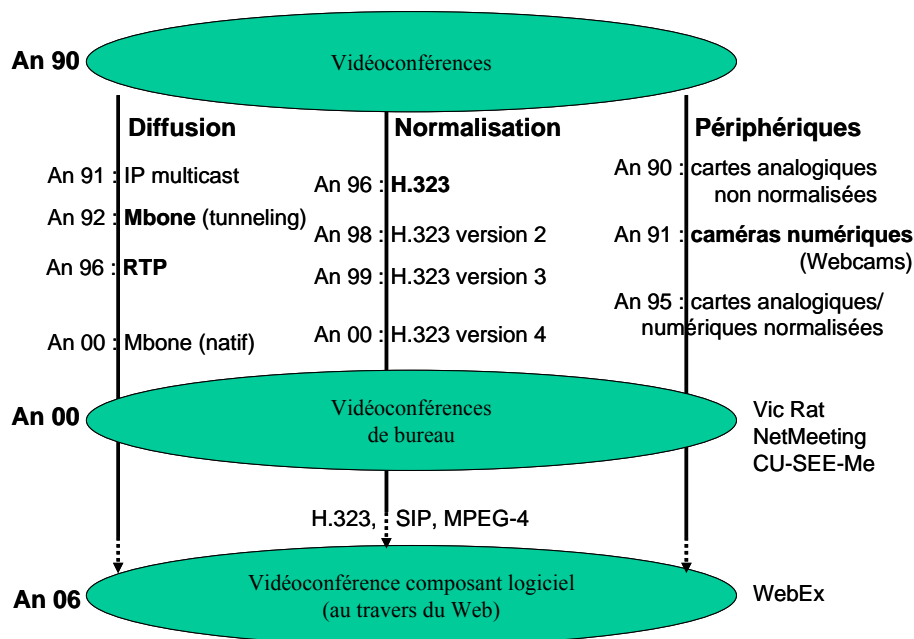


Figure 1.5. Evolution des systèmes de vidéoconférence

Les périphériques de capture et de manipulation de vidéo numérique poursuivent leur évolution en termes de miniaturisation et d'augmentation de la définition des images traitées en nombre de pixels.

Les protocoles de diffusion réseau actuels opérationnels (Protocol Independent Multicast Sparse Mode (PIM-SM) et Multiprotocol Extensions for Border Gateway 4 (MBGP)) ont été établis pour des groupes qui comportent un nombre réduit d'utilisateurs (quelques dizaines) fortement distribués. Ces protocoles sont mal adaptés (i) pour des groupes de taille plus conséquente (de milliers d'utilisateurs et plus) (ii) qui nécessitent des garanties de Qualité de Service ou (iii) qui ont besoin d'accès sécurisés. Ces deux derniers points font partie des évolutions nécessaires pour le support et la mise au point d'outils de vidéoconférence massivement multiutilisateurs.

La normalisation des échanges multimédias pour la vidéoconférence est principalement basée sur la norme H.323. Cependant, cette norme a été développée à l'origine pour des réseaux qui fonctionnent en mode connecté et qui supportent des débits de l'ordre de la centaine de kbits/sec. Le contrôle et la gestion des flux, qui suivent une philosophie connectée, ne sont plus adaptés aux réseaux actuels, qui respectent une philosophie non connectée compatible avec IP. H.323 évolue vers une simplification des protocoles de gestion, avec notamment une intégration progressive et une adaptation au protocole SIP (Session Initiation Protocol) [RFC3261].

De la même façon, les réseaux actuels supportent le transport de tailles d'images bien supérieures à celle définies par le codec H.263. De ce fait, H.263 commence à être supplanté par H.264, ou MPEG-4 Advanced Video Coding, une norme qui a émergé début 2003. H.264 [WIKI05] est basé sur un algorithme de compression MPEG-4, et traite des images de taille et de qualité équivalentes ou supérieures aux formats des films actuels. L'outil Video LAN [VIDE05] qui tend à supplanter "vic" et "rat" pour la diffusion des téléseminaires est l'exemple type de cette tendance.

Finalement, la tendance générale des systèmes de vidéoconférence est de passer de systèmes spécifiques, avec du matériel dédié, à des logiciels de vidéoconférence de bureau, fonctionnant sur des microordinateurs classiques, et de ce fait intégrables avec d'autres applications. Cette tendance se confirme et se renforce en transformant les applications de vidéoconférence en composants logiciels intégrables au travers de navigateurs Web. L'environnement de téléseminaires WebEx [WEBE03], qui comprend des systèmes de vidéoconférence, de partages d'applications et de documents, est un bon exemple de cette évolution.

3.2 Outils de partage d'application

La fonctionnalité de partage d'applications et d'espace de travail est supportée par des outils logiciels qui permettent à plusieurs utilisateurs travaillant sur des ordinateurs différents d'utiliser simultanément et à distance un groupe d'applications hébergé par un autre utilisateur. Les utilisateurs peuvent ainsi travailler de façon collaborative sur des présentations Powerpoint, des programmes graphiques de conception, ou n'importe quel autre applicatif. Les applications partagées tournent chez un seul utilisateur, mais sont rendues visibles chez tous les utilisateurs distants qui se servent du partage d'application. Cette vue est fournie à chaque utilisateur par une fenêtre spéciale de partage d'application qui reproduit exactement l'interface des applications partagées. Au travers de cette vue, chaque participant peut utiliser son clavier et sa souris pour effectuer un contrôle et une commande à distance des applications partagées.

Le partage d'application est très utilisé dans les domaines de :

- La téléassistance, avec l'aide et le dépannage à distance;
- La téléadministration de parc de machines ;
- Le contexte éducationnel, pour partager des applications entre un professeur et un groupe d'élèves.

Les deux exemples de logiciels de partages d'application que nous avons sélectionnés couvrent différemment les domaines d'applications identifiés. Le premier, pcAnywhere est dédié à la téléassistance et la téléadministration, tandis que deuxième, plus universel, peut être utilisé dans les trois domaines d'applications.

3.2.1 Symantec pcAnywhere

PcAnywhere (version 11) [SYMA03] est un des outils de référence pour la prise en main à distance. Combiné à des fonctions de transfert de fichiers et de gestion distante, il permet de résoudre rapidement les problèmes d'assistance et d'administration de serveurs. La fonctionnalité de prise en main à distance, fiable et sécurisée, permet de se connecter et de dépanner les postes de travail et les serveurs distants, afin d'optimiser la résolution des problèmes rencontrés.

La suite d'outils de gestion à distance exploite la connexion sécurisée entre un système Elève et Maître, permettant ainsi de résoudre les problèmes sans avoir à ouvrir une session de prise en main à distance complète. Les fonctions de gestion à distance intègrent des outils classiques tels que le gestionnaire des tâches, la ligne de commandes et la fonction de modification à distance du registre. La fonction ligne de commandes permet aux administrateurs de mettre en file d'attente plusieurs fichiers et commandes DOS et de les traiter dans l'ordre. Il est ainsi possible de réorganiser, de mettre en attente et de supprimer les

fichiers de la file d'attente, le cas échéant. Les transferts de fichiers s'effectuent en tâche de fond afin que les administrateurs puissent travailler sans interruption et continuer à sélectionner des fichiers tout en transférant d'autres. Les administrateurs peuvent même envoyer des fichiers à plusieurs ordinateurs l'un après l'autre. Une interface utilisateur graphique facilite la navigation entre les applications de transfert de fichiers, de prise en main, et de gestion à distance.

Au niveau technique, pcAnywhere ne fonctionne qu'en mode point à point entre un système Elève et un Maître. L'attribution de ces deux rôles entre les systèmes change au cours de sessions pcAnywhere.

3.2.2 Virtual Network Computing (VNC)

L'outil Virtual Network Computing (VNC) [RICH98], [VNC03] a été conçu en 1994 dans les laboratoires AT&T de l'Université de Cambridge (UK). Il s'agit d'un logiciel de contrôle à distance qui permet d'interagir sur un ordinateur (le serveur) en utilisant un autre programme (le visualisateur) qui tourne sur une autre machine connectée à l'Internet (figure 1.6).

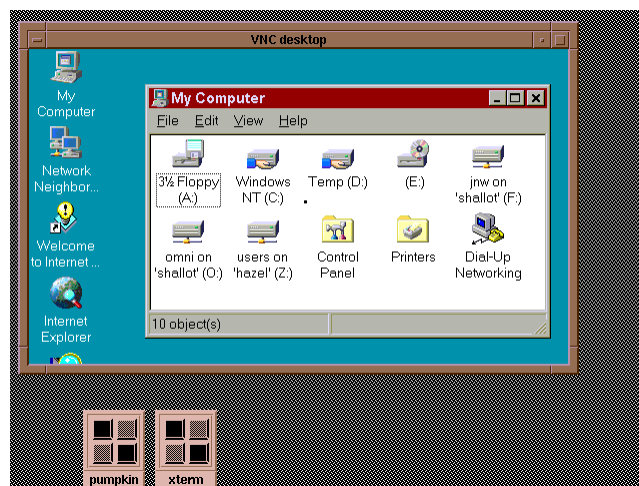


Figure 1.6. Ecran d'un client VNC

Les deux ordinateurs ne sont pas forcément de même type, et il est possible au travers de VNC de visualiser une machine LINUX serveur depuis un PC Windows client. Plusieurs versions interopérables entre elles existent, une pour les systèmes Windows (appelée WinVNC), et une pour chaque plate-forme UNIX de type SUN, HPUX, Silicon Graphics, et Macintosh. La figure 1.6 montre l'écran d'un client VNC qui tourne sur une station UNIX alors que les applications partagées sont sur un serveur Windows.

La technique de visualisation des applications partagées est originale car elle s'appuie directement sur des captures de zones d'écran sous forme de rectangles bitmap qui sont transmises de façon plus moins compressées. Le modèle d'entrée pour les applications distantes se compose d'un clavier générique et d'une souris multiboutons. Cette technique de visualisation garantit une large adaptation de VNC vis-à-vis des types de machines et des systèmes d'exploitation.

La communication entre le serveur et un client se fait avec un lien TCP/IP. VNC possède un mécanisme d'authentification basé sur un login et un mot de passe entre un client et le serveur. Ce mécanisme peut être couplé avec SSH pour augmenter le niveau de sécurité si nécessaire.

Deux types de clients sont disponibles: un client natif écrit en C++ et un client universel écrit en JAVA. Ce dernier choix est garant d'une portabilité maximale, ce client JAVA pouvant être récupéré de façon interactive depuis le serveur VNC via une page Web.

Plusieurs clients peuvent se connecter et dialoguer en même temps avec un seul serveur. Les applications partagées sont dans ce cas, visibles et manipulables par l'ensemble des clients.

VNC est développé sous licence logiciel libre, les codes source sont donc gratuits et accessibles.

3.2.3 Tendances d'évolution des partages d'application

En s'appuyant sur les deux exemple précédents, la figure 1.7 résume les tendances relevées dans les outils de partage d'application :

- Partage en groupe
- Intégration plus forte aux systèmes d'exploitation

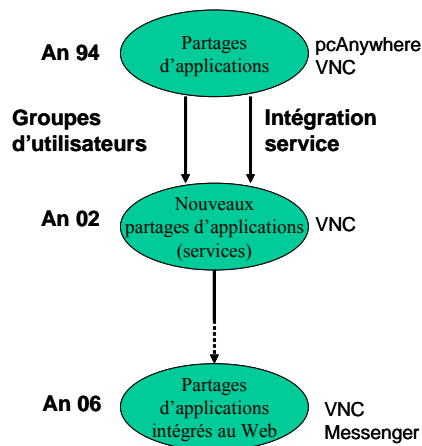


Figure 1.7. Evolution des partages d'application

La première est une évolution vers le partage d'applications en groupe. En effet à un instant donné, l'espace n'est pas uniquement partagé entre un utilisateur serveur et un seul utilisateur client en mode point à point (comme pour pcAnywhere), mais est transmis en même temps et en mode multipoint vers un ensemble de clients. Dans ce dernier cas, l'utilisateur serveur partage sa vue d'écran. Cette vue est rediffusée vers tous les autres clients participant au groupe.

La deuxième tendance est une intégration plus forte avec les systèmes d'exploitation. De simples applicatifs autonomes, les partages d'applications sont de plus en plus souvent associés au démarrage d'un système, sous la forme de démons. Cette tendance, possible avec pcAnywhere et VNC, se renforce avec les nouveaux partages d'applications tels XP Messenger.

Finalement, l'intégration au Web se fait par des applicatifs clients spéciaux, tels ceux développés en JAVA pour l'outil VNC. Cette intégration permet de se connecter au partage d'application au travers de pages Web.

3.3 Outils de partage de documents

La fonctionnalité de partage de document permet à un ensemble d'utilisateurs d'accéder, de modifier voire d'annoter un document mis en commun au sein du groupe synchrone. Trois classes d'outils de partage ont été proposées selon le type de document utilisé et manipulé, type de complexité croissante :

- les éditeurs multiutilisateurs, qui manipulent des informations textuelles
- les tableaux blancs partagés, qui traitent des informations graphiques et des images fixes
- les navigateurs collaboratifs, qui prennent en compte des documents Web pouvant contenir tout type de média textuel, des graphiques, des images fixes et des flux multimédias

3.3.1 Editeurs multiutilisateurs

Les éditeurs multiutilisateurs sont les premiers types d'outils de partage, le document étant de type texte. Un groupe d'utilisateur manipule et modifie de façon interactive un même texte. L'éditeur gère les accès concurrents pour garantir la consistance du fichier partagé.

Un exemple d'éditeur synchrone est le logiciel DistEdit [DIST90] qui permet de partager un fichier entre plusieurs utilisateurs et de maintenir la vue de ce fichier consistante chez chacun d'eux. Les utilisateurs peuvent joindre ou quitter la session d'édition à tout moment. Un seul utilisateur peut modifier le fichier à la fois. Les autres ne sont que de simples observateurs des modifications.

3.3.2 Tableau blanc partagé

Un tableau blanc partagé reproduit le comportement d'un tableau blanc classique. Son but est ainsi de permettre à un groupe d'acteurs de dessiner ou d'annoter des images dans un espace partagé de l'écran, de manière synchrone, les acteurs étant physiquement distants, et interagissant au travers de l'ordinateur.

L'interface utilisateur de cette application se compose d'une fenêtre graphique pour chaque membre. Un membre autorisé du groupe affiche ses transparents. Ce membre peut annoter les transparents affichés. Selon les possibilités de ce tableau, les autres membres du groupe peuvent éventuellement annoter les transparents qui s'affichent.

Cet outil est très utile dans le cas de télé-séminaires ou de télé-réunions. Les supports de la réunion sont visualisés, discutés et annotés au travers du tableau blanc partagé.

Tableau blanc multiutilisateur de l'environnement Platine

Le tableau blanc de l'environnement prototype Platine [PLAT05] est entièrement développé en JAVA (figure 1.8). Plusieurs utilisateurs peuvent travailler ensemble au travers de cet outil. Un membre autorisé du groupe affiche des transparents au format GIF ou JPEG. Il peut ajouter des annotations sur le transparent affiché. Les autres membres du groupe, s'ils en ont l'autorisation, peuvent aussi ajouter d'autres annotations qui sont visibles par l'ensemble des participants connectés. Les annotations sont échangées sous forme d'événements graphiques qui nécessitent peu de bande passante pour leur transmission.

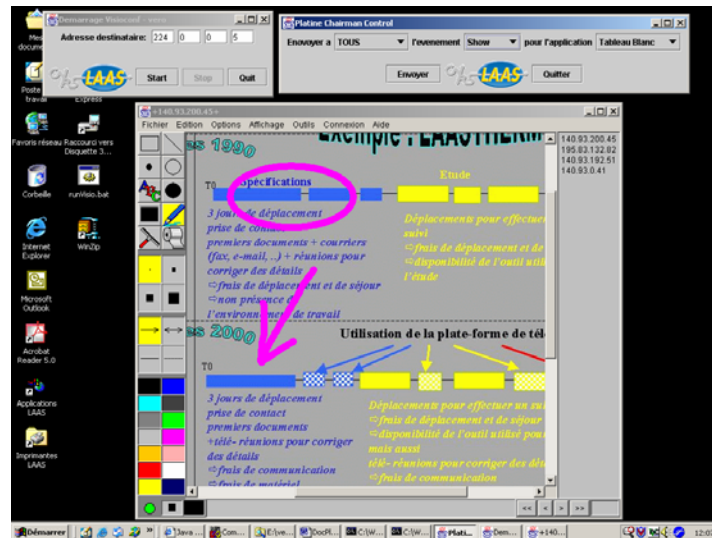


Figure 1.8. Tableau blanc partagé

3.3.3 Co-navigation

L'idée de la co-navigation est une extension de la navigation à un groupe d'utilisateurs, pour que les membres du groupe naviguent en commun sur des documents Web. La manière classique, rendue disponible actuellement par les navigateurs traditionnels, conduit à une interaction inexistante, voire très faible, des utilisateurs entre eux. Chacun accède à un document de façon indépendante de celle des autres. La co-navigation propose un concept de navigation synchrone dans lequel un membre privilégié d'un groupe pilote l'accès à un document Web et force une présentation identique de ce dernier chez tous les autres membres du groupe. L'attribut de membre privilégié peut bien entendu changer de personne dans le temps.

Navigateur collaboratif Colab

Le navigateur collaboratif Colab, développé par G. Hoyos [HOYO02], est un environnement prototype basé sur le Web.

Le principal intérêt de cet environnement prototype est de proposer une extension paramétrable et plus souple, de la façon dont un groupe d'utilisateurs peut naviguer en commun sur des pages Web. En effet, le mode de co-navigation, implanté par les environnements tels que WebEx [WEBE03], force une présentation identique des pages Web chez tous les membres du groupe. Cette présentation est pilotée par un membre privilégié.

La navigation collaborative, implantée par Colab, assouplit le mode de co-navigation. Elle propose ainsi une palette de modes de navigation de groupe possibles, depuis le fortement contraint jusqu'au faiblement contraint en passant par un ensemble de modes intermédiaires définis dynamiquement en fonction des pages visitées, des méta-données associées, du groupe, et de leur évolution conjointe dans le temps.

L'architecture de Colab se compose (i) d'un serveur Web qui joue le rôle de Proxy et (ii) d'un moteur de collaboration. Le serveur Web Proxy se charge de synchroniser l'activité de navigation collaborative. Le moteur de collaboration définit dynamiquement les droits d'accès aux ressources Web, selon des politiques définies sous forme de règles.

Les clients de ce navigateur collaboratif sont des applets JAVA qui sont synchronisées par le serveur Web pour visualiser les pages Web chez chacun des participants.

3.3.4 Tendances d'évolution du partage de documents

La fonctionnalité de partage de documents évolue technologiquement (figure 1.9). On passe de documents texte manipulés dans les années 90 à des partages d'objets graphiques et d'images vers 1995. Par la suite, les documents partagés deviennent des pages Web au format HTML, puis sont remplacés par des documents XML. Actuellement, grâce aux outils de co-navigation, le partage se fait sur des objets plus complexes, tels que des documents multimédias interactifs au format SMILE et des scènes au format VRML. Les co-navigateurs qui manipulent des scènes VRML forment par ailleurs une classe spécifique [LOU03a], [LOU03b]. Ils sont appelés Serveurs Multi-utilisateurs (ou Multi-User servers (MUS)). Un exemple est le MUS Cortona, développé par la société Parallel Graphics [PARA04], qui autorise un groupe d'utilisateurs à visualiser et à interagir sur une scène VRML partagée.

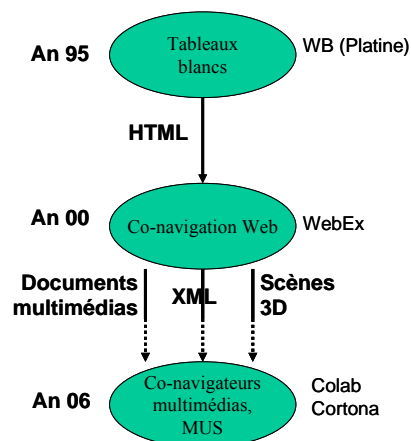


Figure 1.9. Evolution du partage de documents

3.4 Plates-formes de collecticiel synchrone

Les outils collecticiels synchrones sont rarement utilisés directement tels quels. La plupart du temps, ils se trouvent rassemblés dans des plates-formes collecticielles dont le premier but est d'offrir une interface unique pour leur accès.

Le deuxième rôle des plate-formes collecticielles est la mise en œuvre d'un ensemble de services qui coordonnent les outils collecticiels entre eux, et qui gèrent les utilisateurs et les ressources dont ils ont besoin pour accomplir leur travail de groupe.

On peut ainsi citer comme plates-formes commerciales WebEx [WEBE03] ou Microsoft Messenger, et comme plates-formes prototypes les environnements CALiF Multimédia [GARC04], [GARC05] et Platine [PLAT05].

Centre de conférence Active Touch WebEx

La plate-forme WebEx Communications [WEBE03] offre des services de réunion en ligne, de conférence Web, de téléconférence et de vidéoconférence. Ces services intégrés de travail collaboratif répondent à divers besoins de réunions multimédias. Ils sont accessibles à l'aide d'un navigateur et d'un téléphone. S'appuyant sur la technologie MediaTone, WebEx a déployé un réseau distribué dans le monde entier conçu pour faciliter les réunions interactives en ligne, les sessions de support client, les programmes de formation et les séminaires Web.

Toute conférence WebEx tourne dans la fenêtre d'un navigateur classique et ne nécessite aucune installation, configuration ou mise à jour logicielle. Les nouvelles fonctionnalités sont ajoutées automatiquement au navigateur lorsque des nouvelles versions apparaissent. Au niveau sécurité, WebEx fonctionne au travers des pare-feux, offre des réunions privées non

listées, des protections par mot de passe et des possibilités de cryptage des données sensibles. Le système WebEx supporte plusieurs milliers de participants et plusieurs centaines de réunions concurrentes.

Les principales caractéristiques de cette plate-forme sont résumées dans la table 1.1.

Fonctionnalité	Description
Vidéoconférence	La vidéoconférence depuis une caméra de bureau est activée sans autre installation logicielle nécessaire. Les participants peuvent voir les vidéos des autres participants dans leur navigateur.
Partage de présentation	Un participant à une réunion peut spontanément partager une présentation sans charger cette dernière sur le serveur WebEx, où la sécurité du fichier transféré pourrait être compromise. La visualisation peut se faire en mode plein écran.
Partage de document	N'importe quel document ou graphique peut être partagé et étendu par des annotations. Le propriétaire du document peut autoriser ou interdire les copies locales chez les participants.
Partage d'application (ou du bureau)	Des applications peuvent être exécutées et partagées pour des démonstrations ou des apprentissages interactifs.
Contrôle des applications partagées (ou du bureau)	Le présentateur peut transmettre le contrôle des applications partagées à n'importe quel autre participant de la conférence.
Co-navigation par Web	Toute information basée sur le Web peut être partagée. Un participant peut naviguer sur le Web et synchroniser les navigateurs des autres participants. La co-navigation est pleinement interactive et son contrôle peut être passé à n'importe quel participant, permettant ainsi de remplir des formulaires Web ensemble et d'annoter des pages Web.
Transfert de fichier	
Vote	Le présentateur peut solliciter des retours des participants en ligne.

Table 1.1. Caractéristiques de WebEx

3.5 Bilan

L'analyse de l'évolution des grandes classes d'outils collecticiels nous montre que ces outils convergent tous vers les technologies basées sur le Web. En effet, leur conception se fait de plus en plus selon une approche "composant Web". Ils deviennent accessibles soit directement au travers de navigateurs classiques, soit au travers de "plug-ins" qui s'exécutent aussi au travers de navigateurs classiques.

Au niveau de l'évolution des caractéristiques technologiques, les informations multimédias et la dimension coopérative, toutes deux avec des contraintes de temps, incluent des demandes et des garanties de qualité de service pour l'ensemble des utilisateurs des groupes, qui augmentent les difficultés de conception. Ceci justifie par la suite notre approche orientée modèle pour mieux maîtriser ces difficultés.

4. Modèles de session

Conformément à notre démarche, nous présentons, classons et analysons, dans cette section, un ensemble de modèles significatifs qui ont été utilisés pour représenter des sessions de travail collaboratif.

Les modèles que nous avons rencontrés dans la littérature s'appliquent aux deux types de sessions définies, les Sessions Asynchrones (pour laquelle la co-présence des utilisateurs n'est pas nécessaire et n'est pas gérée) et les Sessions Synchrones (qui traitent des utilisateurs en co-présence).

Les premiers travaux, les plus anciens, sont en relation avec les collecticiels asynchrones, tels que les messageries avancées. Comme la suite de nos contributions se place dans le collecticiel synchrone, nous avons volontairement orienté notre analyse et notre classification dans cette direction. Cependant, pour des raisons de complétude, nous citerons deux références qui illustrent les travaux réalisés dans le cadre des sessions asynchrones.

Les modèles d'activité AMIGO [PRINZ89] et les travaux de Benford [BENF93] s'appuient sur la norme d'annuaire X.500 [ISO9594] pour représenter les divers éléments d'une session. AMIGO comprend quatre grandes composantes : rôles, messages, fonctions et règles. Les rôles donnent une description abstraite des caractéristiques de chaque membre. Les fonctions décrivent les opérations que les rôles autorisent. Les règles garantissent les synchronisations entre les échanges de messages, les rôles et les fonctions. Les travaux de Benford organisent les éléments de session sous forme de classes, selon une approche orientée modèle. L'approche X.500 définit des services de nommage pour accéder aux objets définis et instanciés. Ces travaux mettent l'accent sur l'aspect description et organisation des éléments de session. Du fait de la nature asynchrone des sessions, l'aspect coordination entre éléments est peu représenté, voire absent.

L'analyse des solutions proposées pour des sessions synchrones s'appuie sur les critères suivants :

- (1) le formalisme mathématique employé.
- (2) les éléments modélisés : ce critère liste les éléments pris en compte par le modèle.
- (3) le type de session : deux types de sessions sont disponibles, explicites et implicites.

Une session explicite impose que certains utilisateurs privilégiés s'occupent des créations de session. Les autres utilisateurs rejoignent par la suite ces nouvelles sessions, soit d'eux-mêmes, soit par invitation du créateur [MOLI03].

Une session implicite [EDWA94] se crée à partir des actions de chacun des utilisateurs. Lorsque le système détecte des actions concourantes au sein d'une même activité, il crée une session et invite les utilisateurs concernés à se joindre à cette session.

(4) Contrôle de droit de parole : Cette fonction nous semble essentielle pour caractériser le degré de dynamique d'un modèle, ainsi que ses possibilités d'utilisation. Les modèles qui proposent de manipuler cette fonction nous semblent les plus complets et les plus proches d'une utilisation réaliste.

(5) Utilisation du modèle : Ce critère décrit son utilisation courante.

(6) Etat d'avancement : Ce critère indique si le modèle reste au niveau descriptif et abstrait ou s'il a fait l'objet d'une réalisation concrète et aboutie, utilisée dans un système.

4.1 Intermezzo

K. Edwards [EDWA94] présente un modèle de session basé sur le partage d'information par rapport aux utilisateurs et à l'activité du système. Ce modèle, nommé *Intermezzo*, est construit selon le principe que l'analyse des informations d'activité entraîne l'utilisation de mécanismes de gestion de session puissants et flexibles [EDWA95]. En effet, les informations d'activité contiennent le détail des travaux courants qui sont exécutés sur le réseau. Ce détail liste les utilisateurs du système avec les machines sur lesquelles ils sont connectés, les

applications ou les tâches dans lesquelles ils sont actuellement engagés, ainsi que les objets ou données qu'ils manipulent au sein de ces tâches.

Intermezzo définit une activité comme un tuple :

$$A_n = (U_n, T_n, O_n)$$

où A_n représente la n-ième activité. Cette activité est composée de U_n , T_n et O_n qui représentent respectivement le n-ième utilisateur, tâche et objet.

U_n et T_n sont manipulés comme de simples jetons qui identifient de façon unique l'utilisateur et la tâche sur laquelle il travaille. Les correspondances entre utilisateurs, tâches et leurs jetons respectifs U_n et T_n , se font un à un.

Un espace de nom définit un ensemble de données que la tâche utilise. Ses éléments ont une correspondance un à un avec les objets qu'ils représentent. Le jeton O_n est composé d'un identificateur et d'un nom qui ne sont valables que dans cet espace de noms.

Les tâches T_n , qui font partie de la session implicite, doivent publier leur information d'activité pour être visibles par le service de gestion de session.

Dans cette approche, le service de gestion de session détecte des situations collaboratives potentielles en recherchant des chevauchements ou des similitudes dans l'information d'activité publiée à travers le réseau. Quand deux tuples d'activité contiennent le même jeton d'objet (c'est-à-dire, quand les noms de deux objets correspondent exactement dans un espace de noms), alors le service de gestion de session détecte cela et permet aux utilisateurs d'entrer dans une situation collaborative en créant une nouvelle session.

Le système supporte à la fois la gestion de session implicite et explicite.

Il fournit un API pour le stockage et la récupération d'information reliée aux utilisateurs et aux tâches, au travers du réseau.

Le mécanisme de contrôle d'accès est basé sur des politiques (ou ensemble de droits de contrôle d'accès) assignées suivant les rôles des participants. Ce mécanisme est limité parce qu'il ne considère pas d'autres paramètres différents du rôle, tels que le type de plate-forme qu'un participant utilise ou bien la liste des outils qui lui est disponible.

4.2 CoAct

Le Modèle d'Activité Coopératif CoAct [RUSI95] fournit des propriétés transactionnelles appropriées pour des scénarios coopératifs. Le but de ce modèle, basé sur la logique du premier ordre, est de gérer et d'organiser des activités tout en garantissant leur exécution consistante.

La coopération est réalisée par des échanges contrôlés et par la synchronisation du contenu de zones de travail (workspaces). Les utilisateurs de ces zones de travail inscrivent les résultats de leurs activités dans une base de données d'activité commune. La notion d'Activité Coopérative (AC) est décrite par le Type d'Activité Coopérative (TAC). Ce modèle assigne une Activité d'Utilisateur (AU) à chaque participant dans une AC. Une activité d'utilisateur a sa propre zone de travail qui est isolée des autres. La base de donnée d'activité commune est partagée par tous les utilisateurs, qui y stockent des données communes.

Un TAC se compose de cinq parties :

- Déclaration d'activité : définition des paramètres d'une activité.
- Sous-activités constitutives : spécifie un ensemble d'activités (AC ou activité élémentaire).
- États et les transitions d'état d'une activité : décrit l'activité comme un ensemble d'états observables avec les transitions entre états possibles (non exécutée, exécution, faite et détruite).
- Règles d'exécution : elles dirigent l'exécution des sous-activités constitutives. Ces règles sont séparables en deux parties, les règles pour l'exécution en avant (pour diriger l'exécution régulière d'une activité coopérative et les utilisateurs impliqués) et les règles d'exécution en arrière (pour diriger le repositionnement de l'activité). Les règles peuvent être définies en termes d'états observables, de valeurs de production et de variables externes.
- Règles d'entrelacement d'activité : elles définissent l'existence possible de dépendances entre les sous-activités constitutives d'une activité en terme de flux de données.

Une activité utilisateur possède un historique, stocké dans la base de données d'activité commune. Les opérations réalisables sur une activité comprennent :

- *Import (source_activity)* : importation de parties d'une activité dans la zone de travail de l'activité d'utilisateur courante. *source_activity* désigne une activité qui provient soit la base d'activité commune, soit des autres utilisateurs.
- *Delegate (dest_activity)* : délégation de l'activité d'utilisateur de la zone de travail courante vers la zone de travail de *dest_activity*. *dest_activity* désigne un identificateur d'activité de la base d'activité d'un autre utilisateur.
- *Save* : Sauvegarde de l'activité d'utilisateur courante dans la base de données d'activité commune.

Le concept de *compatibilité* se base sur l'historique et la sémantique des activités. Il est utilisé pour déterminer des dépendances entre les sous-activités. Ces dépendances sont déterminées dans les exécutions d'activités d'un seul utilisateur. Dans le cas d'utilisateurs différents qui exécutent chacun une activité identique, la compatibilité ne retient qu'une seule activité. Dans le cas où ces utilisateurs exécutent des activités différentes et indépendantes, ces dernières sont compatibles entre elles.

La compatibilité sert principalement à organiser des sous-activités au travers de la notion de fusion. Trois cas sont possibles: 1) Les sous-activités sont complètement indépendantes et toutes deux peuvent faire partie de la fusion. 2) Les sous-activités ne sont pas compatibles, on n'en retient qu'une seule dans la fusion. 3) Les sous-activités ne sont pas indépendantes, mais grâce à leurs sémantiques spécifiques, elles sont compatibles : il est permis de les avoir toutes les deux présentes dans la fusion.

La fusion se traduit par un état cohérent dans l'historique de l'activité globale. Elle garantit que les historiques d'AU dans la base de données d'activité commune sont corrects, et qu'ils satisfont toutes les règles d'exécution définies dans le type d'activité coopératif. Elle garantit aussi que les échanges d'information se font de manière consistante pour des activités exécutées en concurrence.

Le modèle CoAct gère les sessions de façon implicite. Il ne traite pas le contrôle du droit de parole.

4.3 Espaces partagés

Le modèle d'espaces partagés de Texier et Plouzeau [TEXI99] introduit la notion d'activité qui associe un ensemble d'utilisateurs avec un ensemble d'objets selon la sémantique : chaque utilisateur possède les objets de l'activité. Une session de travail coopératif lie dynamiquement un sous-ensemble d'objets et des utilisateurs d'une activité donnée. Les sessions sont conduites par des règles qui s'exécutent pour coordonner le partage des données et des objets.

L'utilisation d'un objet nécessite au préalable une liaison vers lui. Cette liaison est définie selon deux règles :

- Un membre m d'une activité A est *actif* sur un objet o d' A si et seulement s'il a une liaison (pour lire et/ou écrire) sur o .
- La session $s_A(M,O)$ de l'activité A lie un ensemble de membres (utilisateurs) M à un ensemble d'objets O de A , tel que chaque membre de M est actif sur chaque objet d' O .

Les propriétés de la session comprennent :

- L'ensemble M des membres de la session est maximal s'il contient tous les membres de l'activité.
- L'ensemble O des objets de la session est maximal s'il contient tous les objets de l'activité que les membres de la session partagent.
- Deux sessions peuvent être fusionnées si et seulement si elles ont le même ensemble de membres ou le même ensemble d'objets.

A partir de ces définitions, cette approche propose un algorithme de gestion de session qui procède en deux phases. La première consiste à trouver la session s qui utilise l'objet o . Si une telle session existe, alors on ajoute l'utilisateur u à l'ensemble des membres de s . La deuxième phase est la fusion des sessions.

Cette approche est du type implicite et elle ne prend pas en compte les aspects de contrôle de droit de parole.

4.4 Coordination de groupe

Le modèle de Dommel et Garcia-Luna [DOMM00a] traite de la coordination et de la collaboration de groupe, liés au contrôle de sessions hiérarchiques. La participation aux sessions est basée sur des rôles, dont la prise se fait de façon interactive, à tour de rôle, pour le travail en groupe.

Un environnement de collaboration Γ sur un réseau d'ordinateurs est défini comme un tuple :

$$\Gamma = \langle S, U, R, F \rangle$$

où S est un ensemble de sessions Σ , U est un ensemble d'éléments (des hôtes, des processus, des agents, des participants), R est un ensemble de ressources partagées (des médias) et F est un ensemble de droits (floors) contrôlant les ressources.

Dans ce modèle, les liens entre les éléments sont représentés sous forme de graphe avec des nœuds (stations, hôtes) V envoyant des messages au travers de liens (canaux) $E \subset V \times V$.

Une connexion est une liaison de transmission unidirectionnelle ou bidirectionnelle d'un nœud expéditeur vers un ensemble de nœuds récepteurs [DOMM00b].

Une session fournit l'infrastructure pour la coopération et la collaboration. Une session $\Sigma \in S$ est un tuple :

$$\Sigma = \langle Sid, T_i, T_e, A_S, L \rangle$$

où Sid est un identificateur unique par environnement collaboratif Γ , T_i est le temps d'initiation ou d'annonce, T_e est le temps final et A_S est une liste d'attributs caractérisant la session de niveau hiérarchique L (valeur 0 par défaut).

Un utilisateur $U \in \mathcal{U}$ est un tuple :

$$U = \langle Uid, Sid, Loc, T_j, T_l, A_U \rangle$$

où Uid est un identificateur unique dans la session Sid , Loc est son emplacement donné par l'adresse IP ou par l'identificateur unique d'hôte, T_j est la date à laquelle l'utilisateur s'est joint à la session, T_l est sa date de départ et A_U est une liste d'attributs d'utilisateur (rôle, identité, accès, etc.).

Une ressource $R \in \mathcal{R}$ est un tuple :

$$R = \langle Rid, Sid, Pid, Uid, T_c, T_d, A_R \rangle$$

où Rid est un identificateur de ressource unique appartenant à l'utilisateur Uid dans la session Sid . Pid est l'identificateur du propriétaire de la ressource Rid , T_c est la date de création de la ressource dans la zone de travail collaboratif, T_d est la date de destruction et A_R est une liste d'attributs de ressource (type, utilisation, priorité, QoS, protection, etc.).

Un droit (floor) $F \in \mathcal{F}$ est un tuple :

$$F = \langle Fid, Rid, Uid, T_i, T_d, A_F \rangle$$

où Fid est un identificateur de contrôle de droit unique dans la zone de travail partagée pour une ressource Rid . Ce droit est assigné à l'utilisateur Uid à la date de démarrage T_i et désactivé au temps T_d . A_F précise le type de droit (son état, sa politique, sa modalité, sa stratégie, etc.).

Les droits sont indirectement associés aux sessions via Rid , et les propriétés d'un droit peuvent être héritées d'une ressource maître vers ses sous-composants. Un droit contrôle exactement une ressource Rid , mais une ressource Rid peut se voir assignée de multiples droits Fid avec des contrôles de granularités différents.

Le modèle fournit une taxonomie des éléments essentiels trouvés dans les environnements coopératifs. Cette approche introduit une structure formelle pour des sessions sur Internet et pour la médiation d'accès à des ressources partagées de façon concurrente [DOMM99]. Le modèle de session est du type centré environnement.

4.5 Synthèse des modèles de session

La table 1.2 résume les caractéristiques des modèles analysés. Elle illustre la diversité des approches utilisées, tant du point de vue des formalismes que des buts du modèle. Actuellement, aucune approche ne semble se détacher des autres pour manipuler des sessions synchrones.

	Formalisme mathématique	Eléments modélisés	Type de session	Contrôle de droit de parole	Utilisation du modèle	Etat d'avancement
Intermezzo	Ensembles	Activité qui associe un ensemble d'utilisateurs, de tâches et d'objets	Implicite Explicite	Politiques de contrôle associées aux rôles	Création flexible de sessions	Concret : système Intermezzo
CoAct	Logique du premier ordre	Activité	Implicite	Absent	Gestion et exécution consistante d'activités	Abstrait
Espaces partagés	Ensembles	Activité	Implicite	Absent	Algorithme de gestion de session	Abstrait
Coordination de groupe	Graphes	Sessions comprenant des utilisateurs, des hôtes, des ressources	Explicite	Politiques de contrôle associées aux ressources	Définition de sessions Gestion de droits d'accès à leurs ressources	Abstrait

Table 1.2. Principaux modèles de sessions synchrones

5. Services de communication pour la gestion de session

Une plate-forme de travail collaboratif comprend, en plus des outils collecticiels, un ensemble de services de gestion et de coordination des différents éléments mis en relation (utilisateurs, outils collecticiels, données). Ces services implantent notamment les fonctionnalités suivantes, incontournables dans tout système collaboratif :

- Définition des sessions
 - Définition des utilisateurs, de leurs rôles, de leurs droits d'accès sur les données et de leurs droits de contrôle sur les outils
- Gestion des sessions
 - Activation/désactivation des sessions
 - Gestion de l'accès des membres aux sessions actives
- Coordination en session
 - Contrôle de l'attribution des droits utilisateurs en session
 - Activation/désactivation des outils dans une session

Conformément à notre démarche, nous avons volontairement orienté notre analyse vers des systèmes collecticiels gérant des sessions pour assurer une meilleure relation avec nos contributions et ainsi mieux positionner nos travaux par rapport à ceux existants.

