



HAL
open science

Espaces Collaboratifs Ubiquitaires sur une infrastructure à ressources distribuées

Pascal Dugénie

► **To cite this version:**

Pascal Dugénie. Espaces Collaboratifs Ubiquitaires sur une infrastructure à ressources distribuées. domain_other. Université Montpellier II - Sciences et Techniques du Languedoc, 2007. Français. NNT: . tel-00203542

HAL Id: tel-00203542

<https://theses.hal.science/tel-00203542>

Submitted on 10 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numéro d'identification :

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II
— SCIENCES ET TECHNIQUES DU LANGUEDOC —

THÈSE

présentée à l'Université des Sciences et Techniques du Languedoc
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : **INFORMATIQUE**
Formation Doctorale : **Informatique**
École Doctorale : **Information, Structures, Systèmes**

Espaces collaboratifs ubiquitaires
sur une
infrastructure à ressources distribuées

par

Pascal Dugénie

Soutenue le 7 décembre 2007 devant le Jury composé de :

Stefano A. CERRI, Professeur, Université Montpellier 2, LIRMM Directeur de thèse
Bertrand DAVID, Professeur, Ecole Centrale de Lyon, LIESP Rapporteur
Abdelkader GOUAICH, Maître de Conférences, Université Montpellier 2, LIRMM Examineur
Domenico LAFORENZA, Directeur de Recherche, ISTI-CNR, Pise, Italie Co-directeur de thèse
Jocelyne NANARD, Professeure, Université Montpellier 2, LIRMM Examineur
Thierry PRIOL, Directeur de Recherche INRIA, IRISA Rapporteur
Jean-Pierre PROST, IBM France, Montpellier Examineur
Jean SALLANTIN, Directeur de Recherche, CNRS, LIRMM Examineur

Numéro d'identification :

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II
— SCIENCES ET TECHNIQUES DU LANGUEDOC —

THÈSE

présentée à l'Université des Sciences et Techniques du Languedoc
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : **INFORMATIQUE**
Formation Doctorale : **Informatique**
École Doctorale : **Information, Structures, Systèmes**

Espaces collaboratifs ubiquitaires
sur une
infrastructure à ressources distribuées

par

Pascal Dugénie

Soutenue le 7 décembre 2007 devant le Jury composé de :

Stefano A. CERRI, Professeur, Université Montpellier 2, LIRMM Directeur de thèse
Bertrand DAVID, Professeur, Ecole Centrale de Lyon, LIESP Rapporteur
Abdelkader GOUAICH, Maître de Conférences, Université Montpellier 2, LIRMM Examineur
Domenico LAFORENZA, Directeur de Recherche, ISTI-CNR, Pise, Italie Co-directeur de thèse
Jocelyne NANARD, Professeure, Université Montpellier 2, LIRMM Examineur
Thierry PRIOL, Directeur de Recherche INRIA, IRISA Rapporteur
Jean-Pierre PROST, IBM France, Montpellier Examineur
Jean SALLANTIN, Directeur de Recherche, CNRS, LIRMM Examineur

à mes deux filles Camille et Manon ...

REMERCIEMENTS

En premier lieu, je tiens à exprimer toute ma gratitude envers mon Directeur de thèse PROFESSEUR STEFANO A. CERRI pour toute la confiance qu'il m'a accordée dans le cadre du projet *ELeGI*¹. Son expérience et ses conseils resteront une source d'enrichissement primordial. Un remerciement particulier s'adresse à trois de mes collègues grâce à qui nous avons pu concrétiser plusieurs publications et qui ont apporté, sans relâche, leur expertise et leur soutien amical : DOCTEUR PHILIPPE LEMOISSON sur le modèle conceptuel, DOCTEUR CLÉMENT JONQUET sur la formalisation agents, et DOCTEUR ABDELKADER GOUAICH pour ses précieux conseils avant la soutenance.

Je remercie également mon co-directeur de thèse DOCTEUR DOMENICO LAFORENZA, Directeur de Recherche à l'*ISTI-CNR*² pour son accueil lors de ma visite à Pise, et tous les membres du jury de thèse pour leur disponibilité, et, surtout, pour leur accompagnement ; la précision et la pertinence des remarques ont été déterminantes sur le fond et sur la forme du travail présenté : PROFESSEURE JOCELYNE NANARD, Présidente du jury, PROFESSEUR BERTRAND DAVID de l'*Ecole Centrale de Lyon*³, DOCTEUR THIERRY PRIOL, Directeur de Recherche à l'*INRIA*⁴, DOCTEUR JEAN-PIERRE PROST, Directeur de Recherche à *IBM Montpellier* et DOCTEUR JEAN SALLANTIN, Directeur de Recherche au *CNRS*. Je remercie PROFESSEUR PIERRE LEVY de l'*Université d'Ottawa* qui m'a fait l'honneur de lire attentivement mon manuscrit et pour la généreuse attention de m'écrire une préface.

Ce travail est aussi la contribution d'un ensemble de plusieurs dizaines d'acteurs qui se sont cordialement prêtés aux expérimentations et qui ont apporté des suggestions originales lors de la mise au point du prototype *AGORA*. Je pense plus particulièrement à PROFESSEUR CLAUDE LAURENÇO de l'*ENCM*, PROFESSEUR ALAIN KRIEF et DOCTEUR JULIE HENRY de l'*Université de Namur*, DOCTEUR MONICA CRUBÉZY de l'*Université de Stanford*, PROFESSEUR MARC EISENSTADT de l'*Open University (Milton Keynes, GB)*, PROFESSEUR JOOST BREUKER de l'*Université d'Amsterdam*, PROFESSEUR GUY GOUARDÈRES et SABER MANSOUR de l'*IUT de Bayonne*, et surtout, un remerciement spécial à l'attention de DOCTEUR CATHERINE COLAUX pour sa forte contribution aux essais et aux démonstrations jusqu'au jour de la soutenance. J'aimerais également remercier le laboratoire qui m'a accueilli avec tout le soutien logistique, technique et administratif, indispensable pour mener cette thèse à son terme dans des conditions relationnelles et matérielles privilégiées. L'équipe d'administration des systèmes DOCTEUR JEAN-LUC OMS, DOCTEUR MICHEL JACQUOT et STÉPHANE GEORGES ; ELISABETH GRÈVERIE et tout le personnel administratif ; les étudiants du *Master informatique* et les élèves ingénieurs de *Polytech Montpellier* qui ont dédié leur stage de fin d'études au prototype *AGORA*.

Enfin, l'aboutissement de cette thèse a été rendu possible grâce aux encouragements de toute ma famille, parents et amis durant ces trois années. Sur la dernière ligne droite, je remercie CHRISTOPHE DA COSTA pour les corrections de français et CHRISTINE ROCHE pour la formalisation mathématique. Pour finir, un merci, tout particulier, à ma chère et tendre STÉPHANIE pour son soutien moral et pour ses corrections, ainsi qu'à mes beaux parents JEAN et LUCIENNE VINESSE pour leur implication de tous les instants.

¹Ce travail a été subventionné par le projet *European Learning Grid Infrastructure (ELeGI)*, contrat IST-002205 du 6^e Programme Cadre de Recherche et de Développement (PCRD) européen *Information Society Technologies (IST)*. La communauté européenne n'est pas responsable de l'utilisation faite des résultats publiés dans ce mémoire.

²*Istituto di Scienza e Tecnologia dell'Informazione (ISTI) Consiglio Nazionale delle Ricerche de Pisa (CNR)*.

³*Laboratoire d'Informatique pour l'Entreprise et les Systèmes de Production (LIESP)*

⁴*Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA)*



PRÉFACE DE PIERRE LÉVY

J'ai lu avec beaucoup de plaisir et d'intérêt la thèse de PASCAL DUGÉNIE sur les *espaces collaboratifs ubiquitaires* sur une *infrastructure à ressources distribuées*. Ce texte, non dénué de vertus pédagogiques, a mis à jour mes connaissances sur les architectures *orientées services* en général et sur l'architecture de services GRID en particulier. J'ai notamment été séduit par le fait que l'identification d'un service puisse être réalisée par une couche logique indépendante de la structure physique du réseau. La notion de *service à état* m'a semblé fort prometteuse dans la perspective d'une informatique orientée vers le perfectionnement de l'*intelligence collective* humaine. En ce qui concerne la plateforme AGORA , qui est l'apport propre de PASCAL DUGÉNIE, j'ai été impressionné par la subtilité de la formalisation des agents, des communautés d'agents, des membres et des ensembles de membres pour le service COLLABUREAU . C'est une grande satisfaction, pour un philosophe qui s'intéresse sérieusement à l'informatique, de constater que des chercheurs en informatique peuvent aussi, en retour, puiser une partie de leur inspiration dans des travaux philosophiques. Ainsi, le concept d'*immanence*, essentiel à mon sens dans une théorie de l'*intelligence collective* humaine techniquement augmentée, m'a semblé fort bien intégré et pris en compte dans le travail de PASCAL DUGÉNIE : il correspond ici à l'autonomie décisionnelle et à l'auto-organisation des communautés de collaborateurs. Je souhaite à PASCAL DUGÉNIE la continuation d'une fructueuse carrière de recherche, pleine de découvertes et de collaborations trans-disciplinaires.

PIERRE LÉVY, Ottawa, 13 novembre 2007



TABLE DES MATIÈRES

Remerciements	5
Préface de PIERRE LÉVY	7
Introduction générale	11
Cap vers l' <i>intelligence collective</i>	11
Un contexte technologique favorable	11
Motivation, proposition et approche scientifique	13
Cadre et structure de cette thèse	13
Chapitre 1 Intelligence collective	17
1.1 Fondements	18
1.2 Interactions	18
1.3 Systèmes complexes	22
1.4 Approche socio-constructiviste	24
1.5 Conclusion : mise en relation de compétences réparties	26
Chapitre 2 Espaces collaboratifs ubiquitaires	29
2.1 Panorama	30
2.2 Services d'un <i>espace collaboratif ubiquitaire</i>	32
2.3 Limites des solutions existantes	37
2.4 Caractérisation	39
2.5 Recommandations pour le déploiement	41
Chapitre 3 Infrastructure à ressources distribuées	43
3.1 Evolution conceptuelle et technologique	44
3.2 Principes	52
3.3 Synergie entre GRID et les <i>SMA</i>	61
3.4 Projection future	64
Chapitre 4 Modèle et plateforme AGORA	67
4.1 Cadre conceptuel	68
4.2 Description fonctionnelle	75
4.3 Développement du prototype AGORA	78
4.4 Développement du service COLLABUREAU	80
4.5 Intégration	84

Chapitre 5 Expérimentations	87
5.1 Objectif et méthodologie	88
5.2 Cadre expérimental général	88
5.3 Scénarii d'expérimentation	90
5.4 Résultats majeurs	92
Chapitre 6 Synthèse	99
6.1 Bilan	100
6.2 Discussion	101
Conclusion et perspectives	105
Annexe A : Fonctions AGORA	107
Annexe B : Procédures PISA	113
Table des figures	115
Liste des tableaux	117
Definitions	119
Abréviations	121
Bibliographie	125

INTRODUCTION GÉNÉRALE

Cap vers l'*intelligence collective*

En l'espace de quelques décennies seulement, l'informatique a permis à toutes les sciences d'accélérer leur développement. Elle est aussi devenue l'instrument privilégié d'une nouvelle organisation sociale. Ce constat a suscité de nombreuses questions de nature philosophique comme celles soulevées par PIERRE LEVY qui s'est penché sur l'impact des instruments numériques⁵ dans la société contemporaine. Nous avons sélectionné quelques unes de ses réflexions pour définir notre problématique : D'abord celle concernant les dispositifs techniques, sociaux et sémiotiques qui sont fortement imbriqués [Lévy1990], ensuite celle d'une *intelligence collective* émergeant continuellement de libres relations qui se nouent dans un espace *immanent*⁶ intitulé par son auteur l'*espace anthropologique du savoir*⁷ [Lévy1994].

Ces postulats nous ont amenés à réfléchir sur le potentiel des *TIC*, non pas pour rendre la machine plus intelligente dans le sens de MINSKY⁸, mais plutôt comme catalyseur de l'*intelligence collective*.

Un contexte technologique favorable

L'*intelligence collective* en temps réel et à grande échelle nécessite une infrastructure technique adéquate [Lévy1994]. Dans cette optique, le défi lancé aux chercheurs et ingénieurs du monde entier est de déployer des infrastructures capables non seulement d'acheminer intelligemment des flux de communication, mais aussi d'exploiter ces flux pour ajouter de la valeur aux contenus existants. En quelques années, deux évolutions conceptuelles notables se sont produites de façon totalement indépendante mais avec une forte implication de l'une envers l'autre. La première concerne le niveau d'analyse des informations sur le plan syntaxique mais aussi sur le plan sémantique, et la seconde qui nous intéresse plus particulièrement dans le cadre de cette thèse concerne les architectures.

Un autre facteur qui a eu un impact significatif sur la culture du développement informatique de ces dernières années est la généralisation du logiciel libre. De ce fait, le logiciel n'est désormais plus considéré comme un produit monolithique qui concentre de la matière grise à l'état figé mais celui-ci tend à se métamorphoser progressivement en une variété d'entités dynamiques recomposables : les *services*. Les technologies à base de composants logiciels sont les archétypes de cette nouvelle tendance mais la véritable mutation se situe au niveau de l'architecture.

⁵Diverses appellations sont utilisées pour qualifier ces instruments numériques : *TIC* ou *STIC* pour les *Sciences et Technologies de l'Information et de la Communication*, ou encore, depuis 2006, l'appellation *ST2I (Sciences et Technologies de l'Information et de l'Ingénierie)* remplace *STIC* au CNRS.

⁶Le sens du qualificatif *immanent* (*lat. immanere*, résider dedans) signifie ici que l'activité n'est pas séparable de ce sur quoi elle agit, mais la constitue de manière interne.

⁷Pour PIERRE LEVY, le savoir constitue le quatrième espace anthropologique après la terre, les territoires et l'espace d'échanges de marchandises.

⁸La notion d'intelligence a été initialement introduite en informatique pour désigner l' *Intelligence Artificielle (IA)*. Pour MARVIN MINSKY, l'un de ses fondateurs, l'*IA* est une technique de programmation heuristique dont le but est d'émuler artificiellement un mode de raisonnement semblable à l'humain [Minsky1989, Minsky1991].