De nombreux systèmes collaboratifs sont basés sur le Web et développés en Java. Ces systèmes sont les plus représentatifs que nous avons trouvés dans la littérature.

Nous avons focalisé notre étude vers l'analyse des systèmes qui gèrent des sessions de collaboration synchrones pour deux raisons :

- La première est que ces systèmes ont des contraintes fonctionnelles et temporelles plus difficiles à résoudre que les systèmes manipulant des sessions asynchrones. Les besoins réseau, notamment pour garantir la co-présence distribuée des membres de la session sont aussi plus grands. De ce fait, chronologiquement, les systèmes basés sur des sessions synchrones sont apparus plus tard, à partir des années 1990.
- La deuxième raison est que l'ensemble de nos travaux se situe dans le cadre de la manipulation de sessions synchrones.

Cependant, pour des raisons de complétude, nous citerons les trois travaux CHAOS [BIGN89], COSMOS [DOLL89], [HENN89] et AMIGO [HENN89], qui gèrent des sessions asynchrones, travaux que nous avons trouvés importants et caractéristiques. Le système CHAOS ("Commitments Handling Active Office System") [BIGN89] est un système de messagerie électronique sensible au contexte qui fournit une assistance aux procédures et aux tâches bureautiques. Il classe, répertorie et archive les connaissances échangées au travers d'une messagerie électronique. Les messages échangés au cours de conversations sont semi structurés de façon à ce que le système puisse les interpréter. Ces messages sont organisés en centre d'intérêts, regroupés par la suite en sujets. Cette organisation des connaissances suit aussi la dynamique du groupe et est influencée par la structure de ce dernier. D'autres travaux voisins comme COSMOS [DOLL89], [HENN89], AMIGO [HENN89] offrent un support à des groupes d'utilisateurs au dessus d'un système de messagerie.

L'analyse des systèmes qui gèrent des sessions synchrones a été effectuée en partant des divers éléments de session identifiés. Ces éléments comprennent :

- L'espace utilisateur
- L'espace outil
- La session générale

Nous avons raffiné chacun de ses éléments en un ou plusieurs critères de gestion précis qui nous semblent importants en termes de comparaison et de positionnement de nos travaux. Ainsi :

- La gestion des rôles utilisateur (attachée à l'espace utilisateur). Ce critère définit le type de rôle manipulé par le système
- Le démarrage/arrêt d'un outil (attaché à l'espace outil). Ce critère précise l'instant où un utilisateur voit un outil démarrer (ou s'arrêter). On peut ainsi préciser s'il peut démarrer (ou arrêter) un outil manuellement ou automatiquement
- La politique de contrôle d'accès (attachée à l'espace outil). Ce critère précise comment sont gérés et attribués les droits d'accès
- La sauvegarde de session (attachée à la session générale).

Remarque : Dans cette analyse, nous avons volontairement omis l'espace d'information identifié comme élément de session. Ce choix se justifie par le fait que : d'une part les

données et les flux d'information sont toujours manipulés au travers des outils et qu'ils sont sous la responsabilité directe de ces derniers (et non du système gestionnaire de session) ; d'autre part, la gestion de l'accès aux données nous semble plus du ressort des systèmes de bases de données que des systèmes de gestion de session.

Nous présentons, classons et analysons par la suite un ensemble de systèmes significatifs.

5.1 Tango 2

TANGO 2 [TANG05] (actuellement le prototype CollabWorx), développé au Northeast Parallel Architectures Center (NPAC), est une plate-forme d'intégration écrite en Java, qui permet la mise en oeuvre d'environnements collaboratifs basés sur le Web [COLL05]. Il fournit des moyens pour l'intégration rapide d'applications Web et non Web dans son système collaboratif multi-utilisateur.

TANGO 2 introduit le paradigme de chambre et encapsule toute session dans une chambre. Chaque chambre reproduit une entité abstraite avec des ensembles séparés d'utilisateurs qui manipulent des applications.

Pour chaque session TANGO 2, les utilisateurs connectés respectent un schéma maître/esclave. Le maître de la session possède des privilèges spéciaux pour contrôler l'accès des autres utilisateurs à la session. Les autres participants (les esclaves) peuvent seulement observer les actions produites par le maître. L'utilisateur qui crée une session est automatiquement le maître, c'est-à-dire le coordonnateur. Le mode maître peut cependant être transféré à un autre utilisateur.

Une session TANGO 2 regroupe un ensemble d'applications qui fonctionnent en mode multi utilisateur collaboratif [BECA98]. Toutes les applications appartenant à la même session sont vues de la même façon par les utilisateurs connectés. Elles sont lancées automatiquement chez tout utilisateur se connectant. Elles peuvent être spontanément démarrées ou arrêtées par tout utilisateur présent en session. Le contrôle d'accès est basé sur un mécanisme de verrous qui autorise un contrôle plus fin des applications : un utilisateur de la session peut demander un verrou sur une application pour en prendre le contrôle et libérer ce verrou ensuite.

5.2 Habanero 3.0

L'environnement Habanero 3.0 [HABA05] est le résultat de l'effort de recherche *framework for Integrated Synchronous And Asynchronous Collaboration* (ISAAC) fondé par le programme de Collaboration Intelligente et Visualisation (IC&V) du Bureau de Technologie de l'Information (Information Technology Office, ITO) de l'Agence de Projets de Défense de la Recherche Avancée (DARPA) [NCSA01].

Habanero 3.0 est un environnement logiciel pour la conception et la construction d'applications Java collaboratives synchrones [CHAN98]. Pour cela, il offre aux développeurs un canevas de conception et une interface de communication (API). L'API contient les méthodes nécessaires pour créer ou étendre des applications existantes et pour manipuler des applets dans des applications collaboratives.

Habanero 3.0 définit une session comme une réunion où un utilisateur peut rejoindre d'autres participants. Un utilisateur initialise une réunion. Une session regroupe plusieurs applications et de multiples participants.

Tous les utilisateurs présents dans une session ont des rôles identiques. Un même participant peut appartenir à plusieurs sessions différentes en même temps.

Toute application peut être ouverte dans une session. Elle est vue de la même façon par chaque membre de la session. L'application client Habanero 3.0 fournit à chaque utilisateur une Interface Utilisateur Graphique (GUI) pour visualiser globalement la session. Habanero 3.0 reproduit les applications (appelées applications répliquées) vers chaque client et transmet ensuite tous les changements d'état vers chaque client. Quand un nouveau client joint une session, il est informé et il voit toutes les applications qui s'exécutent dans cette session. Habanero 3.0 s'assure aussi que tous les clients voient les mêmes événements dans le même ordre par une synchronisation d'ordre causal total, ce qui garantit que les vues que les clients ont des applications sont les mêmes.

Habanero 3.0 fournit une suite d'outils de base pour établir une collaboration à distance : un tableau blanc, un navigateur web collaboratif Savina, un logiciel Telnet, un outil de vote, un outil de dialogue textuel (chat), un éditeur collaboratif, etc. Ces outils couvrent les trois cadres de collaboration : production, coordination et communication.

Le contrôle d'accès s'appuie sur un mécanisme de verrous.

Toute session peut être sauvegardée, puis rejouée.

5.3 DOE 2000

Le projet DOE2000 [DOE2000], du département américain de l'énergie, est une initiative pour explorer des nouveaux outils informatiques et pour proposer des bibliothèques pour le Test Informatique Avancé et la Simulation.

L'exploration de la Gestion de la Collaboration en temps réel, avec la définition et le développement d'un environnement sophistiqué, a commencé en 1997. Cette proposition couvre la programmation d'un gestionnaire qui réalise en temps réel l'intégration, la manipulation et la gestion d'outils différents dans un environnement collaboratif commun. Le gestionnaire supporte le développement d'interfaces d'environnement supplémentaires, permettant aux utilisateurs : de joindre une session de collaboration en la choisissant depuis une liste de sessions actives, de se connecter à un instrument partagé, de s'enregistrer dans un carnet partagé, etc.

Les laboratoires Pacific Northwest National Laboratory (PNNL) et National Center for Supercomputing Applications (NCSA) développent ensemble le Moteur de Gestion de collaboration en Java, à partir de l'environnement Habanero. Plus précisément, le laboratoire PNNL développe CORE [MYER97], un système hybride avec des fonctionnalités avancées disponibles à travers deux API standardisées :

- Une API pour la Gestion de la Collaboration : elle fournit l'accès aux fonctionnalités de gestion de collaboration et permet le développement d'interfaces utilisateur supplémentaires
- Une API pour les outils collaboratifs : elle autorise l'intégration de nouvelles applications collaboratives.

L'accent est mis sur la facilité d'utilisation : pour joindre une session collaborative en utilisant CORE, un utilisateur clique simplement sur les boutons appropriés d'une page Web et tout le logiciel nécessaire est lancé automatiquement sur sa machine.

Les caractéristiques de DOE 2000 sont très proches de celles de Habanero 3.0 (utilisateur identiques dans les sessions, démarrage/arrêt des outils possible lorsque l'on joint/quitte la

session ou démarrage/arrêt manuel en session, sauvegarde de sessions et technologie de développement basée sur JAVA).

Par contre, les politiques de contrôle d'accès sont plus riches que celles d'Habanero 3.0. Pour chaque contrôle d'accès dans une session, il est possible de définir et de choisir des mécanismes différents : sans contrôle, prise de contrôle par un utilisateur autorisé, contrôle géré par un utilisateur modérateur, contrôle par passage de bâton.

5.4 DISCIPLE

DISCIPLE (DIstributed System for Collaborative Information Processing and LEarning) [DISC05] est une plate-forme collecticiel synchrone [MARS99], [DORO99]. Il fournit un support pour le développement et le déploiement d'applications conscientes de l'existence du groupe collaboratif et d'applications ignorant son existence, c'est-à-dire des applications à l'origine développées pour un seul utilisateur.

L'architecture de DISCIPLE est hybride, car elle se compose d'applications client répliquées chez chaque utilisateur, de serveurs de ressources centralisés et d'un bus de collaboration. Le bus de collaboration est l'élément central de cette architecture car il rassemble les services de communication nécessaires (garantie de l'ordre des événements et contrôle de la concurrence). Les serveurs de ressources gardent les informations manipulées en session. Les applications se composent d'ensembles de composants JavaBeans répliqués.

Une session DISCIPLE respecte le concept de chambre. Une chambre est définie comme une réunion où deux participants ou plus collaborent sur un sujet particulier et partagent l'ensemble des composants JavaBeans qui fait partie de la session. Les utilisateurs présents dans une session ont des rôles identiques.

Chaque utilisateur possède et exécute une copie de l'application client, copie qui contient tous les composants utilisés en session. Tous les changements qui interviennent dans les applications ou dans la session sont répliqués vers tous les utilisateurs. Ces derniers ont donc des vues consistantes des applications manipulées en session.

Les applications de l'utilisateur sont lancées automatiquement lors de son entrée dans la session. De la même façon, tout participant en session peut lancer ou arrêter une application partagée.

Aucun mécanisme de contrôle d'accès n'est implanté au niveau de l'ensemble du système. Le choix est d'implanter un contrôle d'accès par application. Dans ce cadre, il est à noter un contrôle d'accès original qui fonctionne au moyen d'arbres distribués, implanté pour un éditeur de texte et pour un éditeur d'objets 3D collaboratifs [IONE03].

L'espace d'exécution des applications est persistant. Le système garde toujours le dernier état des applications exécutées. Ceci lui confère des possibilités de sauvegarde des sessions.

5.5 Synthèse des systèmes gestionnaires de session

La table 1.3 résume les caractéristiques des systèmes analysés. Nous notons une assez grande convergence en terme de caractéristiques et de choix techniques.

Les sessions manipulées sont toutes non structurées (les rôles des utilisateurs sont identiques, ou bien ils ne possèdent que peu de choix de rôles). Leur taille reste autour de la dizaine d'utilisateurs en session.

Ces plates-formes se focalisent sur le fait de rendre multi utilisateur des applications. Ces dernières s'exécutent dans un environnement qui permet à un ensemble d'utilisateurs de les voir et de les piloter en commun.

Toutes ces plates-formes sont des réalisations technologiques avancées, et certaines comme Habanero 3.0 et DOE 2000 peuvent presque être considérées comme des produits (et non seulement des prototypes). Tango 2 a évolué en un produit commercial, CollabWorx.

	Gestion des rôles utilisateur	Démarrage/arrêt d'un outil	politique de contrôle d'accès	Sauvegarde de session	Technologie
TANGO 2	{N utilisateurs identiques et un maître} par session	- Au moment de joindre/quitter la session - Manuellement en session	Verrous	Non disponible	Applets JAVA
Habanero 3.0	N utilisateurs identiques par session	- Au moment de joindre/quitter la session - Manuellement en session	Verrous	Disponible	JAVA
DOE 2000	N utilisateurs identiques par session	- Au moment de joindre/quitter la session - Manuellement en session	Variés -utilisateur autorisé -modérateur -passage de bâton	Disponible	JAVA
DISCIPLE	N utilisateurs identiques par session	- Au moment de joindre/quitter la session - Manuellement en session	Aucun au niveau du système	Disponible	JavaBeans

Table 1.3. Principaux systèmes de gestion de session synchrones

Malheureusement, la faiblesse en terme de représentations de sessions se traduit par le peu de contrôle automatique qu'offrent ces plates-formes sur les outils. En effet, les seules interactions possibles avec les outils ne se produisent automatiquement que lors de l'entrée ou de la sortie en session d'un utilisateur. Dans tous les autres cas, les outils doivent être lancés à la main, soit par un utilisateur privilégié, soit par tout utilisateur le souhaitant.

Nous pensons qu'une approche de représentation de session, orientée modèle, et de ce fait plus riche, est une bonne direction pour augmenter le degré d'interaction entre les événements de session et la configuration des outils, et surtout pour faire prendre en charge ces interactions automatiquement par un gestionnaire de session.

Le cadre des travaux exposés par la suite présentera nos contributions en terme de modélisations de session, puis sur les méthodologies et les choix architecturaux retenus pour concevoir certains services de gestion de session.

Chapitre 2. Modélisation de sessions collaboratives

Le chapitre 2 présente nos résultats en terme de modélisation de sessions, conformément au canevas d’instanciation des résultats obtenus (figure 0.2). Il est centré sur l’instance *Diagrammes de Coordination* qui représente notre modèle proposé et utilisé.

La partie 1 présente les trois éléments à co-gérer (utilisateurs, outils, données) pour ces sessions et justifie notre approche orientée modèle. La partie 2 décrit le modèle de *Diagrammes de Coordination*, en correspondance avec l’instance de même nom dans le canevas de résultats. La partie 3 présente l’implantation du modèle de session et son utilisation pour visualiser l’état courant de la session. Ces contributions détaillent le lien entre l’instance *Diagrammes de Coordination* et l’instance *Gestion de Sessions* de la figure 0.2. La partie 4 montre les principaux scénarios modélisés dans plusieurs cadres contractuels. Cette contribution décrit le lien entre l’instance *Diagrammes de Coordination* et l’instance *Téléenseignement/Téléingénierie* de la figure 0.2.

1. Introduction

Le travail à réaliser en collaboration peut être représenté de façon plus ou moins structurée. Ce travail se décompose en un ensemble d’activités qui représentent des unités de travail faiblement couplées entre elles. Ces activités sont instanciées sous forme de tâches, lorsque certains membres les prennent en charge et réalisent le travail associé.

L’ensemble des ressources, tant humaines que matérielles et logicielles, mises en œuvre pour réaliser le travail, est regroupé à l’intérieur de *Sessions Collaboratives*, appelées *Sessions* par la suite.

1.1 Principaux éléments de session

Dans les sessions manipulées, trois principaux espaces peuvent être identifiés (figure 2.1) : *l’espace utilisateur*, *l’espace outil* et *l’espace des ressources d’information*.

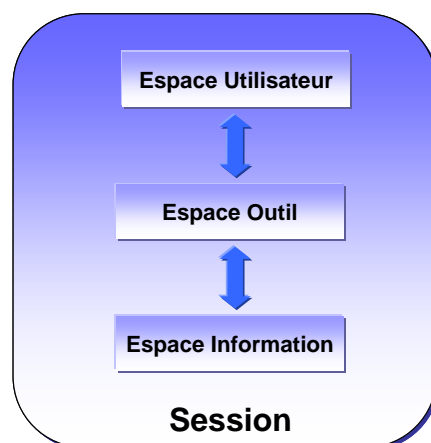


Figure 2.1. Espaces qui composent une session

- *L'espace utilisateur* : il décrit l'ensemble des utilisateurs ou participants qui forme le groupe distribué.

Ces utilisateurs ont des attributs, principalement rôles et contrôle du droit de parole :

- a) le rôle, acquis par un utilisateur lorsqu'il se joint à la session, représente les droits qu'il possède sur l'utilisation des outils, ainsi que sur l'utilisation des ressources d'information.
- b) Le contrôle de droit de parole est un attribut dynamique négociable, attribué à un utilisateur pendant une session. La possession de cet attribut permet à cet utilisateur de modérer, coordonner et éventuellement restreindre les actions des autres utilisateurs sur les outils et sur les informations [BAUD03], [RODR03], [RODR04].

L'accès des utilisateurs à l'espace d'outils permet l'exécution du travail.

- *L'espace outil* : il décrit l'ensemble des outils collecticiels (tableau blanc, outil de dialogue textuel, partage d'applications, éditeurs, etc.) que les membres en session utilisent. Ces outils permettent aux utilisateurs de manipuler les ressources d'information de la session.
- *L'espace des ressources d'information* : Cet espace contient les informations et les flux de données (documents, vidéo, audio, etc.) qui appartiennent aux utilisateurs.

Le lien entre ces trois espaces est fait au moyen de divers formalismes, selon les modèles utilisés par différents travaux.

1.2 Objectifs de la modélisation

Le but recherché par notre approche de modélisation est double :

- Tout d'abord, nous cherchons à décrire les éléments identifiés dans ces sessions, le modèle facilitant et aidant le concepteur de session dans sa démarche. Cette approche descriptive a été exploitée notamment pour décrire des scénarios du domaine de la téléformation.
- Par la suite, nous visons à faire prendre en compte automatiquement les descriptions effectuées par des systèmes ou des plates-formes programmables par modèle. Ceci nécessite de se focaliser sur les langages et techniques informatiques qui supportent l'implantation et la programmation de tels modèles.

2. Diagrammes de coordination

Le modèle que nous proposons, commencé en [VILL95a], est appelé Diagrammes de Coordination Dynamiques (DCD). Il est entièrement basé sur des graphes étiquetés dynamiques qui fournissent à la fois une représentation uniforme des éléments de sessions et une façon de les lier entre eux.

2.1 But du modèle

Ce modèle s'appuie sur des relations de type producteur/consommateur de données. Nous avons volontairement orienté le modèle dans ce sens car nous désirons traiter et représenter des échanges de données dans des sessions de travail synchrones et interactives. De telles sessions manipulent des flux de données interactifs (par exemple vidéo, audio temps réel) qui se représentent naturellement par des relations entre utilisateurs.

Ces relations sont liées aux rôles de chacun des utilisateurs. Elles permettent de les différencier en terme de manipulation des flux d'information et de droits d'utilisation des outils. La notion de rôle se rencontre fréquemment dans les systèmes manipulant des sessions asynchrones, mais est plus rare dans ceux faits pour des sessions synchrones.

Cette orientation flux d'information nous sert aussi à maintenir une relation forte avec le réseau. En effet, ce dernier donne des contraintes en terme de connectivité, de débit et de qualité des liens reliant les membres entre eux. La confrontation des flux d'information décrits à haut niveau dans notre modèle de session, avec les contraintes de bas niveau offertes par le réseau, la prise en compte puis la mise en place de ces flux vis-à-vis des contraintes imposées, est aussi un des objectifs que nous poursuivons dans notre approche de modélisation.

2.2 Modèle

2.2.1 Présentation

Une session est modélisée par des diagrammes de coordination dynamiques qui décrivent les utilisateurs actifs avec leurs relations entre eux. Un diagramme de coordination est un multigraphe étiqueté dont les nœuds représentent les utilisateurs. Ces utilisateurs possèdent un ensemble d'attributs, en particulier un identificateur et un rôle. D'autres attributs, comme les compétences, non obligatoires pour la compréhension de l'exécution du modèle, peuvent leur être ajoutés. Les attributs des utilisateurs apportent des compléments d'information lors d'une vraie visualisation de la session et sont utilisés lors de la gestion de la session.

Les relations du graphe traduisent les échanges de données entre utilisateurs ; ils sont représentés par les flèches du graphe. Les étiquettes utilisées sur chacune des flèches sont composées de deux champs attribut : $\langle nom_outil \rangle . \langle nom_flux_de_donnée \rangle$. Le premier champ donne l'identificateur d'un outil de la session, le second donne le nom d'un flux de donnée produit par cet outil.

Les notations utilisées pour formaliser les diagrammes DCD ont été séparées en deux parties. La table 2.1 donne la liste des informations qui sont globales à la session. Ces informations restent inchangées au cours du temps.

Notations pour les informations générales	
$U_G = \{u_1, \dots, u_n\}$	Liste de tous les utilisateurs qui peuvent faire partie de la session
$Nom = \{n_1, \dots, n_k\}$	Identificateur des utilisateurs
$Role = \{ro_1, \dots, ro_l\}$	Rôles des utilisateurs
$Outil = \{o_1, \dots, o_m\}$	Liste des outils
$Flux = \{f_1, \dots, f_n\}$	Flux de données

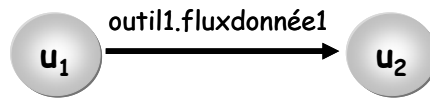
Table 2.1. Informations générales des DCD

La table 2.2 donne les informations nécessaires pour représenter la vue courante de la session, en terme d'utilisateurs présents à un instant donné, de relations entre eux, d'outils et de flux de données manipulés à ce moment là.

Notations pour les informations courantes	
$G = \langle U, R \rangle$	G représente le DCD courant
$U = \{u_1, \dots, u_i\}$	Liste des utilisateurs présents à un instant donné (sous-ensemble de U_G)
$R = \{r_1(u_i, u_j), \dots, r_k(u_l, u_m)\}$	Relations entre utilisateurs présents (les r_k font partie d'éléments du produit cartésien $U \times U$)
Attributs de chaque utilisateur u_i	
$att_util : U \rightarrow \langle Nom, Role \rangle$	La fonction att_util associe un nom et un rôle à chaque utilisateur u_i présent en session à un instant donné
Attributs de chaque relation $r_k(u_l, u_m)$	
$att_rel : R \rightarrow \langle Outil, Flux \rangle$	La fonction att_rel associe un outil et un flux à chaque relation r_k

Table 2.2. Informations courantes des DCD

La figure 2.2 décrit une relation entre deux utilisateurs. L'utilisateur u_1 produit le flux de données 1 qui est consommé par l'utilisateur u_2 . Ce diagramme est décrit par la table 2.3.

Figure 2.2. L'utilisateur U_1 partage le flux de donnée 1 avec l'utilisateur U_2

Informations générales	Informations courantes	
$U_G = \{u_1, u_2\}$	$G = \langle U, R \rangle$	$att_util(u_1) = \langle u_1, utilisateur \rangle$
$Nom = \{u_1, u_2\}$	$U = \{u_1, u_2\}$	$att_util(u_2) = \langle u_2, utilisateur \rangle$
$Role = \{utilisateur\}$	$R = \{r_1(u_1, u_2)\}$	$att_rel(r_1) = \langle outil_1, fluxdonnee_1 \rangle$
$Outil = \{outil_1\}$		
$Flux = \{fluxdonnee_1\}$		

Table 2.3. Formalisme associé à la figure 2.2

2.2.2 Exemple

L'exemple de la figure 2.3 illustre l'utilisation du modèle dans un cas de téléformation. Dans cette session, un cours se déroule avec un professeur et peut être suivi par N étudiants. Trois outils sont disponibles : un navigateur web synchrone (browser), une vidéoconférence (vidconf), et un outil de dialogue textuel (chat). La table 2.4 montre les informations globales à la session.

Informations générales
$U_G = \{p, e_1, e_2, e_3, e_4\}$
$Nom = \{Professeur, Etudiant_1, Etudiant_2, Etudiant_3, Etudiant_4\}$
$Role = \{professeur, etudiant, etudiantprivilegie\}$
$Outil = \{browser, vidconf, chat\}$
$Flux = \{comment, transp\}$

Table 2.4. Informations générales de la session de téléformation

A un instant donné (t0), le professeur diffuse son cours sous la forme de pages Web. Chaque membre du groupe possède deux outils. Le premier, un navigateur Web synchrone, sert à visualiser le cours du professeur. Il est utilisé par le professeur pour diffuser (producteur de données) ses transparents. Les étudiants utilisent une autre instance de ce navigateur Web synchrone pour afficher les transparents (consommateurs de données). L’outil de vidéoconférence sert aux communications informelles entre utilisateurs. Il est utilisé par chaque membre (producteurs et consommateurs) pour envoyer ses commentaires au travers de liens audio et vidéo. La figure 2.3 représente le diagramme de coordination associé à cette session de téléformation lorsque deux étudiants sont présents, avec les deux outils nécessaires. La table 2.5 représente ce même diagramme de façon formelle.

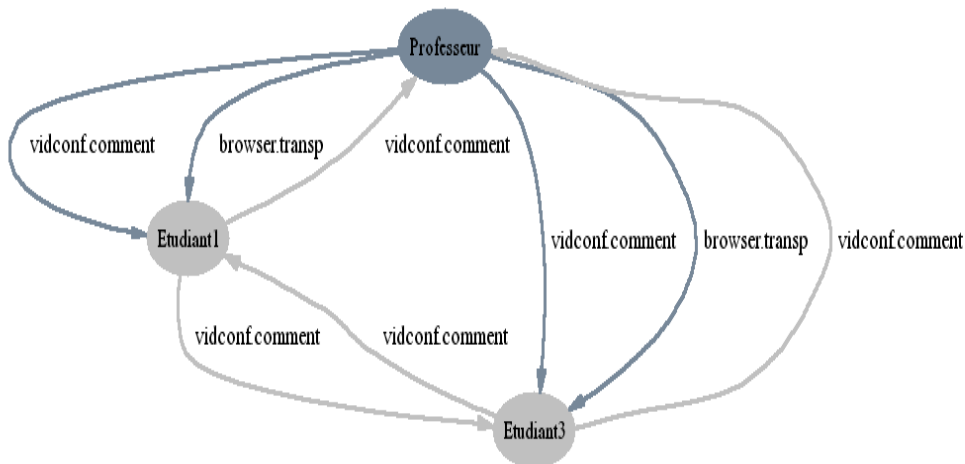


Figure 2.3. Diagramme de coordination pour une session de téléformation à l'instant t0

Informations courantes		
$G_{t_0} = \langle U_{t_0}, R_{t_0} \rangle$	$att_util(p) = \langle Professeur, professeur \rangle$	$att_rel(r_1) = \langle vidconf, comment \rangle$
$U_{t_0} = \{p, e_1, e_3\}$	$att_util(e_1) = \langle Etudiant_1, etudiant \rangle$	$att_rel(r_2) = \langle browser, transp \rangle$
$R_{t_0} = \{r_1(p, e_1), r_2(p, e_1),$	$att_util(e_3) = \langle Etudiant_3, etudiant \rangle$	$att_rel(r_3) = \langle vidconf, comment \rangle$
$r_3(p, e_3), r_4(p, e_3),$		$att_rel(r_4) = \langle browser, transp \rangle$
$r_5(e_1, p), r_6(e_1, e_3),$		$att_rel(r_5) = \langle vidconf, comment \rangle$
$r_7(e_3, p), r_8(e_3, e_1)\}$		$att_rel(r_6) = \langle vidconf, comment \rangle$
		$att_rel(r_7) = \langle vidconf, comment \rangle$
		$att_rel(r_8) = \langle vidconf, comment \rangle$

Table 2.5. Etat courant de la session de téléformation à l'instant t0

2.2.3 Diagrammes de Coordination valides

On considère à présent la façon dont l'activité coopérative associée au groupe peut débiter. La solution triviale correspond au cas où le travail coopératif commence lorsque tous les membres du groupe sont prêts à démarrer. Bien entendu, cette vue est trop restrictive car le travail peut être réalisé dès qu'un sous-ensemble adéquat de participants vient à exister. Cela signifie que, à un certain instant, il n'est pas nécessaire que la totalité des participants ou des ressources soient présentes afin de démarrer ou de mener le travail coopératif. De plus, la façon dont les membres interagissent les uns avec les autres peut être différente à différents moments. Par conséquent, le modèle conceptuel est étendu pour définir quels participants doivent coopérer en même temps, et avec quelles ressources.

L'application coopérative associée au groupe doit définir les configurations qui ont un sens pour la réalisation du travail. Parmi tous les diagrammes de coordination possibles, elle choisit ceux qui sont sémantiquement possibles. Les graphes retenus sont appelés *Diagrammes de Coordination Valides*. On notera cet ensemble G^* (table 2.6).

$G^* = \{G_{ii}\}$	Ensemble des DCD valides
--------------------	--------------------------

Table 2.6. *Configurations valides*

Pour définir quels diagrammes sont valides, l'application doit fournir un ensemble de propriétés, ou critères, que ces diagrammes doivent respecter.

Ces propriétés peuvent être générales aux sessions : par exemple, il est possible d'imposer que, dans chaque configuration courante, chaque membre participant doive être en relation avec au moins un autre membre. Ceci se traduit de façon triviale en théorie des graphes, par la nécessité de ne manipuler que des graphes connexes.

D'autres propriétés dépendent bien entendu du travail coopératif à accomplir. Elles portent en général plus sur les valeurs des attributs des diagrammes. A titre d'exemple, dans le cadre d'une session de téléformation classique, on ne définira comme valides que seules les sessions pour lesquelles le professeur et au moins un élève sont présents.

Dans la suite de notre mémoire, pour des raisons de simplification, on confondra *Diagrammes de Coordination valides* et *Diagrammes de Coordination*.

2.3 Dynamique des sessions

Les changements dynamiques dans le temps décrivent l'évolution de la session. Au niveau du modèle, ils font passer d'une configuration valide à une autre. Les dynamiques identifiées sont au nombre de trois : *dynamique du groupe*, *des rôles* et *des outils*.

2.3.1 Événements avec règles de transformation pour la dynamique

Le déroulement du travail collaboratif au cours du temps entraîne la création d'événements de la part des utilisateurs avec leurs rôles associés, et de la part des outils qu'ils manipulent. Un exemple d'événement est une demande d'entrée en session de la part d'un nouveau membre. Nous avons identifiés trois groupes d'événements : ceux qui concernent le groupe, ceux qui ont un rapport avec les rôles utilisateurs, et ceux qui concernent les outils.

L'arrivée d'un événement provoque la réalisation d'une série d'actions sur la session courante. Cette série d'actions est appelée *règle de transformation de graphe*. Au niveau du modèle, cette règle fait passer d'une configuration valide à une autre.

Chaque règle de transformation de graphe définit les modifications qui doivent être appliquées à la structure du diagramme de coordination courant. Ces modifications se traduisent en terme de :

- Ajout/suppression de nœuds
- Ajout/suppression/modifications de relations
- Ajout/suppression/modifications des fonctions attributs

De façon plus précise, une règle de transformation [DRIR00] se présente sous la forme d'une pré-condition et d'une post-condition, chacune composée d'un ensemble de nœuds, de flèches, et d'étiquettes. Pour que la règle soit applicable, la pré-condition doit être unifiable (au sens de l'unification logique) avec une partie du diagramme de coordination courant. Si la règle est applicable, et si elle est appliquée, le système active les modifications décrites par la post-condition sous forme de changements sur les nœuds, flèches et étiquettes. Le diagramme est ainsi modifié ; il en résulte un nouveau diagramme de coordination.

Dans le cadre de nos travaux sur la modélisation de session, nous n'avons que peu contribué à la proposition de règles de transformation génériques. Nous avons simplement défini un ensemble de fonctions de transformation élémentaires qui opèrent respectivement au niveau d'un nœud, d'une relation ou d'un attribut. Chacune des transformations complexes a ensuite été implantée par programme, en combinant entre elles les fonctions élémentaires.

Cependant, nous avons participé avec K. Drira à la proposition et à l'utilisation de règles de transformations de graphes génériques [DRIR99], lien établi et renforcé dans le cadre d'un Appel d'Offre Télécommunications du CNRS intitulé "Nouveaux Services et nouvelles méthodologies pour le travail coopératif". Les fonctions de transformation élémentaires que nous avons programmées sont utilisables pour implanter ce type de règles génériques.

Nous décrivons plus en détail les règles de transformation de graphes lors de la présentation de notre contribution méthodologique sur la gestion de session, dans le Chapitre 3.

A cause de l'application des règles de transformation, la structure du Diagramme de Coordination évolue dans le temps. Cependant, il est nécessaire que cette évolution maintienne une structure courante qui reste compatible avec le bon déroulement du travail de groupe. Cela signifie que le nouveau diagramme de coordination doit faire partie de l'ensemble G^* des diagrammes de coordination valides.

2.3.2 Groupe

Ce type de dynamique implique l'entrée ou la sortie d'utilisateurs à l'intérieur d'une session. Les changements interviennent dans le groupe d'utilisateurs travaillant de façon synchronisée pendant la session. L'arrivée d'un événement de demande d'entrée ou de sortie, puis l'application de la règle de transformation qui lui est associée, entraîne une modification de la structure du diagramme de coordination courant : l'ensemble de nœuds du diagramme de coordination est modifié selon les changements de participants. Un nœud est ajouté lorsqu'un participant arrive, et est supprimé lorsqu'il quitte le groupe. Par effet de bord, les flèches liées aux nœuds modifiés sont également changées. Ceci induit des changements pour les flux de données des utilisateurs directement connectés au nœud modifié.

Nous repartons du scénario de la figure 2.3 avec un professeur et deux étudiants. Au cours de cette session, nous considérons que les étudiants peuvent se connecter et se déconnecter dynamiquement à la session. L'entrée de l'étudiant 2 dans la session de téléformation à

l'instant t_1 introduit un changement parmi les membres du groupe. La nouvelle structure de la session est donnée par la figure 2.4.

La récupération des outils et du rôle du nouveau membre, qui font partie du contexte du nouvel utilisateur, est traitée par le gestionnaire de session.

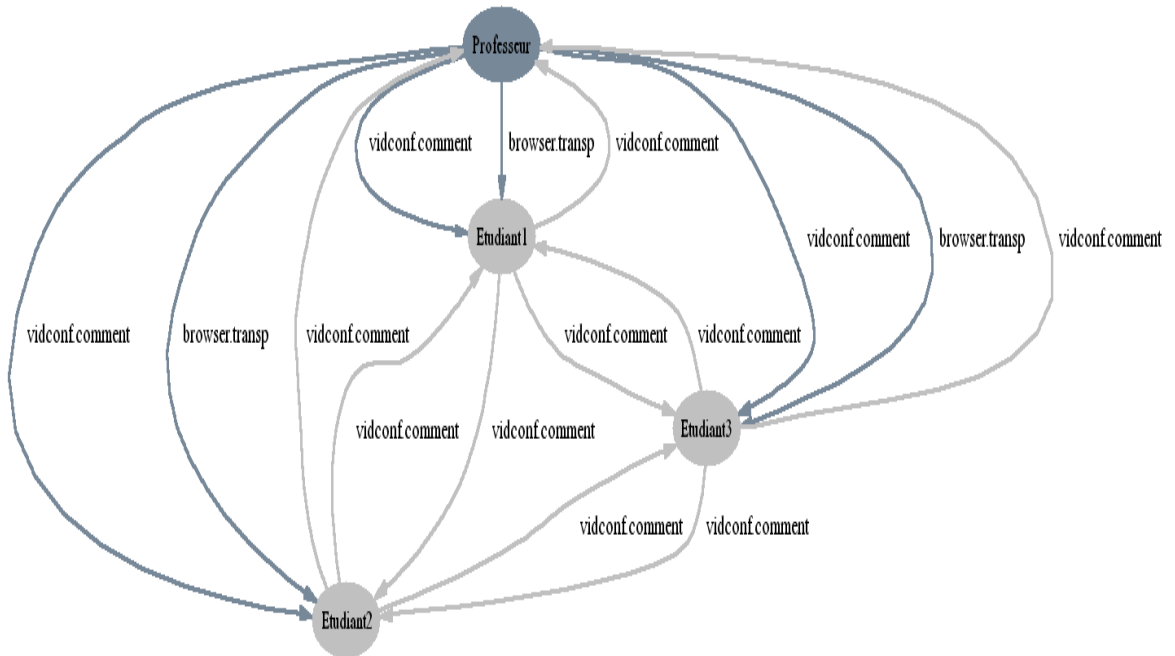


Figure 2.4. Entrée de l'étudiant 2 dans la session de téléformation à l'instant t_1

La table 2.7 montre les modifications dans les relations et les attributs qui interviennent à t_1 . Les parties en gras représentent les relations et les attributs modifiés dans le graphe.

Informations courantes		
$G_{t_0} = \langle U_{t_0}, R_{t_0} \rangle$ $U_{t_0} = \{p, e_1, e_2, e_3\}$ $R_{t_0} = \{r_1(p, e_1), r_2(p, e_1),$ $r_3(p, e_3), r_4(p, e_3),$ $r_5(e_1, p), r_6(e_1, e_3),$ $r_7(e_3, p), r_8(e_3, e_1),$ $r_9(p, e_2), r_{10}(p, e_2),$ $r_{11}(e_1, e_2), r_{12}(e_3, e_2),$ $r_{13}(e_2, p), r_{14}(e_2, e_1),$ $r_{15}(e_2, e_3)\}$	$att_util(p) = \langle Professeur, professeur \rangle$ $att_util(e_1) = \langle Etudiant_1, etudiant \rangle$ $att_util(e_3) = \langle Etudiant_3, etudiant \rangle$ $att_util(e_2) = \langle Etudiant_2, etudiant \rangle$	$att_rel(r_1) = \langle vidconf, comment \rangle$ $att_rel(r_2) = \langle browser, transp \rangle$ $att_rel(r_3) = \langle vidconf, comment \rangle$ $att_rel(r_4) = \langle browser, transp \rangle$ $att_rel(r_5) = \langle vidconf, comment \rangle$ $att_rel(r_6) = \langle vidconf, comment \rangle$ $att_rel(r_7) = \langle vidconf, comment \rangle$ $att_rel(r_8) = \langle vidconf, comment \rangle$ $att_rel(r_9) = \langle vidconf, comment \rangle$ $att_rel(r_{10}) = \langle browser, transp \rangle$ $att_rel(r_{11}) = \langle vidconf, comment \rangle$ $att_rel(r_{12}) = \langle vidconf, comment \rangle$ $att_rel(r_{13}) = \langle vidconf, comment \rangle$ $att_rel(r_{14}) = \langle vidconf, comment \rangle$ $att_rel(r_{15}) = \langle vidconf, comment \rangle$

Table 2.7. Etat courant de la session de téléformation à l'instant t_1

2.3.3 Rôles

La dynamique de rôle intervient lorsqu'un utilisateur devient actif (ou passif) à l'intérieur du groupe, c'est-à-dire devient producteur (ou consommateur) d'information vis-à-vis de certains outils manipulés en session. Les transmissions d'information de ces utilisateurs au travers de certains outils sont modifiées. L'arrivée d'un événement de demande de changements de rôle utilisateur, puis l'application de la règle de transformation qui lui est associée, entraîne une modification de la structure du diagramme de coordination courant : cela se traduit par des ajouts, suppressions ou modifications de flèches qui représentent les flux de données dans le diagramme de coordination. L'ensemble des outils reste le même. De la même façon, l'ensemble des utilisateurs ne change pas : les nœuds du diagramme de coordination sont préservés.

En reprenant le cas de la session de téléformation, la dynamique sur les rôles peut se produire lorsqu'un étudiant est autorisé à intervenir ; à l'instant t_2 (figure 2.5), le professeur cède la parole à l'étudiant 3 en lui conférant le rôle d'étudiant privilégié. L'étudiant 3 peut alors diffuser les transparents liés à sa question vers le reste du groupe. En ce qui concerne l'utilisation de l'outil vidéoconférence, la même configuration est maintenue.

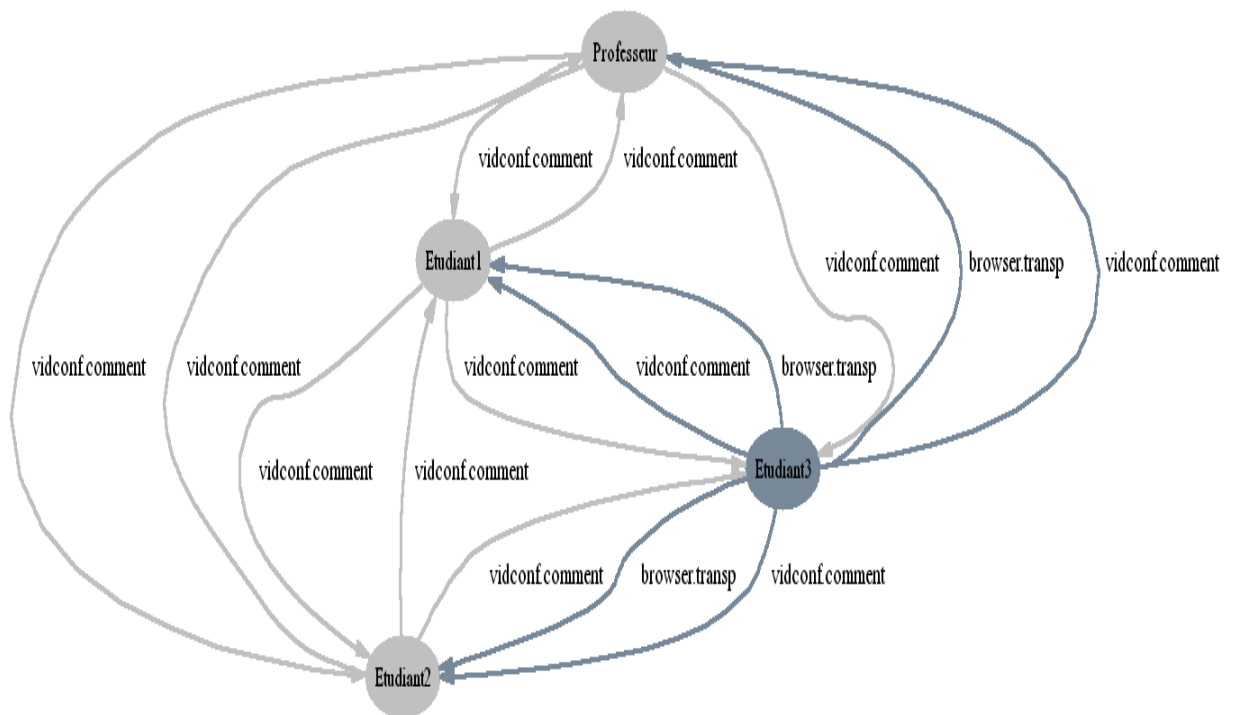


Figure 2.5. Configuration avec un rôle actif de l'étudiant 3 à l'instant t_2

La table 2.8 montre les modifications dans les relations et les attributs qui interviennent à t_2 . Les parties en gras représentent les relations et les attributs modifiés dans le graphe. Le gestionnaire de session fait passer l'outil browser sous le contrôle de l'étudiant 3 qui a un rôle actif sur cet outil.

Informations courantes		
$G_{t_0} = \langle U_{t_0}, R_{t_0} \rangle$ $U_{t_0} = \{p, e_1, e_3\}$ $R_{t_0} = \{r_1(p, e_1), r_2(e_3, e_1),$ $r_3(p, e_3), r_4(e_3, p),$ $r_5(e_1, p), r_6(e_1, e_3),$ $r_7(e_3, p), r_8(e_3, e_1),$ $r_9(p, e_2), r_{10}(e_3, e_2),$ $r_{11}(e_1, e_2), r_{12}(e_3, e_2),$ $r_{13}(e_2, p), r_{14}(e_2, e_1),$ $r_{15}(e_2, e_3)\}$	$att_util(p) = \langle Professeur, professeur \rangle$ $att_util(e_1) = \langle Etudiant_1, etudiant \rangle$ $att_util(e_3) = \langle Etudiant_3, etudiant \rangle$ $att_util(e_2) = \langle Etudiant_2, etudiantprivilegie \rangle$	$att_rel(r_1) = \langle vidconf, comment \rangle$ $att_rel(r_2) = \langle browser, transp \rangle$ $att_rel(r_3) = \langle vidconf, comment \rangle$ $att_rel(r_4) = \langle browser, transp \rangle$ $att_rel(r_5) = \langle vidconf, comment \rangle$ $att_rel(r_6) = \langle vidconf, comment \rangle$ $att_rel(r_7) = \langle vidconf, comment \rangle$ $att_rel(r_8) = \langle vidconf, comment \rangle$ $att_rel(r_9) = \langle vidconf, comment \rangle$ $att_rel(r_{10}) = \langle browser, transp \rangle$ $att_rel(r_{11}) = \langle vidconf, comment \rangle$ $att_rel(r_{12}) = \langle vidconf, comment \rangle$ $att_rel(r_{13}) = \langle vidconf, comment \rangle$ $att_rel(r_{14}) = \langle vidconf, comment \rangle$ $att_rel(r_{15}) = \langle vidconf, comment \rangle$

Table 2.8. Etat courant de la session de téléformation à l'instant t_2

2.3.4 Outils

Plusieurs événements sont liés aux outils. Les actions qui sont associées à ces événements entraînent la modification de l'état des outils et provoquent un changement dans leur configuration distribuée. Ainsi, la dynamique sur les outils se produit :

- Lors de l'arrivée d'un événement de démarrage/arrêt d'un outil. L'outil est activé/désactivé durant l'exécution de la session synchrone. Au niveau du modèle, la règle de transformation appliquée ajoute/efface tous les flux de données étiquetés nécessaires pour modéliser et piloter cet outil du graphe de la session actuelle.
- Lors de l'arrivée d'un événement de modifications de l'accès aux données des utilisateurs. Les actions associées changent la configuration des flux de donnée échangés au travers d'un outil. Au niveau du modèle, la règle de transformation appliquée crée/efface/met à jour les flèches et les étiquettes des flèches qui sont en relation avec les flux de données concernés.
- Lors l'arrivée d'un événement de demande de modification d'une configuration d'outil. Les actions associées sont des commandes de l'outil qui, par contre-coup, modifient certains flux de données. Au niveau du modèle, la règle de transformation appliquée modifie les étiquettes des flèches qui sont en relation avec les flux de données concernés.

En reprenant le même exemple, supposons qu'un problème de transmission audio se produit sur l'outil vidéoconférence à l'instant t_3 . Le groupe décide de changer d'outil. Il utilise à la place un outil de dialogue textuel (chat) pour maintenir la communication entre eux (figure 2.6). Ce changement provoque de la part du gestionnaire de session un arrêt de l'outil de vidéoconférence et un démarrage de l'outil chat. Au niveau du modèle, ceci a comme conséquence de modifier les étiquettes des flèches du diagramme de coordination, les

relations de communication entre les membres du groupe (flèches) sont maintenues dans la même configuration.

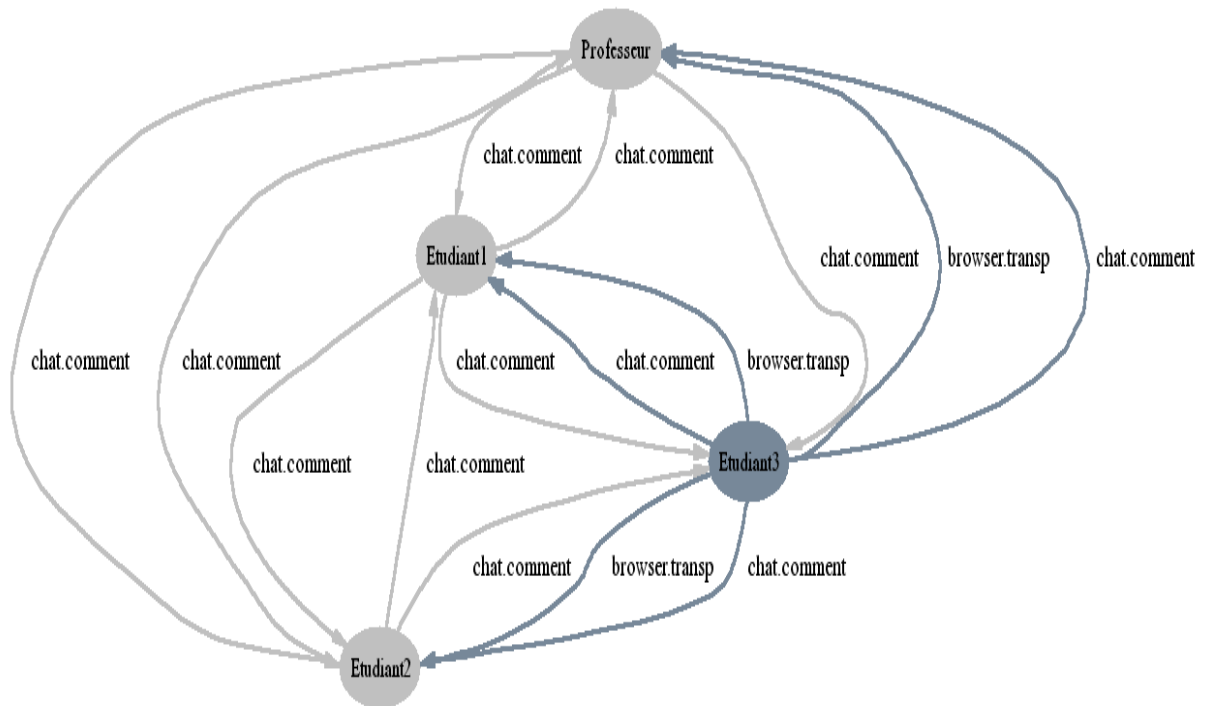


Figure 2.6. Dynamique avec changement d'outil à l'instant t3

Informations courantes		
$G_{t_0} = \langle U_{t_0}, R_{t_0} \rangle$ $U_{t_0} = \{p, e_1, e_3\}$ $R_{t_0} = \{r_1(p, e_1), r_2(e_3, e_1),$ $r_3(p, e_3), r_4(e_3, p),$ $r_5(e_1, p), r_6(e_1, e_3),$ $r_7(e_3, p), r_8(e_3, e_1),$ $r_9(p, e_2), r_{10}(e_3, e_2),$ $r_{11}(e_1, e_2), r_{12}(e_3, e_2),$ $r_{13}(e_2, p), r_{14}(e_2, e_1),$ $r_{15}(e_2, e_3)\}$	$att_util(p) = \langle Professeur, professeur \rangle$ $att_util(e_1) = \langle Etudiant_1, etudiant \rangle$ $att_util(e_3) = \langle Etudiant_3, etudiant \rangle$ $att_util(e_2) = \langle Etudiant_2, etudiantprivilegie \rangle$	$att_rel(r_1) = \langle chat, comment \rangle$ $att_rel(r_2) = \langle browser, transp \rangle$ $att_rel(r_3) = \langle chat, comment \rangle$ $att_rel(r_4) = \langle browser, transp \rangle$ $att_rel(r_5) = \langle chat, comment \rangle$ $att_rel(r_6) = \langle chat, comment \rangle$ $att_rel(r_7) = \langle chat, comment \rangle$ $att_rel(r_8) = \langle chat, comment \rangle$ $att_rel(r_9) = \langle chat, comment \rangle$ $att_rel(r_{10}) = \langle browser, transp \rangle$ $att_rel(r_{11}) = \langle chat, comment \rangle$ $att_rel(r_{12}) = \langle chat, comment \rangle$ $att_rel(r_{13}) = \langle chat, comment \rangle$ $att_rel(r_{14}) = \langle chat, comment \rangle$ $att_rel(r_{15}) = \langle chat, comment \rangle$

Table 2.9. Etat courant de la session de téléformation à l'instant t3

La table 2.9 montre les modifications dans les relations et les attributs qui interviennent à t3. Les parties en gras représentent les relations et les attributs modifiés dans le graphe. Dans notre cas, lors du changement de l'outil vidéoconférence en chat, aucune relation n'est modifiée : seuls les attributs associés à la vidéoconférence le sont.

2.4 Avantages du modèle

D'un point de vue ingénierie, trois avantages principaux peuvent être extraits de cette approche.

Le premier est de donner une représentation à haut niveau des éléments de sessions synchrones générales dont la structure et l'évolution dans le temps seront préprogrammées par un concepteur de session. A cause de sa facilité de compréhension, et de sa structure intuitive de graphes étiquetés, ce modèle est un bon candidat pour une visualisation synthétique et à haut niveau de l'état de la session. Les diagrammes de coordination contribuent ainsi à améliorer la prise en compte de la conscience de groupe.

Plusieurs niveaux de modélisation différents de granularité variable (fine, moyenne ou grossière), au travers desquels les diagrammes sont vus ou manipulés, peuvent être définis. Les flux de données utilisés par les outils sont représentés par des flèches étiquetées. Suivant le niveau de granularité défini par le concepteur de session, une flèche étiquetée peut correspondre à un ou plusieurs flux de données véritables. Par exemple, dans le cas d'une modélisation à grain fin d'un outil de vidéoconférence, deux étiquettes *{conf.audio, conf.video}* sont associées aux deux flux audio et vidéo manipulés par la vidéoconférence. Dans le cas d'une modélisation à grain moyen, une étiquette *{conf.mm-data}* est associée pour les deux flux audio et vidéo qui se trouvent groupés par paire et manipulés ensemble sans aucune distinction. Dans le cas d'une modélisation à grain grossier, tous les flux de données d'un outil sont manipulés sans aucune distinction, lorsque l'outil considéré est démarré. Ce niveau de modélisation adaptable dépend du contrôle de granularité défini par le concepteur de session ou requis par les outils. Il peut être partiellement raffiné et donne une vue adaptable du système.

Le dernier avantage est d'obtenir un modèle programmable qui peut être interprété, compris et pris en compte par une couche logicielle de gestion de session. Cette approche évite une gestion en ligne manuelle de la part des utilisateurs présents en session. Elle crée ainsi une couche de session paramétrable au travers d'un modèle qui est utilisé pour commander et contrôler les outils de la plate-forme.

D'un point de vue plus formel, notre approche possède l'avantage suivant :

Le fait d'utiliser le formalisme mathématique basé sur la théorie des graphes permet de définir des propriétés sur les graphes manipulés.

On peut trouver des propriétés générales aux sessions, comme par exemple, le fait que la configuration courante doit toujours être connexe.

On peut aussi exprimer des propriétés spécifiques, en relation directe avec le travail collaboratif à réaliser. Ces propriétés sont plus liées aux attributs du graphe, par exemple : 1) *la présence obligatoire d'un membre d'un certain rôle*, 2) *le type de contrôle (actif/passif), qu'un utilisateur peut avoir sur un outil*. Elles interviennent pour contrôler et vérifier que les diagrammes de coordination obtenus font bien partie de l'ensemble G^* des configurations valides retenues. Elles peuvent aussi vérifier que les règles de transformation appliquées maintiennent toujours une configuration valide.

Ces ensembles prédéfinis de propriétés servent à vérifier automatiquement la validité des structures de session au cours de leur définition ; elles aident de ce fait le concepteur de la session.

2.5 Positionnement

Le tableau 2.10 résume le positionnement de notre modèle vis-à-vis des autres modèles de gestion de session. Le formalisme proposé s'appuie sur des graphes étiquetés dynamiques pour représenter les principaux éléments d'une session avec leurs liens. Les sessions sont définies de façon explicite, et le droit de parole est associé à la dynamique des rôles. La réalisation concrète du modèle a pour principal objectif de supporter une approche de gestion de session paramétrable par modèle.

	Formalisme mathématique	Éléments modélisés	Type de session	Contrôle de droit de parole	Utilisation du modèle	Etat d'avancement
Intermezzo	Ensembles	Activité qui associe un ensemble d'utilisateurs, de tâches et d'objets	Implicite Explicite	Politiques de contrôle associées aux rôles	Création flexible de sessions	Concret : système Intermezzo
CoAct	Logique du premier ordre	Activité	Implicite	Absent	Gestion et exécution consistante d'activités	Abstrait
Espaces partagés	Ensembles	Activité	Implicite	Absent	Algorithme de gestion de session	Abstrait
Coordination de groupe	Graphes	Sessions comprenant des utilisateurs, des hôtes, des ressources	Explicite	Politiques de contrôle associées aux ressources	Définition de sessions Gestion de droits d'accès à leurs ressources	Abstrait
Diagrammes de Coordination Dynamiques	Graphes	Sessions comprenant des utilisateurs, des outils et des flux de données	Explicite	Politiques de contrôle associées aux rôles	Création de sessions Utilisation par des services gestionnaires de session	Concret : prototype réalisé

Table 2.10. Caractéristiques du modèle de session

3. Contributions pour des services de communication orientés modèle

En accord avec le canevas des résultats obtenus, cette partie décrit le lien entre l'instance *Diagrammes de Coordination* et l'instance *Gestion de Sessions*. L'implantation du modèle de session et son utilisation pour visualiser l'état courant de la session, qui sont présentés, ont contribué à la réalisation de services de communication orientés modèle qui gèrent des sessions. Les caractéristiques conceptuelles et architecturales de ces gestionnaires seront présentées dans le chapitre suivant.

3.1 Implantation des diagrammes de coordination

Dans le contexte de notre travail, nous avons besoin de représenter le modèle de session dans un format standard qui peut être adapté, validé, transmis et visualisé dans des environnements différents [RODR01]. Toutes ces fonctionnalités sont supportées par XML [LIEC98], [MARS00], [XML04] qui implante, pour notre modèle l'état de la session et l'évolution de la session en ligne. A l'opposé du projet DISCIPLE [MARS00] qui emploie XML pour représenter les données "classiques", nous utilisons XML pour coder les informations de contrôle relatives à la session (participants, état de la session, outils, etc.).

Nous avons représenté la structure des graphes étiquetés, décrits au travers de liens hiérarchiques et associatifs (figure 2.7). Pour cela, une DTD (Document Type Definition) appropriée a été définie [RODR01]. Elle définit un canevas générique utile pour instancier et créer des sessions bien formées. Elle permet de vérifier la validité du document XML après toute modification due à un changement de l'état de la session.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ENTITY % entitiesdtd SYSTEM 'Entities.dtd'>
%entitiesdtd;
<!ELEMENT COORD_DIAG (Conference, NODES, LINKS)>
<!ATTLIST COORD_DIAG
  COORD_DIAG_ID CDATA #REQUIRED
  Session_ID CDATA #REQUIRED
  State(CREATED|DELETED|OPENED|CLOSED) #REQUIRED
  %Domain;
  %NotifDateTime; >

<!ELEMENT NODES (NODE*)>
<!ELEMENT NODE (User)>
<!ATTLIST NODE
  NODE_ID ID #REQUIRED
  LABEL CDATA #IMPLIED>

<!ELEMENT LINKS (LINK*)>
<!ELEMENT LINK (DATAFLOW+)>
<!ATTLIST LINK
  SOURCE IDREF #REQUIRED
  DEST IDREF #REQUIRED
  LABEL CDATA #IMPLIED>