Architectures orientées-services WEB et GRID

L'un des facteurs qui a marqué l'évolution des architectures en informatique de ces dernières années, est le passage d'une vision *orientée-système* à une vision *orientée-service*. Les premières architectures *orientée-services* sont apparues avec l'introduction des services WEB [Booth *et al.*2004]. Le terme *service* désigne ici une entité logicielle qui exporte des fonctions accessibles, par un navigateur WEB, à travers une interface et un protocole normalisés. Par la suite, la GRILLE DE RESSOURCES INFORMATIQUES DISTRIBUÉES (GRID)⁹ a introduit de nouveaux concepts comme le *service à état* ou le *service dynamique* [Foster & Kesselman1999b]. Ces concepts découlent de la nature même de l'infrastructure GRID. En effet, si le WEB est l'infrastructure mondiale pour le partage d'informations distribuées, GRID est l'infrastructure pour le partage de ressources distribuées. Cette propriété est fondamentale pour déployer des services autonomes. Parce qu'ils disposent de leur propre contexte d'exécution, la durée de vie de ces services est indépendante des connexions vers ceux-ci. De ce fait, GRID présente une certaine complémentarité avec les *Systèmes Multi Agents (SMA)*, ce qui ouvre la perspective d'une véritable informatique *ubiquitaire*. Afin de bien cerner les enjeux d'aujourd'hui, la table 1 présente une segmentation sommaire des évolutions conceptuelles autour de ces deux paradigmes.

Le WEB sémantique

De par sa popularité, le WEB est la piste privilégiée par la communauté scientifique pour explorer de nouvelles solutions d'indexation des contenus par l'analyse sémantique. Depuis 2001, sous l'impulsion de TIM BERNERS-LEE, le WEB sémantique est le programme de recherche le plus actif dans ce domaine [Berners-Lee *et al.*2001, Berners-Lee2001, Cabral *et al.*2004]. Il propose des normes de codage logique sur la base de documents *eXtended Markup Language (XML)* [XMLurl]. Deux normes aujourd'hui stables ont permis de simplifier les développements du WEB sémantique. Il s'agit de *Resource Description Framework (RDF)* qui est utilisée pour associer des informations sémantiques à un fichier original, et de *Ontology Web Language (OWL)*, un peu plus évolué que *RDF*, qui spécifie les classes et les propriétés dans la forme d'un langage de description à base d'opérateurs booléens [W3C2004, McIlraith *et al.*2001, Martin *et al.*2004]. L'orchestration des services pour le WEB sémantique est analogue au protocole d'interaction entre agents [Buhler & Vidal2003, FIPA2002]. Ceci suscite un intérêt croissant au sein de ces communautés [Singh & Huhns1999, Buhler *et al.*2003, Vidal *et al.*2004, Buhler & Vidal2004]. Une autre norme en cours d'élaboration pour exprimer les nuances sémantiques et pragmatiques des langues naturelles est *Information Economy Meta Language (IEML)* [IEMLurl, Levy2007]. Sa conception tente de répondre au défi de l'interopérabilité sémantique, par l'adressage des données en concordance avec la gouvernance distribuée de l'intelligence collective au service du développement de la pensée humaine.

Le WEB 2.0

Le programme de recherche WEB 2.0 lancé en 2004 par TIM O'REILLY [O'reilly2004] se consacre au développement d'outils collaboratifs sur le WEB. La singularité du WEB 2.0 n'est pas une réalisation technologique mais repose sur le succès de solutions collaboratives ou pair-à-pair dont *wiki* symbolise cette réussite. JOËL DE ROSNAY¹⁰ prédit qu'avec le WEB 2.0 nous allons assister à l'explosion des contenus générés par les utilisateurs avec au premier chef les logiciels collaboratifs [de Rosnay2005].

⁹L'acronyme GRID fait initialement référence au réseau de distribution d'énergie électrique d'Amérique du nord. Une formulation française de *Grid Globalisation des Ressources Informatiques et des Données*, a été proposée en 2001 par le Ministère de la Recherche dans le cadre du lancement d'une *Action Concertée Incitative (ACI)*.

¹⁰JOËL DE ROSNAY est un futurologue renommé qui, depuis les années 1970, a anticipé sur l'impact des *TIC* dans la société [de Rosnay1975, de Rosnay1988]. Selon lui le phénomène va encore s'amplifier et va bouleverser les systèmes de production classique et de distribution pyramidale (TV, radio, édition musicale) fortement concurrencés par les médias transversaux comme des journaux citoyens (Agoravox) ou les encyclopédies de masse (Wiki). Il préfère employer le terme d'intelligence connective ou collaborative pour désigner l' *intelligence collective*.

GRID sémantique pour l'apprentissage collaboratif

GRID sémantique a été lancée en 2001 par une équipe de recherche de l'*Université de Southampton (GB)* sous la direction de DAVID DE ROURE qui décrit GRID sémantique par analogie au WEB sémantique [De Roure *et al.*2001, Goble & De Roure2004, De Roure *et al.*2005, Geldof2004]. Trois composantes fondamentales forment l'infrastructure de GRID sémantique [Jeffery1999] : (i) les services *Données/Traitement*, (ii) les services *Information* et, sur la couche supérieure, (iii) les services *Connaissance*. Ces derniers ont en particulier la charge d'établir les relations ontologiques entre les éléments.

GRID sémantique a donné lieu à de nouvelles pistes de recherche autour de la construction de connaissances [Cannataro & Talia2003, Zhuge2004] et, depuis 2003, de faire de l'apprentissage collaboratif [Cerri2003, Allison *et al.*2004, Ritovato *et al.*2005, Allison *et al.*2005, Nkambou *et al.*2005, Bote-Lorenzo *et al.*2004, Gouaich & Cerri2004].

TAB. 1 – Segmentation des évolutions conceptuelles autour des paradigmes WEB et GRID

Architecture	Approche syntaxique	Approche sémantique
Orientée-système	WEB 1.0	-
Orientée-service (sans état)	WEB 2.0	WEB sémantique
Orientée-service (avec état)	GRID	GRID sémantique

Motivation, proposition et approche scientifique

Notre motivation première est de catalyser l'*intelligence collective* au moyen d'un système fiable, qui tienne compte de tous les aspects dynamiques inhérents à l'*intelligence collective*. Pour cela, nous proposons une architecture composée d'*espaces collaboratifs ubiquitaires* déployables sur une *infrastructure à ressources distribuées*. Rapprocher ces deux concepts dans un modèle unique est un défi majeur. En effet, notre hypothèse est qu'un *espace collaboratif ubiquitaire* est un moyen permettant de dynamiser ostensiblement l'*intelligence collective*. Toutefois, ce concept n'est viable qu'à partir du moment où il peut être déployé dans une large mesure. Ce passage à l'échelle doit s'appuyer sur une infrastructure capable de coordonner et sécuriser les ressources. Notre approche scientifique est fondée sur le paradigme *centré-interaction*. De ce fait, au vu de la dynamique d'un tel système, la preuve de la validité de notre modèle n'est pas envisageable sur la base d'une méthode exclusivement analytique. Nous nous efforcerons plutôt à mettre en évidence le bien fondé de notre intuition de manière expérimentale. La reproductibilité des expériences ne peut toutefois être garantie dans la mesure où le modèle est non-déterministe. Les résultats seront, par conséquent, relatés comme des faits anecdotiques mais ces observations, prises dans leur ensemble, auront une portée significative sur la validité du modèle.

De cette façon, nous espérons obtenir des résultats expérimentaux riches en enseignements par la multiplication des scénarii qui privilégient une utilisation spontanée des services offerts par les *espaces collaboratifs ubiquitaires*.

Cadre et structure de cette thèse

Cette thèse s'inscrit dans le contexte des réflexions menées actuellement sur l'impact des *TIC* dans la société. Plus particulièrement, notre étude vise à analyser les moyens technologiques à mettre en œuvre pour la construction de connaissances dans le cadre de l'enseignement et du travail collaboratif à distance (voir figure 1). Cette thèse comporte six chapitres, décrits ci-dessous, qui ont fait l'objet de publications :

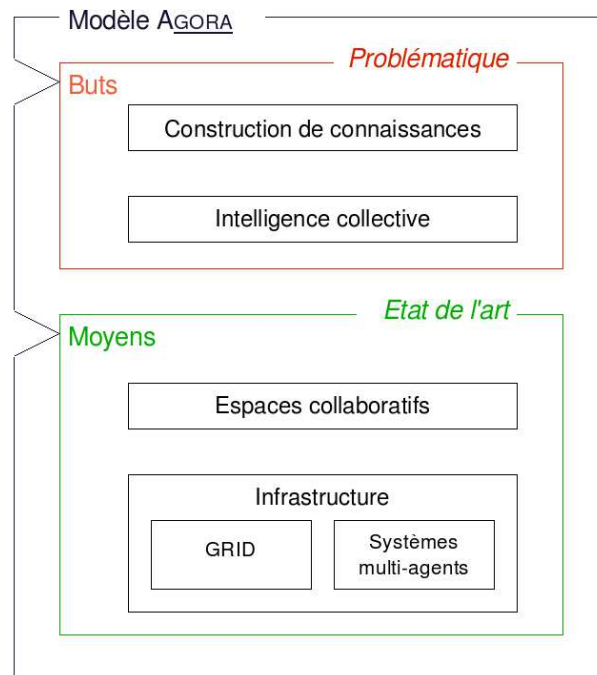


FIG. 1 – Buts et moyens fixés dans le cadre de cette thèse

Chapitre 1 : Intelligence collective

Ce chapitre décrit la problématique. Comment l'*intelligence collective* émerge-t-elle d'un système complexe composé d'humains en interaction par l'intermédiaire de processus potentiellement dotés d'intelligence artificielle. Les notions générales d'interaction, de systémique et de contextualisation y sont décrites.

Chapitre 2 : Espaces collaboratifs ubiquitaires

Ce chapitre dresse un état de l'art pour caractériser le concept *espaces collaboratifs ubiquitaires* et les limites des solutions actuelles. Plusieurs des recommandations décrites dans ce chapitre ont fait l'objet d'une publication [Dugénie & Lemoisson2005].

Chapitre 3 : Infrastructure à ressources distribuées

Ce chapitre dresse un état de l'art et une prospective technologique autour des *architectures* susceptibles d'être utilisées pour déployer des *espaces collaboratifs ubiquitaires*. Ce chapitre fournit une synthèse des propriétés qui nous motivent à privilégier une solution d'*infrastructure* combinant GRID avec les *Systèmes Multi Agents (SMA)*. Ces spécifications ont fait l'objet de trois publications [Dugénie2005, Jonquet *et al.*2006, Jonquet *et al.*2007].

Chapitre 4 : Modèle et prototype AGORA

Ce chapitre décrit le modèle AGORA dont les aspects conceptuels reposent sur les recommandations des chapitres précédents. Le prototype que nous avons développé sur la base de ce modèle vise à mettre en évidence l'intérêt de rapprocher les concepts d'*espaces collaboratifs ubiquitaires* et d'*infrastructure*

à *ressources distribuées* en vue de faire émerger une nouvelle forme d'*intelligence collective*. Ce modèle a partiellement été décrit dans deux publications [Dugénie & Cerri2006, Cerri *et al.*2006].

Chapitre 5 : Expérimentations

Ce chapitre décrit notre méthodologie et nos résultats d'expérimentation lors de la validation du prototype AGORA . Puisque notre méthodologie est essentiellement empirique, ce chapitre vise à corroborer notre argumentation par des résultats expérimentaux. Ces expérimentations ont consisté à amorcer des situations de collaboration puis à laisser les groupes s'auto-organiser. Nous avons ainsi pu identifier les facteurs exerçant une influence positive sur ce processus. Ces résultats sont essentiellement qualitatifs afin de comprendre le point de vue utilisateur. Une partie de ces résultats a fait l'objet de deux publications [Dugénie *et al.*2006, Latorre *et al.*2006].

Chapitre 6 : Synthèse

A travers l'analyse des chapitres précédents, ce dernier tire quelques enseignements. En comparant AGORA avec d'autres modèles, nous avons pu identifier les points forts du nôtre et quelques pistes à explorer comme la normalisation pour faciliter l'évolution du code de ce modèle.

INTELLIGENCE COLLECTIVE

Le chemin se fait en marchant [Machado1936].
Antonio Machado (poète espagnol)

L'INTELLIGENCE COLLECTIVE ou intelligence collaborative selon les différents points de vue qui sous-tendent cette idée, est, au départ, une notion de philosophie sociale qui a pris racine en informatique depuis quelques années. La problématique est de savoir comment émerge l'intelligence collective d'un système complexe composé d'humains en interaction par l'intermédiaire de processus potentiellement dotés d'intelligence artificielle. Ce chapitre a pour but d'étayer l'argumentation sur la méthodologie employée par la suite. Nous avons choisi pour cela de nous appuyer sur des notions de *théorie des systèmes* telles que l'auto-organisation ou l'émergence, et sur des principes comportementaux décrits par le *socio-constructivisme* tels que la contextualisation. L'étude de ces aspects fondamentaux est destinée à orienter notre démarche de modélisation.

Table des matières

1.1	Fondements	18
1.2	Interactions	18
1.2.1	Le paradigme centré-interaction	18
1.2.2	Les deux modes d'interaction	19
1.2.3	Modèles de collaboration	20
1.3	Systèmes complexes	22
1.3.1	Aux origines de la théorie des systèmes	22
1.3.2	Principes de systémique	23
1.4	Approche socio-constructiviste	24
1.4.1	Éléments de socio-constructivisme	24
1.4.2	Contextualisation systémique	25
1.5	Conclusion : mise en relation de compétences réparties	26

1.1 Fondements

L'*intelligence collective* est une vaste notion qui trouve ses fondements dans divers domaines. En philosophie sociale, elle se définit comme « *une intelligence partout distribuée, sans cesse valorisée, coordonnée en temps réel, qui aboutit à une mobilisation effective des compétences* » [Lévy1994]¹. En informatique, elle émane d'études sur les comportements collectifs (insectes sociaux) qui fournissent des méthodes pour la conception d'algorithmes d'optimisation combinatoire² [Bonabeau1994, Bonabeau *et al.*2000], ou encore de travaux de recherche dans le domaine des *Systèmes Multi Agents (SMA)* [Ferber1995].

Nous constatons que la philosophie sociale s'intéresse, pour sa part, à l'*intelligence collective* en tant que valeur émergente d'un système complexe par la collaboration entre humains et grâce aux moyens de communication actuels. Quant à la recherche en informatique, elle se consacre essentiellement à la modélisation des comportements sociaux. Par exemple, la *kénétique* [Ferber1994] consiste à formaliser les principes conduisant à l'auto-organisation d'un SMA par une modélisation des différentes formes d'interaction.

Notre démarche est radicalement différente. Nous ne cherchons pas à reproduire artificiellement un comportement intelligent. Nous cherchons plutôt à définir un modèle pour catalyser le potentiel d'*intelligence collective* présent dans un système complexe et dont l'origine peut être tantôt humaine tantôt artificielle.

Ainsi, comme le montre la figure 1.1, l'*espace anthropologique du savoir*, introduit par PIERRE LEVY, peut se représenter comme un réseau dont les nœuds sont des agents dotés d'une intelligence humaine (agents humains) ou des agents dotés d'une intelligence artificielle (agents artificiels). Tous ces agents interagissent en offrant des services basés sur leurs compétences. Les ressources informatiques réalisent ici soit du calcul, soit une mise en relation pertinente de ces compétences.

1.2 Interactions

1.2.1 Le paradigme centré-interaction

L'étude des interactions entre humains et systèmes informatisés est un domaine qui connaît une popularité croissante sous l'impulsion des travaux menés ces dernières années par PETER WEGNER [Wegner1996, Wegner1997, Wegner & Goldin1999]. Le paradigme centré-interaction diffère de plusieurs façons du paradigme centré-algorithme basé sur le modèle de la machine de TURING [Goldin *et al.*2002, Wegner & Goldin2003]. La notion d'interaction intègre notamment l'idée de réaliser une tâche ou de fournir un service plutôt que de produire, par un procédé algorithmique, des résultats en sortie à partir de données en entrée. D'autres aspects comme l'observation d'un comportement général, ou encore l'importance de l'environnement sur les décisions des agents du système sont des propriétés qui distinguent un système en interaction d'un algorithme.

Une récente classification identifie trois niveaux d'interaction dans un système composé d'humains et de dispositifs informatisés [Beaudouin-Lafon2006] :

1. Interaction directe entre un utilisateur et des objets informatiques. Par exemple, le déplacement de fichiers d'un répertoire à un autre au moyen d'un navigateur.
2. Interaction à travers des processus intelligents qui peuvent travailler en arrière-plan et dialoguer occasionnellement avec l'utilisateur pour indiquer, par exemple, la progression d'une tâche.

¹PIERRE LEVY, *L'intelligence collective* (page 29).

²L'optimisation combinatoire consiste à minimiser une certaine fonction (dite de coût) sur un ensemble fini de configurations. C'est un domaine important en recherche opérationnelle. Par exemple, il peut s'agir de minimiser la distance totale reliant toutes les villes d'un réseau ou le délai d'acheminement de données dans un réseau de télécommunication.

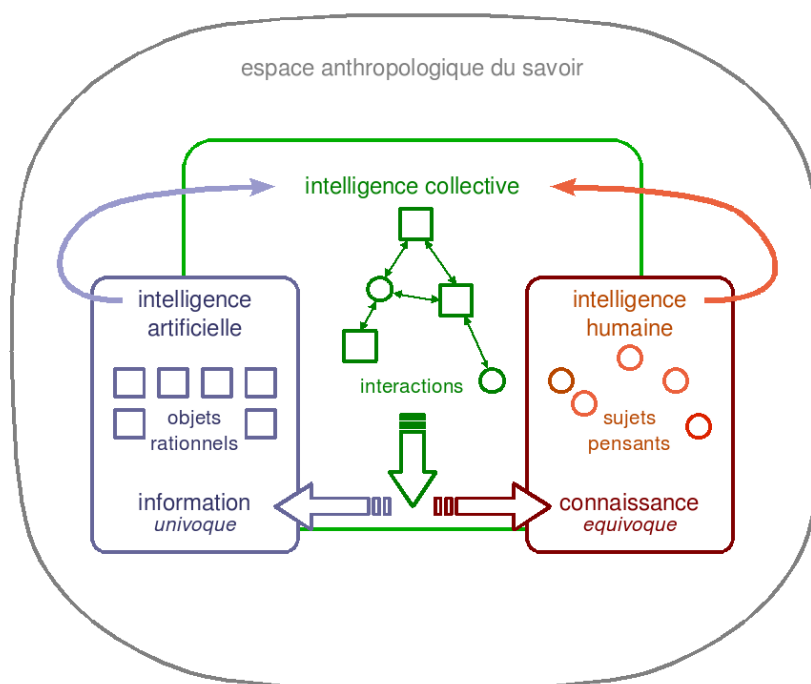


FIG. 1.1 – L'intelligence collective dans l'espace anthropologique du savoir.

3. Interaction entre humains à travers un système informatisé qui est l'objet d'étude du *Travail Coopératif Assisté par Ordinateur (TCAO)* [Ellis1995]³.

Ces trois niveaux d'interaction mettent l'accent sur l'utilisation de dispositifs informatisés en tant que moyen de réaliser une tâche définie et non pas comme une fin. Cet aspect conforte notre vision de l'informatique, autour de laquelle nous avons l'intention de conduire notre démarche de modélisation. Dans tout échange de message entre humains via des processus artificiels, l'informatique joue le rôle d'intermédiaire dans le processus de communication. Toute sa puissance réside dans l'intelligence avec laquelle se nouent ces relations.

1.2.2 Les deux modes d'interaction

L'interaction entre agents peut être liée soit au temps, soit au séquençement d'événements. Dans le cas où l'interaction est liée au temps, il s'agit d'une interaction en mode **synchrone**. Dans ce mode, un agent qui émet un message se met dans l'état d'attente d'une réponse. Il s'agit d'un mode bloquant. Une conversation est un cas typique d'interaction en mode *synchrone*. Deux interlocuteurs ou, plus généralement, deux agents qui entretiennent une conversation, s'échangent des messages et se mettent en état d'attente d'une réponse après chaque message émis. Si les délais d'attente deviennent trop importants, les agents peuvent adopter un autre comportement. Ils peuvent décider de sortir du mode bloquant pour engager d'autres actions pendant les périodes d'attente. Dans ce cas, l'interaction n'est plus liée au temps mais au séquençement des événements car la conversation qui a été interrompue peut reprendre à la suite d'une notification. Il s'agit maintenant d'une interaction en mode **asynchrone**. L'agent peut émettre un message et poursuivre une autre tâche au lieu d'attendre une réponse à son message car ce mode est non bloquant. La diffusion d'un article est un cas d'interaction en mode *asynchrone*. L'auteur de l'ar-

³dont l'équivalent anglais est *Computer Supported Cooperative Work (CSCW)*.

ticle publiée à destination d'un panel de lecteurs en leur accordant un temps suffisamment long pour la réflexion.

En pratique, tout scénario de collaboration est une alternance entre le mode d'interaction *synchrone* et le mode d'interaction *asynchrone*. Par exemple, dans le cas de l'élaboration collaborative d'un contenu documentaire, des personnes peuvent au départ s'échanger des documents en mode *asynchrone* puis discuter de leurs avis respectifs en mode *synchrone* (ex. visualisation partagée d'un document pour valider des modifications en temps réel). De ce fait, un service *synchrone* qui apporte une modalité de communication supplémentaire renforce le couplage de l'interaction entre ces deux personnes. Figure 1.2 illustre le déroulement d'un cycle avec alternance des deux modes d'interaction. La première étape consiste à recevoir les propositions des membres en mode *asynchrone* et de laisser le temps à la réflexion pour la lecture et l'annotation des différentes contributions. Dans la seconde étape, le mode *synchrone* est plus propice à la discussion et vise à entériner ces différentes propositions et leurs amendements respectifs en un document qui peut servir de référence pour la suite. Ce cycle peut se répéter avec, à chaque fois, une nouvelle base de référence construite sur les discussions antérieures. Le mode *synchrone* s'avère plus efficace pour vérifier la convergence ou non des idées proposées par les différents membres.

Le travail collaboratif se déroule en sessions durant lesquelles peuvent être exposés des points de vue et des commentaires mutuels sur les propositions des participants. Cette activité de groupe peut, par exemple, être décomposée en trois périodes :

- **Période de préparation** durant laquelle les membres soumettent leur contribution au groupe.
- **Période de réflexion** durant laquelle chacun se consacre à la lecture et à l'annotation des contributions des autres membres.
- **Période de discussion** durant laquelle les membres se concertent pour entériner ou rejeter des propositions (textes, amendements, etc.).

A l'issue de la troisième période, la discussion qui porte sur les différents avis et annotations peut converger vers un nouveau document consensuel qui entérine l'ensemble des avis. Ce nouveau document peut, à son tour, servir de base de référence dans un nouveau cycle de travail collaboratif. Ceci est un exemple de modèle idéal servant à illustrer le processus de collaboration. Il ne tient pas compte des effets non-déterministes qui se produisent dans tout processus de collaboration.

1.2.3 Modèles de collaboration

L'ordonnancement des tâches à accomplir et la coordination des différents acteurs impliqués dans un processus de collaboration peut se modéliser par un flux opérationnel⁴. Ce flux peut être *directif* s'il applique des procédures pré-établies ou *ad hoc* si les acteurs interviennent dans la décision du cheminement. Ce dernier revêt un intérêt plus particulier dans notre démarche de modélisation car le cheminement est dynamique, ce qui le rend potentiellement auto-adaptatif.

Le modèle *STROBE*

Le modèle *Stream Objects Environment (STROBE)* décrit, de façon formelle, un processus d'interaction entre agents, en tenant compte d'effets comportementaux humains et de leurs conséquences sur le système [Cerri1996, Cerri1999]. Dans *STROBE* un agent peut être soit artificiel soit humain. Chaque agent dispose d'un environnement cognitif (*i.e* une mémoire persistante) de l'état de toutes les conversations menées avec d'autres agents. Lorsqu'un agent pose une question, il s'opère un changement d'état dans le système (il s'agit ici de l'attente d'une réponse au niveau de cet agent). L'absence de réponse à une question peut influencer sur le comportement de cet agent (savoir que l'autre n'est pas en mesure de répondre). Cette absence de réponse a un effet tangible sur l'évolution du système puisqu'elle engendre des comportements non déterministes (par exemple la recherche d'une réponse par un autre moyen).

⁴Le terme anglais *workflow* sert à décrire ce processus.

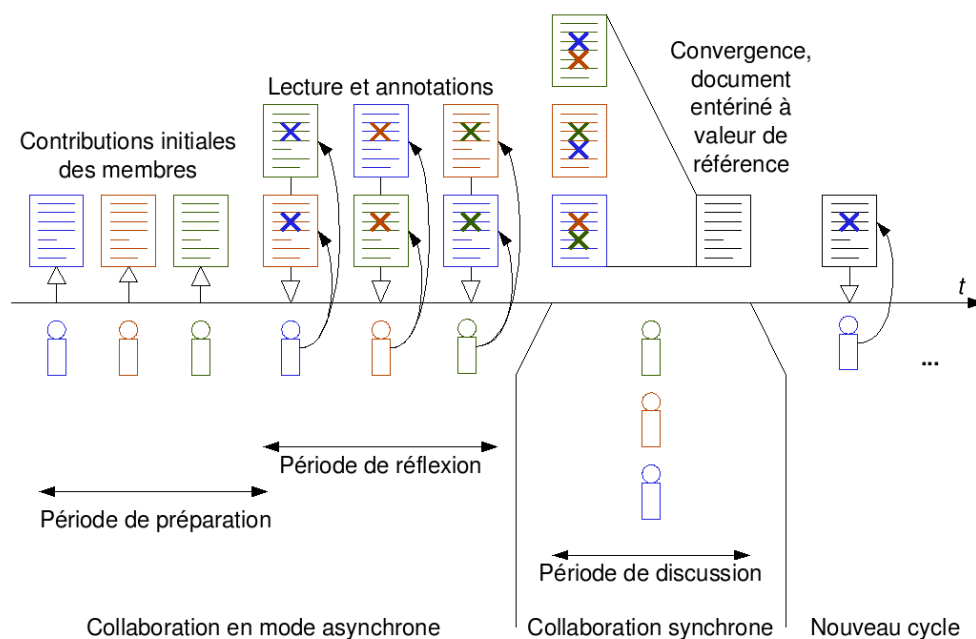


FIG. 1.2 – Déroulement d'un processus de collaboration

Ce type de comportements est illustré par la figure 1.3. La partie gauche de la figure représente une transaction simple. L'agent *A* pose une question à l'agent *B* qui lui renvoie sa réponse. Les changements d'état du système interviennent au niveau des agents. Par exemple, après avoir émis sa question, l'agent *A* est en attente d'une réponse. Sur la partie gauche, la réponse à la question est obtenue. Sur la partie droite, en cas de non aboutissement de la question, la même question est posée à un autre agent *C*.

Le modèle *Orchestra*

Orchestra [David & Chalon2005] propose un formalisme pour exprimer des situations de collaboration, dans lequel les interactions agissent en tant que processus de transformation du modèle. Ce formalisme comprend quatre concepts :

- le concept d'**acteur**, une instantiation d'un ou de plusieurs rôles correspondant chacun à un élément de base du comportement humain dans le système ;
- le concept de **processus** organisés dans un réseau avec des états et des transitions ;
- le concept d'**artefact** qui peut se présenter sous la forme d'un outil ou d'un objet : l'outil est un instrument utilisé pour accomplir une tâche, l'objet est une entrée ou une sortie de la tâche ;
- le concept de **contexte** est composé d'un ensemble de trois aspects : la plateforme, la situation (logique ou physique) et les préférences de l'utilisateur.

Ce modèle est un autre cas typique qui illustre l'importance de l'interaction dans le comportement global du processus de collaboration et, par de là, nécessite une approche non pas de nature proprement analytique mais une approche systémique ou mixte.

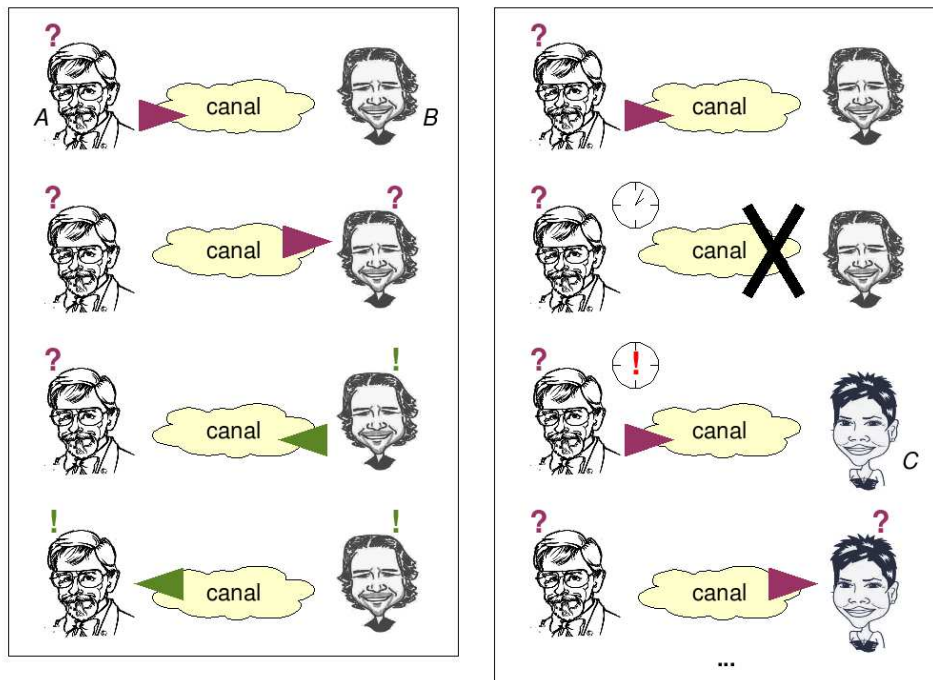


FIG. 1.3 – Interaction de type question-réponse entre deux agents

1.3 Systèmes complexes

Le comportement global d'un groupe d'agents est fortement lié aux interactions entre ces agents. La modélisation d'un tel comportement ne peut être exprimée de façon réductionniste comme c'est le cas avec l'approche analytique car celle-ci ne permet de déterminer des comportements qu'à partir d'hypothèses préalablement établies et acceptées comme telles. Or, il n'est pas envisageable de poser des hypothèses qui anticiperaient sur le comportement d'un groupe comme si ce comportement répondait à une causalité linéaire⁵. Ainsi, il est préférable d'assimiler le comportement d'un groupe à un système complexe dont la structure des relations et les propriétés sont émergentes. Dans un système complexe, les mêmes conséquences peuvent avoir plusieurs causes et les mêmes causes peuvent avoir des effets différents [Le Moigne1990a, Le Moigne1990b]. Pour cela, il nous semble utile de rappeler ici quelques éléments de la théorie des systèmes.