<!ELEMENT DATAFLOW EMPTY>
<!ATTLIST DATAFLOW
  Tool_id CDATA #REQUIRED
  Flow_id CDATA #REQUIRED >
```

Figure 2.7. DTD des diagrammes de coordination

L'aspect dynamique du modèle a été implémenté au moyen de trois modules basés sur les extensions JAVA qui manipulent les documents XML, modules dont l'effet est résumé par la figure 2.8 :

- *Crd_analyser* : Ce paquetage contient l'analyseur de diagrammes de coordination. Il a été implémenté en utilisant Xerces Java Parser 1.4.4 [APA04].

- *Crd_engine* : Ce paquetage contient le moteur des diagrammes de coordination. Il est composé des fonctions élémentaires pour la génération et la manipulation des diagrammes de coordination. L'ensemble des fonctions de transformation élémentaires opère respectivement au niveau d'un nœud, d'une relation ou d'un attribut.
- *Transformation* : Ce paquetage contient l'ensemble des règles de transformation qui sont activées lors de l'arrivée d'un événement et qui, en réponse, réalisent les changements sur les diagrammes de coordination. Ces règles sont composées de fonctions élémentaires et sont implantées par programme.

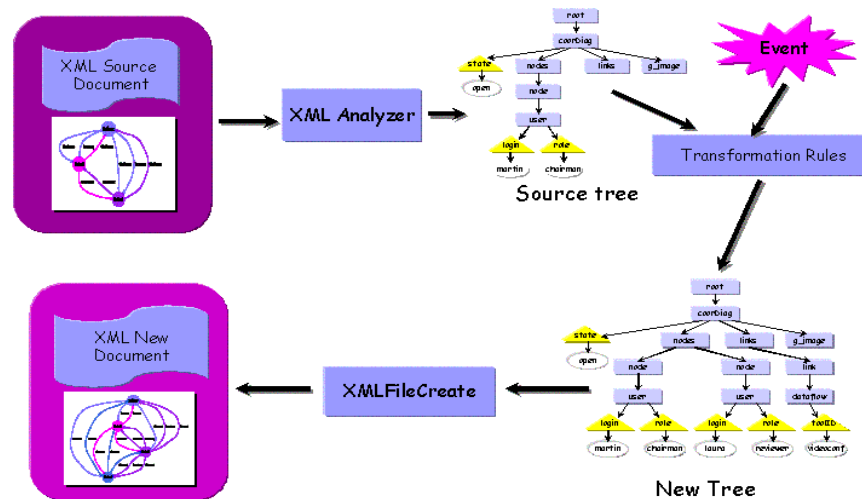


Figure 2.8 Dynamique du modèle XML

Avec XSL (Extensible Stylesheet Language) [XSL04], il est possible de visualiser les documents XML courants manipulés par les composants de session. Par conséquent, XSL peut être utilisé par le programmeur de la session hors ligne pour visualiser l'état de la session. Grâce à XSL, une interface Web générique peut être intégrée comme visualisateur de session en ligne dans le composant gestionnaire de session sans qu'il ne soit nécessaire de développer de logiciel spécifique pour visualiser l'état courant de la session. La vue globale de la session peut ainsi être fournie à chaque participant pour qu'il soit informé des changements intervenus dans la session. Ainsi, les utilisateurs synchronisent plus facilement leur travail.

Les choix de XML et de XSL facilitent le maintien de la conscience de l'état courant de la session pour tous les utilisateurs connectés.

L'ensemble de cette réalisation a été mené dans le cadre de la thèse de Mme Rodriguez-Peralta [RODR03] et valorisé dans le projet IST DSE qui portait sur l'ingénierie distribuée [DRIR01].

3.2 Visualisateur de session

Le visualisateur a pour objectif de montrer l'état d'une session en ligne. Le but recherché est d'offrir un applicatif le plus générique possible et exécutable au travers d'une interface standard. De ce fait, l'utilisation d'un navigateur Web nous a semblé le plus adapté à nos besoins.

Le visualisateur (figure 2.9) offre une représentation simple et graphique de l'état de la session. Le premier tableau montre toute l'information qui appartient aux participants actifs dans la session, tels comme leur nom, login, adresse email, etc. Le deuxième tableau donne les caractéristiques des outils partagés, avec les flux de données qu'ils manipulent. La représentation graphique du diagramme de coordination apparaît en dessous.

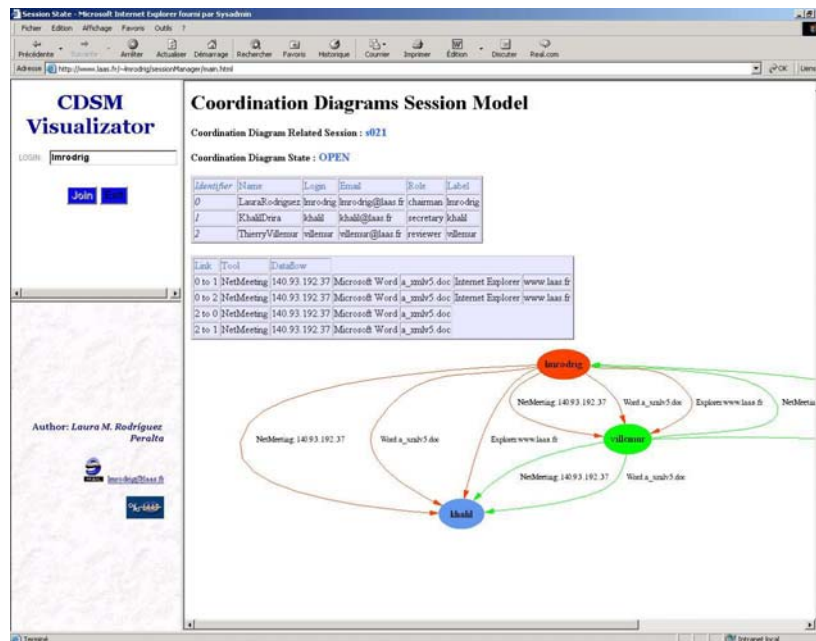


Figure 2.9. Interface du visualisateur

Graphviz est un ensemble d'outils de dessin de graphes disponibles en Unix et MS-Windows (win32) [KOUT02], qui comprend aussi une interface de service Web (webdot). Il s'occupe de la visualisation d'information structurée en partant de représentations géométriques sous forme de graphes et de réseaux abstraits. Graphviz est utilisé pour créer l'image GIF du diagramme de coordination courant.

L'affichage des informations d'état de la session se fait par une applet JAVA, suivant une feuille de style XSL spécifique à chaque utilisateur. Lors de toute modification, cette applet est réactivée automatiquement, de façon à ce que l'affichage soit toujours en correspondance avec l'état courant de la session.

Le visualisateur a été réalisé dans le cadre de la thèse de Mme Rodriguez-Peralta [RODR03] et utilisé pour un scénario de revue de projet, dans le projet IST DSE.

4. Contributions à la modélisation de scénarios pour la téléformation

En accord avec le canevas des résultats obtenus, cette partie décrit le lien entre l'instance *Diagrammes de Coordination* et l'instance *Téléenseignement/Téléingénierie*. Le modèle a été utilisé pour différents scénarios de téléformation dans le cadre de deux projets contractuels.

4.1 Projet TOPASE – Formation de pilotes et d’agents de maintenance

La formation des pilotes et des agents de maintenance aéronautique se déroule traditionnellement de façon centralisée dans des salles de cours équipées d’ordinateurs multimédias connectées en réseau local. Cette formation, qui intervient périodiquement au cours de la carrière ou bien lors de la sortie de nouvelles gammes et de nouvelles versions d’appareils, se déroule de façon intensive sur une semaine. Un des buts du projet TOPASE [VILL97], [VILL99], appel d’offre national Autoroutes de l’Information, a été de distribuer cette formation et de rendre son accès plus flexible. La formation interviendrait ainsi de façon plus étalée dans le temps et se ferait en réseau, au travers d’outils collecticiels multi-utilisateurs, depuis certains lieux privilégiés comme les aéroports ou bien depuis les compagnies clientes.

La formation traditionnelle des pilotes et des agents de maintenance se fait sur une période de quatre semaines. Chaque journée de formation se décompose en trois phases : le briefing, l’auto-apprentissage et le débriefing.

La phase de briefing est similaire à un cours magistral. Durant une vingtaine de minutes, l’instructeur fait une présentation sur le déroulement de la séance et donne quelques explications. Il utilise soit un support manuscrit, soit des transparents qu’il diffuse. Il peut aussi montrer des copies d’écran de l’application de formation que les pilotes et les agents de maintenance vont manipuler.

La phase d’auto-apprentissage est proche d’une séance classique de Travaux Dirigés/Travaux Pratiques. Sa durée s’étale entre une demi-journée et une journée. Chaque élève parcourt à son rythme le contenu des chapitres de l’application de formation stockée sur son poste de travail. Parfois, les élèves pilote consultent la documentation en ligne pour avoir plus de précision sur certains points. Il leur arrive aussi de poser des questions à l’instructeur.

La phase de débriefing est une discussion générale entre l’instructeur et l’ensemble des élèves, sur la séance de formation qui vient de se dérouler. L’instructeur peut revenir sur certaines parties délicates ou sur certains points qui nécessitent un approfondissement. Cette phase dure une vingtaine de minutes environ.

Nous avons utilisé dans ce contexte le modèle de Diagrammes de Coordination Dynamiques pour analyser et représenter tous les cas nécessaires lors de la formation. Deux rôles ont été identifiés : le rôle instructeur et le rôle élève. Chacune des phases nécessaires (briefing, auto-apprentissage, débriefing) a été découpée en situations particulières. Pour chacune de ces situations, nous avons analysé les fonctionnalités requises pour réaliser cette situation de façon distribuée. Les flux de données nécessaires à chaque fonctionnalité ont été ainsi modélisés.

La table 2.11 résume notre contribution pour ce scénario de téléformation. Elle donne en ordonnée les situations particulières à l’intérieur de chaque phase et liste en abscisse les quatre fonctionnalités identifiées. Les croix indiquent quelle fonctionnalité est nécessaire pour supporter quelle situation et chacune d’entre elle a été modélisé par un diagramme de coordination. Nous renvoyons le lecteur à [VILL97], [VILL99b] pour plus de détail concernant les structures des graphes.

	Visualisation de la classe	Partage d'applications	Partage de documents fixes	Diffusion de documents vidéo-audio
Briefing				
Cours de l'instructeur	X	X	X	X
Questions durant le briefing	X	X	X	
Auto-apprentissage				
Suivi de la classe	X	X		
Aide individuelle	X	X	X	X
Aide généralisée	X	X	X	X
Debriefing				
Discussion	X	X	X	X

Table 2.11. *Fonctionnalités pour les scénarios de téléformation de pilotes*

Les graphes ont ensuite été utilisés dans le fonctionnement du gestionnaire de session du projet Topase [VILL99].

4.2 Projet Lab@Future – Laboratoire scolaire distribué

Le projet Lab@Future [BAUD03], [BAUD04a], [BAUD04b], [BAUD04c], [COUR04], soutenu par le programme IST de la CEE, a défini une plate-forme générique et universelle de réalité mixte et augmentée, qui supporte des interactions distantes au travers de réseaux, et qui s'applique aux théories du constructivisme et de l'activité pour du téléenseignement. Ces théories, au travers de canevas tels que la résolution de problèmes en temps réel, l'apprentissage collaboratif exploratoire ou interdisciplinaire, voient l'apprentissage comme un processus de construction fortement interactif au travers d'échanges de connaissances très diversifiées. Elles doivent de ce fait non seulement supporter des canevas d'enseignement classiques (dans lequel le professeur a un rôle central), mais aussi d'autres canevas plus novateurs, plus proches des théories (dans lequel par exemple les élèves s'auto-organisent sans l'aide de professeur).

Du point de vue technique, cette plate-forme doit notamment autoriser l'accès à des expérimentations avancées, mises à la disposition des écoles au travers de réseaux informatiques. Elle sera utilisée pour des expériences d'apprentissage en laboratoire dans les domaines de la mécanique des fluides, de la géométrie, de l'environnement et des sciences humaines, et sera déployée dans huit pays européens.

4.2.1 Rôles des utilisateurs

Quatre rôles ont été définis pour ces scénarios de téléenseignement. Ce sont respectivement les rôles : *professeur*, *élève*, *observateur* et *expert*. Chaque rôle détermine les droits qu'un utilisateur aura pour utiliser les outils de communication et de collaboration définis dans le système.

Les rôles de professeur et d'élève sont auto-explicatifs.

Le rôle d'expert implante une forme d'aide. Il est détenu par un utilisateur en charge des problèmes et des questions qui concernent l'utilisation de la plate-forme de collaboration et/ou des applications spécifiques aux expérimentations. Son aide est volontairement restreinte à ces deux types d'aides. D'autres problèmes qui pourraient survenir des

configurations réseau ou de l'administration système sont en dehors de ses compétences et seront supportées par des procédures standard (c.a.d. non prises en compte ni contrôlées par la plate-forme).

Le rôle d'observateur représente principalement un mode dégradé du rôle étudiant. En effet, typiquement, un observateur aura des accès plus restreints qu'un élève vis-à-vis des outils de la plate-forme. Ceci peut être motivé par deux raisons principales :

- Des raisons en liaison avec la politique de gestion de la plate-forme, politique impliquant que certains utilisateurs ne sont pas autorisés à utiliser toutes les fonctionnalités de la plate-forme.
- Des contraintes de qualité de service réseau qui impliquent que certains utilisateurs distants ne sont pas capables pour des raisons techniques d'utiliser toutes les fonctionnalités de la plate-forme.

L'association entre les rôles et les droits de communication et de collaboration est totalement donnée par l'administrateur du système lorsqu'il définit la session. Cette association, flexible, peut être modifiée pendant l'existence de la session.

4.2.2 Fonctionnalités de communication et de collaboration

Quatre principales fonctionnalités de communication et de collaboration ont été définies comme nécessaires pour supporter les scénarios de téléenseignement de Lab@Future.

- Communications informelles, au travers d'un outil d'audio/vidéoconférence multipoint et d'un outil de dialogue textuel (chat)
- Partage de document, au travers d'un tableau blanc multi-utilisateur
- Partage d'application, au travers d'un outil spécifique
- Partage de scènes 3D virtuelles, au travers d'un serveur 3D multi-utilisateur

4.2.3 Contrôle des fonctionnalités de communication et de collaboration

Pendant la session, chaque fonctionnalité de communication est sous le contrôle d'un (ou de plusieurs) participant(s). Lorsque le participant "Student #1" contrôle la fonctionnalité de partage de documents, il est le producteur des données utilisées pour le partage de documents et diffusées vers les autres participants. Les relations du sous-graphe de collaboration, limité aux données de partage de document, partent du nœud "Student #1" et aboutissent vers les autres participants.

Cependant, l'attribution de ce contrôle est sous la responsabilité d'un participant particulier. Par exemple, le participant "Teacher" a la responsabilité d'attribuer ou de retirer le contrôle du partage de documents au participant "Student #1".

Pour représenter ces possibilités de contrôle des fonctionnalités et d'attribution du contrôle de ces fonctionnalités, on utilise à l'intérieur du diagramme de collaboration un flux de contrôle particulier : cette relation entre deux participants est étiquetée par "Control". Le participant à la source de la relation attribue le contrôle. Le participant destination reçoit le contrôle de la fonctionnalité et de ce fait produit les données associées à la fonctionnalité. Dans notre exemple, on aura une flèche "Control" du participant "Teacher" vers "Student #1".

4.2.4 Flexibilité de l'approche

Les rôles sont détachés des personnes. Une personne peut se connecter avec un rôle, puis par la suite, choisir un autre rôle, selon sa connaissance des mots de passe associés. La seule restriction est qu'une personne ne peut avoir qu'un seul rôle à la fois.

Il n'y a pas à priori de restriction sur le nombre de professeurs, élèves, experts et observateurs qui peuvent être connectés en même temps, la seule restriction étant explicitement donnée au niveau de la définition de la session. Les rôles sont définis une fois par le créateur de la session et ne pourront pas être modifiés durant l'existence de la session. Par contre, l'association entre les rôles utilisateurs et les droits de communication et de collaboration peut être changée par le créateur de la session, au cours de l'existence de la session.

Ces différents niveaux de flexibilité ont été garantis pour être complètement consistants avec des scénarios de téléenseignements novateurs et avancés, scénarios dérivés des cadres pédagogiques provenant des théories de l'activité et du constructivisme.

4.2.5 Scénarios collaboratifs

Le modèle de Diagrammes de Coordination Dynamiques a été utilisé comme base commune pour décrire et analyser les scénarios pédagogiques de Lab@Future. Pour cela, nous sommes partis de quatre canevas de scénarios, plus proches de scénarios de téléenseignement classiques, qui peuvent être réalisés par la plate-forme Lab@Future.

Les quatre scénarios collaboratifs sont les suivants :

- Une session de cours magistral
- Une session de cours modérée
- Un scénario d'expérimentation technique
- Une session de support technique

Pour améliorer la lisibilité du modèle, les nœuds du graphe ont été colorés selon les rôles qui leur sont accordés. Nous avons respecté les conventions suivantes :

- Le rôle professeur est associé à la couleur rouge
- Le rôle expert est représenté en orange
- Le rôle étudiant est défini en vert
- Le rôle observateur est représenté par la couleur jaune

Pour chaque scénario, pour des raisons de clarté, le graphe global de collaboration a été décomposé en plusieurs sous-graphes. Deux sous-graphes sont nécessaires par fonctionnalité de communication : Le premier sous-graphe de contrôle d'accès montre qui attribue et qui détient une fonctionnalité. Le deuxième sous-graphe correspond aux flux de données de cette fonctionnalité.

a) Session de cours magistral

Ce scénario (figure 2.10) correspond à un cours classique où un professeur donne un cours à un groupe d'étudiants. Il se compose d'un professeur, de plusieurs étudiants (au moins un), de quelques observateurs possibles, et d'un expert. Le professeur joue un rôle central dans la session : les flux d'information sont essentiellement unidirectionnels depuis le professeur vers les élèves, et le professeur possède tous les droits d'accès sur les outils de communication génériques.

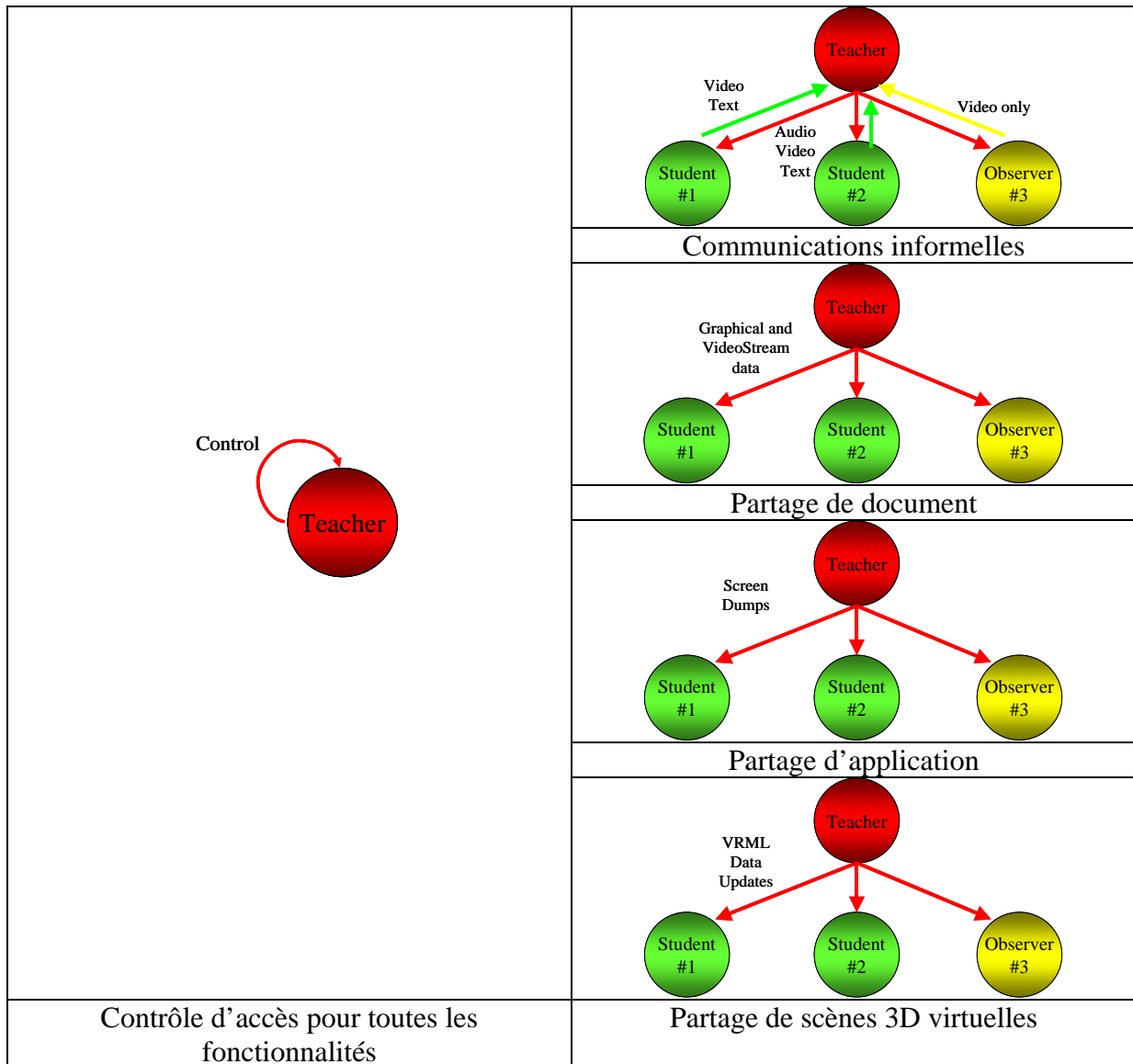


Figure 2.10. Modèle de collaboration pour un scénario de cours magistral

Dans ce scénario, le professeur agit comme le maître et il possède un contrôle complet de la session sans rien déléguer à aucun autre membre du groupe. Le contrôle d'accès est donc représenté par le seul nœud du professeur.

Les communications informelles sont essentiellement unidirectionnelles et la plupart des flux de données (c.a.d. vidéo, audio et texte) sont émis depuis le professeur vers les étudiants, les observateurs et l'expert s'il est présent. Afin de renforcer la conscience de groupe, les flux de donnée vidéo en provenance des étudiants, des observateurs et de l'expert sont renvoyés vers le professeur. Les étudiants, les observateurs et l'expert ne peuvent pas communiquer directement au travers des outils de la plate-forme. Leur conscience de groupe est volontairement restreinte au professeur, et ils ne se voient pas entre eux.

Le partage de document est sous la responsabilité du professeur. Il charge son document dans le tableau blanc et il propose ses annotations.

Le partage d'application est aussi sous l'entière responsabilité du professeur. Ce dernier sélectionne l'application à partager et il est la seule personne autorisée à interagir avec l'application.

Le partage de scènes 3D virtuelles est sous le contrôle du professeur. Ce dernier sélectionne la scène à partager. Chaque client peut librement naviguer dans le monde 3D, mais ne peut pas réaliser de modification. Seules les actions du professeur sont transmises à l'ensemble des membres.

L'expert n'apparaît pas explicitement dans le scénario de la figure 2.10 car il se comporte comme un élève lorsqu'il se joint à la session.

b) Session de cours modérée

Ce scénario se compose d'un professeur, d'un ou de plusieurs étudiants, de plusieurs observateurs possibles, et d'un expert. Il correspond à une situation de téléenseignement où le professeur délègue à un étudiant le contrôle d'une certaine fonction de communication. Par conséquent, les flux d'information ne sont plus unidirectionnels. Le professeur garde cependant le droit d'attribution des contrôles dans la session. Ce scénario correspond à une situation de question/réponses dans un cours.

Le professeur agit comme modérateur. Il gère l'attribution des contrôles parmi les membres actifs de la session, c.a.d. les étudiants. Ni les observateurs, ni l'expert ne peuvent acquérir de contrôle dans ce type de scénario.

Un mécanisme de demande de contrôle permet à un utilisateur d'interagir avec le professeur pour lui demander un contrôle particulier. Sans ce mécanisme, l'interaction avec le professeur se fera au travers des fonctionnalités de communications informelles.

Dans ce scénario, toutes les communications informelles sont propagées à tout le groupe : chacun peut entendre la discussion courante. Cette hypothèse nous semble réaliste car tous les membres du groupe travaillent sur le même sujet. L'information pertinente pour un étudiant intéressera l'ensemble du groupe. De ce fait, le contrôle d'accès de cette fonctionnalité de communication est inutile.

Dans le graphe de collaboration de la figure 2.11, le professeur transmet le contrôle pour le partage de documents à l'étudiant #1. Par conséquent, l'étudiant #1 est autorisé à

- charger un document privé (stocké sur son disque local) dans l'outil de tableau blanc ;
- annoter tout document précédemment chargé dans le tableau blanc (par lui-même, par un autre étudiant, ou par le professeur).

Le contrôle pour le partage d'application a été donné à l'étudiant #2. Par conséquent, l'étudiant #2 est autorisé à contrôler à distance l'application partagée.

Le professeur garde le contrôle du partage de scènes 3D virtuelles. Il sélectionne la scène virtuelle qui est partagée et seules ses actions sont transmises à l'ensemble des participants.

L'expert n'apparaît pas non plus explicitement dans le scénario de la figure 2.11 car il se comporte comme un élève lorsqu'il se joint à la session.

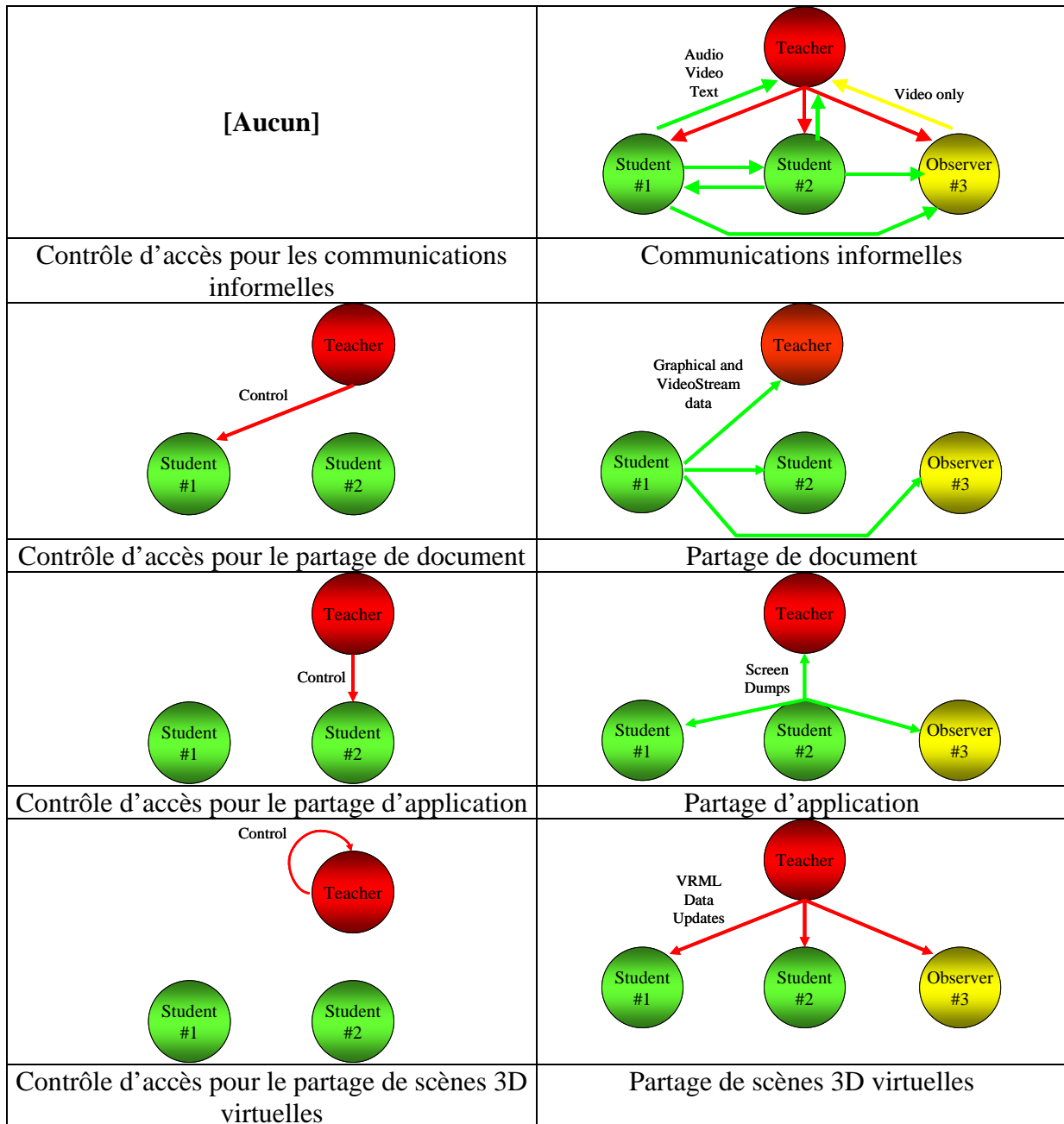


Figure 2.11. Modèle de collaboration pour une session de cours modérée

c) Scénario d'expérimentation technique

Il s'agit d'une session libre (figure 2.12), composée d'étudiants, d'observateurs et d'un expert. Le système laisse un libre contrôle des accès aux membres connectés. Cette situation peut correspondre à une étude faite en groupe. Dans les salles d'étude, les étudiants sont autonomes et ils travaillent sans aucune restriction dans l'attribution des contrôles pour l'accès aux fonctionnalités.

Par conséquent, les flux de données sont sous la responsabilité directe des membres du groupe qui sont connectés. Dans le scénario de la figure 2.12, l'expert contrôle le partage de documents, l'étudiant #1 contrôle le partage d'applications et l'étudiant #3 contrôle le partage d'une scène 3D virtuelle.

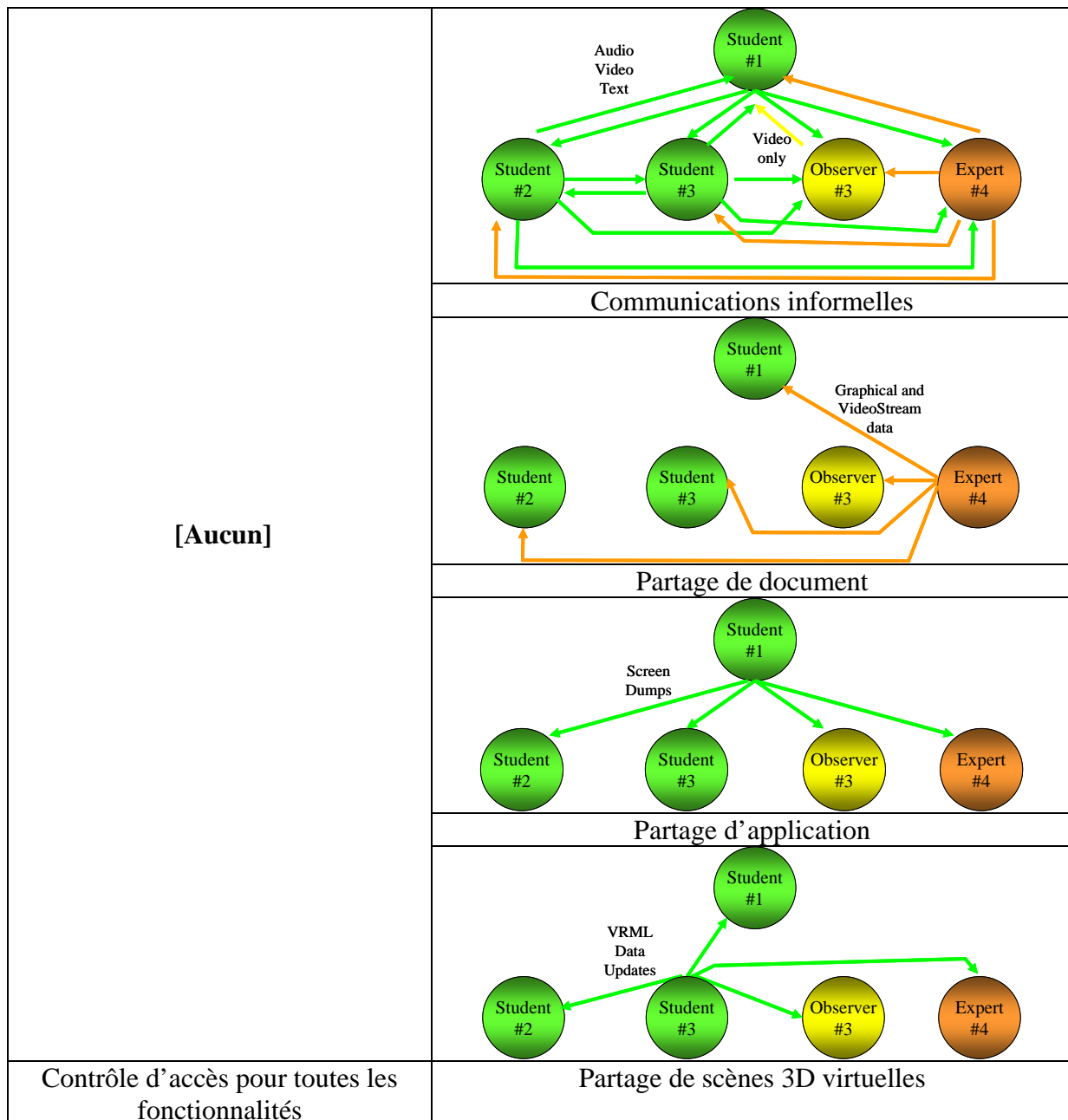


Figure 2.12. *Modèle de collaboration pour un scénario d'expérimentation technique*

d) Session de support technique

La session de support technique n'a pas une fonction pédagogique aussi prononcée que les trois scénarios précédents. Elle correspond à une situation d'aide, en cas de difficulté.

Cette session se compose d'un expert, d'un professeur et d'un ou de plusieurs autres membres du groupe (étudiant, observateur). Elle correspond au cas où des membres du groupe rencontrent des difficultés lorsqu'ils sont confrontés aux outils de communication de la plateforme et aux applications manipulées au travers de ces outils. En commutant dans cette session, le professeur autorise le membre du groupe qui rencontre des difficultés à exposer son problème à l'ensemble du groupe, en lui octroyant les contrôles d'accès nécessaires. L'expert reçoit ensuite les contrôles d'accès qui lui sont nécessaires pour expliquer et clarifier le problème, en utilisant les outils de collaboration requis.

Le professeur agit comme modérateur. Il orchestre l'attribution des contrôles d'accès parmi les membres de la session, plus précisément parmi les étudiants qui rencontrent des difficultés et l'expert. Les interactions définies par ce scénario sont similaires à celles de la figure 2.11, à l'exception que l'expert est maintenant un membre explicite du groupe, au même niveau que les étudiants.

4.2.6 Expériences et bilan

Les scénarios de téléenseignement classique donnent des structures de graphes relativement simples, avec des schémas de type 1 vers N ou N vers N. Ceci peut s'expliquer par les caractéristiques des scénarios décrits qui mettent en œuvre un contrôle de la session très centralisé (schémas 1 vers N) ou qui n'utilisent aucun contrôle (schémas N vers N). En dépit de leur simplicité, des modèles similaires de session de téléenseignement ont servi à programmer le comportement automatique du gestionnaire de session, dans le cadre du projet TOPASE.

Cependant, dans le cadre du projet Lab@Future, ces scénarios ont été utilisés comme cadres expérimentaux lors de évaluations distribuées faites avec des vrais étudiants. Les situations de cours modérée et d'expérimentation technique ont notamment été mises en œuvre de façon distribuée entre le LAAS-CNRS et l'Université de Brème. Deux étudiants français se trouvaient dans deux pièces au LAAS, sans pouvoir communiquer directement entre eux. Un professeur français, localisé dans une autre salle, suivait les étudiants dans la réalisation de leurs exercices. Un expert allemand des outils se trouvait à l'université de Brème. Un observateur psychologue autrichien, lui aussi connecté au système dans une autre pièce, contrôlait le processus expérimental. Les outils de simulation étaient situés à l'Université de Brème et étaient accédés à distance, au travers de l'Internet classique, par les étudiants, le professeur et l'observateur.

Finalement, un des défis du projet Lab@Future, qui était de définir et d'expérimenter des scénarios novateurs issus des théories de l'activité et du constructivisme n'ont été qu'imparfaitement réalisés. Ces théories ont cependant servi d'un côté à la définition de cadres pédagogiques pour la conception de cours et d'exercices et d'un autre côté à la proposition d'une méthodologie d'évaluation des scénarios pédagogiques, méthodologie qui a servi notamment à comparer les gains entre l'apprentissage traditionnel et l'apprentissage distribué au travers d'outils de communication et de collaboration.

4.3 Scénario d'apprentissage collaboratif

Pour mieux illustrer le pouvoir de spécification de notre modèle, nous avons proposé un scénario plus complexe qui pourrait correspondre à un apprentissage collaboratif, plus proche des théories de l'activité et du constructivisme. Ce scénario n'a pas été évalué et testé cependant.

Quatre étudiants travaillent en groupe. Ils sont organisés en binômes et chaque binôme a un dossier de projet à réaliser. Les étudiants utilisent un outil de vidéoconférence et un outil de partage d'applications. Nous ne représenterons pas les interactions du partage d'applications car elles sont identiques à celles des scénarios précédents (un membre du binôme est émetteur vers N destinataires). Un deuxième niveau est formé d'un ensemble de deux professeurs dont le rôle est de surveiller l'avancée du déroulement du travail des binômes. Les trois experts sont des spécialistes de certains domaines ; ils conseilleront et aideront les binômes si besoin est. Les professeurs ont un rôle de médiateur entre les élèves et les experts. Ils surveillent et guident les élèves durant leur travail.

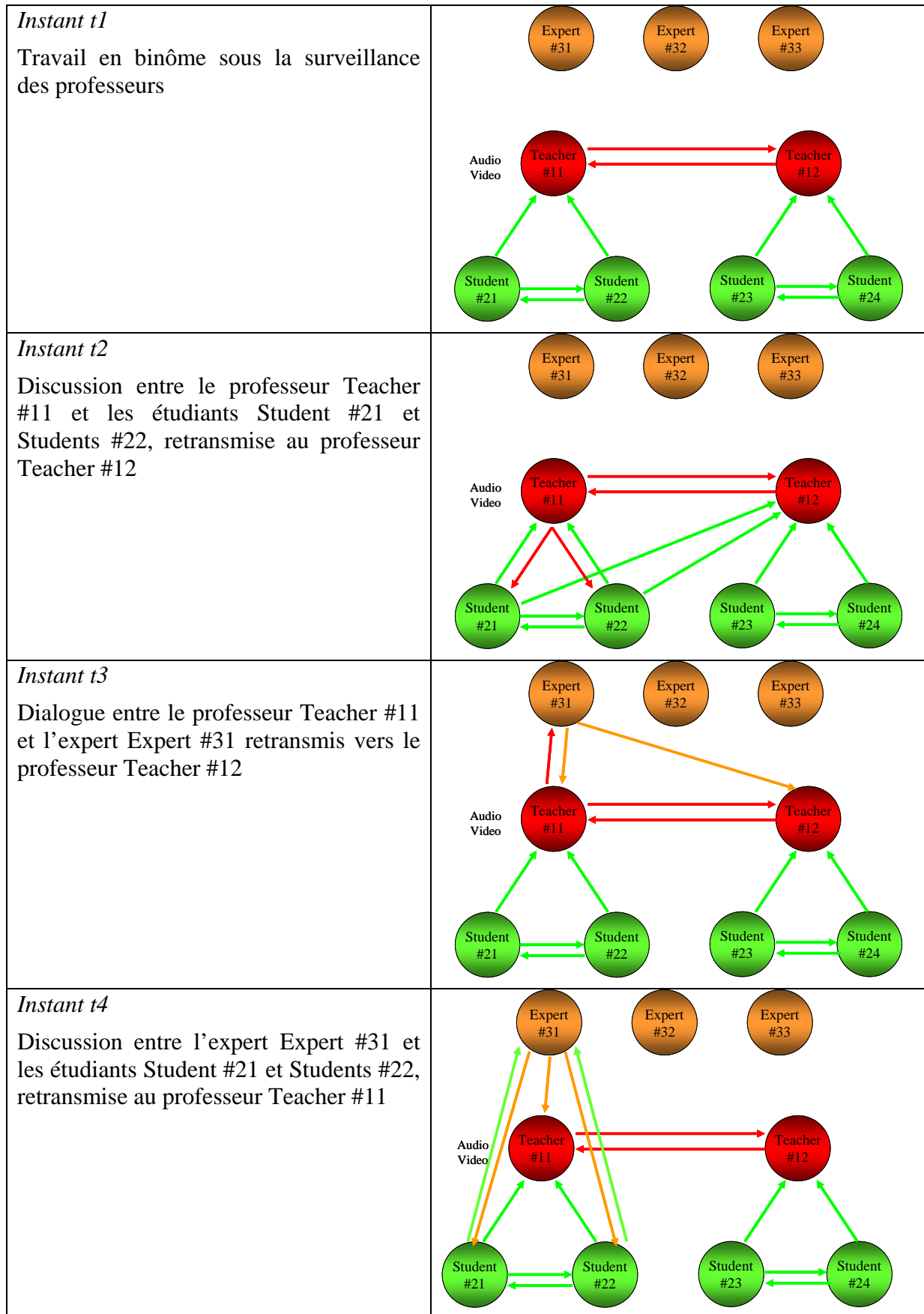


Figure 2.13. Modélisation d'un scénario d'apprentissage collaboratif

L'instant t1 de la figure 2.13 décrit la configuration des flux de la vidéoconférence dans laquelle chaque professeur surveille le travail d'un binôme. Les professeurs peuvent dialoguer entre eux. Les experts ne sont pas sollicités.

A l'instant t2, suite à un problème, le binôme Student #21 et Student #22 contacte le professeur qui les surveille (Teacher #11) et entame une discussion avec ce dernier. Pour des raisons de maintien de la cohérence au niveau des professeurs, l'ensemble de la discussion est retransmise vers tous les professeur : dans notre cas, Teacher #12 suit ce qui se dit.

Après analyse du problème, le professeur Teacher #11 décide de contacter l'expert dont les compétences lui semblent le plus en accord avec le problème. Le dialogue qui s'instaure entre le professeur et l'expert Expert #31 choisi est retransmis vers le professeur Teacher #12, pour maintenir la cohérence des échanges au niveau des professeurs, à l'instant t3.

Finalement, à l'instant t4, l'expert communique avec le binôme Student #21 et Student #22. Les échanges sont suivis par le professeur concerné, ici Teacher #11. Une fois son intervention terminée, l'expert rompt la communication et les étudiants du binôme communiquent à nouveau entre eux sous la surveillance d'un professeur : les flux de la vidéoconférence reprennent la configuration de l'instant t1.

5. Synthèse

La table 2.12 résume de façon synthétique nos contributions techniques. Ces deux contributions ont porté sur des aspects liés à la réalisation du modèle. Les supports technologiques pour la réalisation sont principalement XML, Graphviz et JAVA. Ces deux réalisations ont été valorisées dans le cadre contractuel européen du projet DSE.

Contribution	Réalisation	Support	Cadre contractuel
Eléments pour des services de communication orientés modèle	Implantation du modèle	XML et JAVA	Européen : projet DSE
Eléments pour des services de communication orientés modèle	Visualisateur de session	Graphviz et JAVA	

Table 2.12. Synthèse de nos contributions techniques

La table 2.13 résume nos deux contributions en termes d'utilisation du modèle de Diagrammes de Coordination Dynamiques. Ces contributions ont porté sur la modélisation de scénarios de téléformation et de téléenseignement. Les propositions ont été valorisées dans un cadre national (projet TOPASE) et européen (projet Lab@Future).

Contribution	Réalisation	Support	Cadre contractuel
Modélisation de scénarios	Scénarios de téléformation aéronautique	Diagrammes de Coordination	National : projet TOPASE
Modélisation de scénarios	Scénarios de téléenseignement	Diagrammes de Coordination	Européen : projet Lab@Future

Table 2.13. Synthèse d'utilisation du modèle

Chapitre 3. Gestion de sessions collaboratives

Le chapitre 3 présente nos contributions pour la gestion de sessions collaboratives, conformément au canevas d'instanciation des résultats obtenus (figure 0.2). Il détaille l'instance *Gestion de Sessions* de ce canevas.

La partie 1 présente le cadre architectural qui nous a servi de support pour concevoir nos services et nos protocoles. La partie 2 décrit le service et le protocole qui gèrent des sessions collaboratives synchrones. La partie 3 présente la première version d'implantation que nous avons réalisée du protocole. Cette contribution détaille le lien entre l'instance *Gestion de Sessions* et l'instance *Téléenseignement/Téléingénierie* de la figure 0.2. La partie 4 décrit l'approche méthodologique bi-modèle qui étend notre conception. Cette approche a servi à l'implantation de la version V2 de notre service et protocole de gestion de session. Cette contribution correspond au lien qui boucle sur l'instance *Gestion de Sessions* de la figure 0.2. La partie 5 présente la nouvelle approche architecturale que nous avons proposée pour structurer la nouvelle version V3 de notre protocole de gestion de sessions. Cette architecture, qui introduit un service de modèle, détaille le lien entre l'instance *Gestion de Sessions* et l'instance *Diagrammes de Coordination* de la figure 0.2.

1. Introduction

Cette partie présente nos apports en terme de services pour les systèmes collaboratifs. Gérer les utilisateurs et les ressources organisées en sessions nous semble être une des fonctionnalités essentielles que doit proposer un système collaboratif. Nous avons donc focalisé nos contributions sur l'aspect service de gestion de session qui constitue le point centralisateur de toutes nos démarches. Cette gestion concerne le traitement des graphes d'information au sein de systèmes multimédias distribués. Ce traitement se passe à l'intérieur de sessions synchrones car seules de telles sessions peuvent produire et manipuler des données multimédias interactives telles que celles existant dans les outils de vidéoconférences. La réalisation de services logiciels de gestion de session inclut la prise en compte des contraintes réseau dans le sens des services de communication disponibles, de leurs caractéristiques et de leurs propriétés.

1.1 Cadre de conception des services et des protocoles de communication

Le modèle d'architecture en couches [ZIMM80], ou modèle de référence OSI (Open System Interconnection), sert de base conceptuelle pour concevoir les services et les protocoles de communications des systèmes distribués. Une telle architecture est décrite sous la forme d'un empilement de modules ou couches de protocole, chaque couche définissant un niveau d'abstraction.

Un *service* de niveau N (figure 3.1), représente l'abstraction de la couche N et de l'ensemble des couches inférieures sur lesquelles la couche N s'appuie. Un service N est défini par un ensemble de *primitives de service*. Ces primitives servent pour le dialogue entre les utilisateurs du service N et ce service N. Chaque utilisateur est relié au service N au moyen d'un *Point d'Accès au Service* (SAP) qui lui permet d'envoyer et de recevoir les primitives de service N. Un service N n'est pas un bloc monolithique, mais est composé d'un ensemble d'entités de niveau N. Ces entités communiquent et collaborent entre elles, et leur association fournit le service N. L'ensemble des messages échangés entre les entités N est appelé *Unités de Données de Protocole* (PDUs). Les règles de dialogue entre les entités N, associées aux PDUs, forment le *protocole* N qui réalise le service N. Cette réalisation peut

bien entendu ne pas être unique : il est possible de proposer plusieurs protocoles N. Cependant, le service N attendu doit rester le même. Pour communiquer entre elles, les entités N doivent utiliser un service de communication sous-jacent de niveau N-1 accessible au moyen d'un ensemble de primitives de service de niveau N-1.

Ce type d'architecture est en fait "récuratif". En effet, les utilisateurs du service N peuvent former un ensemble d'entités de niveau N+1, et ainsi fournir un service de niveau N+1. De même, le service N-1 peut être formé par des entités de niveau N-1, qui communiquent en utilisant un service de niveau N-2.

Le premier avantage de ce modèle architectural se présente en terme de réutilisation des couches inférieures. Un service d'une couche inférieure est utilisable par différentes couches supérieures au dessus de lui, ces dernières réalisant des fonctionnalités différentes entre elles. Le deuxième avantage de cette architecture se décline en terme de modularité : le changement d'une couche supérieure n'a pas d'influence sur la conception d'une couche inférieure. Ces deux avantages, en terme de modularité et de structuration en niveaux de fonctionnalités différents sont très importants et conditionnent fortement la conception de nombreux systèmes de communication.

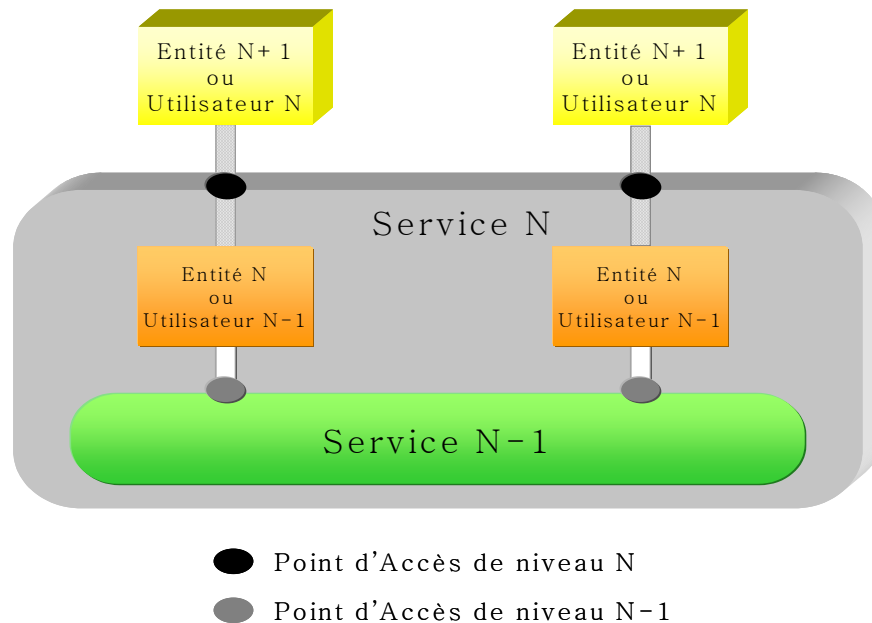


Figure 3.1. Modèle d'architecture en couches

A partir du découpage fonctionnel proposé par l'ISO en quatre couches basses (physique, liaison de données, réseau et transport) qui traitent de la fiabilité et de l'acheminement des données et en trois couches hautes (session, présentation et application) qui s'occupent de la sémantique des données et de leur représentation, ce modèle architectural a été utilisé pour concevoir des services et des protocoles point à point.

1.2 Objectifs

Notre premier objectif a été d'appliquer le cadre architectural en couches au domaine du travail collaboratif. Ce cadre a été étendu à la conception de services et de protocoles de session et de coordination. De tels services, multipoints, s'appuient sur des diffusions de messages.

Le deuxième objectif est de proposer un service de gestion de session programmable par modèle comme élément fédérateur et comme support commun de réalisation. Le but des différentes versions de ce service est de montrer l'évolution technique, technologique et conceptuelle de notre démarche.

Le troisième objectif est, au travers de ce service et de ses trois réalisations (versions V1, V2 et V3), de mettre en avant nos trois principales contributions en terme de :

- Implantation formelle
- Méthodologie de conception
- Support architectural orienté service de modèle

2. Gestion de session

Les services de gestion de session font partie des fonctionnalités de base requises par de multiples environnements collaboratifs. De tels services assurent le lien et la coordination entre les utilisateurs en session et les divers éléments manipulés.

2.1 Service de gestion de session

Les Diagrammes de Coordination présentés au chapitre précédent donnent les configurations qui sont requises pour la réalisation du travail collaboratif associé au groupe. Ils définissent ainsi les structures valides de la session. Le but premier du *Service de Gestion de Session (SGS)* que nous avons proposé est de garantir que, à chaque instant, la session qu'il manipule respecte une configuration valide. Cette propriété reste garantie pendant la dynamique de la session, dynamique qui survient au niveau du groupe, des rôles des membres, et des outils qu'ils manipulent. Pour cela, le service de gestion de session SGS doit coordonner la prise en compte des événements de session avec le passage d'une configuration valide vers une autre.

2.1.1 Description

Le service proposé contrôle l'évolution du groupe coopératif dans le temps. Il cherche, dès que possible, à passer d'une configuration valide (dans laquelle les membres collaborent en mode synchrone) vers une autre configuration valide, en prenant en compte les requêtes d'entrée et de sortie de la coopération, qui proviennent des membres du groupe. Ce service se charge de l'évolution dynamique de la session, et garantit que, après la prise en compte de requêtes d'entrées/sorties, la structure courante de la session respecte toujours une configuration valide prédéfinie.

Lors de l'arrivée de requêtes d'entrée/sortie, le système propose la nouvelle configuration valide possible à l'ensemble des membres déjà présents en coopération synchrone. Plusieurs possibilités sont envisageables pour changer de configuration. La première est de changer de configuration chaque fois que cela est possible. Le système qui gère la coopération réalise la modification pour chaque nouvelle configuration. La deuxième possibilité, qui a été retenue ici, est de considérer le changement de configuration comme une décision coopérative et d'utiliser un mécanisme générique pour implanter une technique de décision de groupe. Par conséquent, la possibilité de changement se voit proposée à l'ensemble des membres présents en coopération. Ces derniers votent pour accepter ou pour refuser de changer de configuration. Le type et les règles de vote sont spécifiés par l'application coopérative. Suivant le résultat, la configuration valide courante est changée ou non. La structure courante de session reste inchangée si le vote est refusé.

2.1.2 Phases d'évolution et cohérence

Le service SGS gère deux étapes, lorsque la coopération est en cours : la première est l'étape de coopération proprement dite, dans laquelle les membres réalisent le travail de groupe et échangent des informations en suivant la structure de la session.

La seconde étape est celle de la prise de décision lorsque les membres présents votent. Le service de gestion de session supervise le processus de vote. Le changement potentiel de la structure de la session doit être accepté par les membres qui sont actuellement présents en coopération. A la fin de cette étape, le service leur transmet le résultat.