1.3.1 Aux origines de la théorie des systèmes

Pendant que DESCARTES divisait les problèmes en une série d'éléments et conduisait par ordre sa pensée du plus simple au plus complexe [Descartes1996], PASCAL s'interrogeait sur la pertinence de comprendre les parties sans en comprendre le tout et inversement⁶. A sa manière, PASCAL a posé les bases de la théorie des systèmes.

⁵La causalité linéaire et la causalité circulaire sont deux notions qui s'opposent. La première est une caractéristique des processus déterministes, à savoir la relation de cause à effet. La seconde est une caractéristique des systèmes complexes qui décrit l'effet de rétroaction d'un système.

⁶Donc toutes choses étant causées et causantes, aidées et aidantes, médiates et immédiates et toutes s'entretenant par un lien naturel et insensible qui lie les plus éloignées et les plus différentes, je tiens impossible de connaître les parties sans connaître le tout, non plus que de connaître le tout sans connaître particulièrement les parties (Pensées, Oeuvres Complètes, p.527 [Pascal1963])

La théorie des systèmes, baptisée plus tard la *systémique*, a véritablement été introduite par un biologiste LUDWIG VON BERTALANFFY pour qui le *déterminisme causal* s'oppose aux *processus stochastiques* [von Bertalanffy1973]. Dans le *déterminisme*, l'état final dépend des conditions de départ. Dans un *processus stochastique* le principe d'*équifinalité* et d'organisation au hasard des circonstances favorables permet d'atteindre un état final à partir de différents états initiaux et quels que soient les voies et moyens.

L'approche systémique a été par la suite étendue au domaine de la théorie de la communication par un anthropologue GREGORY BATESON qui compare le comportement d'un algorithme avec celui d'une stratégie [Bateson1977]. Dans un algorithme, le comportement est exhibé à partir de règles et de réglages fixés *a priori*. Dans une stratégie, les ajustements sont successivement établis selon l'expérience acquise *a posteriori*. Une stratégie a un comportement moins prévisible qu'un algorithme car la stratégie se fixe un but quel que soit le parcours, alors que l'algorithme est déterministe, il dépend des conditions initiales. La définition d'un système nous est donné par WATZLAWICK [Watzlawick *et al.*1972] :

SYSTÈME

Un système se définit comme un ensemble d'objets et de relations entre ces objets et leurs attributs ; les objets sont les composants ou éléments du système, les attributs sont les propriétés des objets, et les relations sont ce qui fait tenir le système. Si les objets sont des êtres humains, les attributs qui permettent de les identifier dans le système sont leurs comportements de communication (WATZLAWICK, *Une logique de la communication*, p. 120).

1.3.2 Principes de systémique

Un système complexe est caractérisé par le fait que son comportement est non-déterministe car il est induit par ses interactions. Une communauté ou plus généralement un groupe de personnes est un exemple de système complexe où chaque individu peut prendre des décisions, avoir un effet sur le comportement du groupe et être en retour influencé par le groupe lui-même. Un groupe est aussi un système ouvert car il existe des relations internes et externes entre les membres de ce groupe : les membres communiquent entre eux et communiquent aussi avec des personnes étrangères au groupe. Plusieurs principes de la systémique marquent la rupture avec la méthode cartésienne. En nous basant sur une étude approfondie [Calderoni2002], nous pouvons en citer quelques-uns parmi les plus importants :

- Le principe **holistique** exprime que le tout est plus que la somme de ses parties du fait de propriétés émergentes qui n'appartiennent à aucune des parties, mais qui naissent de leurs interactions. Le membre d'un groupe agit sur le groupe et réciproquement.
- Le principe **téléologique** est le principe systémique qui exprime que tout système évolue afin d'accomplir sa vocation intime en fonction des opportunités offertes par sa stratégie. Cela signifie qu'un système évolue comme s'il poursuivait un but, une finalité mais ces finalités sont relatives et variables dans le temps. La constitution d'un groupe se fait sur la base d'objectifs et de motivations qui peuvent être clairement exprimés ou tacites.
- **L'homéostase** (ou l'homéostasie) est une propriété d'un système capable de maintenir sa structure et ses fonctions dans un état stationnaire par l'intermédiaire d'une multitude d'équilibres dynamiques dont le but est de préserver l'unité du système [Cannon1932]. L'équilibre d'un groupe s'établit par le fait d'actions qui ont un effet de régulation des comportements. Ce sont des boucles de rétroaction du système.
- **L'autopoïèse** est un principe par lequel tout système est vu comme un réseau de processus dont les transformations et les interactions régénèrent continuellement le réseau qui les a produits. L'idée d'autopoïèse s'appuie sur l'homéostasie et la développe dans deux directions importantes : d'abord dans le fait de transformer toutes les références de l'homéostasie en références internes au système

lui-même, ensuite par l'affirmation que l'identité du système, que nous appréhendons comme une unité concrète, provient de l'interdépendance des processus. L'autopoïèse ou réorganisation permanente est une catégorie applicable à l'ordre biologique, et, par extension, à l'ordre social humain. Un groupe d'individus, vu comme un ensemble de systèmes naturels et artificiels, est un système autopoïétique [Maturana & Varela1984].

- **L'auto-organisation** est l'aptitude d'un système à produire ses propres principes d'organisation à partir d'un modèle d'interactions conduisant à une structuration de l'ensemble. A la suite des premières recherches formelles sur la logique des systèmes auto-organiseurs, NORBERT WIENER a introduit la **cybernétique** comme principe auto-organisateur pour les systèmes à temps et à dimension définis. [Wiener1948]. De cette manière, un système se réorganise de façon permanente en maintenant sa cohésion interne. Sa structure et ses interactions se perpétuent d'elles-mêmes. Comme elles s'auto-renouvellent, elles sont extrêmement autonomes [Morin2005].
- **L'émergence** désigne l'apparition de propriétés d'un système complexe dont la connaissance détaillée de ses parties ne renseigne pas complètement. Tout comme l'idée *cybernétique* de rétroaction, l'émergence permet d'appréhender les raisons qui assurent l'organisation et l'autonomie d'un système. Une organisation sociale dépend de ses individualités en tant que parties intégrantes d'un tout. L'intelligence des rapports sociaux est la clé de la dynamique du système. L'émergence, résultante de la nature organisationnelle du système, n'est pas un processus déterministe ni chaotique sensible aux conditions initiales du système. Cette notion d'émergence en systémique suscite un intérêt profond dans la pensée contemporaine [Morin1991, Morin2005].

Si l'on applique ces principes de systémique à un groupe d'individus, nous pouvons en tirer un certain nombre de constats utiles pour la suite. Tout d'abord, l'*auto-organisation* est une propriété émergente de l'activité d'un groupe liée essentiellement au principe *holistique* (les interactions provoquent une *causalité circulaire* et continue entre les membres et leur groupe d'appartenance) et au principe *téléologique* (chaque membre peut apporter un élément de contrôle de la finalité des objectifs fixés par le groupe et donc influencer sur la stratégie). Ensuite, la cohérence d'un groupe tient du fait de la solidité de ses relations et de leur adaptabilité dans le temps pour atteindre un *état homéostatique* et une réorganisation dynamique qui explique qu'un groupe d'individus est un *système autopoïétique*. Tant que ces conditions ne sont pas réalisées, un groupe ne demeure qu'une entité hiérarchisée arbitrairement et, par conséquent, figée et sans aucune capacité d'adaptation au changement de ses conditions d'environnement. Ces résultats visent à enrichir l'étude des caractéristiques de l'*espace collaboratif ubiquitaire* (cf. chapitre 2).

1.4 Approche socio-constructiviste

La théorie des systèmes met en exergue l'importance de la vision centrée-interaction dans l'étude d'un comportement social. Ceci doit être complété par une meilleure compréhension des facteurs influant cette dynamique. L'approche socio-constructiviste nous apporte quelques bases clés telle que la contextualisation qui place le contexte en préalable à toute interaction.

1.4.1 Eléments de socio-constructivisme

Apprendre met en jeu le fonctionnement mental [Piaget1967] et social [Vygotsky1978]⁷. Au départ, le *constructivisme* dénote la construction de connaissance lorsque le sujet impose ses propres structures mentales à l'objet qu'il construit [Piaget1967]. Etant donné que la réflexion d'une personne isolée ne peut mener aussi loin que celle d'un groupe d'individus, l'apport de l'élément social au *constructivisme* devient une évidence. Le social influence aussi sur l'échange de résultats d'expériences, ouvre de nouveaux champs exploratoires et forge ainsi une connaissance nouvelle par voie de conséquence. Le

⁷Deux autres facteurs sont parfois cités : le fonctionnement affectif et le fonctionnement culturel.

socio-constructivisme conçoit ainsi la construction de connaissances à travers un système d'interactions sociales [Vygotsky1978]. L'acquisition de connaissances *a posteriori* s'opère sur deux plans :

- sur le **plan individuel** par la réflexion et le raisonnement de façon introspective (conscience individuelle) ;
- sur le **plan collectif** à travers un système d'échanges, de conversations, dans un contexte de collaboration ou de confrontation (conscience collective).

Par la suite, la recherche dans le domaine de la communication s'est intéressée à la dynamique du *socio-constructivisme*. Des théories sur les mécanismes d'échange de connaissances ont été élaborées au *Mental Research Institute de Palo Alto (MRI)* sous l'impulsion notamment de WATZLAWICK [Watzlawick *et al.*1972]. Parmi ces théories, les modèles sont des éléments fondamentaux du socio-constructivisme pour représenter et communiquer une vision du monde. Le langage naturel n'est pas suffisant, la modélisation abstraite de phénomènes permet de mieux expliquer des réalités concrètes. Un autre cadre d'étude du *MRI* est la conversation. Que l'on veuille apprendre ou enseigner quelque chose, nous sommes sans cesse amenés à conduire des conversations. Au cours d'une conversation, quel que soit le rôle des parties en présence, l'acquisition de nouvelles connaissances s'opère chez tous les interlocuteurs. Lorsqu'un jeune professeur enseigne une discipline, il valide sa méthode pédagogique au travers d'interactions avec ses étudiants par le jeu de questions et de réponses, par exemple. L'étudiant acquiert ainsi la connaissance de la discipline alors que le professeur acquiert une meilleure connaissance de la validité de sa méthode pédagogique.

1.4.2 Contextualisation systémique

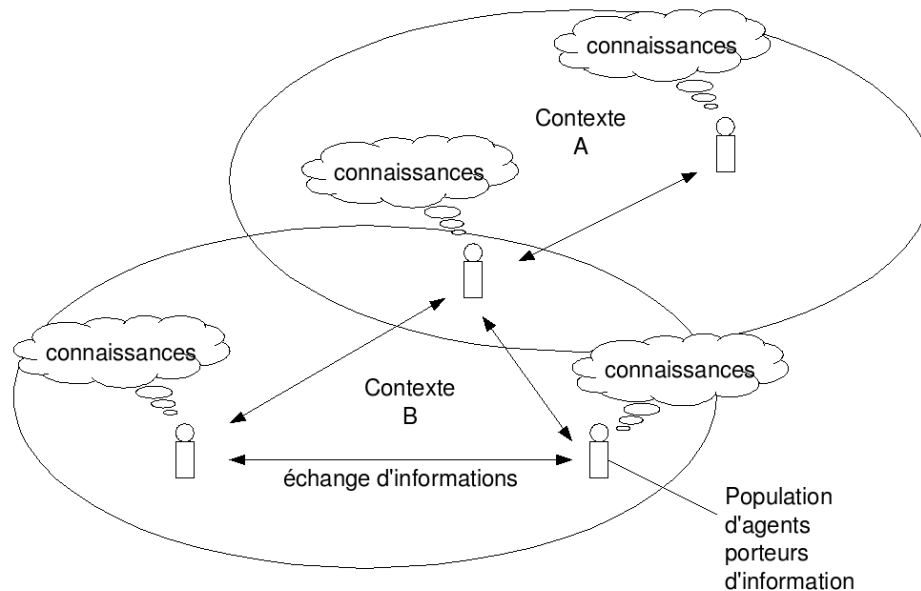


FIG. 1.4 – Contextualisation

Aucun phénomène ou objet scientifique ne peut être connu isolément, dans un vide environnemental. Le sens émerge toujours d'une contextualisation, c'est-à-dire d'une mise en relation de quelque chose avec quelque chose d'autre [Berger & Luckmann1986]. Ainsi, un processus de contextualisation effectue la mise en relation d'un phénomène avec des éléments provenant de l'environnement. Ce processus est

fondamental dans le cadre d'un travail collaboratif puisque des significations peuvent émerger à partir d'interactions qui se produisent dans un référentiel commun. Nous adoptons le terme *espace collaboratif* pour qualifier ce référentiel. La contextualisation a donné lieu à l'élaboration de différentes méthodes [Mucchielli & Noy2005]. Celle qui présente le plus d'intérêt pour l'approche socio-constructiviste est la méthode de *contextualisation systémique*. En effet, toute forme de communication s'insère dans un système au sein duquel un phénomène prend son sens par la mise en relation avec des phénomènes concomitants. Comme le montre la figure 1.4, la *contextualisation systémique* est un travail d'échanges récurrents entre des acteurs placés en situation de communication.