Lors d'une analyse plus fine du comportement du service de gestion de session, il apparaît que les échanges de données peuvent interférer avec les changements de la structure de la session dans le temps, et de ce fait ces deux éléments doivent être synchronisés entre eux. Deux principales synchronisations des données sont nécessaires : avant de changer de structure de session, les données manipulées par les membres doivent atteindre un état cohérent, état qui respecte la structure du diagramme de coordination courant. L'hypothèse minimale retenue par ce service est qu'un état cohérent est obtenu lorsque tous les échanges de données non volatiles entre les membres présents en coopération sont terminés : les membres accèdent alors aux même valeurs pour toutes les données.



Lorsqu'un changement de groupe a été décidé, les nouveaux membres en coopération doivent échanger entre eux l'ensemble des informations initiales qui composent le contexte de chacun, en suivant la structure de la session, ceci avant de commencer la nouvelle coopération. En effet, les nouveaux arrivants n'ont pas connaissance du contexte des autres coopérants avec lesquels ils sont en relation suivant le diagramme de coordination. De la même façon, les anciens coopérants, qui restent dans le groupe, doivent obtenir les contextes des nouveaux arrivants.

Ces besoins en synchronisation se décomposent en quatre phases différentes :

- (1) Echange des contextes initiaux entre les membres de la nouvelle configuration.
- (2) Réalisation du travail en session.
- (3) Suspension des échanges de données avec état cohérent pour terminer la configuration courante valide.
- (4) Restructuration de la session

2.1.3 Primitives de service

Les primitives de service pour définir le service SGS, sont regroupées par unités fonctionnelles dans le tableau 3.1.

Dans la colonne sens d'utilisation, les flèches  indiquent que la primitive va d'un utilisateur vers le service de gestion de session. Les flèches  indiquent que la primitive va du service vers l'utilisateur.

Les primitives des unités fonctionnelles d'entrée et de sortie des participants provoquent l'évolution de la session dans le temps. Les primitives de décision permettent à chaque participant présent en session d'accepter ou de refuser un changement. Lorsqu'un changement est accepté, la primitive de changement de configuration informe de la nouvelle configuration de session mise en place. Les primitives d'échange des contextes initiaux interviennent dès la mise en place d'une nouvelle configuration de session. Les primitives d'échange de données

sont utilisées pendant le travail collaboratif associé au groupe. Les primitives de cohérence des échanges permettent de synchroniser les participants avant tout changement de configuration et informent de la cohérence des contextes après la mise en place d'une nouvelle configuration de session.

Nom	Sens d'utilisation	Rôle
Entrée d'un participant		
<i>joinSessionReq</i> (id: session_id; role: role_type)	↘	Un participant demande à entrer dans la session courante.
<i>joinSessionInd</i> (id: session_id; result: boolean, session: session_type)	↗	Cette indication signale si l'entrée a été acceptée ou non. Dans l'affirmative, "session" contient la configuration courante.
Sortie d'un participant		
<i>leaveSessionReq</i> (id: session_id)	↘	Un participant demande à quitter la session courante.
<i>leaveSessionInd</i> (id: session_id; result: boolean)	↗	Cette primitive signale si la sortie a été acceptée ou non.
Prise de décision		
<i>voteInd</i> (id: session_id; newsession: session_type)	↗	Une nouvelle session courante est possible et les participants présents votent pour accepter ou refuser le changement.
<i>voteResp</i> (id: session_id; result: boolean)	↘	Décision de chaque participant.
Changement de configuration		
<i>newSessionInd</i> (id: session_id; result: boolean; newsession: session_type)	↗	Cette primitive signale si le changement dans la session courante a été accepté ou non. Dans l'affirmative, "newsession" contient la nouvelle configuration courante.
Echange des contextes initiaux		
<i>contextReq</i> (id: session_id; data: data_type)	↘	Un participant transmet son contexte initial aux membres avec lesquels il est en relation.
<i>contextInd</i> (id: session_id; data: data_type)	↗	Réception des contextes initiaux des autres participants.

Echange de données		
<code>dataReq (</code> id: session_id; data: data_type)	↓	Un participant envoie une unité de donnée.
<code>dataInd (</code> id: session_id; data: data_type)	↑	Réception d'une unité de donnée.
Cohérence des échanges		
<code>noNewDataInd (</code> id: session_id)	↗	Suspension des échanges de données d'un participant pour atteindre un état cohérent.
<code>dataCoherentInd (</code> id: session_id)	↗	Les contextes des participants sont cohérents : le travail peut continuer

Table 3.1. Primitives du service SGS

2.2 Protocole

Le protocole proposé est un protocole semi centralisé. Toutes les requêtes qui concernent l'entrée et la sortie de la session sont contrôlées par un module gestionnaire. Les échanges de données sont par contre distribués entre les utilisateurs, selon la structure de la session. Cette conception permet de séparer les rôles : chacun des membres coopérants est ainsi responsable des données qu'il véhicule et se charge de leur acheminement. Le gestionnaire se charge de contrôler l'évolution dans le temps de la structure de coopération.

L'architecture de la spécification, présentée sur la figure 3.2, est basée sur le modèle de référence en couches OSI.

Au niveau application, les modules "Util i" représentent les membres participant à une session synchrone. En utilisant les primitives du service de gestion de session, chaque utilisateur envoie à son module protocole respectif des requêtes pour joindre (resp. quitter) la session, envoie (resp. reçoit) des informations vers (resp. depuis) les autres membres coopérant, et participe au processus de vote.

Les modules "Protocole i", chacun connecté à un utilisateur, fournissent le service défini. Chaque module transmet les requêtes pour joindre (resp. quitter) la session au module gestionnaire. Ils se transmettent entre eux les données manipulées en session et gèrent les conflits de cohérence entre les échanges de données et les changements de structure de la session. Chaque module envoie aussi au gestionnaire la réponse de son utilisateur connecté, lors d'un vote.

Le module "Gestionnaire" contrôle la dynamique de la session. Il a connaissance des configurations de session valides, détenues par le module "Configurations valides". Le gestionnaire reçoit les requêtes qui proviennent des modules "Protocole i" pour joindre (resp. quitter) la session. Il gère le processus de vote en indiquant aux modules "Protocole i" si un changement de structure a été accepté ou refusé. Dans l'affirmative, il envoie un ordre vers les modules "Protocole i" pour arrêter leurs échanges de données et pour mettre leurs données dans un état cohérent. Il met en place la nouvelle structure de la session et donne l'ordre aux

nouveaux agents coopérants d'échanger leurs données initiales avant de reprendre le travail collaboratif.

Le service de diffusion sous-jacent doit respecter un ordre FIFO entre les paires connectées, c'est-à-dire qu'un message transmis par une entité ne doit pas dépasser les messages précédents émis par cette même entité. De plus, tous les messages de contrôle doivent être transmis de façon fiable, les messages de données supportant d'éventuelles pertes.

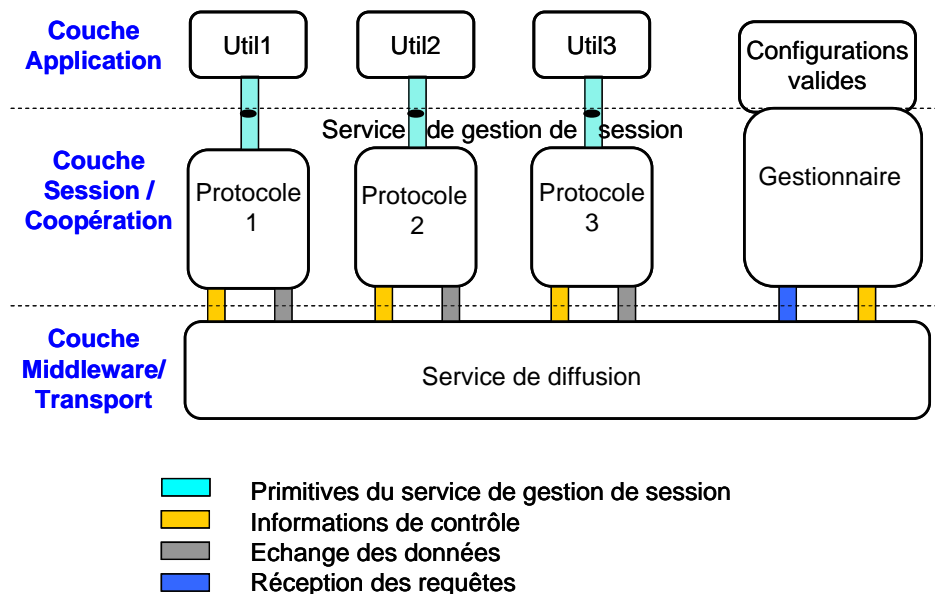


Figure 3.2. Architecture du protocole de gestion de session

2.3 Réalisation

L'ensemble de l'étude de ce service et du protocole proposé a été réalisé dans le cadre de ma thèse.

Au niveau du service, nous avons notamment :

- Traité les cas de déconnexions brutales avec reconfiguration de la session sur erreur

Au niveau protocolaire, nous avons :

- Codé les entités de protocole et les modules gestionnaire nécessaires
- Simulé le comportement de l'ensemble des entités
- Validé les comportements décrits par des techniques d'analyse du graphe d'accessibilité

Nous renvoyons le lecteur à [VILL95b] pour plus de précision.

2.4 Positionnement

La table 3.2 illustre le positionnement de notre service de gestion de session vis-à-vis des autres systèmes de gestion synchrones que nous avons analysés dans le chapitre 1.

Du point de vue technologique, le développement de ces systèmes s'appuie sur des éléments proches de ceux que nous avons utilisés pour coder les versions les plus récentes de notre protocole (versions V2 et V3, présentées par la suite).

Du point de vue fonctionnel, nous rappelons que la première faiblesse de ces systèmes est que l'absence de structuration et de formalisation des sessions (utilisateurs identiques, peu ou pas de rôles) entraîne une non différenciation des utilisateurs en terme de manipulation d'outils collecticiels (tous voient et accèdent à un même espace partagé). La deuxième faiblesse due également à cette absence de structuration de session est de ne pouvoir supporter qu'un contrôle automatique très sommaire des outils collecticiels (uniquement au moment de joindre/quitter une session).

	Gestion des rôles utilisateur	Démarrage/arrêt d'un outil	politique de contrôle d'accès	Sauvegarde de session	Technologie
TANGO 2	{N utilisateurs identiques et un maître} par session	- Au moment de joindre/quitter la session - Manuellement en session	Verrous	Non disponible	Applets JAVA
Habanero 3.0	N utilisateurs identiques par session	- Au moment de joindre/quitter la session - Manuellement en session	Verrous	Disponible	JAVA
DOE 2000	N utilisateurs identiques par session	- Au moment de joindre/quitter la session - Manuellement en session	Variés -utilisateur autorisé -modérateur -passage de bâton	Disponible	JAVA
DISCIPLE	N utilisateurs identiques par session	- Au moment de joindre/quitter la session - Manuellement en session	Aucun au niveau du système	Disponible	JavaBeans
SGS	Rôles variés	- Au moment de joindre/quitter la session - Tout changement de structure de session contrôlé par modèle	Contrôle par modèle	Non disponible	V1 (TOPASE) : Estelle + C V2 (AO Telecom) : Statechart + JAVA V3 (DSE) : JAVA

Table 3.2. Caractéristiques de notre gestionnaire de session

La dernière ligne de la table 3.2 met en évidence les gains de l'approche paramétrable par modèle que nous avons retenue pour notre gestionnaire. Au niveau des sessions, il est possible de définir et de faire prendre en compte par le service de gestion des structures de session

complexes, qui comprennent des utilisateurs avec des rôles et des possibilités différentes. Au niveau des outils, cette approche nous permet un contrôle automatique plus fin : comme précédemment, un outil est démarré (resp. arrêté) lors de l'entrée (resp. la sortie) d'un membre. Cependant, il peut en plus recevoir toute notification de changement dans la structure de la session et ainsi se reconfigurer pour être en conformité avec ce changement. L'utilisation de modèle permet aussi de faire prendre en compte automatiquement par le système certaines politiques de contrôle d'accès aux outils.

3. Contribution à la gestion du travail collaboratif

En accord avec le canevas des résultats obtenus, cette partie décrit le lien entre l'instance *Gestion de Sessions* et l'instance *Téléenseignement/Téléingénierie*.

La contribution décrite dans cette partie porte sur l'implantation formelle du service et du protocole de gestion de session que nous avons proposé. Dans cette première contribution, la version V1 est utilisée pour réaliser effectivement toute la gestion des sessions dynamiques pour le prototype réalisé dans le cadre du projet TOPASE qui concerne une expérimentation de téléformation professionnelle pour des pilotes et des agents de maintenance du domaine aéronautique.

Nous avons privilégié l'utilisation de techniques de description formelle (plus précisément de la technique de description formelle Estelle présentée par la suite) pour leur apport en termes de :

- facilité de description de comportements
- qualité de réalisation
- gain de temps pour l'implantation

3.1 Langage et méthode

Le service et le protocole de gestion de session ont été décrits, codés puis simulés en utilisant le langage Estelle [VILL95b].

Estelle ("Extended State Transition ModEL") [ISO9074], [DIAZ89], [BUDK87] est un langage de description formelle qui se prête bien aux spécifications de services et de protocoles décrits suivant le modèle de référence OSI.

Estelle est basé sur le concept d'une hiérarchie de machines à états qui sont étendues par l'utilisation de files d'attente «First-In-First-Out» et par le langage PASCAL. Brièvement, une spécification Estelle peut être décrite comme un ensemble de modules. Ces modules sont en fait des instances d'une classe à laquelle est associés un type de comportement générique. Une spécification Estelle peut donc être dynamique car le nombre d'instances ou le comportement des instances peut changer dans le temps.

Les modules peuvent être emboîtés entre eux. En effet, un module peut contenir un ensemble de sous-modules. Une spécification Estelle forme donc une hiérarchie de modules.

L'interface de module décrit les possibilités de communication externes d'un module. Elle définit notamment les points d'interaction qui permettent aux différentes instances de communiquer entre elles au moyen de canaux.

Un canal est une file d'attente bidirectionnelle qui relie deux instances de modules. La structure des messages en transit dans un canal est décrite en utilisant le langage PASCAL.

Chaque instance de module contient un ensemble de données et de sous-programmes PASCAL qui lui est propre et qui n'est visible et accessible qu'à l'intérieur de cette instance. Le comportement de chacun de ces modules est défini par une machine à états étendue. Cette machine à états est décrite au moyen d'un ensemble de transitions de la forme :

<Condition/Action>

où le premier terme donne la condition de sensibilisation de la transition, parfois aussi appelée sa garde. En Estelle, cette condition peut être l'attente d'arrivée d'un message particulier et/ou un test booléen sur des variables internes ou exportées du module. La partie «Action» donne l'ensemble des actions réalisées de façon atomique lorsque la transition est tirée. Ces actions peuvent être essentiellement des envois de messages dans les files d'attente associées aux points d'interaction du module, des modifications des valeurs de variables, des créations ou des destructions dynamiques de sous-modules.

Le langage Estelle peut être compilé. En particulier, l'environnement de programmation EDT ("Estelle Development Toolset") utilisé possède un générateur de code C [EC92], c'est-à-dire qu'il est capable de traduire une spécification Estelle en C. De plus, il est possible, à l'intérieur de la spécification Estelle, d'inclure des parties de code écrites directement en C et de faire appel à des routines ou fonctions C. Cette propriété se révèle intéressante lorsque l'on désire utiliser et récupérer la spécification réalisée à des fins d'implantation.

La technique de modification de la spécification Estelle en vue de son implantation consiste essentiellement à modifier les connexions utilisées entre les modules. Dans une spécification Estelle, les modules sont reliés entre eux par des files bidirectionnelles. Il s'agit de remplacer certaines files Estelle par un canal de communication réel, canal qui permet de faire effectivement communiquer les modules qui peuvent être répartis sur plusieurs sites physiques. Les créations, les connexions, et les communications réelles ne se font plus avec les primitives d'envoi et de réception fournies par Estelle, mais font appel à des fonctions interface qui réalisent les appels système pour l'envoi et la réception de messages sur les connexions établies.

Dans un environnement UNIX, les communications sur une même machine peuvent être basées sur les "Inter-Process Communications" comme les tubes ou les files de message. Par contre, les canaux Estelle entre machines peuvent être remplacés par l'utilisation de primitives de transport comme les "sockets" s'appuyant sur les protocole TCP ou UDP, pour des communications distantes.

Chaque instance de module Estelle communiquant par un moyen de communication réel est alors incluse dans un processus UNIX. Certaines transitions font alors appel à des fonctions externes interface écrites en C. Puis, le code du moteur d'implantation est alors automatiquement ajouté au processus pour assurer l'exécution de ce dernier. Dans un cas plus général, un ensemble de modules Estelle peut être regroupé dans un même processus UNIX.

3.2 Architecture d'implantation

L'environnement prototype support de téléformation se compose de deux parties principales : Un ensemble de trois outils collecticiels (une vidéoconférence N-TSVS, un logiciel de partage d'application et un tableau blanc) assure les échanges d'information pour chacune des fonctionnalités identifiées et fournit les interactions nécessaires entre l'instructeur et les élèves distribués géographiquement. Un ensemble générique de services de gestion de session et de coordination des outils se charge de maintenir la structure du groupe dans le temps et de contrôler la plate-forme support du travail de groupe.

Parmi ce deuxième ensemble, le service de gestion de session a été codé entièrement en Estelle, implanté automatiquement, puis intégré au sein de l'environnement prototype. Ce service représente la version V1 de notre contribution.

L'architecture du prototype est décrite par la figure 3.3. Les modules en gris représentent les parties implantées de façon automatique à partir de la conception formelle. La gestion des outils et des diverses phases de coopération se fait au niveau applicatif par le lanceur, tandis que la gestion de la dynamique du groupe se fait au niveau session/coopération par les protocoles et le gestionnaire de coopération. Cette architecture, respecte la séparation entre la partie outils et la partie générique de coordination du groupe. Cette approche modulaire peut facilement s'adapter aux modifications des outils ainsi qu'aux changements dans la structure des groupes, sans reconsidérer la conception de l'ensemble du système. Toute modification de l'ensemble des outils collecticiels n'aura que peu, voire aucune, influence sur le service de gestion de session. Tout changement dans la structure du groupe ne modifiera pas l'ensemble des outils collecticiels, mais seulement leur configuration dynamique.

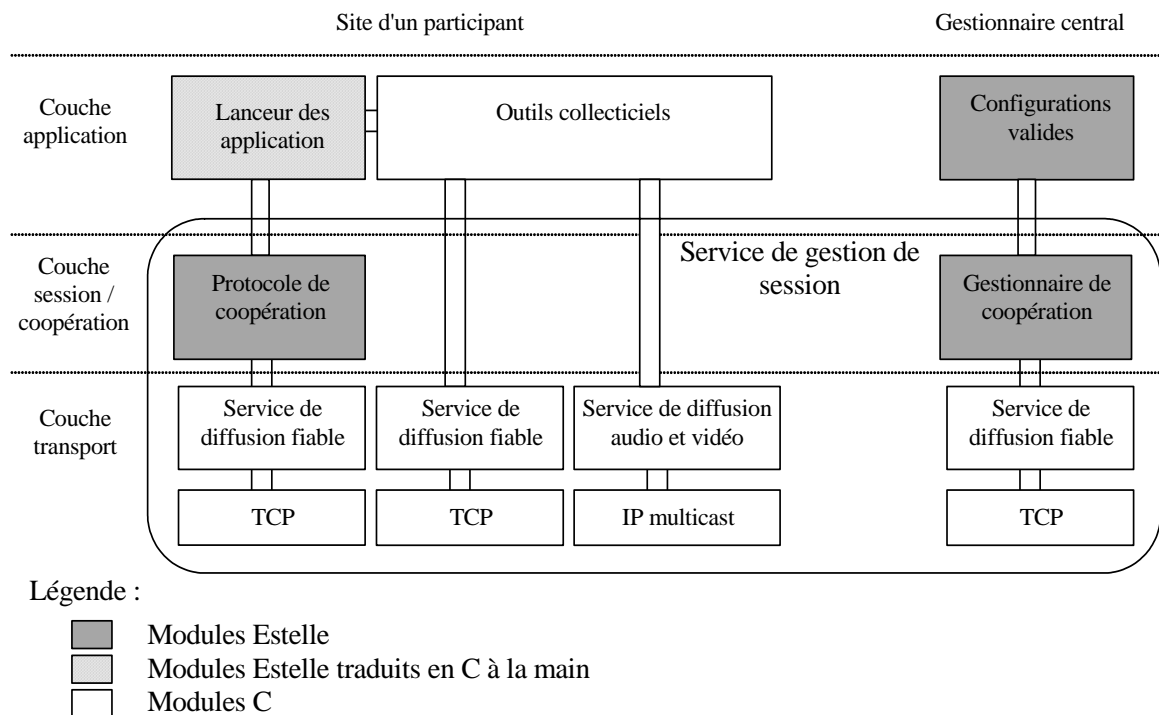


Figure 3.3. Architecture du prototype avec intégration du service de gestion

Les trois outils collecticiels (vidéoconférence, logiciel de partage d'application et tableau blanc) de l'environnement de téléenseignement font partie du niveau application.

Le lanceur des applications, au niveau applicatif, a quatre rôles principaux pour chaque utilisateur : tout d'abord, il gère les différents outils en leur fournissant toutes les informations nécessaires à leur propre configuration (configuration des flux de données respectifs). Il donne l'ordre de démarrage et de fin des outils suivant la phase en cours, maintient le contexte de chaque outil, et les informe lorsqu'un problème survient. En deuxième lieu, il gère les changements de phase. Grâce aux diagrammes de coordination, il surveille la mise en place et la modification des flux de données, et informe ainsi les outils concernés. Son troisième rôle concerne l'affichage des informations sur le contexte courant du groupe coopératif (les

personnes présentes en coopération, leur nombre, la configuration courante du groupe, l'état des différents outils...). Finalement, il maintient la valeur courante des estampilles locales, nécessaires pour assurer la synchronisation des données des outils. Pour faciliter l'interfaçage du lanceur avec les outils développés en C, nous avons opté pour une traduction manuelle de sa spécification formelle. Cependant, cette traduction n'a pas présenté de grande difficulté, du fait de la simplicité du comportement codé.

Le module de configurations valides partage ses données avec le service de gestion de session sous-jacent, et plus précisément avec le gestionnaire de la coopération. En utilisant ces conditions qui sont dépendantes de l'application, le gestionnaire sait si un changement dans la dynamique du groupe est possible. En plus des règles de validité, ce module contient le type de vote utilisé, avec les poids affectés à chaque utilisateur. Ce module a été généré automatiquement depuis la spécification Estelle.

Le service de gestion de session [VILL95b] coordonne la session et la configure selon ses changements de structure (phases de briefing, d'auto apprentissage, débriefing) et selon sa dynamique (entrées/sorties d'utilisateurs et changements de rôle). Il se compose des modules protocole de coopération et du gestionnaire de coopération. Il gère la dynamique du groupe en contrôlant les entrées et les sorties des membres pour maintenir le groupe dans un état cohérent, tout en surveillant les relations de données entre les membres. Il supervise le processus de vote et synchronise les changements de configuration avec les échanges de données, lorsque le groupe est modifié. Les modules protocole et gestionnaires ont été automatiquement générés à partir de la spécification Estelle.

Le service de gestion de session, construit au-dessus d'un service de transport multipoints, est considéré comme un service de niveau session/coopération [NGUY96].

Les échanges de données audio et vidéo pour les applications se font de manière non fiable, avec une bibliothèque de communication construite en utilisant IP multicast.

Les transmissions des informations de contrôle internes de chaque outil et des PDUs du protocole de coopération nécessitent des communications fiables. Ces informations sont petites et échangées en rafale, lorsqu'un changement se produit dans la session. Elles n'ont pas besoin d'un débit rapide, mais leur transport doit être garanti. Nous avons opté pour une implantation au-dessus de TCP, avec un processus centralisateur. Cependant, les primitives de service proposées sont assez génériques pour pouvoir être remplacées par n'importe quel autre service de diffusion fiable de niveau transport.

3.3 Prototype

La figure 3.4 montre l'aspect du prototype sur le poste instructeur, lorsque deux élèves sont présents. L'instructeur se sert de deux écrans sur la même station de travail, car l'ensemble des outils et des interfaces gestionnaires de la session (lanceur des applications) ne contiennent pas sur un seul écran. Pour les élèves, un seul écran suffit car la quantité d'information affichée par leurs interfaces gestionnaires a été réduite.

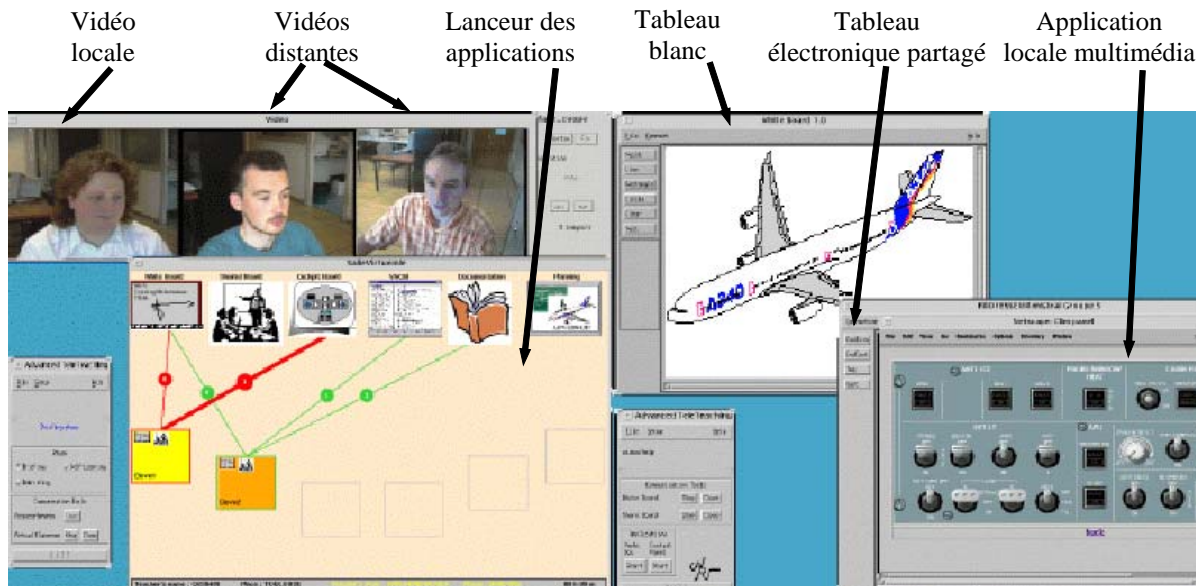


Figure 3.4. Prototype TOPASE

La visualisation de tous les membres de la classe se fait au travers d'une visioconférence synchronisée appelée N-TSVS (N-Timestamp Synchronized Videoconference System), et développée dans le cadre de la thèse de P. Owezarski [VILL98], [OWEZ98]. Les modules de N-TSVS, un par participant, définissent, gèrent, et synchronisent les différents médias audio et vidéo transmis et reçus, en garantissant leur validité temporelle au niveau du groupe. Au niveau interface, les fenêtres vidéo affichent les images reçues des autres participants. Toute fenêtre vidéo entourée par un cadre rouge indique que son participant émet de l'audio. Ce signal visuel, très utile pour faciliter le suivi des conversations, permet aussi de faciliter la résolution des collisions audio, lorsque plusieurs participants parlent en même temps.

Le tableau électronique partagé, développé par V. Baudin, permet de visualiser et de piloter à distance des applications mises sous son contrôle. L'interface de chaque membre se compose d'une fenêtre vidéo. Un membre sélectionné place ses applications à partager dans la zone de son tableau électronique. Tout autre membre autorisé peut alors contrôler à distance les applications partagées. Le tableau supporte aussi un mécanisme d'annotation.

Le tableau blanc, développé par V. Baudin, permet à un membre sélectionné de diffuser des transparents au format GIF ou JPEG. Son interface utilisateur se compose d'une fenêtre graphique. Le tableau blanc supporte des annotations publiques faites par un membre sélectionné et autorise des annotations privées.

La fenêtre du lanceur des applications, développé par V. Baudin, affiche l'état courant de l'ensemble de la session. De façon plus précise, il montre l'état de l'ensemble des applications collectives utilisées et indique quelle est la configuration courante du groupe. Il participe à la mise en place de la fonctionnalité de conscience de groupe et facilite à l'instructeur le suivi du travail de groupe.

Au niveau technologique, l'ensemble des outils collectifs et le lanceur des applications ont été réalisés en C, selon une architecture «multithreadée», où chaque thread a en charge une opération élémentaire sur un seul flux de donnée identifié. La synchronisation des

données utilise une technique à base d'estampilles temporelles. Les formats audio et vidéo manipulés sont propriétaires (car les développements ont été faits entre 1996 et 1998). Les échanges de données se font au-dessus d'IP multicast.

Le support matériel du prototype consiste en un ensemble de stations de travail SUN SOLARIS 2.5, équipées de cartes vidéo Parallax, et connectées au travers de deux réseaux Ethernet et ATM. Nos tests de l'environnement se sont déroulés en utilisant IP au-dessus d'Ethernet et d'ATM. Bien entendu, la bande passante d'Ethernet, de 10 Mb/s à l'époque, est rapidement atteinte avec seulement trois participants : ATM supporte plus d'utilisateurs.

L'ensemble des outils précédents (N-TSVS, tableau électronique partagé, tableau blanc) sont sous le contrôle du lanceur des applications, lui même connecté par l'intermédiaire du module Protocole de Coopération, au service de gestion de session. De ce fait, l'ensemble des flux de données manipulés au travers de ces outils, respectent et suivent la structure courante de la session, en accord avec la structure du diagramme de coordination qui modélise la session. De façon plus précise, le nombre d'affichages vidéo fait par le module N-TSVS de chaque utilisateur est en conformité avec la phase de la session et la position de chaque utilisateur. Par exemple, en phase de briefing, un utilisateur élève ne voit que l'image de son professeur, alors que ce dernier voit tous les élèves. En phase de question durant le briefing, la vidéo de l'élève qui pose une question apparaît automatiquement chez tous les élèves. De la même façon, les flux audio de N-TSVS respectent la structure de la session : par exemple, en phase de briefing, N-TSVS n'autorise que l'audio venant du professeur. En phase de debriefing, tout le monde peut discuter. Les outils tableau électronique partagé et tableau blanc ont un comportement similaire. En phase de briefing, seul le professeur peut partager ses applications et diffuser ses supports. En phase de debriefing, plus informelle, chaque membre peut diffuser et partager ses documents.

Ces comportements dynamiques et adaptables en session ont été autorisés grâce à la conception d'outils collecticiels ouverts et configurables au travers d'APIs, APIs que nous avons connectées au lanceur des applications. Le lanceur a ensuite eu pour charge de traduire les configurations de session, génériques et définies sous forme de graphes, en appels de méthodes spécifiques, pour chaque API d'outil.

3.4 Bilan

L'implantation en Estelle + C du service et du protocole V1 de gestion de session s'est révélée possible, sans grande difficulté, une fois la spécification mise au point et simulée. Le service et le protocole se sont avérés robustes vis-à-vis de déconnexions brutales des utilisateurs, avec reconfiguration automatique en mode dégradé de la session courante, et vis-à-vis de la prise en compte des outils.

Cette première contribution illustre l'utilisation de techniques de descriptions formelles et de leur apport pour faciliter la réalisation, et surtout l'implantation complète de services et de protocoles multi utilisateurs complexes.

Le premier gain s'exprime en terme de facilité de description. Il est beaucoup plus facile de décrire des comportements sous forme d'automates communicants que directement dans un langage.

Le deuxième gain (mis en œuvre dans ma thèse) est sur la qualité des comportements décrits lors de la conception et sur la confiance mise dans leur codage, comportements qui sont simulés et validés formellement, ceci pour en détecter les erreurs. Ces approches de

simulation et de validation aident grandement à détecter les erreurs, toujours difficiles à provoquer, à déterminer, à analyser et à reproduire dans les systèmes distribués.

Le dernier gain que nous avons illustré ici est l'énorme gain en temps de réalisation pour l'implantation du protocole. Aucune erreur supplémentaire n'est introduite car on ne procède à aucun recodage manuel des comportements. L'utilisation de moteurs d'implantation pré codés sous forme de bibliothèques logicielles facilite l'activation automatique des comportements simulés et évite de perdre du temps à recoder leurs fonctionnalités. On effectue ainsi une retouche minimum du code, retouche toujours éventuellement source d'introduction de nouvelle erreur.

Une remarque tempère ces bénéfices : L'approche compilée utilisée par l'environnement de programmation et nécessaire pour lier Estelle avec C, introduit des difficultés d'interfaçage. Certaines variables doivent être mises en commun, il est nécessaire de mélanger des instructions C et Estelle. Ces difficultés proviennent du fait que deux langages différents sont utilisés pour l'implantation finale. Nous pensons que ces problèmes pourraient être levés par l'orientation suivante : utiliser une approche bibliothèque de macro instructions pour la technique formelle initiale sur un langage unique.

4. Contribution méthodologique

En accord avec le canevas des résultats obtenus, cette partie décrit le lien qui boucle sur l'instance *Gestion de Sessions*.

La construction de services génériques de coordination au-dessus de couches middleware peut être réalisée au moyen de méthodologies qui guident le processus de conception logicielle du système distribué en cours de développement. Notre but est de généraliser les études précédentes pour des couches quelconques au dessus de services middleware.

Le développement et l'application de cette contribution ont abouti au développement de la version V2 du protocole de gestion de session. Ces travaux ont été faits dans le cadre d'un Appel d'Offre Télécommunications du CNRS. Le projet s'intitule "Nouveaux Services et nouvelles méthodologies pour le travail coopératif".

4.1 Motivations

4.1.1 Contexte

Les méthodes formelles sont largement utilisées en télécommunications et en ingénierie des protocoles pour la conception des services et des protocoles de communication [GOTZ98]. Par une tendance normale, leur application s'étend maintenant aux domaines des algorithmes et des systèmes distribués, dans lesquels les protocoles de communication forment une sous-classe particulière.

Parmi les différents types de méthodes formelles, celles basées sur des états/transitions, dont les principaux représentants sont les machines à états finis étendues [SARM96] et les réseaux de Petri [BERT91], [DIAZ01], sont utilisées pour décrire formellement les comportements des différents modules et entités. Cette base formelle est bien adaptée pour détecter les comportements incohérents, inattendus ou mal spécifiés (interblocages, livelocks, événements non prévus,...). Par effet de bord, leur description procédurale des composants distribués peut facilement s'adapter à la génération automatique du code d'implantation, créant par conséquent une base de conception formelle pour le processus de développement logiciel [VILL95b].

Une autre approche formelle s'appuie sur l'utilisation de formalismes basés sur des règles, le plus souvent composés d'extensions de logiques [DIAZ03]. Ces formalismes décrivent un système sous la forme d'un ensemble de propriétés ou de règles que ce système distribué doit respecter. Cette approche basée sur des règles est plus abstraite que celle qui respecte les modèles états/transitions car elle décrit le système global à un plus haut niveau, sans considérer en détail le comportement de chaque composant distribué. Elle s'applique facilement à l'analyse et à la vérification des systèmes distribués.

4.1.2 Proposition

L'analyse des services de coordination proposés ont laissé apparaître deux composantes orthogonales qui, bien que liées, peuvent être découplées pour des facilités de conception :

- l'aspect structurel comprend des problèmes de gestion d'architectures, de dépendances et de relations, comme par exemple les relations d'échange d'information entre les membres du groupe
- l'aspect comportemental est lié aux actions venant de chaque membre, comme les demandes d'entrée/sortie en session

La méthodologie que nous proposons cherche à prendre en compte cette séparation fondamentale lors de la conception, tout en essayant d'y introduire et de maintenir un lien indispensable entre elles.

Afin de profiter de tous les bénéfices apportés par les méthodes formelles dans le domaine de l'ingénierie des systèmes distribués, notre approche méthodologique se base sur un cadre formel bi modèle :

- Le premier modèle, basé sur des règles, sert à gérer la distribution à haut niveau pour l'aspect structurel. Il se rapporte à la distribution dynamique des entités logicielles invoquées.
- Le deuxième modèle, plus classique à base d'états et de transitions, est utilisé pour programmer le comportement "fin" des entités distribuées. Il se réfère à l'aspect comportemental de chaque composant logiciel.

Notre contribution vise à fusionner les deux aspects, supportés par les deux modèles, à l'intérieur d'une méthodologie globale pour le processus de conception logiciel de services collaboratifs distribués. Certains supports de conception comme UML ou environnements comme Telelogic Tau [TELE05] utilisent plusieurs modèles pour la conception globale et détaillée des systèmes développés. Cependant, dans ces divers cas, la conception préliminaire est faite au travers d'un modèle informel faiblement couplé au modèle formel détaillé, utilisé uniquement pour spécifier certaines parties critiques du système.

4.2 Méthodologie bi modèle

En plus des avantages des méthodes formelles pour faciliter la programmation et la mise au point des systèmes distribués, cette approche a été choisie pour deux raisons principales : la première est d'apporter une technique modulaire et raffnable au concepteur logiciel, partant de la topologie grossière des composants, et terminant par la programmation détaillée de chaque composant. La deuxième raison est de maintenir une indépendance relative entre la conception de la topologie de l'ensemble du système et celle de chaque composant local [GRAD99].

La problématique posée est donc de faciliter la conception globale du système décrit, tout en s'appuyant sur des bases formelles pour bénéficier de leur apport théorique en terme de validation des comportements définis.

4.2.1 Modèle structurel formel

Le modèle retenu est basé sur des transformations de graphes. Il s'agit d'une approche qui s'appuie sur des graphes qui décrivent la topologie des composants distribués. Ce modèle donne une vue abstraite de la topologie, associée à des règles qui décrivent son évolution possible.

Les transformations de graphes consistent en un graphe initial et en un ensemble de règles de production appelées "règles de réécriture de graphes". Plusieurs notations de transformations de graphes se rencontrent dans la littérature. Celle que nous nous proposons d'utiliser est une spécialisation de l'interface delta de [KAPL93]. Cette spécialisation a été initialement proposée par K. Drira, et utilisée pour divers services de coordination [DRIR99], [ROBL02].

Le graphe de coordination est un graphe typé dont les nœuds sont les entités qui coopèrent. Chaque nœud se voit associé un ensemble d'attributs. Pour chaque nœud, l'attribut obligatoire "état structurel", permet de représenter dans le graphe l'état du composant associé au nœud. Les valeurs des attributs "état structurel" interviennent dans l'architecture dynamique du système. Elles seront modifiées lors des changements d'états du système et de ses composants associés. Elles autorisent cette architecture à être gérée par des actions qui sont en relation avec les règles de transformation. Ces actions dépendent de la sémantique finale donnée par l'application dont les composants distribués sont liés par le graphe de coordination. D'autres attributs optionnels, qui représentent des paramètres définis par l'utilisateur, peuvent aussi être associés aux nœuds du graphe.

Les règles de transformation forment un ensemble qui s'applique sur le graphe de coordination. Une règle de transformation de graphe est un motif de graphe dans lequel nous distinguons trois fragments : les fragments "Requis et Retrait", "Requis et Préservation" et le fragment "Insertion". Les deux premiers fragments forment la partie requise pour l'application de la règle. Les règles peuvent avoir des paramètres. Des gardes, qui décrivent les conditions booléennes nécessaires aux valeurs des attributs, aux paramètres et aux variables des graphes de règles, peuvent être utilisées.

Une transformation de graphe se réalise comme suit :

- (1) un sous-graphe isomorphe aux fragments "Requis et Retrait", "Requis et Préservation" est recherché dans le graphe de coordination courant.
- (2) Si un tel sous-graphe est trouvé, le sous-graphe qui correspond au fragment "Requis et Retrait" est supprimé du graphe de coordination courant. Un sous-graphe isomorphe au fragment "Insertion" est alors ajouté au graphe de coordination précédemment transformé.

La sélection d'une règle est sous la dépendance d'un événement du système qui lui est associée. Lors de la production de cet événement, la règle correspondante est sélectionnée et les transformations qu'elle induit sont réalisées sur le graphe de coordination courant.

Une règle simple de transformation de graphe est présentée par la figure 3.5. Cette règle nécessite deux nœuds consécutifs entièrement connectés entre eux. Lorsqu'elle est appliquée, elle supprime une connexion, puis crée un nouveau nœud qu'elle connecte aux précédents.