1.5 Conclusion : mise en relation de compétences réparties

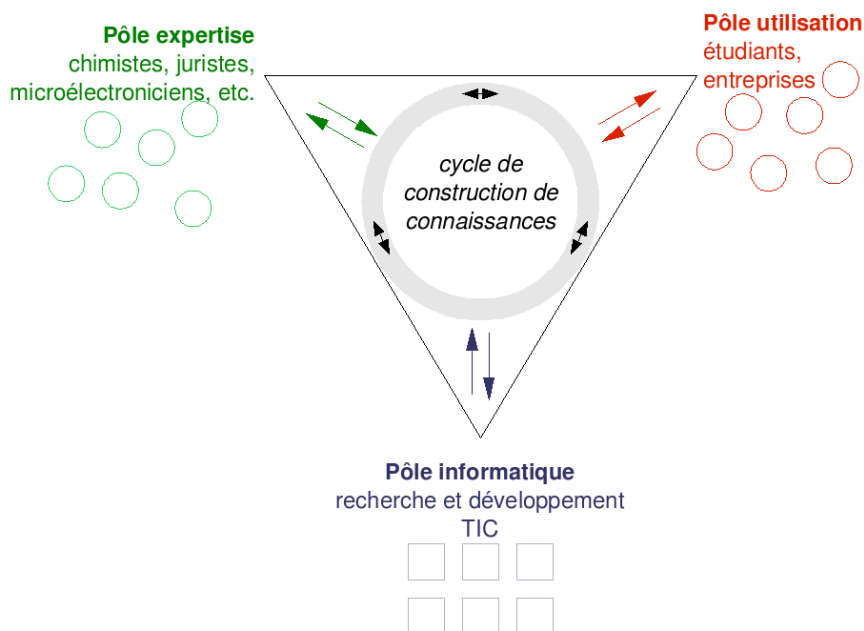


FIG. 1.5 – Mise en relation de compétences

Nous avons tenté dans ce chapitre de dégager la quiddité de l'*intelligence collective* de différentes manières. D'abord, en la replaçant dans son cadre philosophique, nous avons identifié trois facteurs (le technique, le social et le sémiotique) intervenant dans l'émergence de l'*intelligence collective*. Puis, en étudiant différents modèles d'interaction entre des utilisateurs humains et un système informatisé, nous avons conclu qu'il s'agit d'un phénomène non déterministe, donc difficilement étudiable par une approche purement analytique. Ensuite, nous nous sommes focalisés sur le facteur technique en caractérisant la notion de système dans lequel se produisent ces interactions pour comprendre les effets du système, notamment l'auto-organisation, sur la condition d'émergence de l'*intelligence collective*. Et enfin, nous nous sommes focalisés sur le facteur social en explorant la notion de contextualisation et son importance dans le processus socio-constructiviste. En conclusion, nous percevons l'*intelligence collective* comme le résultat d'une activité *socio-constructiviste* pour laquelle le groupe est un système ouvert dont les membres interagissent dans un référentiel commun. Par la suite, nous définissons ce référentiel comme l'*espace collaboratif* du groupe. Selon le principe de la *cartographie des compétences* [Authier & Lévy1992], il est envisageable de voir émerger une *intelligence collective* d'un tel système. A

des fins de présentation, nous avons procédé à une simplification de ce principe avec trois pôles distincts et leurs relations représentés sur la figure 1.5.

- **pôle expertise** : Ce pôle est constitué d'agents fournisseurs de l'expertise d'un domaine (enseignants, scientifiques, ingénieurs, juristes, etc.) qui collaborent continuellement pour construire et améliorer un matériel pédagogique de base.
- **pôle utilisation** : Ce pôle est constitué d'agents demandeurs d'une expertise (entreprises, étudiants, etc.) qui exploitent le matériel pédagogique et qui peuvent aussi demander au **pôle expertise** de clarifier certains points obscurs du matériel pédagogique.
- **pôle informatique** : Ce pôle est constitué de services interactifs qui concourent à ajouter de la valeur aux contenus produits par les deux autres pôles. Ces services réalisent diverses opérations telles que l'analyse syntaxique et sémantique pour en dégager des structures permettant de favoriser la mise en relation des agents. Il peut s'agir, par exemple, de moteurs de recherche mais aussi d'automates de génération de contenus.

Ces trois pôles vont nouer des relations, facilitées par un *espace collaboratif*, pour accomplir un objectif de construction collaborative de connaissance. Il peut s'agir d'un groupe d'expert qui réalise des simulations numériques. Dans ce cas, les deux pôles expertise et informatique seront sollicités. Il peut s'agir aussi d'un groupe d'étudiants qui recherche une expertise sans savoir d'où celle-ci provient. Dans ce cas, les deux pôles utilisation et informatique seront sollicités. Enfin, il peut s'agir d'un groupe d'étudiants et d'experts dont les membres interagissent pour améliorer un matériel pédagogique. Dans ce cas, les deux pôles expertise et utilisation seront plus fortement sollicités.

Une condition essentielle pour que ces scénarii se réalisent, est que l'*espace collaboratif* soit accessible en tout lieu et à tout instant pour que chacun puisse y apporter sa contribution et y exercer ses compétences. Sur cette base, nous allons définir le concept d'*espace collaboratif ubiquitaire* au chapitre suivant (*cf.* chapitre 2).



CHAPITRE 2

ESPACES COLLABORATIFS UBIQUITAIRES

C'est l'inconnu qui m'attire. Quand je vois un écheveau bien enchevêtré, je me dis qu'il serait bien de trouver un fil conducteur.
(Le Monde du 23 octobre 1991)
Pierre-Gilles de Gennes

LE TRAVAIL Coopératif Assisté par Ordinateur^a a donné lieu, depuis peu, à l'émergence de nouveaux concepts. Le *collecticiel* est apparue comme une extension du concept d' *Environnement Numérique de Travail (ENT)* qui a commencé à se répandre depuis 2002. Dans ce chapitre, nous analysons les solutions existantes en termes de fonctionnalités et de niveau d'intégration, puis nous examinons les limites de ces solutions. Enfin, sur la base de cette analyse, nous décrivons les caractéristiques du concept d' *espace collaboratif ubiquitaire* et nous posons quelques recommandations pour en favoriser le déploiement.

^aTCAO ou CSCW (Computer Supported Cooperative Work).

Table des matières

2.1	Panorama	30
2.1.1	Environnements Numériques de Travail	30
2.1.2	Espaces collaboratifs à l'échelle européenne	31
2.1.3	Vers des <i>espaces collaboratifs ubiquitaires</i>	31
2.2	Services d'un <i>espace collaboratif ubiquitaire</i>	32
2.2.1	Typologies	32
2.2.2	Administration des groupes, membres, et droits	33
2.2.3	Visualisation partagée	34
2.2.4	Partage de contenus	35
2.2.5	Communication audio-visuelle multiplexée	36
2.2.6	Notifications	37
2.3	Limites des solutions existantes	37
2.3.1	Intégration partielle des services	37
2.3.2	Rigidité dans l'organisation des groupes	37
2.3.3	Harmonisation technologique pénalisante	38
2.3.4	Insuffisance opérationnelle	39
2.4	Caractérisation	39
2.4.1	Caractère multimodal de l'IHM	39
2.4.2	Caractère d'immanence	40
2.4.3	Caractère d'ubiquité	41
2.5	Recommandations pour le déploiement	41
2.5.1	Sécurité	41
2.5.2	Disponibilité du service	41
2.5.3	Passage à l'échelle	41

2.1 Panorama

De plus en plus d'activités comme l'ingénierie, la maintenance ou encore l'enseignement ont recours au *Travail Coopératif Assisté par Ordinateur (TCAO)*¹ [Borghoff & Schlichter2000] (c.f section 1.2.1). Le *TCAO* se caractérise par un ensemble d'applications et d'utilisateurs, répartis géographiquement, qui se retrouvent en situation de coopérer par l'usage de moyens leur permettant d'être virtuellement localisés au même endroit [Applegate1991]. Nous distinguons deux types de moyens mis en œuvre pour le *TCAO* :

- L' *Environnement Numérique de Travail (ENT)*
- Le *collecticiel* ou l'*espace collaboratif*

Ce panorama présente succinctement les *ENT* dont il existe des solutions opérationnelles. Ceci a pour but de fixer le contexte afin de déterminer au mieux les exigences techniques requises par les *espaces collaboratifs* qui sont plus représentatifs des objectifs fixés dans le cadre de cette thèse.

2.1.1 Environnements Numériques de Travail

Le terme *ENT* recouvre toute une terminologie assez hétérogène autour des notions d'*apprentissage collaboratif*, *eLearning*, *ePortfolio*, *eScience*, *espace virtuel*, etc. De manière générale, ces concepts visent à améliorer le processus d'apprentissage par des méthodes formelles ou informelles. Au niveau international, un rapport de 2003 de l' *Institut de l'Audiovisuel et des Télécommunications (IDATE)* nous renseigne sur les choix adoptés par quelques pays [IDATE2003]. Les systèmes éducatifs de ces pays ont des projets visant au déploiement et à l'amélioration des *TIC* pour l'enseignement. Dans tous les cas, il s'agit d'environnements d'apprentissage dédiés davantage à l'enseignement et, à un degré moindre, à la collaboration. La France, par son volontarisme politique en la matière, se place en position honorable sur le plan international. A la différence des autres pays, la France, est le seul pays à adopter un schéma directeur général. Trois rapports font date sur l'engagement du ministère de l'*Education Nationale* du point de vue du déploiement des *ENT* :

- *Le Schéma Stratégique des Systèmes d'Information et de Télécommunication 2002-2004* qui est le premier document de référence pour le cadrage et l'évolution des dispositifs *TIC* à l'*Education Nationale* [S3Ia2002, S3Ib2002].
- *Le Schéma Directeur des Infrastructures* qui donne les grandes orientations de l'*Education Nationale* pour son architecture réseau. L'un des objectifs est de soutenir le développement des réseaux régionaux de télécommunications à haut débit avec des capacités de raccordement à *Renater 3* d'au moins 155 Mbits par seconde pour l'ensemble des universités et des établissements de recherche [SDI2002].
- *Le Schéma Directeur des ENT* a été établi afin de dresser les grandes lignes du programme cadre des *UNR* [SDEurl].

Ainsi, des initiatives d'*ENT* se sont multipliées depuis 2002 dans l'enseignement supérieur avec le soutien du programme des *Universités Numériques en Région (UNR)*. Deux d'entre elles ont attiré notre attention.

- *ENCORA*, en région Rhône-Alpes, a été lancée à l'issue du premier appel à projet *Campus Numériques*. L'*Université de Lyon 2* propose, depuis 2004, un *ENT* permettant aux étudiants un accès libre à tout un panel de services collaboratifs [Encoraur].
- *ENTICE*, en région Languedoc-Roussillon auquel est associé le *LIRMM*, a été lancée à l'issue du deuxième appel à projets *Usages de l'Internet* [Enticeurl]. Ce projet s'est achevé fin 2006.

¹En *TCAO*, l'Interface Homme-Machine classique évolue vers une Interface Homme-Machine-Homme de type WYSIWIS (What you see is what I see, ce qui se traduit littéralement par ce que tu vois est ce que je vois). Il s'agit d'une interface multi-utilisateurs selon laquelle les modifications opérées par un utilisateur apparaissent en mode synchrone chez les autres utilisateurs.

2.1.2 Espaces collaboratifs à l'échelle européenne

Si les *espaces collaboratifs* héritent à la base des principes des *ENT*, il s'en démarquent à plusieurs titres. Le terme *espace* plutôt qu'*environnement* exprime mieux l'idée d'une *souplesse de structure*. Le terme *collaboratif*² plutôt que *travail* suggère davantage de *liberté d'action*. Ils en étendent fortement les capacités en matière d'organisation, de dimensionnement et se caractérisent par des propriétés telles que l'*Interface Homme-Machine (IHM) multimodale* ou encore l'*immanence* que nous décrirons plus loin. Par la suite, l'idée d'*espace collaboratif ubiquitaire* prend tout son sens lorsqu'une volonté de déploiement d'infrastructure est envisagée. Toutefois, avant de choisir une infrastructure, il est souhaitable de bien identifier les contraintes imposées à l'architecture.

Dans le cadre du projet européen *European Learning Grid Infrastructure (ELeGI)*, nous avons tenté d'identifier ces contraintes et avons exploré les potentialités techniques de GRID. La méthodologie employée a consisté à dérouler des scénarii d'apprentissage collaboratif tout en évaluant diverses technologies de façon itérative. Les développements ont été menés parallèlement aux évaluations. Ceci a permis d'analyser les comportements des utilisateurs dans leur contexte d'apprentissage. Cette méthodologie s'est révélée être efficace dans la mesure où nous avons pu tirer, dès la première année du projet, des enseignements utiles à réinjecter dans les spécifications techniques. Nous avons observé, par exemple, un fort engouement pour un apprentissage collaboratif de nature *socio-constructiviste*. Ce mode d'apprentissage que nous avons décrit dans la section 1.4 considère l'apprentissage comme un effet de bord de la collaboration [Breuker *et al.*2005] et s'est montré complémentaire aux méthodes formelles d'apprentissage. Alors, plutôt que de structurer des environnements dédiés à un apprentissage formel, nous avons opté pour développer un *espace collaboratif* le plus générique et le plus évolutif possible. Sur le plan technique, la collaboration fait intervenir des interactions en mode *asynchrone* et *synchrone* [Dugénie & Lemoisson2005]. Ceci a définitivement écarté l'idée d'utiliser une technologie WEB traditionnelle.

Les projets comme *ENCORA* et *ENTICE* adressent surtout le partage d'informations réparties avec des accès en mode *asynchrone* et peuvent se contenter d'une technologie WEB. Le service de visualisation partagée en mode *synchrone* adopté par le projet *ELeGI* a demandé un effort supplémentaire pour trouver des solutions.

2.1.3 Vers des espaces collaboratifs ubiquitaires

Selon l'idée qu'une collaboration efficace requiert une architecture combinant les modes d'interaction *synchrone* et *asynchrone*, le projet *ELeGI* s'est intéressé aux services collaboratifs sur GRID comme alternative au WEB. Des solutions partielles d'*espaces collaboratifs* foisonnent sur le WEB. Toutefois ces solutions offrent un panel restreint de services dûs à une limitation intrinsèque du WEB. Bien que certaines caractéristiques innovantes du WEB SÉMANTIQUE et du WEB 2.0 apportent quelques améliorations, de nombreux verrous technologiques persistent et restent sans réponse. Nous avons ainsi conceptualisé les *espaces collaboratifs* avec les propriétés de *multimodalité* pour décrire la capacité à combiner les modes de communication *asynchrones* et *synchrones* et d'*ubiquité* pour décrire la capacité à être accessible en tout lieu et à tout instant. Ces caractéristiques complétées par des propriétés propres à l'*IHM*, telle que la *plasticité*³ ou encore la *capillarité*⁴ [David *et al.*2003b, David *et al.*2003a]. Avec la propriété d'*ubiquité*, les *espaces collaboratifs* ou *collecticiels* visent à l'abolition des dimensions espace et temps. Les principaux objectifs des collecticiels sont d'obtenir des gains de performance, de capitaliser des

²Nous nous référons à la connotation usuelle du terme plutôt qu'à sa dénotation étymologique (collaborer, du latin *cum labore*, travailler avec).

³La *plasticité* décrit la capacité d'une interface à s'adapter à différents types d'environnements [Thevenin1999].

⁴La *capillarité* est une propriété émergente des systèmes de *TCAO* qui inclut l'usage d'interfaces nomades, légères, conscientes du contexte dans lequel elles évoluent (*context-aware*) et disposant d'une capacité d'autonomie décrite par la notion de *proactivité*.

connaissances, d'améliorer des temps de réponse, de partager des compétences ou plus généralement de faciliter le travail à distance [David2001].

2.2 Services d'un espace collaboratif ubiquitaire

Un *espace collaboratif ubiquitaire* se distingue d'un *ENT* par son caractère évolutif à la convenance du groupe qui l'utilise. A titre de comparaison, une métaphore est représentée sur la figure 2.1. D'un côté un labo est prééquipé d'instruments comme c'est le cas avec les *ENT*, de l'autre côté, une salle est vide mais laissée libre aux groupes de l'équiper avec les instruments de leur choix. Cette salle vide correspond à l'état initial d'un *espace collaboratif ubiquitaire* qui se remplit selon les choix des membres au fur et à mesure que le groupe se constitue.