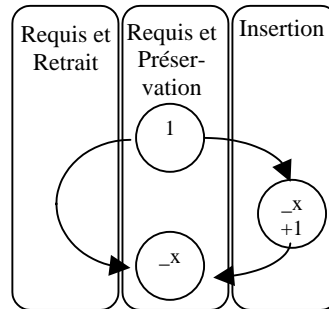


Figure 3.5. Représentation graphique d'une règle de transformation R1

La figure 3.6 décrit l'application consécutive de R1 sur le graphe de coordination initial G1. Pour des raisons de simplification, comme la règle R1 n'intervient pas sur les attributs des nœuds de G1, nous n'avons pas représenté les attributs de chaque nœud. Lors de la transformation de G1 en G2, le fragment requis est mis en correspondance avec celui du graphe originel en associant le nœud $_x$ de la règle avec le nœud 2 du graphe et les flèches entre 1 et $_x$ de la règle avec celles entre 1 et 2 du graphe. Dans la transformation suivante de G2 à G3, l'unification R1 est appliquée entre le nœud $_x$ de la règle avec le nœud 3 du graphe et les flèches entre 1 et $_x$ de la règle avec celles entre 1 et 3 du graphe.

En fonction de la sémantique associée au graphe de coordination, l'application de la règle sélectionnée introduira des transformations dans l'architecture des composants du système.

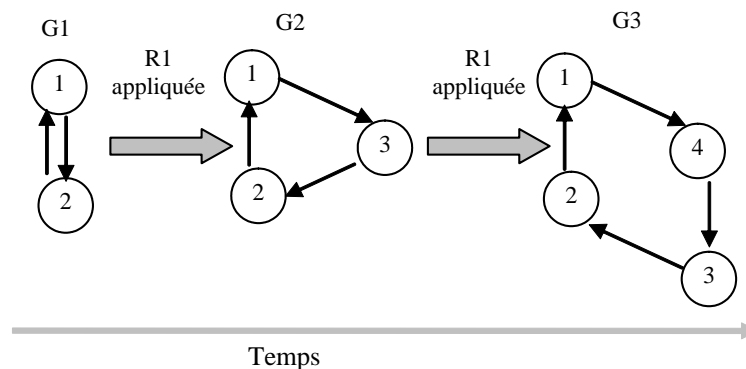


Figure 3.6. Transformations de graphe successives

4.2.2 Modèle comportemental formel

Une façon classique de modéliser et de programmer des composants distribués réactifs est d'utiliser des approches états/transitions telles que les machines à états étendues [SARM96], [HARE90]. Les modèles états/transitions supportent et facilitent la spécification du comportement interne de chaque composant.

Parmi cette classe de modèles, le formalisme statechart a été retenu pour plusieurs raisons : tout d'abord, il s'agit d'un modèle très complet pour spécifier facilement des comportements complexes. Une statechart est composée d'un ensemble hiérarchique de machines à états finis

étendues, combinées ensemble. Deux sortes d'opérations de composition sont données par ce modèle [HARE90]. Le premier est un raffinement hiérarchique. Un état majeur peut être raffiné en une machine à états plus détaillée, en conformité avec les approches descendantes utilisées en génie logiciel. La seconde opération de composition est un produit de machine à états, piloté par les événements reçus. Ce dernier opérateur est très utile pour spécifier des comportements indépendants qui peuvent avoir lieu en même temps dans un même composant logiciel. De plus, il évite de programmer des états intermédiaires qui peuvent apparaître à cause de l'entrelacement des comportements indépendants. Une autre raison de ce choix est la large utilisation des statecharts dans l'industrie avec plusieurs environnements logiciels support, et son appartenance au support de conception UML, pour modéliser des systèmes concurrents avec éventuellement des contraintes de temps.

4.2.3 Articulation des deux modèles

L'articulation des deux modèles structurels et comportementaux se fait par une association entre :

- d'un côté les valeurs des attributs "état structurel" liés à la signification des nœuds du modèle structurel et
- de l'autre certains états considérés comme plus importants, ou états majeurs, dans le modèle comportemental.

Au travers de ses attributs "état structurel", le modèle structurel peut donc se voir comme une vue globale, ou instantanée, de certains états de l'ensemble du système, ces états étant considérés comme les plus importants. La signification des valeurs de ces attributs, (ou "états majeurs" du modèle comportemental) dépend bien entendu du type de système auquel cette méthodologie est appliquée.

Les flèches entre les nœuds du modèle structurel donnent les relations qui existent entre les entités distribuées lorsque ces dernières se trouvent dans l'état majeur donné par le modèle comportemental, et identique à la valeur de l'attribut "état structurel" du modèle structurel. De la même façon, les flèches du modèle structurel peuvent avoir diverses significations (sens de communication, ordre de précedence...) suivant le type de service envisagé.

L'évolution du graphe structurel au moyen d'une règle de coordination donne la nouvelle vue globale que l'ensemble du système se doit d'atteindre. L'évolution réelle du système est faite par les entités logicielles qui sont définies au travers du modèle comportemental. Ces dernières partent d'un ensemble d'"états majeurs" en conformité avec les valeurs des attributs "état structurel" des nœuds du modèle structurel. Elles se coordonnent de façon distribuée, passent éventuellement par des états intermédiaires ou transitoires définis dans leurs comportements, et aboutissent finalement dans de nouveaux "états majeurs" qui correspondent exactement à la vue globale donnée par le nouveau graphe structurel.

Pour clarifier nos propos, reprenons l'exemple des figures 3.5 et 3.6. Le service associé correspond à la gestion d'un anneau de communication logique et dynamique entre utilisateurs. En partant du modèle structurel, chaque utilisateur est représenté par un nœud du graphe, les flèches donnant les communications possibles entre eux. La valeur de l'attribut "état structurel" que l'on associe à chaque nœud est "PRESENT", qui indique que l'utilisateur associé est présent dans l'anneau. A l'instant t_0 , seuls les utilisateurs 1 et 2 sont présents. A l'instant t_1 , l'utilisateur 3 souhaite se joindre à l'anneau. Il crée pour cela l'événement d'entrée de l'utilisateur 3, événement qui provoque l'activation de la règle R1 qui lui a été associée. Le graphe structurel G1 se transforme sous l'action de la règle et aboutit à la

nouvelle configuration G2 qui inclut l'utilisateur 3 à l'instant t2. La figure 3.7 résume l'évolution du modèle structurel.

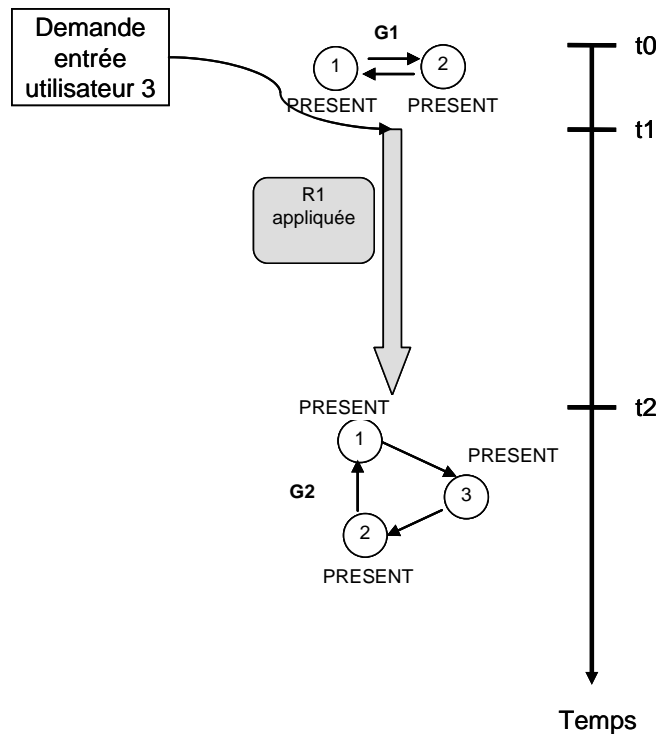


Figure 3.7 Evolution du modèle structurel

Le comportement de chaque utilisateur est décrit par la figure 3.8. Dans ce modèle comportemental simple, un utilisateur est absent de l'anneau lorsqu'il se trouve dans l'état "INACTIF". Il peut alors demander à entrer (état "ATT_ENT"), et passe dans l'état "PRESENT" lorsqu'il est accepté dans l'anneau. Toute demande d'entrée de la part d'un autre utilisateur le fait passer dans l'état "RECONFIG". Il revient ensuite dans l'état "PRESENT" lorsque le nouvel utilisateur est entré. De façon symétrique, une demande de sortie le fait passer dans l'état "ATT_SORT". Il revient à l'état "INACTIF" lorsqu'il a quitté l'anneau.

Dans ce cas, la valeur "PRESENT" de l'attribut "état structurel" du modèle structurel correspond à l'état majeur "PRESENT" du modèle comportemental.

Les états intermédiaires, "ATT_ENT", "RECONFIG" et "ATT_SORT" ne font partie que du modèle comportemental et ne correspondent à aucune valeur d'attribut du modèle structurel.

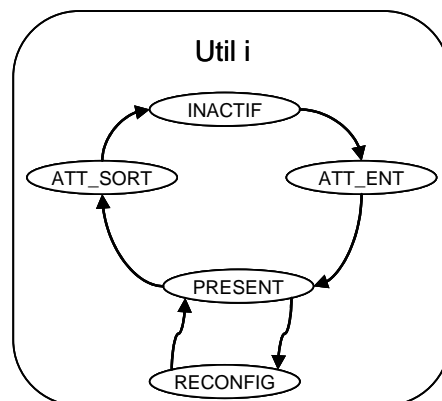


Figure 3.8. Modèle comportemental

La figure 3.9 donne l'évolution des comportements des trois utilisateurs au sein du système. A l'instant t_0 , seuls les utilisateurs 1 et 2 sont présents dans l'anneau (eux seuls sont dans l'état "PRESENT"). La demande d'entrée de l'utilisateur 3, à l'instant t_1 , entraîne son passage à l'état "ATT_ENT". Entre l'instant t_1 et t_2 , les utilisateurs passent par une suite d'états intermédiaires (état "RECONFIG"), puis ils se coordonnent entre eux jusqu'à ce que l'état global du système corresponde à la vue logique du modèle structurel. A l'instant t_2 , les trois utilisateurs sont dans l'état "PRESENT", ce qui est conforme au nouvel anneau logique G2.

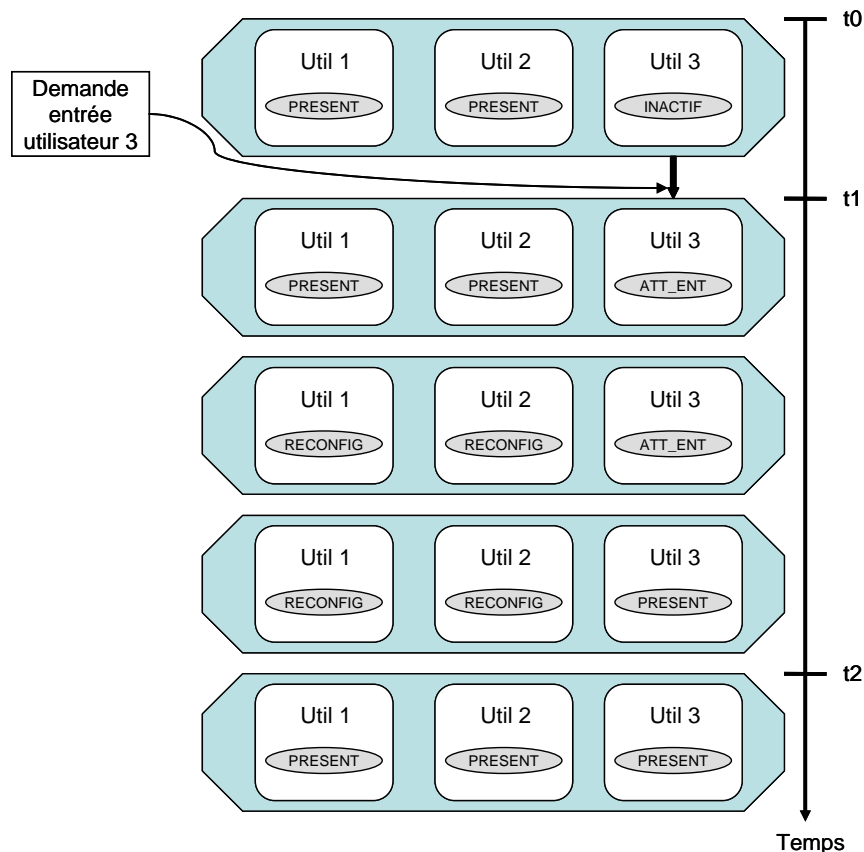


Figure 3.9. Comportement des utilisateurs dans le temps

4.2.4 Synchronisation des deux modèles

L'ensemble de ces travaux est basé sur la nécessité d'une synchronisation entre l'état courant du modèle structurel et l'état global courant du modèle comportemental. Cette synchronisation se décompose en deux parties :

P1 : Avant d'appliquer une règle de transformation, il est nécessaire que les deux modèles partent de deux états courants coordonnés. Par exemple, sur le modèle structurel, la règle associée à la demande d'entrée de l'utilisateur 3 ne s'appliquera que si le nœud 3 est absent du graphe et que tous les nœuds du graphe sont dans l'état PRESENT. Cette condition d'application est garantie par le motif de la règle sélectionnée. Au moment de l'application de la règle, il est nécessaire dans le modèle comportemental que l'utilisateur 3 ne soit pas en coopération et que tous les autres utilisateurs en coopération soient dans l'état PRESENT. Or, suite à l'avancée du travail associé au groupe, il se peut que certains des utilisateurs en coopération soient dans des états intermédiaires différents de l'état PRESENT. Il est

nécessaire, dans ce cas, qu'ils reviennent dans un état en conformité avec celui du modèle structurel pour pouvoir appliquer la règle de transformation associée à la demande d'entrée de l'utilisateur 3. Soit il faut attendre la conformité entre l'état courant du modèle structurel et du modèle comportemental, soit on peut imaginer que l'on force la cohérence du modèle comportemental en demandant aux utilisateurs en coopération de se rendre à l'état PRESENT ou bien en les y remettant arbitrairement.

P2 : Lors de l'application d'une règle de transformation, il est nécessaire de maintenir l'évolution conjointe cohérente des modèles structurels et comportementaux. Cela signifie que, lorsque le modèle structurel évolue et passe d'un état vers un autre, il est nécessaire que le modèle comportemental converge dans le futur vers le même état que le modèle structurel, en passant éventuellement par une suite d'états intermédiaires. Cette garantie doit être vérifiée lors de la conception en utilisant par exemple des techniques d'analyse d'accessibilité.

Pour des raisons de clarification, nous avons basé nos explications sur les besoins de synchronisation des deux modèles en accordant la priorité au modèle structurel, de plus haut niveau :

- Le modèle de haut niveau est d'abord activé. Il contraint les entités de bas niveau et synchronise le modèle comportemental

Ce cas correspond à un service et à un protocole de gestion de session dans lequel l'entrée ou la sortie d'un membre n'est pas soumise à l'influence et à la décision des autres membres présents dans le groupe.

Une deuxième possibilité est de prendre la position inverse et d'accorder la priorité au modèle comportemental, de plus bas niveau :

- Les entités de bas niveau se coordonnent d'abord lors de tout changement. Elles contraignent et synchronisent ensuite le modèle de haut niveau.

Ce cas correspond à un service et à un protocole de visualisation de session. Lorsque l'entrée d'un nouvel utilisateur se produit, les entités du modèle comportemental informent le modèle structurel qui modifie sa structure courante pour être en conformité avec la nouvelle session.

Le troisième choix de conception est de n'avoir pas de priorité entre les deux modèles. Selon le cas :

- soit le modèle structurel contraint le modèle comportemental, soit l'inverse se produit, soit les deux modèles s'influencent de façon conjointe et entrelacée.

Un exemple d'influence entrelacée se produit à l'intérieur d'un autre service de gestion de session qui supporte les déconnexions brutales. En cas de déconnexion brutale d'un utilisateur en coopération, les entités du modèle comportemental informent le modèle structurel de reconfigurer la session selon une nouvelle configuration valide, suite à la panne survenue. Par la suite, le modèle structurel impose sa nouvelle configuration.

Le choix du type de synchronisation parmi les trois propositions dépend du système à concevoir. Il est fait au moment de la conception du système.

4.3 Technologie middleware

L'environnement middleware Charis Actor Portfolio (CAP) [ALLE98], [CHAR97] est une plate-forme distribuée formelle qui s'appuie sur le paradigme acteur/statechart. Au vu de ses caractéristiques, cet environnement de développement nous a semblé le mieux adapté pour concevoir des nouvelles versions des services et des protocoles de coordination au dessus de services middleware.

Une spécification CAP est basée sur un ensemble d'acteurs coopérants complexes et distribués, dont chaque comportement est décrit formellement en utilisant le modèle statechart [HARE90]. Brièvement, une spécification CAP se compose de classes d'acteurs qui décrivent les comportements génériques et les ressources des acteurs. Les instances des classes d'acteurs sont les éléments distribués actifs de la spécification, qui s'exécutent dans un environnement middleware.

Un acteur CAP se compose de deux parties principales (figure 3.10). a) Un cortex externe gère les communications de l'acteur avec l'extérieur au travers d'une liste de ports de communication. b) Une statechart définit le comportement normal et courant de l'acteur. Elle est encapsulée dans le cortex. Elle se compose d'un ensemble hiérarchique de machines à états finis étendues qui accèdent toutes au contexte commun de l'acteur, composé des variables internes créées pour cette instance d'acteur.

Les acteurs distribués communiquent en mode point-à-point grâce à leurs points d'interaction en utilisant un mode de communication synchrone (rendez-vous) ou asynchrone.

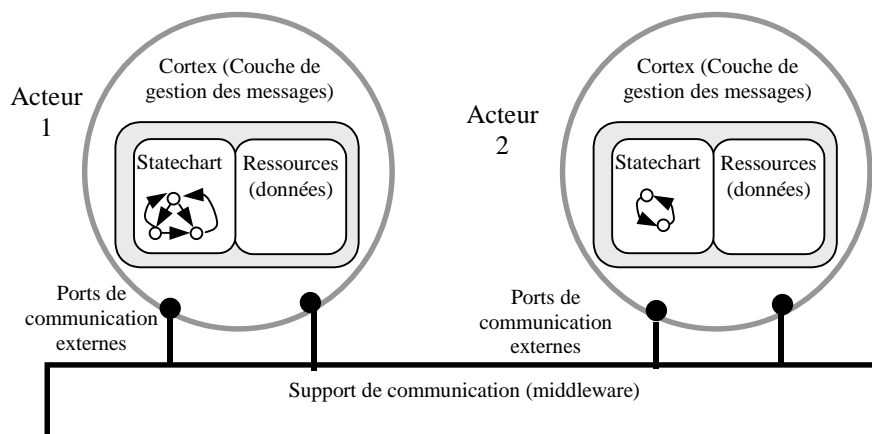


Figure 3.10. Paradigme acteur/statechart CAP

L'environnement CAP permet le développement unitaire de chaque acteur. Une fois compilé, il est possible d'injecter des événements à un acteur et de suivre ainsi pas à pas son évolution. Des traces d'exécution peuvent aussi être gardées pour faciliter la mise au point. En dernier lieu, il est possible de procéder à une intégration partielle de certains acteurs et d'observer leur comportement de façon distribuée. Un programme, appelé observateur, permet de sélectionner un ensemble d'acteurs et de tracer sous la forme de suite de séquences de messages, leurs interactions distribuées.

L'implantation des acteurs CAP est entièrement basée sur le langage JAVA.

CAP contient un langage de spécification formel appelé Actor Description Language (ADL). ADL contient un ensemble de méta instructions spécifiques (par exemple trans, state, from, to...) qui décrivent les états et les transitions de chaque statechart.

Toutes les ressources manipulées par les acteurs (données, objets, sous-programmes...) sont décrites en JAVA. Chaque acteur est ensuite pré compilé en un programme source JAVA, qui est ensuite compilé en bytecode en utilisant un compilateur JAVA standard. Les instances des classes d'acteurs, qui sont les éléments distribués actifs de la spécification, s'exécutent finalement à l'intérieur de machines virtuelles JAVA.

4.4 Implantation bi modèle pour le service V2 de gestion de session

La nouvelle version V2 du service et du protocole de gestion de session a été conçue selon la méthodologie bi modèle proposée : le modèle de graphes de coordination représente la configuration courante des utilisateurs en session, tandis que le modèle statechart code le comportement de chacune des entités mises en œuvre. Les états du modèle comportemental sont codés par les états des statecharts.

L'architecture de la figure 3.2 a servi de base à l'implantation bi modèle (figure 3.11).

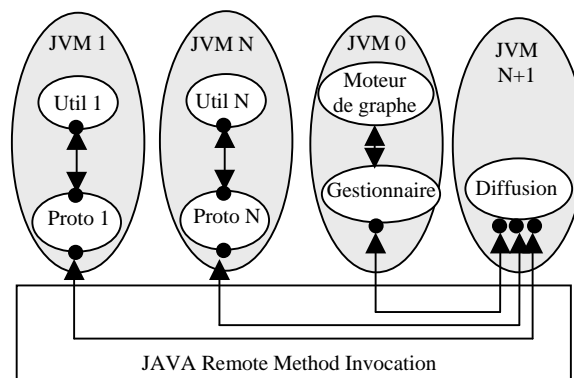


Figure 3.11. Architecture d'implantation bi modèle

L'interface entre le module moteur de graphe et le gestionnaire de coopération réalise physiquement le lien entre le modèle structurel basé sur des règles et le modèle comportemental état/transition.

Notre implantation respecte cependant un maximum d'indépendance et de faible couplage entre les règles structurelles de la session et le modèle comportemental : la configuration retenue pour la session peut ainsi être changée et reprogrammée sans reconsidérer ni changer le code interne des acteurs distribués qui gèrent la session.

Dans ce cadre méthodologique retenu pour la conception du protocole de gestion de session, les deux modèles évoluent de façon conjointe et entrelacée. Cet entrelacement s'illustre notamment par les deux étapes suivantes : lors de la possibilité d'entrée (resp. sortie) d'un nouveau membre en session, le modèle à haut niveau informe les entités distribuées. Finalement, ces dernières fixeront par leur vote la nouvelle configuration retenue et en informeront le modèle à haut niveau.

La synchronisation entre l'état du modèle structurel et l'état global du modèle comportemental se fait en attendant que l'état global du modèle comportemental soit en correspondance avec celui du modèle structurel. Dans notre implantation, il n'est pas nécessaire de forcer la correspondance car le système ne reste que peu de temps dans les états intermédiaires.

Tous les développements de la version V2 ont été faits avec CAP. Les différents modules ont été directement traduits en acteurs CAP qui ont été placés dans des Machines Virtuelles JAVA (JVM), selon l'architecture distribuée de la figure 3.11. Les deux acteurs requis pour

un utilisateur (i.e. l'utilisateur et son entité de protocole pour la gestion de session) sont regroupés dans une même JVM. Le gestionnaire de coopération et le moteur de graphe sont encapsulés dans une autre JVM.

Le support middleware de JAVA, "Remote Method Invocation" (RMI), ne supporte pas la diffusion. Par conséquent, notre service de diffusion est implanté par un acteur CAP centralisateur qui tourne sur une machine spécifique.

Le moteur de graphe pour les configurations valides est écrit avec des méthodes spécifiques JAVA. Le graphe de coordination dynamique et les règles de coopération sont définis par des classes JAVA. Ces méthodes, invoquées par le gestionnaire, implantent directement le lien entre la partie basée sur des règles et la partie distribuée états/transitions de notre système.

4.5 Bilan

Sur le plan technologique, la conception du service et du protocole V2 de gestion de session s'est avérée beaucoup plus aisée que celle de la version V1 : un seul langage (JAVA) est utilisé à la fois pour l'ensemble du codage. Ceci évite les problèmes d'apprentissage de plusieurs langages et d'interfaçage précédemment évoqués, toujours source d'erreur. Nous pensons que l'approche librairie avec méta instructions qui représentent les instructions du modèle initial est beaucoup plus facile à utiliser et à implanter car au final elle ne porte que sur un seul langage.

De plus, le choix de JAVA en tant que langage support permet de bénéficier de tous les avantages que cet environnement apporte. La compilation complète des acteurs en JAVA garantit leur portabilité et profite de la gestion de l'hétérogénéité garantie par les environnements JAVA. La communication entre acteurs distribués s'appuie sur les extensions middleware que JAVA fournit. En dernier lieu, l'utilisation d'un modèle formel au-dessus d'un langage parallèle distribué comme JAVA évite beaucoup des problèmes créés par la programmation parallèle. Le programmeur peut de ce fait se concentrer sur la conception du système distribué, et non sur les outils nécessaires pour le réaliser.

Sur le plan méthodologique, le principal apport consiste, à notre avis, en une tentative de fusion de deux approches complémentaires pour des environnements coopératifs, à savoir une approche plus structurelle et une approche plus comportementale, dont les deux composantes sont nécessaires pour tout service et protocole de coordination.

En terme de gain, l'aspect structurel intervient lors d'une approche de conception de haut niveau, l'aspect comportemental lors de la conception de niveau plus fin, avec nécessité de synchroniser les deux niveaux.

Dans le cadre contractuel de l'Appel d'Offre Télécommunications du CNRS, d'autres services ont été proposés selon cette approche : un service de coordination d'objets distribués [DRIR99] et un service d'édition partagée [ROBL02].

Pour des raisons de complétude, il nous semble nécessaire d'approfondir et de poursuivre cette étude selon plusieurs axes. Voici les principales directions que nous avons identifiées :

- Comparer cette méthodologie avec d'autres approches de conception de systèmes distribués
- Intégrer cette approche dans les démarches de conception classiques de type UML

5. Contribution architecturale

En suivant le canevas des résultats obtenus, cette partie décrit le lien entre l'instance *Gestion de Sessions* et l'instance *Diagrammes de Coordination*.

La contribution architecturale décrite dans cette partie porte sur la proposition d'un service et d'un protocole de gestion de modèle, dont le rôle est de ne contrôler que les éléments du modèle structurel. Le but est, à partir de cette gestion du modèle, d'obtenir un service de coordination générique. Cette proposition a été développée dans le cadre de la thèse de Mme Rodriguez Peralta [RODRO3]. Ceci a abouti à définir une nouvelle architecture pour notre service de gestion de session. La nouvelle implantation réalisée, la version V3, a été valorisée dans le cadre du projet Européen DSE qui porte sur la télé-ingénierie dans le domaine des systèmes spatiaux.

5.1 Motivations

Dans les contributions architecturales précédentes, deux principaux éléments sont apparus (figure 3.12) : le premier élément *Gestion de Session* regroupe les entités qui s'occupent de cette gestion de session ; le second élément *Diagrammes de Coordination* modélise la structure de la session. Ce deuxième élément est implanté à l'intérieur d'un module centralisé, le moteur de graphe qui contient les configurations valides et qui n'est qu'en relation directe avec le gestionnaire de session.

L'idée est de séparer ce deuxième élément de haut niveau, et de le transformer en un service particulier sous la forme d'une couche protocolaire. Ce nouveau service de gestion de graphes est défini comme étant une couche de coordination générique.

La nouvelle architecture proposée figure 3.12 comprend alors deux services indépendants : un qui gère la session et un service de gestion de graphes.

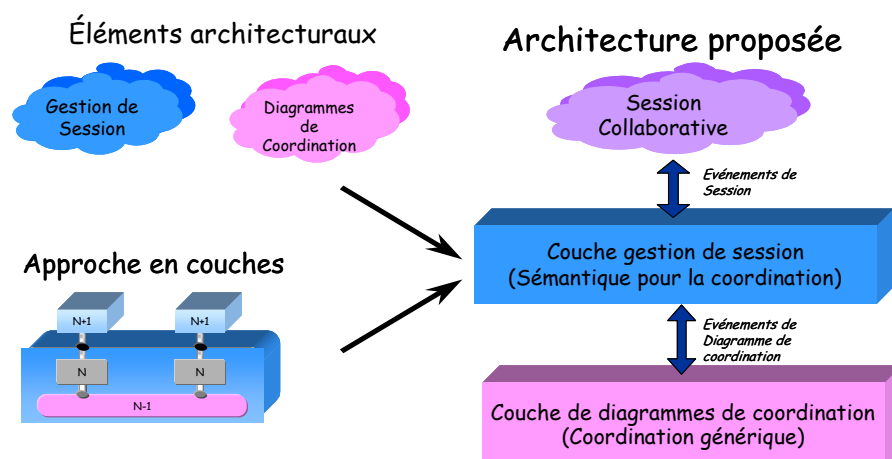


Figure 3.12. Nouvelle organisation en couche des éléments architecturaux

La partie la plus basse de l'architecture (couche diagramme de coordination) implante un service de gestion de graphe entièrement générique. Elle se charge de gérer la structure des diagrammes de coordination. Elle est associée à l'état du diagramme de coordination courant et aux transformations de diagrammes (ajouter/effacer nœuds, liens, étiquettes, etc.) pour son évolution dynamique. Cette partie bénéficie de tous les avantages provenant des manipulations de diagrammes en terme de compréhensibilité, de structuration et de

programmation à haut niveau, ceci grâce à XML. Nous avons choisi de développer cette partie sous la forme d'une couche indépendante.

Cette couche de diagrammes de coordination est définie comme implantant une coordination générique car elle n'a pas de signification précise quand elle est prise d'une manière isolée ; elle se contente de manipuler un graphe, et de le transformer selon les règles appliquées. Ce sont les couches plus hautes qui définissent et associent une sémantique à cette manipulation de graphes. Un exemple de sémantique est de coordonner des objets distribués [DRIR99] ou de partager des zones de texte dans un service d'édition distribuée [ROBL02].

La deuxième partie de l'architecture comprend la couche de gestion de session qui implante le service de gestion de session. Plus précisément, cette couche se charge de contrôler les différents composants de la session collaborative (utilisateurs, outils, données), en fonction de leurs états. Lors de l'évolution de la session dans le temps, cette couche interprète les événements de session et exécute les actions qui leurs sont associés.

Cette deuxième couche se situe au dessus de celle des diagrammes de coordination. Elle donne une "sémantique de session" à la couche diagrammes de coordination générique sous-jacente.

5.2 Architecture détaillée

5.2.1 Présentation

L'architecture détaillée du gestionnaire de session est donnée par la figure 3.13. Les couches qui composent l'architecture proposée sont décrites par ordre d'importance.

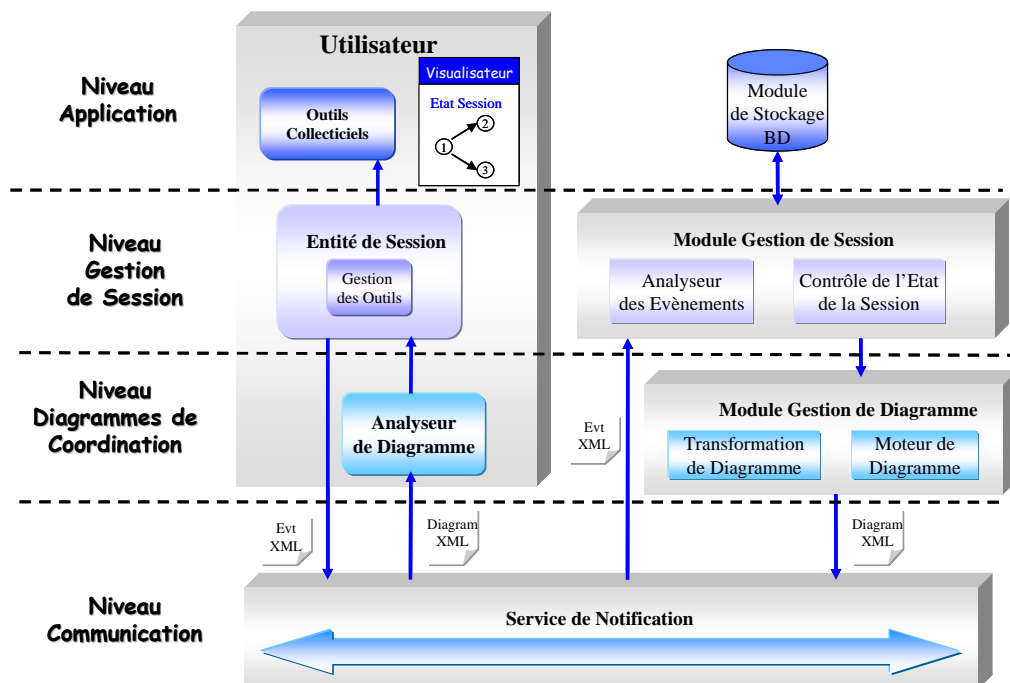


Figure 3.13. Architecture multi niveaux

La couche *Diagrammes de Coordination* implante le service de gestion de graphe de coordination qui manipule les graphes dynamiques.

Le module *Gestion de Diagramme* modifie et génère le diagramme de coordination associé à l'état de la session. Il se charge de toutes les transformations nécessaires dans ce diagramme. Ces transformations sont accomplies par les modules *Moteur de Diagramme* et *Transformation de Diagramme*. Le *Moteur de Diagramme* contient la configuration courante du diagramme de coordination et il fournit les fonctions de transformation élémentaires qui permettent de rechercher, ajouter, retirer ou modifier un nœud, un attribut d'un nœud, un lien ou un attribut de lien. Il permet aussi de traduire et de garder le diagramme de coordination courant dans un document XML. Le module *Transformation de Diagramme* contient l'ensemble des règles de transformation qui sont appliquées à la structure du diagramme de coordination courant lorsque l'action (ou événement) associée est activée. Ces règles sont réalisées à partir des fonctions de transformation élémentaires du module *Moteur de Diagramme*.

Le module *Analyseur de Diagramme*, un par utilisateur, traite les documents XML reçus et extrait l'information des diagrammes de coordination contenus dans les documents *Diagram.XML*. Avec cette information, chaque utilisateur associé à la session, peut accéder et mettre à jour son état local, et coordonner ses actions avec celles des autres participants du groupe.

La couche *Gestion de Session* contrôle les composants de la session. Elle traite les demandes venant des utilisateurs présents à l'intérieur de la session, et informe les membres de n'importe quel changement d'état de la session.

Le module *Module Gestion de Session* contrôle le déroulement de la session. Il analyse les événements reçus, il contrôle les changements de l'état de la session et transmet les informations sur ces changements. C'est un module actif qui dirige le déroulement de la session quand les participants sont en ligne. Il contient les modules *Analyseur des Événements* et *Contrôle de l'Etat de la Session*. Le module *Analyseur des Événements* reçoit les événements à traiter dans le format XML, par l'intermédiaire des documents *Evt.XML*. Ces événements proviennent des utilisateurs actifs ou du module *Gestion de Session*. Le module *Contrôle de l'Etat de la Session* se charge de stocker l'état courant de la session. Il maintient aussi l'historique des événements de session survenus. Il informe finalement les membres du groupe de tout changement dans l'état de la session.

Le module *Entité de Session*, un par utilisateur, implante un ensemble de mécanismes pour la participation des membres à la session et pour gérer leurs changements de rôle. Il permet d'informer le service de chaque changement dans l'état local respectif de chaque utilisateur. Il informe les utilisateurs des changements survenus dans l'état de la session. Le sous module *Gestion des Outils* génère les commandes exécutées par les outils collecticiels. Il transforme ainsi le contrôle abstrait des outils, donné par le niveau coordination sous la forme du diagramme de coordination, en un contrôle réel sous la forme de commandes pour la couche application.

La couche *Communication*, la plus basse, sert à l'échange des messages pour les couches de plus haut niveau. Tous les messages transmis et reçus par ce service codent des informations de contrôle, échangées en rafale lorsqu'un changement se produit dans la session. Ils représentent des informations complexes, typées et structurées dans des documents XML de quelques centaines d'octets. Pour échanger de façon correcte ces

messages de contrôle, les communications entre les utilisateurs et les modules gestionnaires nécessitent un service de diffusion fiable et ordonné, implanté par le *Service de Notification*.

La couche *Application* représente le niveau le plus haut de notre architecture. Elle contient l'ensemble d'*Outils Collecticiels* qui sont utilisés pendant la session, le *Visualisateur* de session en ligne et le *Module de Stockage BD*.

Chaque utilisateur présent durant la session collaborative synchrone se sert des *Outils Collecticiels* pour réaliser le travail de groupe. Chacun possède aussi un *Visualisateur* qui lui montre la configuration courante de la session. Ce *Visualisateur* est un navigateur Web classique, qui supporte XML et XSL. Ceci est un fort avantage lié à XML : une interface Web générique est utilisée directement pour visualiser la session, sans qu'il ne soit nécessaire de développer de nouveau logiciel ou outil spécifique qui se charge d'accomplir cette fonctionnalité.

Le *Module de Stockage BD* contient toute l'information de la session, stockée au format XML de façon persistante.

5.2.2 Correspondance session/coordination

Le service de gestion de session traite les événements de la session et maintient l'état de la session durant son évolution dans le temps.

Le service de coordination fournit un service de diagrammes de coordination générique, représenté en XML. Les événements que reçoit ce service sont associés à des règles de transformation de la structure du diagramme de coordination [DRIR00].

Lors de l'arrivée d'un événement de session, l'événement de coordination associé est créé, puis le service de coordination est activé. La couche de coordination sélectionne la règle de transformation qui correspond à cet événement de coordination. Elle effectue ensuite, sous l'action de cette règle, la transformation de diagramme de coordination courant.

De façon inverse, lorsque le diagramme de coordination courant est modifié, la couche de coordination notifie à la couche supérieure de gestion de session les changements survenus dans la structure du diagramme. La couche session sélectionne alors les actions de session qui sont associées à ces modifications. Ensuite, elle exécute toutes ces actions. Quelques exemples de telles actions sont : l'activation, la désactivation ou le changement de configuration de certains outils.

Les deux services principaux de notre architecture, le service de plus haut niveau gestion de session et le service générique de coordination doivent être mis en relation pour que leur évolution conjointe se déroule correctement. Une correspondance biunivoque doit donc être établie entre les événements de session et les événements de coordination associés aux règles de transformation de graphes, pour réaliser correctement le lien entre les services de session et de coordination.

5.2.3 Exemple

Pour clarifier notre description, nous prenons l'exemple suivant : nous définissons une session, dont l'identificateur est *s1*, avec un ensemble d'utilisateurs connectés en anneau. Chacun dialogue avec le suivant en utilisant un outil de dialogue textuel appelé « chat ».

Dans la correspondance biunivoque établie, l'événement de session *ajouter un utilisateur* « *joinSession* » est associé à la règle de coordination identifiée par « *joinSession_rule_id* ». Le

tableau 3.3 nous montre l'effet de cette règle sur une partie du graphe de coordination, entre deux nœuds X et Y.

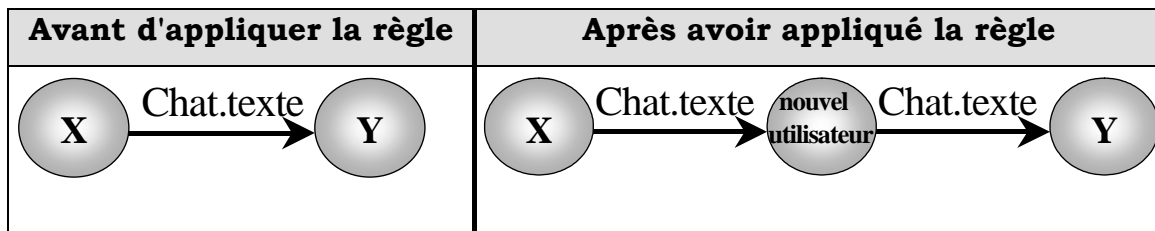


Table 3.3. Règle de transformation *joinSession_rule_id*

Le tableau 3.4 montre l'application de cette règle et son influence parallèle sur le diagramme de coordination courant. A l'instant t_0 , seuls les utilisateurs 1 et 2 sont présents et sont connectés en anneau. A l'instant t_1 , l'utilisateur 3 souhaite se joindre à la session. Il effectue sa demande au service de gestion de session en créant l'événement *joinSession(s1, 3)*. Au niveau coordination, la règle de transformation « *joinSession_rule_id* », qui a été associée au préalable à l'événement *joinSession*, est activée : le nœud 3 est ajouté entre les nœuds 1 et 2, et la nouvelle structure du diagramme de coordination forme un anneau entre les nœuds 1, 2 et 3. De la même façon, l'utilisateur 4 souhaite se joindre à la session à l'instant t_2 . Il effectue sa demande au service de gestion de session en créant l'événement *joinSession(s1, 4)*. Au niveau coordination, la règle de transformation « *joinSession_rule_id* » est activée, le nœud 4 est ajouté et le nouveau diagramme de coordination forme un anneau entre les quatre membres présents.

Temps	t_0	t_1	t_2
Événement de session		<i>joinSession(s1, 3)</i>	<i>joinSession(s1, 4)</i>
Règle de transformation appliquée		<i>joinSession_rule_id</i>	<i>joinSession_rule_id</i>
Diagramme de coordination courant			

Table 3.4. Exemple d'application de la règle de transformation pour une entrée en session

Selon les différents types de session que l'on souhaite gérer dans le temps (hiérarchique, maître/esclave, membres avec les mêmes droits, etc.), les besoins demandés au service de coordination générique seront différents. En effet, une session hiérarchique ne fera pas appel aux mêmes règles de transformation qu'une session symétrique. A chaque fois, le format des règles, leur structure et leur codage seront différents. Cependant, il est possible de dégager un ensemble canevas de règles de transformations, obligatoire et associé à toutes les opérations de session. Ce canevas liste les propriétés que doit respecter toute règle de coordination qui a été associée à un événement de session particulier. Par exemple, toute règle de transformation associée à l'événement d'ouverture de session doit connecter tous les nœuds du diagramme de coordination courant.

La table 3.5 établit la correspondance entre les événements de session et le canevas de règles.

Couche Session	Couche Coordination
Événements de session généraux	Canevas de règles de transformation
openSession (id : session_id)	Connecte tous les nœuds du diagramme de coordination
closeSession (id: session_id)	Déconnecte les nœuds du diagramme de coordination
Événements des participants	Canevas de règles de transformation
joinSession (id: session_id; p: participant)	Crée un nœud et si la session est ouverte et le connecte au diagramme de coordination lorsque l'état du diagramme le permet
leaveSession (id: session_id; p: participant)	Détruit un nœud et ses liens lorsque l'état du diagramme de coordination le permet
Événements des outils	Canevas de règles de transformation
startTool (id : tool_id ; flow : dataflow_id)	Ajouter les liens étiquetés TOOL depuis le participant p producteur
stopTool (id : tool_id)	Stopper l'outil TOOL et supprime les liens contenant TOOL du participant p

Table 3.5. Événements de session associés aux canevas de règles de transformation

Dans notre architecture, la correspondance "événements de session - règles de transformation" se fait au niveau des gestionnaires. De façon plus précise, le sous module règles de *Transformation de Diagramme* contient une table qui garde les éléments en correspondance et qui maintient ainsi le lien biunivoque entre les événements de session et les règles de graphe.

5.3 Implantation

L'architecture de la figure 3.13 a été développée directement en JAVA 1.2, pour bénéficier des avantages de ce langage en terme de portabilité et de comptabilité avec les navigateurs. A cause de l'évolution technologique, nous n'avons pas utilisé la plate-forme Charis Actor Portfolio car elle n'est plus maintenue et parce qu'elle n'est pas compatible avec JAVA 1.2. Cette nouvelle réalisation, faite par Mme Rodriguez-Peralta, forme la version V3 du service de gestion de session initial. La partie décisionnelle, proposée sous forme de vote, pour modérer les changements de session, n'a cependant été implantée que partiellement, l'accent ayant été mis sur l'aspect architectural.

Le prototype décrit figure 3.14 est composé des paquetages suivants, organisés par niveaux.

Au niveau *Communication*, nous trouvons deux paquetages principaux :

- *Jsd* : Le paquetage Java Shared Data Toolkit 2.0 (JSDT) est un service de notification pour des applets et des applications JAVA [JSDT99]. Il autorise la diffusion multipoint et fiable de messages quelconques, au travers d'un canal, auxquels s'abonnent les applications émettrices et réceptrices au travers de l'interface sockets.
- *Notification_Service* : Ce paquetage réalise la diffusion mutipoint et fiable des messages XML.

Le niveau *Diagrammes de Coordination* se compose de trois paquetages :

- *Crd_analyser* : Ce paquetage contient l'analyseur de diagrammes de coordination qui a été implémenté en Xerces Java Parser 1.4.4 [XERC00]. Ce dernier contient deux types d'analyseurs, l'analyseur Document Object Model (DOM) pour récupérer la structure du document et l'analyseur Simple API for XML (SAX), pour récupérer les attributs du document XML. Ces analyseurs sont modulaires et configurables.
- *Transformation* : Ce paquetage contient l'ensemble des règles de transformation qui sont activées lors de l'arrivée d'un événement et qui, en réponse, réalisent les changements sur les diagrammes de coordination.
- *Crd_engine* : Ce paquetage implante le moteur de diagrammes de coordination. Il fournit les fonctions élémentaires pour la génération et la manipulation des diagrammes de coordination.



Figure 3.14. Paquetages pour l'implantation

Au niveau *Gestion de Sessions*, nous trouvons deux paquetages :

- *EventXML* : Ce paquetage code les événements reçus dans un format XML. Il procède à l'analyse et à la validation des événements manipulés. L'analyseur syntaxique des Événements XML est aussi implémenté avec Xerces Java Parser 1.4.4.
- *Toolscontrol* : Ce paquetage contient deux fonctions principales. (1) Pour chaque participant, il extrait du diagramme de coordination courant l'ensemble des outils actifs. (2) Il manipule les outils, c'est-à-dire qu'il ouvre et ferme les applications chez chaque utilisateur. Grâce aux relations de flux de données, les outils peuvent être configurés localement en mode producteur ou consommateur d'information, pour chaque participant.

Le niveau *Application* nous sert à bâtir les applications finales et les interfaces d'accès au service de Gestion de Session. Il se compose des deux paquetages suivants :

- *Applet* : Ce paquetage standard Java nous sert à afficher les résultats de l'exécution de code Java dans un navigateur Web.
- *Visualizatorweb* : Ce paquetage contient le code du visualisateur de session. Pour cette implantation, nous avons utilisé des applets Java, le langage XSL et le navigateur Internet explorer.

L'implantation réalisée est organisée en 35 classes. Elle a nécessité la production de 5000 lignes de code.

Cette réalisation a été valorisée dans le cadre du projet Européen Distributed System Engineering (DSE) [DSE02]. DSE avait pour objectif de réaliser un support logiciel pour l'ingénierie distribuée dans le domaine de l'espace. Il cherchait à promouvoir l'utilisation à grande échelle de plates-formes de technologie collaborative dans l'industrie européenne, qui supportent le cycle de vie complet de l'ingénierie des systèmes [DRIR01].

5.4 Bilan

Le développement d'un service de modèle dynamique (service de coordination), indépendant de son utilisation, constitue la principale caractéristique de cette contribution.

Le premier gain de cette approche architecturale est de maintenir l'indépendance entre une couche de coordination générique et son domaine d'utilisation. Il est ainsi possible de définir des styles de coordination (linéaires, hiérarchiques) réutilisables pour divers domaines, comme proposé par K. Drira [DRIR99], [ROBL02].

Un autre avantage de notre proposition est de faciliter l'évolutivité de notre système. En effet, du fait de notre approche en couches qui fournit clairement une séparation entre la coordination et son utilisation, il est possible de faire évoluer le service utilisateur (ici la gestion de session) sans modifier la couche coordination.

La séparation entre ces deux éléments : diagrammes de coordination et gestion de session, garantit la modularité de notre architecture ainsi que son adaptation potentielle à d'autres domaines qui seront associés à d'autres services de niveau supérieur. Dans ce cas, le service de coordination basé sur des diagrammes de coordination reste le même. Il suffit d'ajouter une nouvelle couche logicielle qui donnera une autre sémantique à la couche de diagrammes de coordination. D'autres services ont été conçus selon cette approche : un service de coordination d'objets distribués au dessus d'une plate-forme middleware CORBA [DRIR99] et un service d'édition partagée [ROBL02].

Cette démarche architecturale peut finalement s'inscrire dans une problématique plus vaste de proposition de services à haut niveau au-dessus d'un service basé sur des graphes dynamiques, au sein d'une méthodologie associée.

6. Synthèse

La table 3.6 résume de façon synthétique nos trois contributions.

Contribution	Gestionnaire de session	Support technologique	Cadre contractuel
Implantation formelle	Version V1	Estelle et C	National : projet TOPASE
Méthodologie de conception	Version V2	Statechart et JAVA	National : Appel d'offre Télécommunications du CNRS
Support architectural orienté service de modèle	Version V3	JAVA	Européen : projet DSE

Table 3.6. Synthèse de nos contributions

Chacune de ces contributions a porté sur un aspect lié à la conception et/ou à l'implantation de services et de protocoles de session et de coordination. Chacun de ces aspects a été illustré par la réalisation d'une version du service de gestion de session. Chaque version a fait appel à des supports technologiques différents et chacune a été valorisée dans un cadre contractuel qui lui est propre.

L'évolution de nos trois réalisations a été dirigée par une amélioration de la généricité de l'architecture. Nous sommes ainsi passés de modules de coordination pour les versions V1 et V2 à la réalisation d'un service de coordination générique pour la version V3.

De façon parallèle, sur le plan technologique, l'utilisation de langages et d'environnements de plus en plus portables et interopérables, associée à l'utilisation, la plupart du temps, de méthodes formelles, nous a permis un gain en termes de facilité et de qualité de conception des différentes versions.

Conclusion et perspectives

En conclusion, nous récapitulons les principales contributions et les principaux résultats auxquels nous sommes arrivés ces dix dernières années.

1. Conclusion

Le premier chapitre de notre mémoire a situé le contexte de nos travaux. Cette partie préambule a développé le canevas de recherche en trois classes, *Travail Collaboratif*, *Modèles*, *Services de Coordination* que nous avons proposées, et qui nous ont servi de fil conducteur pour ce mémoire. La figure 4.1 rappelle ce canevas.

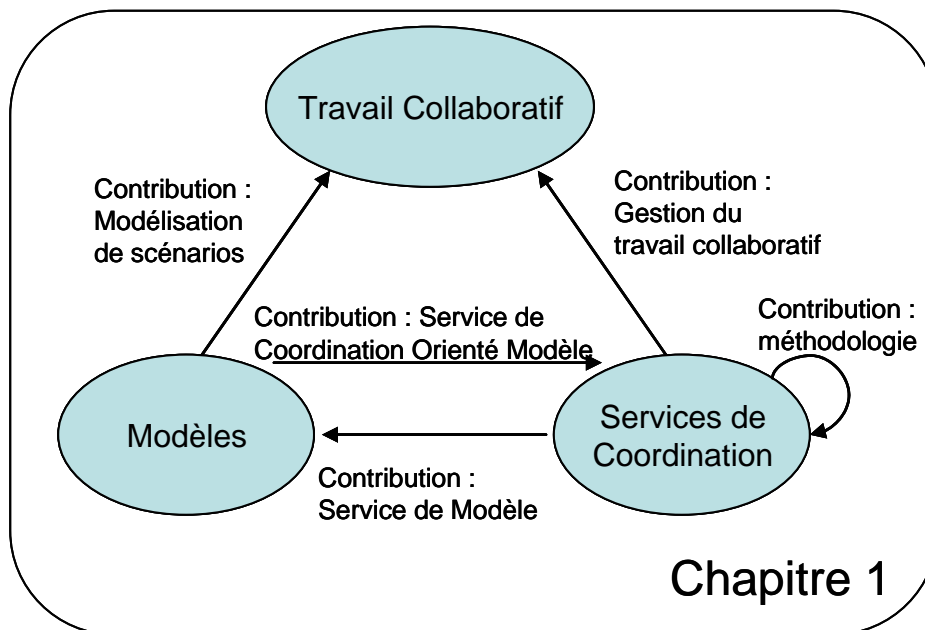


Figure 4.1. Canevas de notre projet de recherche

Pour détailler la classe *Travail Collaboratif*, nous avons tout d'abord défini les principaux termes et mots-clés qui font partie du travail collaboratif (Collaboration, Sessions, Interactions synchrones/asynchrones, Collecticiel). Par la suite, nous avons procédé à une classification des principaux outils collecticiels selon leur domaine d'interaction asynchrone ou synchrone, et nous avons analysé leur évolution au cours des dix dernières années, sur les plans fonctionnels, architecturaux et technologiques. Sur le plan fonctionnel, on peut noter une convergence entre les domaines asynchrones et synchrones qui sont traités à la fois par les plates-formes collecticielles courantes. Sur le plan architectural, on se dirige vers l'utilisation de composants téléchargeables ou de services intégrés au système. Sur le plan technologique, on remarque une forte convergence vers les technologies basées sur le Web.

Pour décrire la classe *Modèles*, nous avons ensuite analysé et classé un ensemble de modèles significatifs qui ont été utilisés pour représenter des sessions. En conclusion, aucune approche ne semble se détacher des autres pour manipuler des sessions synchrones.

Pour développer la classe *Services de Coordination*, nous avons ensuite analysé et classé les principaux gestionnaires de sessions existants. Ces systèmes convergent en terme de caractéristiques et de choix techniques. Ils manipulent tous des sessions non structurées, tous

permettent de rendre multiutilisateurs les applications sous leur contrôle, et certains d'entre eux sont des réalisations technologiques très avancées. Malheureusement, la faiblesse en terme de représentations de sessions se traduit par le peu de contrôle automatique qu'offre ces plates-formes sur les outils, ce contrôle étant effectué à la main par un utilisateur privilégié.

Le canevas de notre projet de recherche s'instancie sur la figure 4.2 par les principaux résultats que nous avons obtenus. De cette façon, la classe *Travail Collaboratif* est instanciée par les résultats *Téléenseignement/Téléingénierie*, la classe *Modèles* est instanciée par le résultat *Diagrammes de Coordination*, et la classe *Services de Coordination* est instanciée par le résultat *Gestion de Sessions*. Les flèches de la figure 4.2 donnent les contributions de chacun de ces résultats, vers une autre partie qui est utilisatrice de ces résultats.

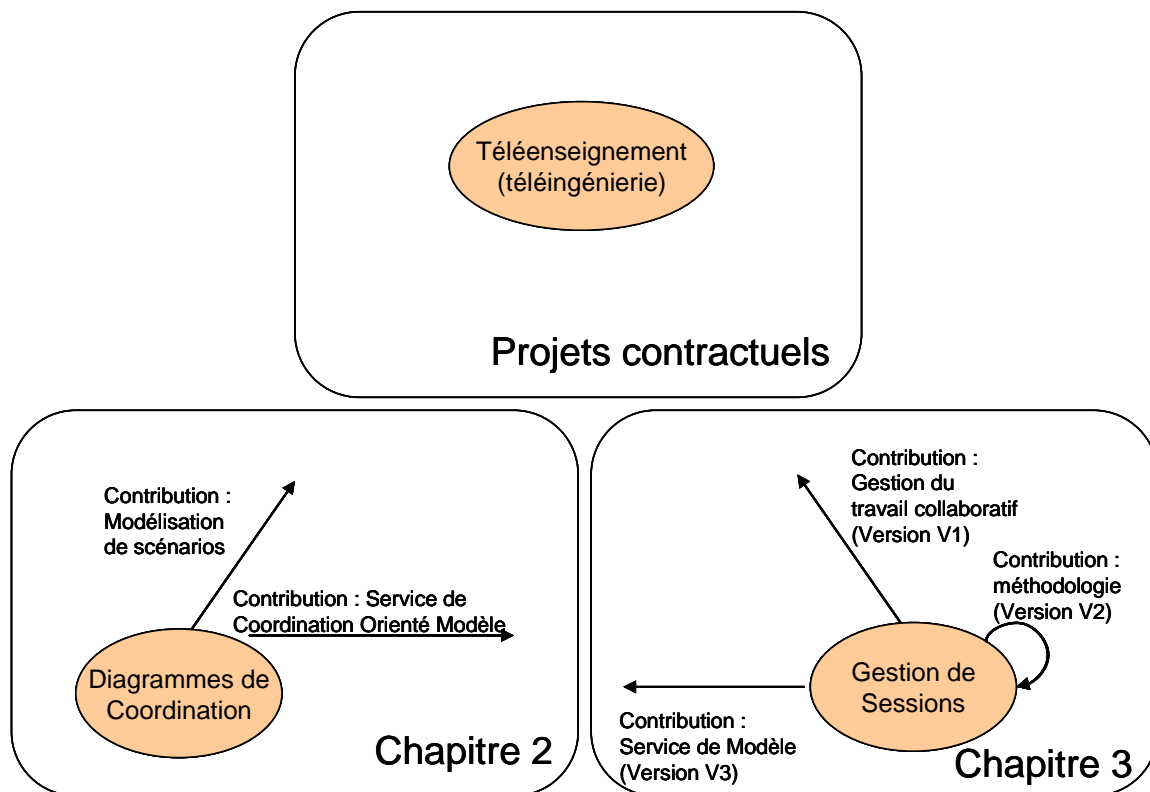


Figure 4.2. Résultats obtenus

Le deuxième chapitre a décrit le résultat *Diagrammes de Coordination* et a détaillé les deux contributions vers la *Gestion de Session* et vers le *Téléenseignement/Téléingénierie*. Ce chapitre a décrit l'approche orientée modèle pour représenter des sessions collaboratives. Il a présenté les trois éléments à co-gérer (utilisateurs, outils, données) pour ces sessions.

Pour détailler le résultat *Diagrammes de Coordination*, nous avons proposé un modèle de session collaborative synchrone (*Diagrammes de Coordination Dynamiques*) basé sur des graphes étiquetés. Ce modèle donne une représentation assez intuitive et facile à comprendre des éléments de session. Il peut profiter de la théorie des graphes en terme de vérification de propriétés.

La première contribution, appelée *Service de Coordination Orienté Modèle*, a porté sur l'aspect programmable par modèle, à destination de services de gestion de session. Elle a consisté en l'implantation en XML du modèle de *Diagrammes de Coordination*. Ce choix a facilité l'affichage de l'état de la session au travers de n'importe quel navigateur XML ; il a

renforcé de ce fait la prise de conscience du groupe par les utilisateurs. Cette implantation a finalement rempli le but principal recherché par notre contribution : obtenir un modèle programmable et utilisable de façon automatique par une couche logicielle de gestion de session. Ce travail a été valorisé dans le cadre du projet européen DSE sur l'Ingénierie Distribuée.

La deuxième contribution, intitulée *Modélisation de Scénarios*, a porté sur l'aspect descriptif de notre modèle. Elle a été illustrée par la spécification de plusieurs scénarios provenant du domaine du téléenseignement : formation de pilotes d'avion et d'agents de maintenance, puis accès à des laboratoires virtuels distribués pour des élèves. Nous avons pu ainsi évaluer, au travers du modèle proposé, la facilité de représentation de l'état de la session et de son évolution dans le temps. Le premier groupe de scénarios a été valorisé dans le cadre du projet national TOPASE sur la téléformation de personnels aéronautiques. Le second groupe a été proposé pour le projet européen Lab@Future sur un laboratoire scolaire pour des expérimentations avancées.

Le troisième chapitre a décrit le résultat *Gestion de Sessions*. De façon plus précise, il a été centré sur la conception d'un service de gestion de session programmable par modèle. Ce service gère l'évolution du groupe dans le temps et garantit que l'état courant de la session, donné par le modèle, correspond bien à l'état réel des utilisateurs connectés en session.

La première contribution, appelée *Gestion du travail collaboratif*, a porté sur le point de vue conception. Elle a mis en exergue l'apport des techniques de descriptions formelles en terme de qualité de conception pour la description et l'implantation automatique de services et de protocoles de gestion de session. La réalisation de ce service (version V1) a été intégrée au sein d'un prototype d'environnement support de travaux collaboratifs synchrones. Elle a été valorisée dans le cadre contractuel du projet national TOPASE qui portait sur la téléformation professionnelle.

La deuxième contribution, appelée *Méthodologie*, a porté sur le point de vue méthodologique en isolant deux composantes orthogonales des systèmes collaboratifs : un aspect structurel du système et un aspect comportemental des entités distribuées qui le composent. Nous avons proposé deux modèles interdépendants pour gérer ces deux aspects : le premier basé sur un modèle de transformations de graphes, le deuxième basé sur des modèles états/transitions. Cette deuxième contribution, illustrée par une deuxième réalisation (version V2) du service et du protocole de gestion de session, a été implantée dans un environnement acteur. Elle a été valorisée dans un cadre contractuel d'Appel d'Offre Télécommunications du CNRS.

La troisième contribution, appelée *Service de Modèle*, a porté sur le point de vue architectural. Afin d'augmenter la modularité et la réutilisabilité des éléments de notre système, nous avons proposé un nouveau découpage en introduisant une couche de modèle, générique et utilisable par des services de plus haut niveau. Ce service générique de coordination gère le modèle de graphe et est utilisé par des services de plus haut niveau et plus spécialisés, comme par exemple un service de gestion de session. Nous avons appliqué cette nouvelle architecture sur une troisième version de notre service de gestion de sessions. Cette troisième réalisation (version V3) a été valorisée dans le projet européen Distributed System Engineering.

2. Perspectives à moyen terme

Conformément à notre canevas de recherche, les perspectives porteront sur les classes *Modèles, Services de Coordination et Travail Collaboratif*.

2.1 Modèles

2.1.1 Groupes massivement multiutilisateurs

Les sessions que nous avons manipulées dans nos travaux sont de l'ordre d'une dizaine de membres. Une perspective porterait ainsi sur le traitement de sessions de taille plus conséquentes (par exemple de milliers d'individus ou plus). Plusieurs pistes semblent possibles :

- (i) définir des Sessions Hiérarchiques, en structurant les sessions ;
- (ii) utiliser des modélisations de plus haut niveau d'abstraction, en manipulant les entités au niveau des rôles et non explicitement au niveau de chaque utilisateur ;
- (iii) combiner les deux approches précédentes.

Ces pistes seront explorées dans le cadre de la thèse de Mme Sara Mota, thèse qui porte plus précisément sur l'étude et la modélisation de groupes de milliers d'utilisateurs communiquant de façon sécurisée.

2.1.2 Sessions spontanées

Les contributions que nous avons présentées ne portent que sur des sessions dynamiques explicites, qui sont préprogrammées par un concepteur de session. Une autre direction de recherche serait de considérer des sessions spontanées, dans lesquelles les membres s'auto organiseraient spontanément selon certains intérêts, buts ou critères mis en commun. Ces sessions, hautement dynamiques, admettent des compositions différentes et voient leur structure changer globalement dans le temps, au travers de politiques qui leur sont associées.

Cet axe, plus prospectif, permettrait de couvrir de façon plus complète l'espace de notre projet de recherche.

2.1.3 Multi sessions

En dernier lieu, dans un cadre à plus long terme, nous prévoyons de manipuler des sessions multiples, ces dernières pouvant être préprogrammées ou bien spontanées, et dans lesquelles les utilisateurs seront fixes ou nomades, dans des environnements Internet ubiquistes. Des problèmes d'appartenance en même temps à plusieurs sessions, ainsi que des problèmes de cohérence inter-sessions, se posent à ce niveau et restent à résoudre.

2.2 Services de coordination

La majorité de nos réalisations, et toutes celles présentées dans ce mémoire, a été centrée sur des services et des protocoles de gestion de sessions collaboratives synchrones. L'avantage de cette stratégie est d'offrir un cadre comparatif assez unifié pour illustrer les différentes contributions que nous avons apportées. Cependant, ce seul service apparaît comme trop restrictif, si l'on se situe dans une approche "boîte à outils de services pour des systèmes collaboratifs". Nous prévoyons donc d'étendre à moyen terme nos contributions dans les deux directions suivantes :

2.2.1 Déploiement automatique de composants

Dans les sessions que nous avons manipulées jusqu'à présent, les outils de communication, qui servent aux utilisateurs à réaliser le travail associé au groupe, étaient supposés disponibles et accessibles par tout nouveau membre se joignant au groupe. Du fait de l'augmentation de l'hétérogénéité au sens des réseaux, des machines et des systèmes d'exploitation, et du nomadisme des utilisateurs, cette hypothèse nous semble de moins en moins réaliste. Nous avons commencé à diriger nos recherches sur la proposition, la spécification, puis l'implantation d'un service de déploiement automatique d'outils et de composants, applicable à des sessions collaboratives. Au vu des besoins de ces sessions et des caractéristiques de chaque utilisateur, ce service se chargera de localiser les outils et les composants nécessaires à la session, puis de les rapatrier automatiquement avant tout démarrage de session. Ce travail fait partie de la thèse en cours de réalisation de M. Emir Hammami.

Contexte

Le déploiement des composants concerne l'ensemble des activités permettant de rendre les composants coopératifs disponibles sur les sites interconnectés dans des sessions collaboratives.

Bien que plusieurs systèmes de déploiement existent d'ores et déjà, les principes et concepts qui les sous-tendent considèrent le déploiement manuel et statique qui ne prend pas en compte les contraintes sur les sites et les environnements support. Avec le développement de l'informatique, de nouvelles exigences apparaissent :

- **Déploiement à large échelle.** Le nombre de sites formant une session peut croître rapidement ce qui rend impossible le déploiement manuel.
- **Evolution des sessions.** La structure de la session peut évoluer dans le temps. Les composants requis pour le nouvel état ne sont pas nécessairement déployés à l'avance.
- **Contextes variés.** Les sites de déploiement offrent des environnements d'exécution hétérogènes. Il est de plus nécessaire de prendre en compte les contraintes de chaque site (logicielles et matérielles).

Partant de ces constats, un support automatisé pour le déploiement est alors nécessaire pour identifier et déployer les composants coopératifs les plus aptes à répondre aux besoins de chaque site.

L'objectif est de concevoir une plate-forme générique pour le déploiement qui tient compte de ces exigences en permettant de générer des configurations de déploiement valides décrivant la distribution des composants à déployer sur les sites.

Caractérisation d'une plate-forme pour le déploiement

Toute plate-forme de déploiement doit contenir au moins les éléments regroupés dans la figure 4.3. Les *sites de déploiement* constituent les hôtes qui vont accueillir les composants. Ils sont liés par un réseau de communication. Ces sites forment la session collaborative et sont organisés selon un schéma de coopération défini à l'avance donnant l'état cohérent à atteindre, c'est-à-dire l'objectif du déploiement. Ce schéma décrit la *structure de la session*, en particulier les relations entre les sites. Ces derniers sont hétérogènes allant des ordinateurs de bureau vers les terminaux de poche en passant par les ordinateurs portables et ils offrent des contextes d'exécution variés (les contraintes sur les sites). Les composants sont développés par une tierce partie qui les rend disponibles à travers les *fournisseurs de composants*. Chaque composant requiert un ensemble d'exigences pour son fonctionnement

optimal. La plate-forme de déploiement forme le middleware entre ces différentes entités. L'algorithme de déploiement tente de trouver une configuration de déploiement valide qui respecte les exigences et les contraintes en tenant compte de la structure de la session et procède à la distribution réelle des instances des composants depuis les sites fournisseurs de composants vers les sites de déploiement.

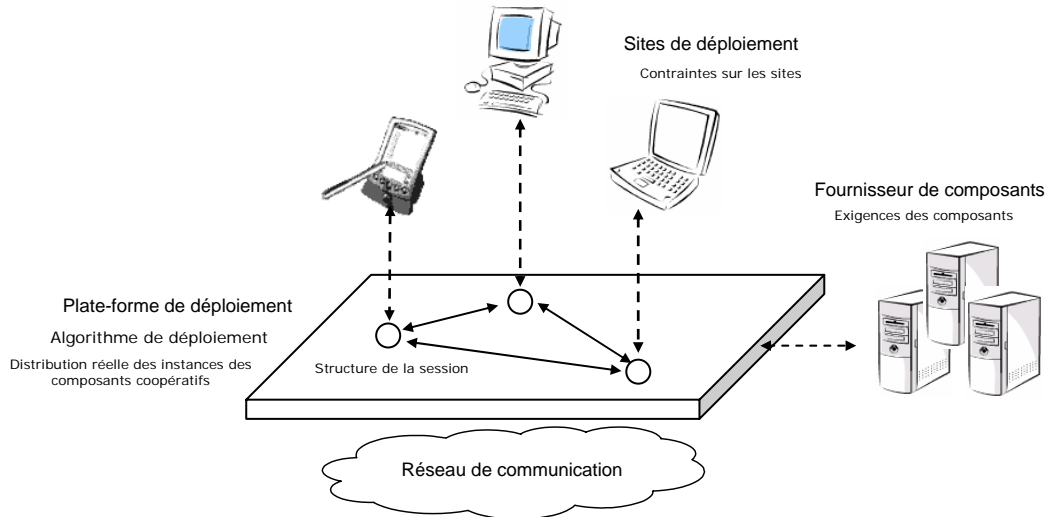


Figure 4.3. *Caractérisation d'une plate-forme pour le déploiement*

Dans le but d'être générique et permettre le déploiement automatique, une solution de déploiement doit :

- Opérer sur une variété d'environnements réseaux ;
- Fournir un modèle sémantique pour décrire un grand ensemble de composants ;
- Fournir un modèle sémantique pour les sites afin de décrire le contexte de déploiement ;
- Offrir un modèle définissant la structure de la session ;
- Permettre le déploiement initial et le déploiement dynamique suite aux événements extérieurs.

Modèles pour automatiser le déploiement

La session, les composants coopératifs et les sites sont fortement impliqués dans le processus de déploiement. Pour rendre le déploiement automatique, nous avons besoin de modèles sémantiques décrivant ces entités.

Le modèle de session, défini à partir des Diagrammes de Coordination, décrit de façon abstraite la topologie de la session. Ainsi, une session est modélisée par un graphe étiqueté orienté qui décrit tous les sites connectés avec leurs relations entre eux. Ce modèle sert de point d'entrée à notre service de déploiement. A partir de ce modèle, il est possible d'extraire l'ensemble des composants qui doivent être déployés sur chaque site. Cette approche orientée modèle offre deux avantages : (1) le premier est de donner une représentation à haut niveau des éléments de sessions de coopération dont la structure et l'évolution dans le temps seront pré-programmées par un concepteur de session. (2) Le second avantage est d'obtenir un modèle programmable qui puisse être interprété, compris et pris en compte par une couche logicielle de gestion de déploiement qui décharge les utilisateurs connectés aux sites d'une gestion en ligne faite à la main.

Il existe dans la littérature plusieurs définitions de composant. Szyperski définit un composant comme une unité de composition avec des interfaces contractuellement spécifiées et des dépendances de contexte entièrement explicites. Il peut être déployé de façon indépendante et il est sujet à une tierce partie de composition. Dans le cadre de notre travail, nous avons mis l'accent sur une classification des composants par rapport au traitement des flux de données. (1) Les *Composants Monolithiques Monomédias* ne produisent ou bien ne consomment qu'un seul flux d'un même type de donnée. (2) Les *Composants Composites Monomédias* traitent un flux en production et un flux en consommation tous deux de même type de donnée. (3) Les *Composants Composites Multimédias*, les plus complexes, manipulent plusieurs flux qui possèdent éventuellement des types de données différents.

Avant de déployer un composant coopératif qui répond aux types de composant requis sur un site, nous devons vérifier d'abord si le site qui va l'accueillir permet d'offrir les ressources et les services requis pour fonctionner correctement ou non. Il est donc nécessaire de fournir une description sémantique des exigences du composant. Le composant est décrit par un ensemble d'attributs et de propriétés encapsulées dans le descripteur de composant. Deux types d'attributs sont définis :

- Les attributs locaux pour caractériser les ressources exigées sur un site (ex. CPU, mémoire, espace disque) ainsi que les paramètres de Qualité de Service (persistance, sécurité et transaction).
- Les attributs distants utilisés pour vérifier l'interopérabilité avec les autres composants déployés sur les sites distants (ex. H.263, MPEG).

Les sites constituent l'environnement de déploiement des composants. Chaque site offre un contexte d'exécution varié. Le modèle de site doit contenir les informations sémantiques sur les caractéristiques logicielles du site comme le système d'exploitation, les services de Qualité de Service offerts ainsi que les caractéristiques matérielles du site telles que la vitesse du processeur, la taille de la mémoire, et la bande passante de la connexion au réseau. En combinant ces informations avec les exigences des composants coopératifs, il est possible de vérifier si un composant peut fonctionner correctement ou non et donc d'autoriser ou de refuser son rapatriement.

Algorithme de déploiement

Une brique principale de la plate-forme consiste en la conception de l'algorithme de déploiement. L'objectif de l'algorithme de déploiement est d'assurer la cohérence de la session c'est-à-dire de fournir à chaque site les composants coopératifs conformément au modèle de session défini par le graphe tout en respectant les contraintes locales sur les sites et les contraintes d'interopérabilité entre les composants.

Le déroulement de l'algorithme devrait nécessiter cinq étapes, comme le montre la figure 4.4. Partant d'un état incohérent où chaque site de la session ne possède aucun ou seulement une partie des composants nécessaires au travail, l'algorithme qui s'exécute sur chaque site, tente de trouver les composants adéquats et génère une configuration de déploiement valide. Cette configuration est utilisée pour rapatrier les composants vers les sites cibles.

Lorsque la session évolue, le déroulement de l'algorithme de déploiement se répète en prenant en compte le nouveau graphe de coopération décrivant le nouvel état de la session.

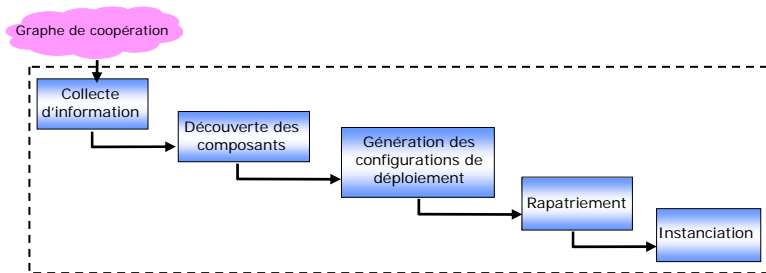


Figure 4.4. *Etapes de l'algorithme de déploiement*

La première étape *Collecte d'Information* consiste à énumérer le nombre maximal de types de composant monolithique à trouver sans tenir compte des différentes contraintes. Cette information est extraite à partir du graphe de session.

Etant donné le type de composant et le type de donnée, l'étape *Découverte des Composants* de l'algorithme consiste à découvrir les composants qui répondent à ces besoins. La découverte est effectuée de façon dynamique pendant la phase de déploiement. A la fin de cette étape, chaque site se retrouve avec une liste des descripteurs des composants candidats.

Un premier filtrage élimine les composants incompatibles avec le contexte d'exécution de chaque site. En combinant les exigences citées dans le descripteur de chaque composant découvert avec le modèle de site, l'algorithme définit des ensembles locaux qui couvrent la solution sur chaque site. Ces configurations ne sont pas toutes valides du fait que certains composants situés sur un site peuvent ne pas être interopérables avec les autres composants qui se trouvent sur les sites en relation avec eux. Un deuxième filtrage est donc nécessaire pour assurer la compatibilité distribuée. Il s'agit d'éliminer les ensembles non satisfaits par les voisins. En respectant les contraintes locales et d'interopérabilité, l'étape *Génération des Configurations de Déploiement* fournit l'ensemble des configurations de déploiement valides qui décrit la correspondance entre les composants capables de fonctionner correctement et les sites hôtes.

L'étape *Rapatriement* de l'algorithme consiste à choisir une configuration valide et à effectuer les rapatriements des composants vers les sites concernés conformément à cette configuration. Le choix de cette configuration est guidé par le fait que l'on cherche à minimiser le nombre de rapatriements à faire.

La dernière étape *Instanciation* de notre algorithme consiste à instancier les composants déjà rapatriés et à les démarrer.

2.2.2 Services de communications sécurisées de groupes

Dans nos travaux, le contrôle d'accès de la session se fait selon la configuration courante du diagramme de coordination et selon le vote des utilisateurs. Dans le cadre de sessions contenant de grands nombres d'utilisateurs nomades, il faut plus de sécurité. Nous compléterons ces mécanismes par des approches plus traditionnelles de gestion de clés de groupe qui pourraient, selon les besoins, assurer des fonctions d'authentification ou de confidentialité. Du fait de leur complexité, l'étude et l'analyse de ces services et protocoles de groupes sécurisés feront appel à des techniques formelles pour valider leurs propriétés en terme de sûreté, de vivacité et de robustesse. L'ensemble de ces études, plus prospectif, est actuellement effectué par Mme Sara Mota, dans sa thèse.

Contexte

Domaine d'études

La problématique étudiée par le projet SAFecast, projet du Réseau National de la Recherche en Télécommunications porte sur l'étude, la conception et la validation de fonctionnalités et de mécanismes garants de communications sécurisées à l'intérieur de groupes dynamiques. Ce projet, qui représente le cadre de nos recherches, porte sur des communications dans des réseaux de diffusion multicast de type Private Mobile Radiocommunication (PMR.) utilisés dans des zones d'intervention (incendies, attentats, situation de crise, etc.). Les utilisateurs communiquent à l'intérieur de leur groupe mais aussi entre des groupes. Du fait de la dynamique des groupes, les éléments de sécurité devront s'actualiser en ligne pour garantir la sécurité établie lors des spécifications du système.

La modélisation et la vérification de ce système doivent donc prendre en compte les trois éléments suivants : le premier élément est le développement d'une architecture compatible avec les besoins définis pendant l'étape d'analyse du système. Le deuxième élément contient le choix de la méthodologie de conception et des outils qui la supportent. A partir des deux éléments précédents, le troisième élément s'attache à la modélisation et à la vérification du système.

Contexte scientifique

La nécessité de systèmes sécurisés de communication pour des groupes d'utilisateurs pose le problème de la protection des données. Cette protection, qui s'appuie sur un chiffrement de l'information, doit garantir les propriétés d'intégrité et de confidentialité. Ce chiffrement utilise des algorithmes asymétriques ou bien symétriques. Les algorithmes asymétriques, qui utilisent une paire de clés publique et privée, sont bien adaptés à des échanges point-à-point. Cependant, leur utilisation dans des communications de groupes s'avère trop lourde au vu de la combinatoire des couples de clés et de leur charge de calcul. De plus, la complexité du chiffrement asymétrique compromet le passage à l'échelle. Ceci explique le choix des partenaires du projet SAFecast en faveur d'algorithmes symétriques à clé secrète qui utilisent une clé partagée (clé de groupe) à l'intérieur d'un même groupe de communication.

De manière classique, on se base sur l'algorithme de Diffie et Hellman. Cet algorithme cryptographique symétrique pour la gestion de clés distribué et contributif constitue le meilleur choix.

Approche utilisée

L'approche d'analyse que nous nous proposons de suivre s'appuie sur une méthodologie qui utilise UML 2.0 (Unified Modeling Language) comme langage de modélisation. Tout d'abord, des diagrammes de contexte sont utilisés pour définir les acteurs qui participent au système. A partir d'eux, nous définissons les fonctionnalités qui composent le système, en s'appuyant sur des diagrammes de cas d'utilisation. Ensuite, cette description des cas d'utilisation nous servira à extraire des ensembles de scénarii décrits sous la forme de Diagrammes de Séquences Système.

Pour la phase de conception, nous utiliserons le profil UML TURTLE (Timed UML and RT-LOTOS Environment) supporté par l'outil TTool. Le modèle du système est complété par des observateurs en charge de guider la vérification du protocole. Cette vérification s'applique à des systèmes bornés. Elle repose sur une analyse énumérative suivie d'une minimisation du graphe d'accessibilité qui débouche sur un automate quotient plus facile à interpréter lorsqu'il

s'agit de montrer que la conception satisfait les propriétés du système identifiées en phase d'analyse.

Fonctionnalités du système

Les fonctionnalités et rôles que nous avons isolés sont mis en œuvre lorsque la session est active, c'est-à-dire lorsque les membres sont présents dans le groupe et qu'ils communiquent entre eux. Nous n'avons pas décrit les fonctionnalités nécessaires en dehors des sessions (par ex. la création des droits des membres et des types de groupes) car elles ont moins de contraintes en terme d'interactivité et de garanties de sécurité.

Pendant une session active, le système se décompose en trois grandes classes de fonctionnalités : (i) celles de sécurité, (ii) celles intra-groupes et (iii) celles inter-groupes.

Les *Fonctionnalités de Sécurité* supportent les mécanismes de contrôle d'accès et de confidentialité des échanges dans les groupes pendant la durée d'une session, y compris lors de son évolution dynamique. Ces mécanismes sont basés sur des clés de groupe symétriques. Les principales fonctions sont le renouvellement et la distribution des clés nécessaires.

Les *Fonctionnalités Intra-Groupes* gèrent les groupes de communication et traitent de la dynamique à l'intérieur d'un même groupe de communication. Les fonctionnalités élémentaires sont l'entrée d'un membre, sa sortie, sa descente ou sa montée dans la hiérarchie, son exclusion et sa reconnexion.

Les *Fonctionnalités Inter-Groupes* traitent des relations entre plusieurs groupes de communication. Elles font intervenir deux ou plusieurs groupes lors de leur exécution. Ces fonctionnalités comprennent les opérations fondamentales de fusion et de séparation des groupes. Les groupes fusionnés ou séparés pourront posséder des structures hiérarchiques semblables ou différentes. Dans ce dernier cas, il faudra passer par une structure hiérarchique commune.

Chacune des classes de fonctionnalités isolées sera représentée par un diagramme de cas d'utilisation.

Architecture proposée

L'architecture de modélisation du protocole s'appuiera sur les fonctionnalités précédemment identifiées. Cette architecture, organisée selon une approche en couche, servira à l'analyse et à la vérification des propriétés de l'ensemble du système.

Quatre niveaux, développés sous la forme de couches indépendantes, seront étudiés afin de composer l'architecture de modélisation (Figure 4.5).

La couche *Trusted Medium* se compose de deux sous-niveaux.

Au niveau le plus bas, le sous-niveau *Medium (PMR)* offre les services de communication élémentaires de la couche physique radio PMR, et se charge de la diffusion des messages. Trois types de services ont été définis dans ce médium : un service de communication point-à-point, un service multipoint (de 1 vers N) et un service de diffusion (1 vers tous).

Le sous-niveau *Security Operators (SO)*, offre les actions élémentaires de sécurité qui garantissent l'intégrité, la confidentialité et le non rejeu. Elle gère les opérations de sécurité liées aux communications. On trouve des opérations de Hashing pour garantir l'intégrité des

données, des opérations de Chiffrement qui servent à la confidentialité et d'empêchement de rejeu avec l'utilisation des index.

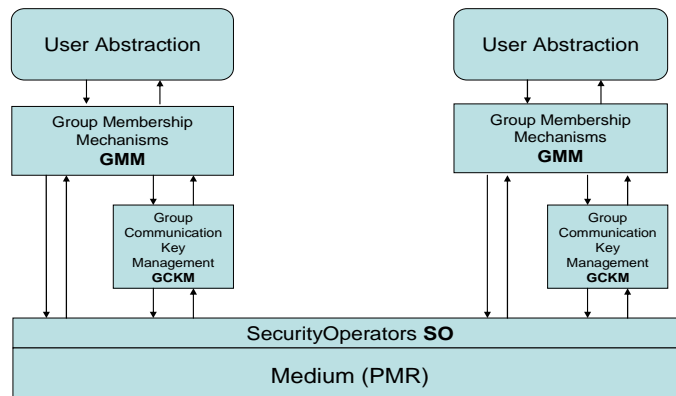


Figure 4.5. Architecture de modélisation

La couche *Group Communication Key Management (GCKM)* gère les opérations liées à la sécurisation des groupes, opérations liées aux clés de session.

Le service de renouvellement de clés inclut les deux sous-services : la génération et la distribution des clés. Les clés de session générées sont destinées à chaque classe ou hiérarchie d'un groupe de communication. Ce groupe peut aussi être un groupe fusionné. La distribution des clés se fait selon trois cas : entre des chefs d'un groupe, à l'intérieur d'un groupe, et dans un groupe fusionné.

Le renouvellement des clés doit se faire en maintenant ainsi l'intégrité du système vis-à-vis des changements des rôles et des entrée/sorties des utilisateurs.

La couche *Group Membership Mechanisms (GMM)* se charge de gérer les groupes en contrôlant leur structure, leur évolution et leur dynamique. Elle intègre l'ensemble des fonctionnalités inter groupes et intra groupe qui affectent la composition des groupes. Les services donnés dans cette couche (join, leave, downgrade, upgrade, exclusion, merging, splitting) reprennent la liste des fonctionnalités intra et inter groupes identifiées. Le fonctionnement de ces services se fait en relation avec les mécanismes de gestion des clés de session sous-jacents et l'utilisation des certifications qui donnent des droits aux membres.

Travaux en cours

A court terme, nous devons poursuivre la modélisation du système avec le profil UML TURTLE. Cette modélisation sera faite de façon à minimiser le plus possible la taille de l'espace des états du système complet.

A moyen terme, nous procéderons à l'évaluation et à la vérification des contraintes temporelles du système, en nous appuyant sur une méthodologie qui utilise des observateurs. Des scénarii généraux seront utilisés pour vérifier des propriétés qui s'appliquent à l'ensemble du système.

2.3 Travail collaboratif

Au niveau contractuel, nous souhaitons étendre nos contributions à des groupes indifférenciés d'utilisateurs réels et d'agents logiciels fonctionnant sur des machines autonomes. Cette contribution s'inscrit sous l'appellation services ubiquistes de type machine à machine.

Le but des services réseau de type machine à machine (M2M) ubiquistes est d'étendre l'accès des services réseau à des machines, qui peuvent être des capteurs et des actionneurs complexes, et qui sont connectées au travers de réseaux IP quelconques. Les services M2M englobent des services de communication tels que la capture, la transmission et le traitement d'information qui servent à établir un système hautement interactif en mettant en relation les machines concernées de façon dynamique. De tels systèmes doivent finalement être intégrés à un système logiciel de gestion M2M, qui peut ou non requérir une interaction humaine.

Nos prévoyons d'étudier les systèmes M2M dans des domaines d'applications tels que la surveillance et le contrôle d'infrastructures, la maintenance et le contrôle de machines mobiles, les systèmes mobiles télématiques sans fil avec de multiples clients agissant comme consommateur d'information.

De façon plus précise, nous souhaitons focaliser nos contributions sur les besoins architecturaux de tels systèmes, en définissant une architecture conceptuelle pour de tels services :

- La plate-forme proposée devra supporter l'intégration dynamique de services M2M fournis par divers dépositaires.
- Elle devra spécifier des infrastructures de services génériques, pour que les services M2M puissent être utilisés dans différents domaines d'applications.
- L'accès aux services M2M doit rester générique et commode pour tout type d'utilisateur, et plus particulièrement les utilisateurs mobiles.
- Les communications M2M devront se faire au-dessus de tout type de réseau supportant une connectivité IP, depuis des réseaux sans fils et filaires locaux jusqu'aux longue distance.

3. Perspectives générales

Nos perspectives de recherche à plus long terme s'inscrivent dans l'axe stratégique des Architectures et des Protocoles Auto- adaptatifs. Nous présentons par la suite les principaux points sur lesquels porteront nos contributions futures, avec leurs liens et intégration dans cet axe.

3.1 Thématique

Ces dernières années, les évolutions de l'informatique et des télécommunications ont conduit à une modification radicale du paysage de la communication informatique, et en conséquence de l'Internet et de ses services. Les applications sont maintenant à la fois multimédias, multi utilisateurs et coopératives. Les réseaux sont désormais à haut débit et ont évolué vers des communications sans fil, en particulier ad hoc ; les types de terminaux et de point d'accès sont multiples et les services offerts par les opérateurs de l'Internet multi domaines se généralisent.

Pour le futur, la communication devrait être diffusable beaucoup plus largement par une utilisation massive de processeurs embarqués, de type capteurs ou actionneurs. L'Internet sera

alors extrêmement diffus, mobile et hétérogène, au moins à sa périphérie, c'est-à-dire où se situeront les utilisateurs.

L'horizon de ces avancées est celui d'un **Internet ambiant, ubiquitaire, pervasif et intelligent**, au travers duquel :

- l'utilisateur sera mobile et donc susceptible d'utiliser des ressources en différents endroits, sans devoir considérer son environnement courant ;
- la communication sera diffusable partout, avec des caractéristiques différentes en fonction des ressources disponibles ;