FIG. 2.1 – Métaphore pour comparer *espace collaboratif* et *ENT*

ESPACE COLLABORATIF OU COLLECTICIEL

composition de plusieurs services dont l'objet est de gérer des entités (groupes, membres, droits, services, contenus) ainsi que les interactions et les flux d'information (visualisations, parole, notifications) entre ces entités.

2.2.1 Typologies

Il existe plusieurs typologies de *collecticiels*. Nous avons abordé une typologie basée sur la coordination dans la section 1.2.3 avec la notion de *flux opérationnel* qui peut être *directionnel* ou *ad hoc*. D'autres typologies peuvent être basées sur la *coopération*, sur la *granularité*, ou encore sur la *matrice espace-temps* [David2001]. La *coopération* indique le nombre de participants et l'orientation des interactions

(un vers tous, tous vers un, tous vers tous). La *granularité* peut être de type *temporelle* ou *informationnelle*. La *granularité temporelle* se rapporte au temps qui sépare les interventions des différents acteurs. Dans le cas d'une interaction *asynchrone*, la *granularité temporelle* est conséquente, car l'intervalle de temps entre chaque intervention est très important. La granularité temporelle est minimale dans le cas d'une interaction *synchrone* car, avec un effet dynamique, le basculement des interventions des acteurs peut être plus ou moins rapide. La *granularité informationnelle*, exprime le degré de simultanéité d'accès aux informations. Elle peut être élevée dans le cas d'une interaction *asynchrone*, car il peut s'agir, par exemple, de l'échange d'un document volumineux. Dans le cas d'une interaction *synchrone*, il est préférable de réduire considérablement la granularité informationnelle pour accroître la dynamique. Chaque acteur peut ainsi se focaliser sur une partie précise d'un ensemble plus conséquent traité au niveau du groupe.

Ces typologies nous permettent de caractériser un certain nombre de services indispensables à un *espace collaboratif ubiquitaire*. La figure 2.2 représente une classification de services selon leur mode d'interaction : *asynchrone* sur la partie inférieure et *synchrone* sur la partie supérieure de la figure.

- service de gestion des contenus partagés, données et fichiers ;
- service de gestion des entités membres et autorisations ;
- service de notifications ;
- service de visualisation partagée ;
- service de communication audio-visuel multiplexé ;

Il est à noter sur cette figure que le service de notifications peut être amené à gérer des interactions dans les deux modes. En effet, une notification peut avoir lieu dans le cadre d'une collaboration synchrone (ex. messagerie instantanée) ou dans le cadre d'une collaboration asynchrone (ex. modification d'un contenu partagé). La figure 2.3 est un exemple typique de scénario de collaboration s'appuyant sur plusieurs de ces services. Sur cette figure est représenté deux utilisateurs qui communiquent au moyen de services synchrones de visualisation partagée couplée avec un service de communication audio-vidéo. Leur objectif est d'élaborer un modèle de calcul et de réaliser des simulations en discutant directement des résultats.

2.2.2 Administration des groupes, membres, et droits

Pour gérer des entités avec une relation d'appartenance (membre de tel groupe) et une relation de privilèges (autorisations sur tel service), plusieurs options peuvent se présenter. La première option est de gérer ces entités dans une structure arborescente. Cette option a l'avantage d'être simple mais elle présente une structure peu évolutive. C'est pourquoi les solutions qui s'appuient sur des bases de données relationnelles sont de plus en plus répandues. Ce service doit principalement tenir compte de deux contraintes :

1. L'appartenance à un groupe n'est pas exclusif. Les membres d'un groupe peuvent aussi être membres d'autres groupes avec des privilèges différents sur les services de ces autres groupes.
2. La gestion des identités des membres d'un groupe n'est pas forcément centralisée.

La première contrainte nécessite un service qui gère une relation ternaire entre les entités agent, groupe et service. Cette relation d'autorisation peut être un jeu de permissions sur des services ou, à un niveau de granularité plus gros, cette relation peut être assimilée à un rôle. La seconde contrainte nécessite un service capable de propager et de retrouver les identités des agents. Ceci consiste à interroger l'établissement d'origine de l'agent et obtenir les attributs pour gérer l'accès à des services ou personnaliser le profil.

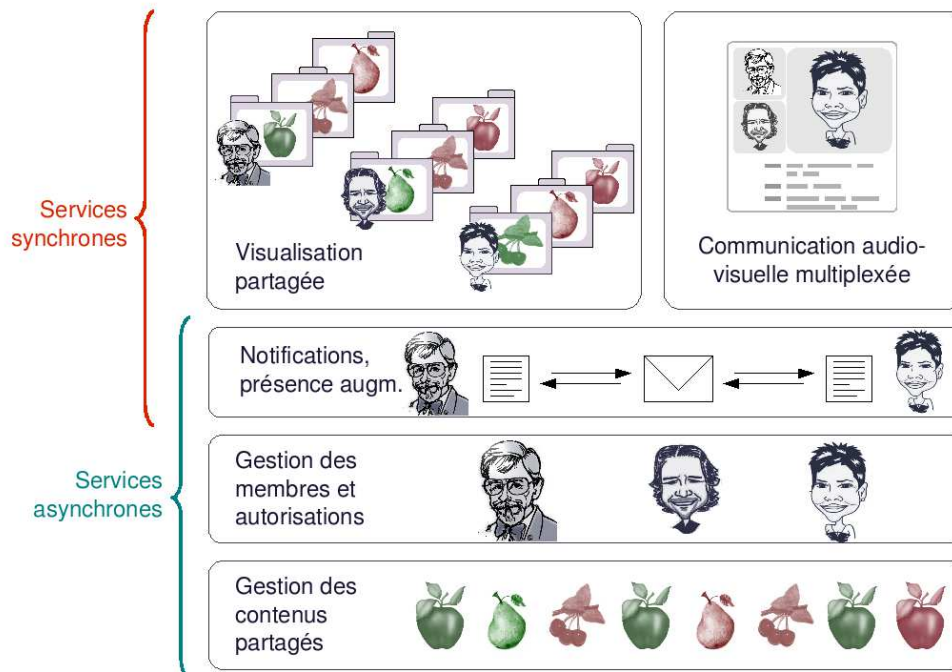


FIG. 2.2 – Structure des services d'un *espace collaboratif ubiquitaire*

2.2.3 Visualisation partagée

Un service de visualisation partagée consiste à diffuser le contenu d'un écran et transmettre les événements clavier et souris à travers une connexion réseau de façon à pouvoir agir depuis différents endroits sur un espace visuel commun.

Solutions existantes et leurs inconvénients

Les solutions existantes aujourd'hui consistent en une suite d'outils à installer sur un poste client pour partager un bureau local. Nous avons identifié de nombreuses solutions de ce type dont les plus connues sont :

- GotoMeeting (<http://www.gotomeeting.com>)
- GatherPlace (<http://www.gatherplace.net>)
- Glance (<http://www.glance.net>)
- Mikogo (<http://www.mikogo.com>)

Toutefois, ces solutions ont été conçues pour que des entreprises puissent effectuer des réunions en lignes avec au maximum une dizaine de participants. De ce fait, elles présentent de nombreux inconvénients pour couvrir le besoin correspondant à notre vision conceptuelle d'*espace collaboratif*. D'une part, certains de ces environnements utilisent des technologies lourdes, onéreuses et quelque peu dépassées (c'est le cas avec *Gotomeeting* qui utilise *Citrix*), ils sont limités à une plateforme unique (*Microsoft* pour la plupart) et restreints à l'usage de quelques applications de bureautique. D'autre part, comme le montre la figure 2.4, le partage de visualisation depuis un poste personnel comporte des risques inhérents à la sécurité et à la qualité de service.

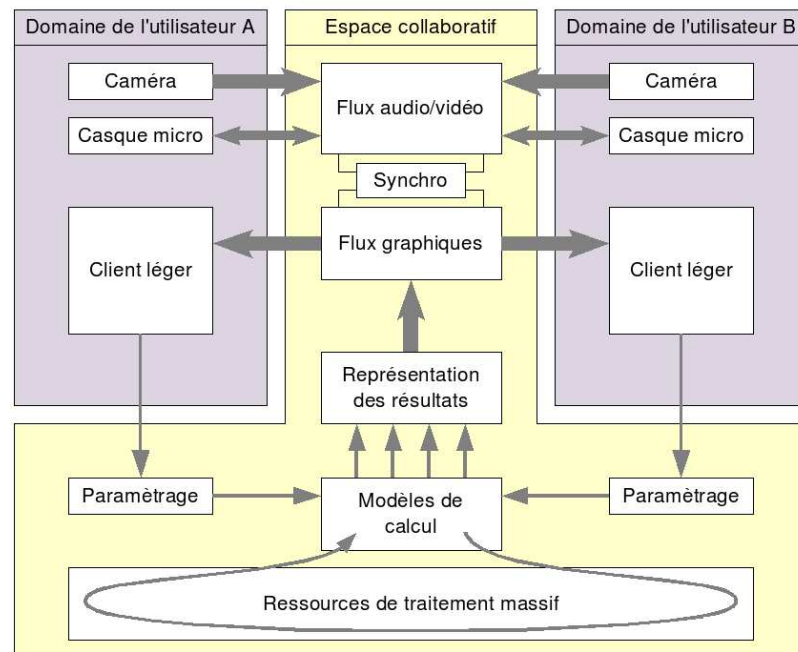


FIG. 2.3 – Exemple caractéristique d'espace collaboratif ubiquitaire

Comment remédier à ces inconvénients

Toutes les solutions de visualisation partagées, citées plus haut, présentent de nombreux inconvénients d'ordres opérationnels. Pour comprendre les contraintes techniques mises en jeu, sur la figure 2.4 la partie haute du diagramme montre comment les solutions qui font appel à de la ressource du poste client sont dans l'incapacité de servir convenablement les autres clients. En effet, sur ce diagramme, avec une collaboration à quatre, chacun peut prendre la main à tour de rôle. En d'autres termes, l'un des agents occupe le rôle de serveur (celui qui diffuse son écran) et les trois autres occupent le rôle de client. L'intensité du trafic va devenir importante au niveau du serveur et ceci proportionnellement au nombre de participants (clients à servir). La capacité de la ligne aux terminaisons du réseau capillaire est dans la plupart des cas insuffisamment dimensionnée. Sur une liaison de longue distance, le goulot d'étranglement se situe le plus souvent à cet endroit du réseau. Un autre inconvénient à ne pas négliger est la sécurité. Ouvrir son poste de travail à toutes sortes d'accès extérieurs n'est souvent pas autorisé par les administrateurs de réseaux. Ceci se justifie par le fait qu'une intrusion mal intentionnée est facilitée par les failles de sécurité connues des systèmes d'exploitation des ordinateurs personnels.

La partie basse du diagramme figure 2.4 montre une solution d'architecture qui rationalise l'utilisation des ressources système et réseau. De cette façon, le trafic est distribué entre les services et redirigé vers les agents. Les clients sont sans état car c'est le service qui gère l'état des conversations. Un agent X_i dans l'état actif interagit avec les autres agents par l'intermédiaire de son service associé S_i . Par la suite, nous verrons comment les serveurs peuvent être virtualisés et remplacés par des *services à état* afin de rendre cette architecture parfaitement ubiquitaire (cf. chapitre 3).

2.2.4 Partage de contenus

Le partage de contenus distribués, sous forme de données brutes ou de fichiers, est un service collaboratif essentiel dont la mise en œuvre n'est pas triviale. En effet, ces contenus sont, la plupart du temps,

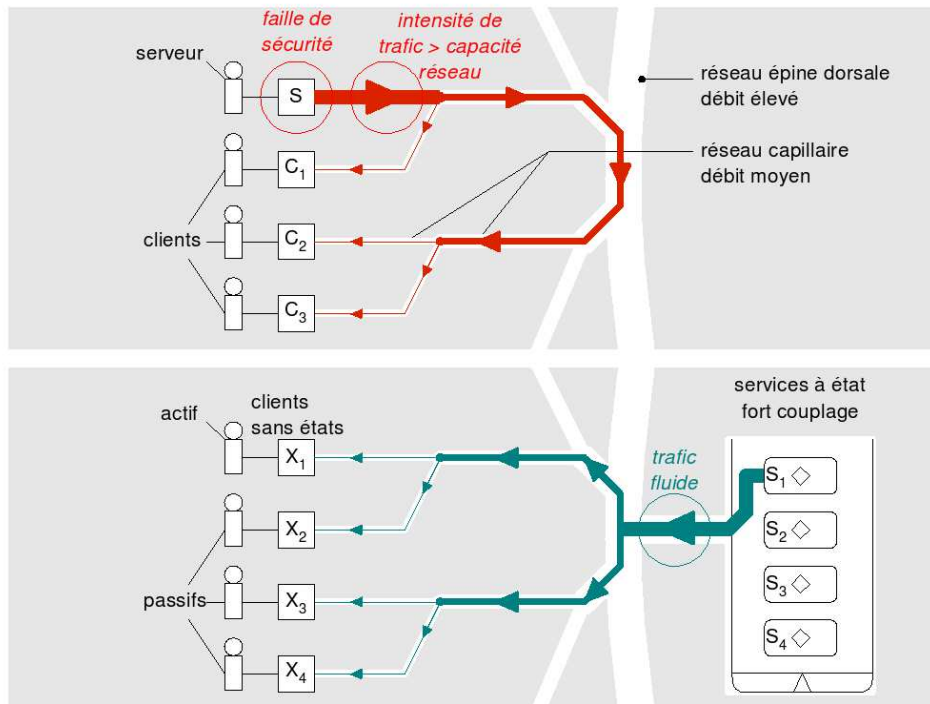


FIG. 2.4 – Comparaison de deux approches pour la visualisation partagée

saisis puis traités à partir de copies résidentes sur l'espace disque des postes des utilisateurs. Dans un contexte collaboratif, ces contenus peuvent être saisis directement dans l'espace partagé comme c'est le cas avec *wiki*⁵. Lorsque des contenus sont saisis et traités localement, il doivent être ensuite transférés vers une zone partagée. Toutefois, le suivi des versions et l'interopérabilité des formats ne va pas toujours de soi. Idéalement, un tel service devrait, à partir d'une interface conviviale, permettre le transfert de données vers l'espace partagé disposant de ses propres règles de sécurité et des processus de conversions dynamiques pour harmoniser les formats de données et pour indexer ces contenus dans une base de données relationnelle. Les solutions les communément rencontrées ne le permettent pas. Par exemple, la solution qui combine *Webdav* et *SubVersion (SVN)*⁶ consiste à monter un système de fichiers distant qui est accessible à partir d'un navigateur et qui permet de gérer les versions à la façon de *Concurrent Versions System (CVS)*⁷. Cette solution ne permet toutefois pas d'indexer les contenus. Pour ce faire, il est nécessaire d'utiliser conjointement un service d'indexation avec une base de données relationnelle telle que *MySQL*.

2.2.5 Communication audio-visuelle multiplexée

La communication audio-visuelle multiplexée consiste à distribuer des flux audio et vidéo à partir de un ou plusieurs émetteurs, vers un ou plusieurs destinataires. Ce service forme, avec le service de visua-

⁵Un *wiki* sert à faciliter l'écriture collaborative de documents en ligne par les utilisateurs autorisés. Le *wiki* a été inventé en 1995 par WARD CUNNINGHAM. Ce mot vient du redoublement hawaïen *wiki wiki*, qui signifie rapide. Les Wikis sont depuis 2002 associés au WEB 2.0.

⁶*SubVersion (SVN)* est un système de versionnage plus récent (2004) et plus évolué que *CVS* (déplacement et renommage des dossiers, support Unicode).

⁷*CVS* est un système de versionnage concurrentiel datant de 1986 et très populaire. Il a été conçu à partir de *Revision Control System (RCS)*, en lui ajoutant principalement la capacité de gérer des projets entiers (modules) plutôt que seulement des fichiers, ainsi que la gestion multi-utilisateurs.

lisation, une composition de services synchrones qui permet de reconstituer virtuellement les conditions d'une salle de réunion et qui augmente ainsi la sensation de présence dans une très large mesure. Pour sa mise en œuvre sur un réseau à commutation de paquets comme *internet*, la contrainte principale réside dans le transfert des flux en temps-réel. Du point de vue de l'interface utilisateur, ce service doit être le plus transparent possible. Idéalement accessible à partir d'un navigateur. Il n'est donc pas utile de devoir installer des applications spécifiques.

2.2.6 Notifications

Le service de notification considéré ici concerne les notifications vers des agents humains. La notification est un service indispensable dès lors qu'une collaboration doit se mettre en place mais aussi pour indiquer tout type d'événement particulier (alerte, changement de configuration, etc.). Le service de notification qui s'est imposé *de facto* est le *courrier électronique*. Les autres services, basés sur l'écoute d'événements sur côté client, présentent tous l'inconvénient d'accéder aux ressources du poste de l'utilisateur. Ils peuvent par conséquent introduire des failles de sécurité.

2.3 Limites des solutions existantes

Dans la section 2.2, nous avons composé un *espace collaboratif ubiquitaire* avec cinq services pour permettre la mise en place d'une collaboration effective et contextualisée. Dans cette section, nous décrivons les limites conceptuelles, généralement observées avec les solutions existantes. Ces limites s'expriment en termes d'intégration, d'autonomie des groupes ou encore en termes opérationnels (harmonisation technologique, sécurité, qualité de service et passage à l'échelle).

2.3.1 Intégration partielle des services

Un *espace collaboratif ubiquitaire* est composé de services *asynchrones* mais aussi de services *synchrones*. Or, pour intégrer tous ces services dans une même solution, il est nécessaire d'avoir recours à de la ressource dédiée, notamment pour les services synchrones. Le choix de l'architecture va donc être déterminant. Une architecture adéquate doit être en mesure de fournir de la ressource aux services, tout en décorrélant les deux concepts, service et ressource. Si le service et la ressource sont liés, cette architecture risque de manquer d'évolutivité.

Ainsi, nous comprenons mieux pourquoi l'intégration des services n'est que partielle dans les solutions existantes. Le choix d'une architecture porte soit sur une solution *ad hoc* qui peut fournir de la ressource mais dont les services sont fortement liés à la plateforme sur laquelle ils sont déployés, soit sur une solution WEB plus souple mais sans apport de ressource. Ceci ne permet pas le déploiement de services dynamiques comme les services conversationnels synchrones.

2.3.2 Rigidité dans l'organisation des groupes

En nous référant au chapitre 1, nous constatons que l'autonomie décisionnelle est une condition *sine qua non* à l'auto-organisation d'un groupe de personnes. Un groupe peut se voir comme un système à la fois ouvert et fermé [Watzlawick *et al.* 1972]. Il est ouvert car chacun des membres qui le compose peut communiquer avec l'extérieur et apporter des éléments nouveaux sur son évolution. Par contre, un groupe est un système fermé du point de vue décisionnel. Toute décision qui influe sur le devenir du groupe doit être prise par le groupe lui-même et non pas par une entité extérieure. Par ailleurs, nous avons vu dans le chapitre 1 que le contexte joue un rôle majeur dans une activité collaborative de type *socio-constructiviste*. Cette activité se comporte comme un système complexe dont le comportement ne peut être déterminé par avance. A partir du moment où l'humain est intégré au processus, l'intelligence collective émergente des interactions dans ce système repose sur des critères liés à la compréhension,

la confiance, le naturel, l'accessibilité ou encore l'interprétation. Autant de critères subjectifs qui ne peuvent pas être déconnectés des aspects culturels et sociaux dans lesquels les valeurs et les rôles des intervenants sont capitaux pour assurer le fonctionnement du système. Ainsi, toute rigidité introduite dans l'organisation d'un groupe peut nuire à son développement. Dans le cas des systèmes les plus répandus, cette rigidité est habituellement introduite par une mauvaise interprétation de ses concepts constitutifs ou à une nécessité d'appliquer des règles fixées *a priori* qui n'auront pas été souhaitées par le groupe. La mauvaise interprétation des concepts constitutifs d'un groupe (utilisateur, membre, droits et permissions) provient d'une confusion largement répandue avec l'héritage des modèles multi-utilisateurs classiques. Par exemple, le membre d'un groupe dans le modèle *Unix* peut être tantôt un *utilisateur* tantôt un *super-utilisateur*. La notion de *super-utilisateur* se distingue d'abord de la notion d'*utilisateur* par le fait qu'elle n'est pas nominative. Ensuite et surtout, elle est confondue avec une notion particulière de privilèges (ex. le *super-utilisateur root* a des droits absolus sur tous les services d'un système *Unix*).

Une autre source de confusion vient de la notion même de groupe. Dans le cas d'*Unix*, comme dans la plupart des systèmes actuels ⁸, le concept de groupe peut avoir une signification particulière vis-à-vis de certains utilisateurs : il peut s'agir de leur groupe principal. Or, ce même groupe peut être vu comme un simple groupe vis-à-vis d'autres utilisateurs. Cette caractéristique de groupe principal implique plusieurs situations particulières : un groupe principal est le groupe par défaut de l'utilisateur après qu'il se soit authentifié et un groupe principal ne peut être supprimé.

Ces cas de figures illustrent toute la rigidité d'une vaste majorité des systèmes multi-utilisateurs actuels. Ainsi, nous proposons de bâtir notre modèle sur les conditions suivantes :

1. adopter le concept de rôle et une relation ternaire entre le groupe, l'utilisateur et le rôle de l'utilisateur en tant que membre d'un groupe donné ;
2. décliner le concept de groupe en deux autres concepts : la catégorie d'utilisateurs (ex. invité) à la place du groupe principal (en autorisant plusieurs catégories) et la communauté d'utilisateurs.

2.3.3 Harmonisation technologique pénalisante

Pour qu'un espace collaboratif puisse émerger, il doit au préalable être **technologiquement harmonisé**. C'est à dire qu'un groupe qui décide de collaborer doit se mettre d'accord sur le choix des outils et des formats de fichiers utiles à la collaboration. De par l'obligation d'installer des logiciels spécifiques à tous ces outils, les solutions actuelles réduisent cette harmonisation au plus petit dénominateur commun des possibilités d'interopérabilité. Un exemple de situation est souvent rencontré avec des utilisateurs de plateformes différentes. Ceux-ci peuvent s'échanger des fichiers dans un format portable comme le pdf ou encore exécuter le même code écrit dans un langage générique tel que Java mais ils ne pourront toutefois pas exécuter des programmes spécifiquement compilés pour une plateforme sur une autre plateforme. Pour éviter de se retrouver dans une situation trop contraignante, les groupes se voient dans l'obligation d'imposer à tous leurs membres une configuration type de plateforme cliente que chaque membre devra adopter. Il est fréquent de voir qu'un service est accessible uniquement pour tel ou tel navigateur configuré de telle ou telle manière. Il persistera toujours des situations frustrantes où des utilisateurs devront faire preuve d'un effort accru d'adaptation pour se soumettre à une plateforme adoptée par d'autres membres du groupe. Bien que cette approche affecte peu les utilisateurs qui accèdent toujours au service par le même poste client et qui font partie d'un seul groupe, elle ne résout pas le problème des utilisateurs nomades ou membres de nombreux groupes. En d'autres termes, cette approche est une entrave au principe d'ubiquité que nous cherchons à promouvoir. En outre, ces effets cumulés, par une multiplication de barrières d'acceptation, risquent de nuire à la motivation des utilisateurs.

Un problème récurrent avec les modèles existants est qu'ils imposent des règles d'harmonisation technologiques contraignantes dès lors qu'une collaboration doit se mettre en place. Les modèles existants

⁸Il est important de noter ici (nous le verrons plus loin) que les solutions de gestion des organisations virtuelles dans Grid (en particulier VOMS), hérite des défauts du modèle Unix.

tants utilisent la ressource informatique du poste client, notamment la ressource de stockage. Ceci a pour effet de mettre le poste client dans un état statique, complexe à administrer et, parfois, avec des comportements imprévisibles. Ainsi, gérer la ressource au niveau du client ôte toute possibilité d'ubiquité (*i.e* attachement de l'utilisateur à son poste).

Si l'on se place dans un contexte collaboratif où les utilisateurs peuvent être membres de plusieurs groupes, chaque groupe effectue un choix d'applications commun à l'ensemble des membres du groupe mais ce choix peut varier selon les groupes. L'utilisateur se retrouve ainsi dans l'obligation de se soumettre à la maîtrise d'un environnement technologique qu'il n'a pas choisi, et ceci pour autant de groupes dont il est membre, au lieu de consacrer son temps à son domaine de compétence.

2.3.4 Insuffisance opérationnelle

La domination du modèle d'architecture *orienté-système*, avec toutes ses limites, pèse aujourd'hui sur le développement des *espaces collaboratifs ubiquitaires*. Ce modèle gravite autour du paradigme de l'ordinateur (que chacun possède à titre personnel) interconnecté au réseau internet et sur lequel sont installés des outils de communication, des périphériques et d'autres applications *ad hoc* que l'utilisateur a délibérément choisi ou non d'installer. D'abord, ce modèle a un impact négatif sur le coût et le poids des équipements terminaux. Ensuite, par un effet combinatoire, l'utilisateur doit sans cesse, à son niveau, s'adapter à une complexité informatique croissante pour gérer convenablement son outil. Cette complexité peut rapidement devenir démesurée dès qu'il s'agit de constituer un espace collaboratif entre plusieurs utilisateurs interconnectés, du fait de l'hétérogénéité de leurs outils.

Par conséquent, l'objectif de collaboration doit prendre le dessus sur les contraintes induites par les solutions techniques adoptées. Ainsi, il s'agira de tenir compte de ces risques pour définir les contours d'une infrastructure adéquate.

2.4 Caractérisation

Sur la base de notre analyse, nous avons identifié trois caractéristiques essentielles qui spécifient un *espace collaboratif ubiquitaire* :

1. le caractère **multimodal** de l' *Interface Homme-Machine (IHM)* pour une interaction renforcée ;
2. le caractère d'**immanence** pour une meilleure auto-organisation des groupes ;
3. le caractère d'**ubiquité** pour une indépendance de l'utilisateur vis-à-vis de l'environnement ;

2.4.1 Caractère multimodal de l'IHM

Une *Interface Homme-Machine (IHM) multimodale* permet d'accroître l'efficacité de l'activité collaborative par un rapprochement avec la notion de *réalité augmentée*. Ceci suppose la coexistence de médias de communication complémentaires entre les différents services, qu'ils soient de type *asynchrones* ou de type *synchrones* (*c.f* section 1.2.2). L'une des difficultés est liée à l'ergonomie de l'environnement. Développer des interfaces multimodales, à la fois conviviales et intuitives [Franck Tarpin-Bernard1999] peut devenir une gageure. Par exemple, un mode combiné de communication bi-directionnel oral et écrit requiert plusieurs canaux, plusieurs files d'attentes avec plusieurs niveaux de sécurité. L'utilisateur doit parfois se familiariser avec des concepts liés à la collaboration tels que la prise de tour qui peut sembler moins intuitive à travers une *IHM* que dans une situation réelle.

La figure 2.5, nous montre que l'accès à l'*espace collaboratif ubiquitaire* peut s'effectuer à travers un service *synchrone* ou *asynchrone* à des fins de consultation ou de modification des contenus. Un service collaboratif *asynchrone* adopte le plus souvent un mode de communication écrit comme *wiki* ou des formes basiques d'échange de documents comme le *courrier électronique*. Dans ce cas, le mode oral est le plus souvent sous forme pré-enregistré. Pour maintenir une traçabilité de la collaboration, les services

asynchrones requièrent un moyen de suivi des modifications sous forme plus ou moins synthétique. Il existe des gestionnaires de suivi par versionnage incrémental comme c'est le cas pour *wiki*, *CVS* ou encore *Subversion*. Néanmoins, un service de suivi de modification ne sera pas efficace s'il n'est pas couplé intelligemment avec un service de notification.

Les services collaboratifs *synchrones* sont plus fortement concernés par une *IHM multimodale*. L'une des conditions essentielles au bon fonctionnement de services collaboratifs *synchrones* est la notion d'*awareness* qui se traduit littéralement par avoir conscience de l'état de présence ou de l'activité d'autrui. Cette notion est très utile, par exemple, lorsque les utilisateurs souhaitent se joindre pour discuter en direct de leur travail. Dans ce mode de communication, outils et protocoles diffèrent de ceux utilisés par les services *asynchrones*. La messagerie instantanée, la téléprésence ou la visioconférence sont des modes de communication couramment utilisés par les services *synchrones*. L'écran partagé est un mode de communication encore peu utilisé de par les exigences techniques qui rendent complexes leur mise en œuvre sur les architectures actuelles.