- le réseau et ses services seront à même de satisfaire, voire d'anticiper, les besoins des utilisateurs, en tirant au maximum partie des capacités de l'environnement (incluant les utilisateurs eux mêmes) dans lequel ceux-ci évolueront à chaque instant.

Dans ce nouveau contexte, l'application au support des activités de groupe dans l'Internet multi domaines et ambiant est importante pour plusieurs raisons :

- d'une part, elle cible un domaine applicatif qui influencera fortement le déploiement et l'utilisation des réseaux, l'Internet en particulier, et où il y a une compétition et un effort de recherche internationaux pour anticiper et maîtriser cette orientation ;
- d'autre part, il y a une forte demande des utilisateurs pour des solutions de communication et de coopération dans différents domaines d'activité (coopération humaine, coopération de composants logiciels, ...), qui, malgré leurs différences, peuvent partager la plupart des solutions présentées dans le cadre de cette thématique de recherche.

Les enjeux scientifiques spécifiques aux solutions proposées couvrent la maîtrise des technologies relatives aux réseaux de capteurs, à la conception d'architectures et de protocoles pour la communication et la coopération, en tenant compte des contraintes d'adaptabilité, de configurabilité dynamique et de forte interactivité. Du point de vue de la coordination, il s'agit de maîtriser la conception des solutions qui couvrent plusieurs niveaux fonctionnels tout en garantissant la cohérence et la stabilité des politiques d'adaptation intra et inter niveaux dans ces architectures distribuées.

3.2 Contexte et positionnement

La communication et la coopération en environnement ambiant sont des domaines d'investigation récents et nécessitent des compétences à la fois en ingénierie des protocoles, en réseau, et en technologies logicielles.

Les défis à relever portent sur la conception de systèmes communicants coopératifs, sur les technologies réseaux sous-jacentes et sur leurs applications.

3.2.1 Initiatives internationales

Différentes initiatives, projets et groupes de travail adressent tout ou partie de ces défis, parmi lesquels les initiatives Wireless-World Initiative (WWI) et Internet 2, ainsi que le projet américain OXYGEN du MIT. On trouve aussi les organismes de standardisation, en particulier l'IETF, au travers des groupes AVT, SCTP, DCCP et NSIS.

3.2.2 Initiatives européennes

Au niveau européen, le contexte stratégique se situe au niveau de plusieurs programmes du 6^{ème} et 7^{ème} PCRD, qui couvrent :

- les logiciels et les services (*software and services*) ;
- les environnements de travail collaboratifs (*collaborative working environments*) ;
- la simulation de propriétés émergentes dans les systèmes complexes (*simulating emergent properties in complex Systems*) ;
- la communication, le calcul et les technologies logicielles (*communication, computing and software technologies*) ;
- les plates-formes des réseaux de la recherche (*research networking testbeds*).

Parmi ces programmes, nous pouvons mettre en avant les projets européens, WS-Diamond, CONTEXT, End-to-End Reconfigurability (E2R), Winner, Ambient Networks, et EuQoS.

3.2.3 Initiatives nationales

Au niveau national, le contexte stratégique se situe :

- d'une part au niveau du pôle de compétitivité Aéronautique, Espace et Systèmes Embarqués ;
- d'autre part au niveau des projets RNRT et ARA de l'Agence Nationale de la Recherche, dont les thèmes portent respectivement sur : la sécurité, les systèmes embarqués et l'intelligence Ambiante (ARA SSIA) ; la sécurité, la mobilité et l'intelligence ambiante (RNRT).

Au niveau animation scientifique, le pôle ResCom du GDR Architectures, Systèmes et Réseaux (ASR) a pour objectif, dans cet horizon, de fédérer des thématiques de recherche en particulier liées aux aspects algorithmiques et structurels, aux outils de modélisation et d'évaluation de performance et aux aspects protocolaires. L'action "Adaptation dynamique aux environnements d'exécution", animée par F. André dans le cadre du pôle Grille, Système et Parallélisme (GSP) du GDR ASR, vise à fédérer les initiatives autour du terme de l'auto-adaptation.

3.3 Objectif scientifique

Nos perspectives s'inscrivent dans un double contexte :

- un **contexte applicatif étendu, mobile et évolutif** par ses cas d'applications et d'utilisation : applications de type ingénierie coopérative ou opération d'intervention d'urgence ; utilisation de différents types de terminaux et de points d'accès (filaire/sans fil), besoin en qualité de service (QoS) différents d'un cas à l'autre, besoin en sécurisation des communications, ...
- un **contexte réseau hétérogène, ouvert et à ressources variables**, interconnectant des logiciels à architectures complexes, sans contrôle du nombre de participants aux activités supportées. Les ressources correspondent aux équipements, en particulier aux serveurs et aux routeurs, ainsi qu'aux liens d'interconnexion, à leurs classes de service et à leur bande passante.

L'objectif scientifique de nos perspectives porte sur la qualité des services de communication et de coopération, qu'il s'agit de maintenir au plus haut degré en tenant compte d'une dynamique du contexte à la fois au niveau utilisateur (mobile et coopérant) et au niveau du réseau (aux ressources variables).

De façon plus précise, les objectifs se déclinent en :

- **un traitement des activités de communication et de coopération à très grande échelle** ; depuis sa création, le succès de l'Internet résulte de son aptitude à connecter des grands nombres d'utilisateurs et son futur sera également conditionné par cette capacité ;
- **l'auto adaptabilité des systèmes de communication et de coopération**, pour satisfaire les différents cas d'utilisation des futures plates-formes de coordination qui seront composées d'entités autonomes aptes à communiquer de façon spontanée et sensible au contexte de communication et de coopération.
- **la compréhension et l'analyse des implication coopératives des technologies basées sur les réseaux sans fil et ad hoc**, qui sont de plus en plus présentes dans les activités quotidiennes, et qui conditionneront le développement des futures plates-formes de communication ;

Plusieurs fonctions de gestion seront abordées, au niveau :

- de l'évolution des contextes ;
- de la coopération ;
- de la mobilité au niveau applicatif et réseau ;
- de la maîtrise de la continuité des services et de leur qualité.

3.4 Méthodes et approches

Les approches reposeront sur l'adaptabilité des protocoles de communication et de coopération aux contraintes de niveau réseau et aux exigences du niveau des applications. Elles exploreront en particulier l'adaptabilité des architectures intégrant les deux niveaux, c'est-à-dire assurant un contrôle et une cohérence des solutions d'adaptabilité sur chaque niveau.

L'adaptabilité implante la sensibilité au contexte : il s'agit par exemple d'adapter la distribution des composants de protocoles aux contraintes du réseau, ou d'adapter les canaux de communication à la mobilité des utilisateurs et aux changements dans l'espace de coopération.

L'étude de l'adaptabilité soulève des problèmes relatifs aux contraintes d'ordre structurel (architectures et topologies) et comportemental (fonctions et protocoles). Les méthodes associées concernent les techniques de conception des systèmes coopératifs.

Ces perspectives se déclinent en trois thèmes de recherche :

- le premier, pour lequel nous aurons une activité moins centrale, cible les études sur les nouvelles technologies réseaux pour les futures applications de l'Internet multi domaines et ambiant.

- les deux derniers, dans lesquels nous aurons une participation active, couvrent l'auto adaptabilité au travers des protocoles configurables, des architectures dynamiques et des politiques de coordination associées ;

3.4.1 Conception et émulation d'architectures multi domaines

Ce thème cible l'étude de nouvelles plates-formes réseau, sous jacentes aux futures applications de la communication et de la coopération ambiantes. Deux directions principales sont explorées, les réseaux sans fil et la signalisation en environnement multi réseaux :

- pour les réseaux sans fil, il s'agit particulièrement d'étudier la connectivité, l'évaluation des performances et la continuité de la QoS au sein de ces réseaux ; des solutions seront élaborées à l'échelle du réseau isolé puis connecté à l'Internet ;
- pour la signalisation, l'étude porte sur l'élaboration de protocoles pour la réservation de ressources en environnement sans fil et multi-réseaux. Ces protocoles doivent être indépendants des politiques de gestion de la QoS au sein de chaque réseau et doivent pouvoir être interfacés avec tout type de signalisation applicative.

Les besoins couverts par cette partie sont la gestion de l'hétérogénéité des réseaux, la gestion de l'évolution des contextes et la gestion de la mobilité au niveau applicatif et réseau.

Il s'agit pour nous d'évaluer les impacts de ces nouvelles technologies et solutions à QoS sur les contraintes, les fonctionnalités et les possibilités de comportement coopératifs, afin d'en garantir un certain ensemble de propriétés.

3.4.2 Architectures dynamiques et protocoles auto configurables

Ce thème concerne :

- l'élaboration de protocoles de coopération configurables dynamiquement pour la gestion de sessions coopératives spontanées ou prédéfinies ; ceci inclut la découverte dynamique de partenaires et de services dans des architectures distribuées (peer-to-peer, web services, middleware), la gestion de sessions et de partage ;
- la conception d'architectures dynamiques à la fois dans une couche de protocole, en particulier la couche Transport, et entre plusieurs couches, de façon locale ou distribuée ;
- l'élaboration de protocoles de communication configurables dynamiquement pour offrir une Qualité de Service maximale, compatible d'une part avec la nature des informations échangées et les priorités relatives des communications, et d'autre part avec les performances du réseau.

Les besoins couverts par ce thème sont la gestion la gestion de l'évolution des contextes, de la coopération, de la mobilité au niveau applicatif et réseau et de l'hétérogénéité des réseaux.

Il s'agit ici de suivre et de représenter l'évolution des différentes parties des architectures.

3.4.3 Politiques de coordination pour l'adaptabilité

Ce thème cible l'élaboration de politiques d'adaptabilité pour les couches communication et coopération, selon une approche de coordination orientée modèle, le modèle de départ étant défini par des grammaires de graphes enrichies. Ces politiques sont guidées par les exigences applicatives et les contraintes du réseau, et agissent d'une part sur la (re)configuration des

protocoles, et d'autre part sur la transformation d'architectures (déploiement dynamique de composants et/ou de protocoles de coopération et de communication).

Les besoins couverts par ce thème sont la gestion de l'évolution des contextes, de la coopération et de la mobilité au niveau applicatif et réseau.

Ce thème devra permettre de coordonner des politiques différentes, d'en assurer la cohérence et de fournir les approches et mécanismes qui permettront de les implanter.

Références

Chapitre 1

- [AALS02] W. van de Aalst and K. van Hee. Workflow management. Models, methods and systems. N°ISBN 0-262-01189-1, MIT Press, 2002.
- [ARIS03] Séminaires X-Aristote. <http://www.aristote.asso.fr/>
- [BECA98] L. Beca, G.Cheng, G. C. Fox, T. Jurga, K. Olszewski, M. Podgorny, P. Sokolowski and K. Walczak. Java enabling collaborative education, health care, and computing. *Concurrency: Practice and Experience*, Volume 9, Issue 6 , Pages 521 – 533, John Wiley & Sons, Ltd, Décembre 1998.
- [BENF91] S. Benford, A. Bullock, P. Harvey, H. Howidi, A. Shepherd and H. Smith. The Grace project. Group Communication in an Open Systems Environment. Final Report. Communications Research Group. Nottingham University. Nottingham NG7 2RD. Septembre 1991.
- [BENF92] S. Benford, H. Smith et A. Shepherd, A. Bullock and H. Howidi. Information Sharing Approach to CSCW: The Grace Project. *Computer Communications*, 15(8):502-508, Octobre 1992.
- [BENF93] S. Benford and J. Palme. A Standard for OSI Group Communication. *Computer Networks and ISDN Systems*, 25(8):933-946, Mars 1993.
- [BIGN89] C. Bignoli and C. Simone. AI Techniques for Supporting Human to Human Communications in CHAOS. . In *Studies in Computer Supported Cooperative Work : Theory, Practice and Design*, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991.
- [BSCW04] Basic Support for Collaborative Work Web site. http://www.bscw.de/index_en.html
- [CHIV03] Sridhar Chirravuri, Mike Meeks, Matt Rutherford and Ken Taylor. Lotus Notes synthesis. University of Georgia, Athens GA. – <http://www.cs.uga.edu/~chirravu/notes/>
- [CAMB03] First use of Webcam. The Trojan Room, University of Cambridge, UK. – http://radio.cbc.ca/programs/asithappens/STEAM/webcam_030701.html
<http://www.cl.cam.ac.uk/coffee/coffee.html>
- [CAMP03] Portail du Campus de France – <http://www.campusdefrance.prd.fr>
- [CHAN98] A. Chanbert, E. Grossman, L. Jackson and S. Petrovicz, NCSA Habanero-Synchronous collaborative framework and environment, Software Development Division at the National Center for Supercomputing Applications, white paper, 1998.
- [COLL05] CollabWorx Web site. <http://www.collabworx.com>
- [CONF03] H.323 Support in Exchange 2000 Conferencing Server. White Paper. May 2000. http://www.microsoft.com/china/exchange/zip/ECS_H323.doc
- [CUSE03] CUSeeMe Web site. <http://www.cuworld.com/>
- [CYBE03] Site de la Cyberlicence de Droit – <http://www.univ-tlse1.fr/TeleCours/>

- [DOMM99] H. P. Dommel and J. J. Garcia Luna. Aceves, Group Coordination Support for Internet collaboration. In: IEEE Internet Computing Magazine, Special Issue on Multimedia and Collaborative Computing over the Internet, Vol. 3, No. 2, Mars-Avril 1999, pp. 74-80.
- [DIAN98] Projet DIANE. – <http://www.silogic.fr/diane/Default2.htm>
- [DISC05] Disciple Web site. <http://www.caip.rutgers.edu/disciple/>
- [DOE2000] DOE2000 Web site. <http://www-unix.mcs.anl.gov/DOE2000>
- [DOLL89] J. Dollimore and S. Wilbur. Experiences in Building a Configurable CSCW System. In Studies in Computer Supported Cooperative Work : Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991.
- [DOMM99] H. P. Dommel and J. J. Garcia Luna Aceves, Group Coordination Support for Internet collaboration. In: IEEE Internet Computing Magazine, Special Issue on Multimedia and Collaborative Computing over the Internet, Vol. 3, No. 2, Mars-Avril 1999, pp. 74-80.
- [DOMM00a] H. P. Dommel and J. J. Garcia Luna Aceves, Network Support for Group Coordination. In: 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000), Orlando, FL, 23-26 Juillet 2000.
- [DOMM00b] H. P. Dommel and J. J. Garcia Luna Aceves, Networking Foundations for Collaborative Computing at Internet Scope. In: International ICSC Congress on Intelligent Systems and Applications, Symposium on Interactive and Collaborative Computing (ICC'2000), Wollongong, Australia, 12 – 15 Décembre 2000.
- [DORO99] B. Dorohonceanu and I. Marsic. A Desktop Design for Synchronous Collaboration. Proceedings of the Graphics Interface '99 (GI'99), pages 27-35, Kingston, Ontario, Canada, Juin 1999.
- [DRIR01] K Drira, A. Martelli and T. Villemur (Eds.). Cooperative Environments for Distributed Systems Engineering, The Distributed System Engineering Report, State of the Art Survey. In : Lecture Notes in Computer Sciences, Vol. 2236, N°ISBN 3-540-43083-0, Springer, 2001.
- [DIST90] M.J. Knister and A. Prakash. DistEdit: A Distributed Toolkit for Supporting Multiple Group Editors. Proceedings of the ACM: 3rd International Conference on Computer Supported Cooperative Work, pages 343-355, Octobre 1990.
- [EDWA94] W. K. Edwards, Session Management for Collaborative Applications. In: Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'94), Chapel Hill, NC, 22-26 Octobre 1994.
- [EDWA95] W. K. Edwards, Coordination Infrastructure in Collaborative Systems. Dissertation of Doctor of Computer Science, Georgia Institute of Technology, 22 Novembre 1995. Available:
<http://www2.parc.com/csl/members/kedwards/pubs/dissertation.pdf>
- [ELLI91] C. Ellis, S. Gibbs et G. Rein. Groupware. Some issues and experiences. Communications of the ACM, 34(1):38-58, Janvier 1991.
- [EXCH03] Microsoft Exchange Web site. – <http://www.microsoft.com/exchange/default.asp>
- [FVC03] First Virtual Communications Web site. – <http://www.fvc.com>

- [GARC04] E. Garcia, H. Guyennet, J.-C. Lapayre, et N. Zerhouni. A new Industrial Cooperative tele-maintenance Platform. Special Issue of Computers & Industrial Engineering, Elsevier Science, 46(4):851--864, 2004.
- [GARC05] E. Garcia, H. Guyennet, J. Henriet, et J.-C. Lapayre. Research Approach Advance in Concurrency Management Algorithms for Cooperative Work. International Journal of Computer Science and Network Security, Special issue on Computer Science, 5(10):1--12, Octobre 2005.
- [G711] G.711 International Telecom Union Recommendation. Pulse code modulation (PCM) of voice frequencies. November 1988.
- [G726] G.726 International Telecom Union Recommendation. 40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM). December 1990.
- [GROU03] Produit Novell Group Wise. – <http://www.novell.com/groupwise/index.html>
- [GSM03] ETSI I-ETS 300 036 ed.1. European digital cellular telecommunications system (Phase 1);Full rate speech;Transcoding. Reference:GSM 06.10 Source:SMG2 4176 bytes (90 Pages). Februray 2002.
- [HABA05] Habanero 3.0 Web site. <http://www.isrl.uiuc.edu/isaac/Habanero>
- [HENN89] P. Hennessy. Information Domains in CSCW. In Studies in Computer Supported Cooperative Work : Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991.
- [H261] H.261 International Telecom Union Recommendation. Video codec for audiovisual services at p x 64 kbit/s. 1993.
- [H263] H.263 International Telecom Union Recommendation. Video coding for low bit rate communication. February 1998.
- [H323] H.323 International Telecom Union Recommendation. Packet-based multimedia communications systems. November 2000.
- [HOYO02] G.J. Hoyos-Rivera, R. Lima-Gomes and J.P. Courtiat. A Flexible Architecture for Collaborative Browsing. 11th IEEE WETICE, Workshop on Web-Based Infrastructures and Coordination Architectures for Collaborative Enterprises, 2002, Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA.
- [IONE03] M. Ionescu and I. Marsic, Tree-Based Concurrency Control in Distributed Groupware, Computer-Supported Cooperative Work, ACM / Kluwer Academic Publishers, Vol.12, No.3, pp.329-350, 2003.
- [INSA03] Formation en ligne de l'INSA Toulouse. – <http://enligne.insa-tlse.fr>
- [IRC88] Internet Relay Chat (IRC) help archive. – <http://www.irchelp.org/>
- [ISO10021] Message Oriented Text Interchange System (MOTIS), IS 10021 (CCITT X.400), 1990.
- [ISO9594] The Directory, IS 9594 (CCITT X.500), 1990.
- [JAVV04] Société Javvin. Cartes des protocols H.7xx. <http://www.javvin.com/protocolG7xx.html>
- [KARS94] A. Karsenty. Le collecticiel : de l'interaction homme-machine à la communication homme-machine-homme. Technique et science informatiques, 13(1):105-127, 1994.
- [KRAE88] K. L. Kraemer et J. L. King. Computer-Based Systems for Cooperative Work and Group Decision Making. ACM Computing Surveys, 20(2):115-146, Juin 1988.
- [LEAR03] Lotus – IBM – <http://www.lotus.com/home.nsf/welcome/learnspace>

- [LOGI03] Logitech Web site. – <http://www.logitech.com>
- [LONC03] J. Lonchamp. Le travail coopératif et ses technologies. Hermès Science, 320 pages, 2003.
- [LOTU03] Produit Lotus Notes. – <http://www.lotus.com>
- [LOU03a] S. Louis Dit Picard, "Plate-forme de communication distribuée pour les Environnements Virtuels Collaboratifs 3D à fort couplage d'activité synchrone", Thèse de Doctorat, Université de Lille I, 2003.
- [LOU03b] S. Louis Dit Picard, S. Degrande, C. Gransart, G. Saugis et C. Chaillou. VRML97 Distributed Authoring Interface. Proceeding of Web3D'03, Saint Malo, France, pp. 135-145, Mars 2003.
- [MARS99] I. Marsic, DISCIPLE: A Framework for Multimodal Collaboration in Heterogeneous Environments. In: ACM Computing Survey, Vol. 31, N° 2, Juin 1999, pp. 34-40.
- [MCCL00] B. McClelland. Digital Teaching, Learning and Program Supports: An Examination of Developments for Students in Higher Education. 26th EUROMICRO Conference, Maastricht, the Netherlands, Volume II, pages 43-49, Septembre 2000.
- [McCA97] Steven McCanne, Martin Vetterli, and Van Jacobson. Low-complexity Video Coding for Receiver-driven Layered Multicast. IEEE Journal on Selected Areas in Communications, vol. 16, no. 6, pp. 983-1001, August 1997.
- [MESS04] Microsoft Messenger Web site. – <http://messenger.msn.com/>
- [MIRC03] mIRC Web site. – <http://www.mirc.com>
- [MOLI03] J.M.Molina-Espinosa. Modèle et services pour la coordination des sessions coopératives multi-applications : application à l'ingénierie système distribuée. Thèse de Doctorat, Institut National Polytechnique de Toulouse, France, 175 pages, 1er Juillet 2003.
- [MYER97] J. Myers, N. Chonacky, T. Dunning and E. Leber, Collaboratories: Bringing National Laboratory into the undergraduate classroom and Laboratory via the Internet, Council on Undergraduate research Quarterly vol. 17, N° 3, Mars 1997.
- [NCSA01] National Center for Supercomputing Applications, NCSA Habanero - Version 3.0 & ISAAC Version 1.0, Users Guide, University of Illinois, 2001, URL: <http://www.isrl.uiuc.edu/isaac/Habanero/3.0/habanero3.0.html>
- [NETM03] Microsoft Netmeeting Web site. – <http://www.microsoft.com/windows/netmeeting/default.asp>
- [PARA03] Parallax Web site. – <http://www.jlw.com/~woolsey/parallax>
- [PARA04] Parallel Graphics Web site – <http://www.parallelgraphics.com>
- [PLAT05] Platine Web site. – <http://www.laas.fr/PLATINE>
- [PREV03] J. Prévost. Visioconférence et H.323 : dernières nouvelles. <http://www.renater.fr/Video/VideoNum/H323-Juillet2003/Prevost/Prevost.pdf>. Juillet 2003.
- [PRINZ89] W. Prinz and P. Pennelli. Relevance of the X.500 Directory to CSCW Applications. In Studies in Computer Supported Cooperative Work : Theory, Practice and Design, Eds. J. M. Bowers and S. D. Benford, Elsevier, 1991.
- [PORT03] Stephen Gance. Description de WebCT.Portland State University. – <http://www.webct.pdx.edu/faculty/basics.html>

- [RAT03] Robust Audio Tool Web site. <http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/>
- [RFC1112] S. Deering. IETF RFC 1112. Host Extensions for IP Multicasting. August 1989.
- [RFC1889] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson. IETF RFC 1889. RTP: A Transport Protocol for Real-Time Applications. Janvier 1996.
- [RFC3261] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. IETF RFC 3261. SIP: Session Initiation Protocol. Juin 2002.
- [RICH98] T. Richardson, Q. Stafford-Fraser, K. R. Wood and A. Hopper. Virtual Network Computing. IEEE Internet Computing Volume 2, Number 1, January/February 1998.
- [RUSI95] M. Rusinkiewicz, W. Klass, T. Tesch, J. Wäscher and P. Muth, Towards a Cooperative Transaction Model – The Cooperative Activity Model. In: Proceedings of the 21th VLDB Conference Zurich, Switzerland, 1995, pp. 194-205.
- [SCHL03] C. Schlatter. Basic Architecture of H.323. Swiss Education and Research Network. http://www.switch.ch/vconf/ws2003/h323_basics_handout.pdf. 2003.
- [SCOL03] Infostance– <http://www.scolastance.com>
- [SYMA03] Symantec pcAnywhere. – <http://enterprisesecurity.symantec.fr>
- [TANG05] TANGO 2 Web site. <http://www.npac.syr.edu/tango>. - Java projets for collaboration <http://www.npac.syr.edu/Projects/Java/>
- [TERM04] Grand dictionnaire terminologique
http://www.olf.gouv.qc.ca/ressources/gdt_bdl2.html
- [TEXI99] G. Texier and N. Plouzeau, Automatic Management of Sessions in Shared Spaces. In : Proceedings Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas, NV, Juin/Juillet 1999, pp. 67–73.
- [TUB03] GSM compression. <http://kbs.cs.tu-berlin.de/~jutta/toast.html>
- [TUY02] B. Tuy. Protocoles de routage multicast.
<http://www.renater.fr/Video/IPmulticast/MulticastFev02/P/index.htm>. Janvier 2002
- [VINC02] P. Vincent. Voix sur IP. <http://www-lor.int-evry.fr/~vincent/ppt/voixIP/index.htm>. Mars 2002.
- [VIDE05] VideoLAN Web site. – <http://www.videolan.org>
- [VIEW03] Viewcast Web site. – <http://www.viewcast.com>
- [VISU03] Visual IRC Web site. – <http://www.visualirc.net/>
- [VRVS03] Virtual Room Videoconferencing System. <http://www.vrvs.org>
- [VNC03] Virtual Network Computing Web site. – <http://www.realvnc.com>
- [WEBC03] Produit WebCT. – <http://www.webct.com>
- [WEBE03] WebEx Web site. – <http://www.webex.com>
- [WIKI05] H.264 presentation – <http://fr.wikipedia.org/wiki/H.264>

Chapitre 2

- [BAUD03] V. Baudin, J.P. Courtiat, R.L. Gomes, G.J. Hoyos Rivera and T. Villemur. Lab@Future architecture specification. Rapport LAAS No03313. Lab@Future, Project IST-2001-34204, Juillet 2003, 84p.
- [BAUD04a] V. Baudin, J.P. Courtiat, R.L. Gomes, G.J. Hoyos Rivera and T. Villemur. Lab@future development prototype. Rapport LAAS No04351. Lab@Future, Project IST-2001-34204, Juin 2004, 57p.
- [BAUD04b] V. Baudin, M. Faust, H. Kaufmann, V. Litsa, D. Mwanza, A. Pierre and A. Totter. The Lab@future project. Moving towards the future of e-learning. Actes de la conférence IFIP 18th World Computer Congress. Workshop on Technology Enhanced Learning (TEL'04), Toulouse (France), 22-27 Août 2004, pp.3-18
- [BAUD04c] V. Baudin, J.P. Courtiat and T. Villemur. Lab@Future evaluation prototype. Rapport LAAS No04686. Lab@Future, Project IST-2001-34204, Décembre 2004, 37p.
- [COUR04] J.P. Courtiat, C. Davarakis, M. Faust, H. Kaufmann, D. Mwanza, A. Totter and T. Villemur. Evaluating lab@Future, a collaborative e-learning laboratory experiments platform. Actes de la conférence European Distance and E-Learning Network (EDEN'2004), Budapest (Hongrie), 16-19 Juin 2004, 6p.
- [DRIR99] K. Drira, F. Gouezec and M. Diaz. Design and implementation of coordination protocols for distributed cooperating objects. A general graph-based technique applied to CORBA. Actes de la Troisième Conférence Internationale IFIP sur Formal Methods for Open Object-based Distributed Systems (FMOODS'99), Florence, Italie, 15-18 février 1999.
- [DRIR00] K. Drira and M. Diaz. Graph-grammar based coordination in inter-corporate computer supported collaborative activities. Annual Review of Scalable Computing, Vol.2, Series on Scalable Computing, Ed. YC.Kwong, Singapore University Press, N°ISBN 981-02-4413-4, 2000, Chapter 1, pp.1-27
- [DRIR01] K Drira, A. Martelli and T. Villemur (Eds.). Cooperative Environments for Distributed Systems Engineering, The Distributed System Engineering Report, State of the Art Survey. In : Lecture Notes in Computer Sciences, Vol. 2236, N°ISBN 3-540-43083-0, Springer, 2001.
- [KOUT02] E. Koutsofios, S. North, Drawing graphs with dot – dot, User's Manual, AT&T Lab Research, Février 2002. URL : <http://www.research.att.com/sw/tools/graphviz/>
- [LIEC98] O. Liechti, M. J. Sifer and T. Ichikawa, Structured graph format: XML metadata for describing Web site structure, In: Computer Networks and ISDN Systems, N° 30, (1998), 11-21
- [MARS00] I. Marsic: Real-time Collaboration in Heterogeneous Computing Environments, In: Proceedings of the international Conference on Information Technology, ITCC200, Las Vegas, NV, USA, (March 27-29, 2000), 222-227
- [RODR01] L. M. Rodríguez Peralta, T. Villemur and K. Drira: An XML on-line session model based on graphs for synchronous cooperative groups. In: Proceedings of the international Conference on Parallel and Distributed Processing Techniques and Applications PDPTA'2001, Volume III, Session: Coordination

- in Component-Oriented Systems Architecture, Models, Languages and Applications, Las Vegas, Nevada, USA (June 25 - 28, 2001), 1257-12
- [RODR03] L. M. Rodríguez Peralta, "Service de Gestion de Session Orienté modèle pour des groupes collaboratifs synchrones". Thèse de Doctorat, Institut National Polytechnique de Toulouse, France, 166 pages, Avril 2003.
- [RODR04] L. M. Rodríguez Peralta, T. Villemur and K. Drira: A model-driven and web-enabled coordination approach for managing synchronous collaborative sessions. In: Proceedings of 5e Conferência da Associação Portuguesa de Sistemas de Informação (CAPSI), Lisbonne (Portugal), 3-5 Novembre 2004, 10p.
- [VILL95a] T. Villemur, Conception de services et de protocoles pour la gestion de groupes coopératifs, Thèse de Doctorat, Université Paul Sabatier, Toulouse France, 3 janvier 1995.
- [VILL97] T. Villemur. La coopération dans TOPASE Rapport LAAS No97228. Projet TOPASE N° 3464, Juin 1997, 30p.
- [VILL99] T. Villemur, V. Baudin, S. Owezarski, M. Diaz. Multimedia tools supporting the work of distributed synchronous cooperative groups. Cluster Computing, Vol.2, N°1, pp.61-74, 1999.
- [XML04] W3C consortium, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, (6 October 2000) <http://www.w3.org/XML/>
- [XSL04] W3C consortium, Extensible Stylesheet Language (XSL) Version 1.0, W3C Recommendation 15 October 2001, <http://www.w3.org/TR/xsl/>

Chapitre 3

- [ALEN98] A. Allen. Actor-Based Computing: Vision Forestalled, Vision Fulfilled. Proceedings of 2nd International Conference on Autonomous Agents (Agents'98), 1998.
- [BERT91] B. Berthomieu and M. Diaz. Modeling and verification of time dependant distributed systems using time Petri nets. IEEE Transactions on Software Engineering, 17(3):259-273, mars 1991.
- [BUDK87] S. Budkowski and P. Dembinski. An Introduction to Estelle: A specification Language for Distributed Systems. Computer Networks and ISDN Systems, 14(1):3-23, 1987.
- [CHAR97] Charis Actor Portfolio. Actor Description Language Programmer's Reference. Charis Software Systems Inc., Copyright 1997.
- [DIAZ89] M. Diaz and C. Vissers. SEDOS: Designing Open Distributed Systems. IEEE Software 6(6):24-33, Novembre 1989.
- [DIAZ01] M. Diaz. Les réseaux de Petri : modèles fondamentaux. Traité IC2, série informatique et systèmes d'information, Hermès Science, 384 pages, mai 2001.
- [DIAZ03] M. Diaz. Vérification et mise en oeuvre des réseaux de Pétri. Traité IC2, série Informatique et systèmes d'information, Hermès Science, 388 pages, janvier 2003.

- [DRIR99] K. Drira, F. Gouezec and M. Diaz. Design and implementation of coordination protocols for distributed cooperating objects. A general graph-based technique applied to CORBA. Actes de la Troisième Conférence Internationale IFIP sur Formal Methods for Open Object-based Distributed Systems (FMOODS'99), Florence, Italie, 15-18 février 1999.
- [DRIR00] K. Drira and M. Diaz. Graph-grammar based coordination in inter-corporate computer supported collaborative activities. Annual Review of Scalable Computing, Vol.2, Series on Scalable Computing, Ed. YC.Kwong, Singapore University Press, N°ISBN 981-02-4413-4, 2000, Chapter 1, pp.1-27
- [DRIR01] K. Drira, A. Martelli and T. Villemur (Eds): Cooperative Environments for Distributed Systems Engineering, The Distributed System Environment Report, State of the Art Survey, Lectures Notes in Computer Sciences Vol. 2236, Springer 2001.
- [DSE02] DSE Web site. <http://cec.to.alespazio.it/DSE/>
- [EC92] EC ESTELLE to C compiler. Version 3.0. User reference manual. BULL S.A Copyright 1989, 1990. INT Copyright 1991, 1992.
- [GRAD99] P. Gradit, F. Vernadat and P. Azema. A Layered Delta-net Specification of a workshop. Proceedings of the International Conference on Parallel and Distributed Processing, Techniques and Applications (PDPTA'99), Las Vegas (U.S.A), 28 Juin - 1 Juillet 1999, Volume VI, pp.2808-2815.
- [GOTZ98] R. Gotzhein and J. Brederke. Trends in Formal Description Techniques. Computer Networks and ISDN Systems 30(9-10), Mai 1998.
- [HARE90] D. Harel, H. Lachover, A. Namaad, A. Pnuelli, M. Politti, R. Sherman, A. Shtull-Trauring and M. Traktenbrot. STATEMATE: A Working Environment for the Development of Complex Reactive Systems. IEEE Transactions on Software Engineering, 16(4):403-414, Avril 1990.
- [ISO9074] ISO/IEC ISO 9074 : 1989 (E). Information processing systems. Open System Interconnection. Estelle: A formal description technique based on an extended state transition model, ISO IS 9074, Juin 1989.
- [JSDT99] Sun Microsystems, Inc., Java Shared Data Toolkit 2.0, User Guide, October 4, 1999, URL : <http://java.sun.com/products/java-media/jsdt/>
- [KAPL93] S. Kaplan, J. P. Loyall and S. K. Goering. Specifying Concurrent Languages and Systems with Delta Grammars. Research Directions in Concurrent Object-Oriented Programming. MIT Press, 1993.
- [NGUY96] M. H. Nguyen and M. Schwartz. MCMP: A Transport/Session Level Distributed Protocol for Desktop Conference Setup. IEEE Journal on Selected Areas in Communications, 14(7):1404-1421, Septembre 1996.
- [OWEZ98] P.Owezarski, M. Diaz et C. Chassot. A Time-Efficient Architecture for Multimedia Applications. IEEE Journal on Selected Areas in Communications, pages 383-396, avril 1998.
- [ROBL02] E.Roblet, K.Drira and M.Diaz. Formal design and development of a CORBA-based application for cooperative HTML group editing support. Journal of Systems and Software, Vol.60, N°2, pp.113-127, 2002
- [RODR03] L. M. Rodríguez Peralta, "Service de Gestion de Session Orienté modèle pour des groupes collaboratifs synchrones". Thèse de Doctorat, Institut National Polytechnique de Toulouse, France, 166 pages, Avril 2003.

-
- [SARM96] A. Sarma. An introduction to SDL-92. *Computer Networks and ISDN Systems* 28(1996):1603-1615.
- [TELE05] Telelogic Tau Web site. <http://www.telelogic.com/products/tau/index.cfm>
- [VILL95a] T. Villemur. Conception de Services et de Protocoles pour la Gestion de Groupes Coopératifs. Thèse de Doctorat de 3ème cycle, Spécialité Informatique, Université Paul Sabatier, Toulouse III, Janvier 1995.
- [VILL95b] T. Villemur, M. Diaz and F. Vernadat. Validated design of dynamic membership services and protocols for cooperative groups. *Annales des Télécommunications*, N°11-12, pp.859-873, Novembre-Décembre 1995.
- [VILL98] T. Villemur, P. Owezarski, and M. Diaz. N-TSVS: a Videoconference Tool for Generic Cooperative Groups. *Proceedings of the International Conference on Multimedia Modeling MMM'98*, Octobre 1998.
- [XERC00] Apache XML Project, Xerces Java Parser 1.4.4, The Apache Software Foundation (1999-2000), URL: <http://xml.apache.org/xerces-j/index.html>.
- [ZIMM80] H. Zimmerman. OSI Reference Model, the ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, vol. COM-28, Avril 1980.

Modèles et services logiciels pour le travail collaboratif

Résumé

Ce mémoire se compose de deux parties.

La première partie présente le cursus que nous avons suivi en termes d'activités d'enseignement, de recherche et de responsabilités administratives, depuis l'obtention de la thèse de troisième cycle.

La deuxième partie synthétise nos principales contributions dans le cadre de la conception de services et de protocoles pour les systèmes distribués collaboratifs.

Le premier chapitre de cette partie présente le cadre de notre travail, le travail collaboratif, vu essentiellement selon l'angle de l'informatique répartie. Au travers d'une étude, analyse et classification, nous mettons en évidence, sous forme de bilan, les principales avancées et tendances d'évolution des systèmes collaboratifs. Nous présentons une analyse et une classification des principaux modèles de session rencontrés, que nous comparons et classifions. Finalement, nous procédons à une analyse et une comparaison des principaux gestionnaires de sessions existants.

Le second chapitre de cette partie décrit nos contributions en terme de modélisation de sessions collaboratives. Après avoir décrit les principaux éléments qui composent une session collaborative, nous décrivons le modèle de session que nous avons proposé et nous le positionnons vis-à-vis des autres formalismes. Nous terminons par nos contributions détaillées pour l'implantation et l'utilisation de notre modèle de session.

Le troisième chapitre de cette partie traite de la gestion des sessions collaboratives. Nous décrivons le service de gestion de session que nous avons proposé et nous comparons ses caractéristiques à celles des autres gestionnaires. Finalement, nous présentons le détail de nos contributions dans les domaines de la conception formelle et de l'implantation des services de gestion de session, et plus généralement de services collaboratifs.

Mots clés :

- Travail collaboratif, Téléenseignement
- Modèles pour des sessions collaboratives, Diagrammes de coordination
- Services de coordination, Gestion de sessions, Méthodologies et architectures de conception, Implantation formelle

Models and software services for collaborative work

Abstract

This document is composed of two parts.

Part One presents our curriculum vitae, including teaching activities, research activities and administrative responsibilities since our PhD.

Part Two synthesizes our main contributions in the framework of design of services and protocols for distributed collaborative systems.

The first chapter of part Two presents our framework, the collaborative work, by taking the distributed systems viewpoint. The main advances and evolution trends of collaborative systems are studied, analyzed and classified. The main models of session are presented, analyzed, compared and classified. Then, the main existing session management systems are analyzed and compared.

The second chapter of part Two presents our contributions for modeling collaborative sessions. The main elements that compose a collaborative sessions are firstly presented. The proposed session model is described and compared to other formalisms encountered. The detailed contributions are composed of implementation and use of the session model proposed.

The third chapter of part Two presents the management of collaborative sessions. The proposed management service is described. Its characteristics are compared with other existing session management service characteristics. At end, the contributions are detailed in the domain of formal design and implementation of session management services and more generally collaborative services.

Keywords:

- Collaborative work, E-Learning
- Models for collaborative sessions, Coordination diagrams
- Coordination services, Session management, Design methodologies and architectures, Formal implementation