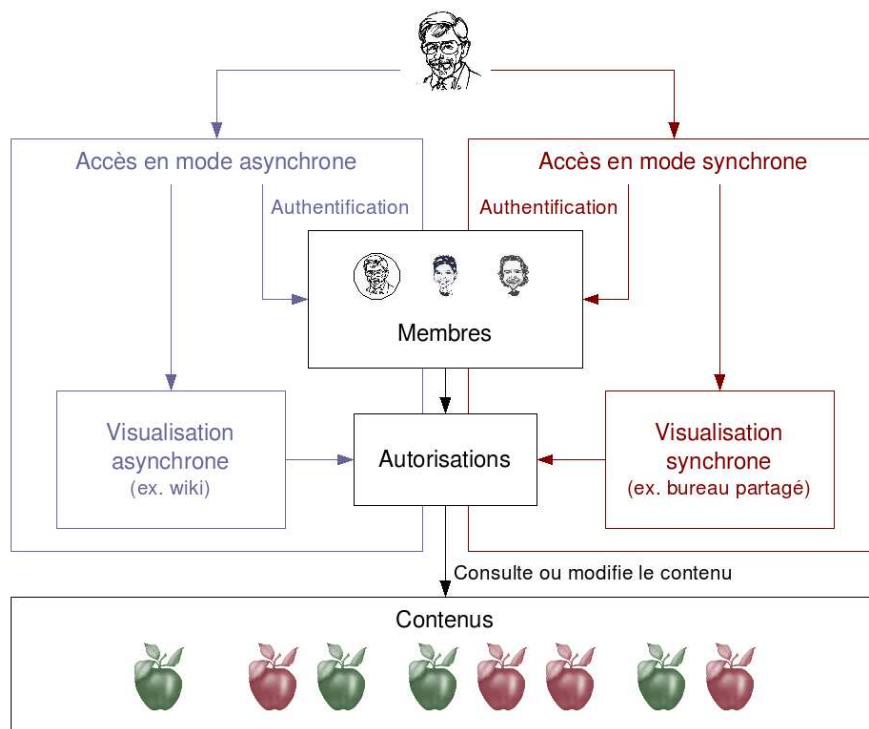


FIG. 2.5 – Accès aux contenus par un service synchrone et asynchrone

2.4.2 Caractère d'immanence

Un groupe a besoin de règles pour s'auto-organiser. Toutefois, ces principes organisateurs ne doivent pas être définis à l'extérieur du groupe. Le caractère d'immanence implique que toute décision ayant un effet sur le développement du groupe doit pouvoir être prise au sein du groupe. Réciproquement, aucune décision prise à l'extérieur du groupe ne doit influencer sur le devenir de celui-ci. Il s'agirait, dans ce cas, d'une action transcendante qui n'est pas souhaitable pour conserver l'autonomie décisionnelle du groupe. Ce sont des groupes *moléculaires* qui élaborent en continu leur projet, leurs ressources et leurs compétences. Ils se réorganisent en temps réel [Lévy1990]. Ainsi, le principe d'immanence agit ici comme un lien social émergent de la relation de chacun à tous.

2.4.3 Caractère d'ubiquité

La notion d'*ubiquité informatique* a été introduite par MARC WEISER de Xerox PARC en 1991 [Weiser1991], suivi quelques années plus tard par un ouvrage en collaboration avec SEELY BROWN *Designing Calm Technology* [Weiser & Brown1995]. Le terme *calm* est une vision d'architecture qui signifie que la technologie se retrouve déplacée en arrière plan. WEISER distingue trois époques de la relation entre l'humain et l'ordinateur :

- **L'époque des superordinateurs.** Utilisés par des experts dans des salles spécialisées, les ordinateurs étaient vus comme des objets dispendieux. Cette époque marque le début de l'informatique avec une relation de plusieurs humains pour un seul ordinateur.
- **L'époque de l'ordinateur personnel.** La relation humain-ordinateur est équilibrée dans le sens où la relation est un-pour-un. Cette situation a créé une forme d'intimité entre l'humain et l'ordinateur, mais aussi une dépendance technologique parfois pénalisante.
- **L'époque de l'informatique ubiquitaire.** Ici une personne peut disposer de plusieurs ordinateurs. La notion d'ordinateur n'est plus une notion physique. Le couplage entre l'humain et le réseau prend une importance considérable au détriment du poste de travail. A terme, l'utilisateur peut accéder à toutes ses données et ses ressources de traitement depuis un terminal sans état, voire avec seulement une mémoire volatile.

2.5 Recommandations pour le déploiement

Afin d'orienter la définition d'une infrastructure, ce chapitre tire en conclusion trois recommandations, qui portent sur les aspects liés au déploiement des *espaces collaboratifs ubiquitaires*, à savoir :

- la sécurité ;
- la disponibilité du service ;
- le passage à l'échelle.

Le respect de ces recommandations contribue à établir un meilleur rapport de confiance entre les utilisateurs et leur environnement technologique.

2.5.1 Sécurité

La *sécurité* englobe à la fois la notion d'*authentification* des utilisateurs et aussi les notions de *confidentialité* et d'*intégrité* des données.

La confidentialité, l'intégrité et la sauvegarde des contenus et des flux d'un *espace collaboratif ubiquitaire* doit être totalement assuré par l'infrastructure.

2.5.2 Disponibilité du service

En plus des critères de *qualité de service* liés à la vitesse d'acheminement des flux sur le réseau, la disponibilité du service est essentielle.

L'indisponibilité d'un *espace collaboratif ubiquitaire* doit être imperceptible.

2.5.3 Passage à l'échelle

Le passage à l'échelle signifie que le service peut être dimensionné quel que soit le nombre d'utilisateurs, le nombre d'*espaces collaboratifs ubiquitaires* instanciés et la quantité de ressources mise à disposition.

Pour le passage à l'échelle, le déploiement du service doit être une opération reproductible, réversible et automatisable.

INFRASTRUCTURE À RESSOURCES DISTRIBUÉES

Le savoir est devenu la nouvelle infrastructure [Serres1994].

Michel Serres

SUR LA base des recommandations du chapitre 2, il s'agit de caractériser une infrastructure adéquate permettant de déployer des *espaces collaboratifs ubiquitaires*. L'état de l'art présenté dans ce chapitre concerne de manière générale les architectures *orientée-services* et plus particulièrement GRID. Si GRID apporte une réponse plus satisfaisante que le WEB, en termes de coordination de ressources distribuées et en termes de sécurité, celle-ci demeure encore insuffisante en termes de gestion des *organisations virtuelles*. Ainsi, nous avons plus longuement exploré la question de la gestion des relations entre membres et services d'une *organisation virtuelle*. Ceci nous a permis d'identifier une synergie entre GRID et les *Systèmes Multi Agents (SMA)*. L'étude de cette synergie marque une étape clé dans nos recherches et constitue une contribution originale que nous allons tenter d'exploiter lors de l'élaboration de notre modèle AGORA (*cf.* chapitre 4).

Table des matières

3.1	Evolution conceptuelle et technologique	44
3.1.1	Du WEB à GRID	44
3.1.2	Du calcul à la collaboration	45
3.1.3	Les solutions GRID	49
3.2	Principes	52
3.2.1	Généralités sur les architectures distribuées	52
3.2.2	Architectures <i>orientées-services</i>	53
3.2.3	Gestion des ressources réparties	55
3.2.4	Sécurité	56
3.2.5	Organisations virtuelles	59
3.3	Synergie entre GRID et les SMA	61
3.3.1	Un modèle GRID simplifié	61
3.3.2	Analogies	62
3.3.3	Modèle d'intégration	63
3.4	Projection future	64
3.4.1	Bilan de l'activité de recherche	65
3.4.2	Ubiquité informatique	65
3.4.3	Infrastructure idéale pour <i>espaces collaboratifs ubiquitaires</i>	66

3.1 Evolution conceptuelle et technologique

Le partage de ressources et, plus généralement, de tout dispositif informatique distribué est une vision en plein essor dans le domaine de l'*informatique opérationnelle* dont l'image la plus évocatrice dans le monde scientifique est la métaphore du labo *sans murs*¹. Bien que l'usage de GRID demeure, depuis son existence, l'apanage du calcul intensif, ce paradigme présente un intérêt croissant dans la perspective de l'*informatique ubiquitaire*.

3.1.1 Du WEB à GRID

Alors que le WEB a montré toute la puissance d'une infrastructure pour le partage d'informations distribuées, GRID ouvre un nouveau champ de possibilités avec une infrastructure mutualisant des ressources informatiques réparties géographiquement. De ce fait, GRID apporte de nouvelles réponses à certaines limites intrinsèques du WEB, telle que la coordination des ressources pour délivrer une qualité de service à la demande [Foster2002]. Bien qu'il existe des points de convergence entre ces deux architectures [Talia2002], ce qui distingue fondamentalement GRID du WEB sont des différences d'ordre topologique entre les nœuds de provision des ressources et les nœuds de consommation de ces ressources. Pour expliquer cette évolution, la figure 3.1 nous donne une représentation schématique des trois familles de topologies qui sont apparues depuis l'existence des réseaux informatiques. Les cercles indiquent les terminaisons côté utilisateur tandis que les carrés décrivent les terminaisons du côté des ressources informatiques. Le premier cas correspond à la période qui précède l'ère internet ; les ressources sont fortement concentrées en un point et découplées des terminaisons utilisateurs. Le second cas correspond à la période qui suit l'apparition des architectures *client-serveur* et du WEB ; les ressources commencent à être réparties mais restent fortement couplées physiquement aux terminaisons des utilisateurs. C'est la principale raison pour laquelle il n'est pas possible de coordonner les ressources réparties sur une infrastructure WEB. Le troisième cas correspond à une utilisation généralisée de GRID avec des ressources réparties et virtualisées. Les terminaisons des utilisateurs ne sont plus couplées physiquement à des ressources informatiques mais elles sont en interaction avec des *services à état*, un concept clé de GRID dont nous détaillerons les principes dans ce chapitre.

HTTP ET DNS

Le fonctionnement du WEB repose sur une infrastructure de serveurs interconnectés qui communiquent grâce aux protocoles *Domain Name Service (DNS)* et *Hyper Text Transfer Protocol (HTTP)*. Le protocole *DNS* permet d'établir une correspondance entre une adresse *Internet* numérique de la forme 193.49.113.146 et un nom de domaine (ex. *agora.lirmm.fr*). Cette association réside de façon entièrement statique dans des tables de correspondance des serveurs *DNS*. Les modifications sur ces tables peuvent être effectuées à titre exceptionnel. Ce nom de domaine permet d'identifier les différents serveurs *HTTP*. Ces serveurs fournissent un service d'échange d'information. L'indication dans un navigateur de l'adresse complète `http://www.lirmm.fr` déclenche automatiquement l'appel du protocole *HTTP*. L'inconvénient de cette architecture est son manque de fiabilité : un serveur *HTTP* qui s'écroule est autant d'information inaccessible.

GRID permet grâce au *service à état* de virtualiser toute forme d'échange d'information et de minimiser les risques de pannes. De cette façon la fonction serveur *HTTP* est réalisée de façon logique par

¹Le livre blanc de l'informatique opérationnelle du *Comité des Services Informatiques de l'Enseignement Supérieur et de la Recherche (CSIESR)* [CSIESR2001] décrit l'organisation de l'informatique pour la recherche selon trois niveaux sur la base d'un modèle antérieur du *Ministère de la Recherche*. Au premier niveau se situent les équipements du laboratoire. Au second niveau se situe le mésocentre, un centre de calcul intermédiaire regroupant des équipements partagés par plusieurs laboratoires avec des ressources humaines compétentes pour les gérer. Au troisième niveau, se situent les grands centres de calcul dotés d'équipements de grande puissance et interconnecté par des liaisons à très fort débit sur le réseau *RENATER*.

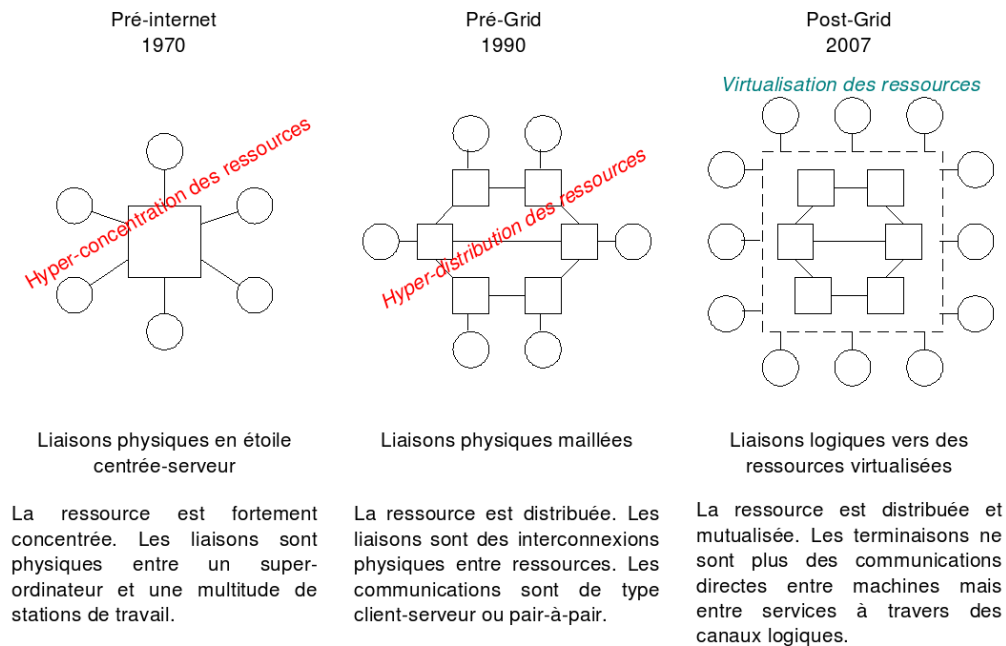


FIG. 3.1 – Trois familles de topologies.

un *service à état* accessible par un identifiant, au lieu d'être matérialisé par un *hôte* (ie. une machine réelle ou virtuelle) avec une *adresse TCP/IP* comme c'est le cas pour le WEB. Ce principe permet de déployer de façon simple et efficace des réseaux de services en interaction avec des règles globales pour une meilleure tolérance aux pannes et pour décentraliser les politiques de sécurité.

3.1.2 Du calcul à la collaboration

A l'origine, GRID désigne des plates-formes de calcul à grande échelle composées de ressources réparties et hétérogènes. L'objectif est de mutualiser ces ressources en une capacité informatique globale puis de redistribuer cette capacité informatique sur le même mode décentralisé qu'est l'électricité. Cette idée date des années 1970, en revanche, ce sont les avancées technologiques d'une dizaine d'années qui ont permis les réalisations concrètes. La motivation des chercheurs et ingénieurs est portée par ce fameux paradoxe : « *en informatique, il est compliqué de rendre simple* ». Le défi étant de faire coopérer des matériels distants et hétérogènes, diminuer la probabilité de défaillance, garantir la confidentialité, tout en rendant ces opérations totalement transparentes et intuitives pour l'utilisateur. Depuis environ cinq ans, nous assistons à une transition d'une vision exclusivement *orientée-calcul* à une vision *orientée-collaboration*.

Infrastructures de calcul

Les projets qui ont vu le jour dans les années 1990 se sont, pour la plupart, orientés vers le *High Throughput Computing (HTC)* : *Condor* [Condorurl], l'un des premiers projets supportant des traitements massifs de données distribuées, *Unicore* [Unicoreurl], une infrastructure couplée avec un portail scientifique pour l'accès aux supercalculateurs et *Legion* [Legionurl], un projet de l'*Université de Virginie* visant à supporter le parallélisme dans une large mesure. **Legion** est décrit une architecture et un intergiciel basé sur une technologie objet, permettant à des utilisateurs d'accéder à des ressources distri-

buées de façon transparente et coordonnée, et avec une forte tolérance aux pannes. Parmi ses objectifs, *Legion* vise à déployer des sites autonomes du point de vue de leur ressource et de leur politique de sécurité avec des composants logiciels évolutifs et recomposables.

Comme la plupart de ces projets provenant des Etats-Unis, le projet *TeraGrid* ambitionne de bâtir la plus puissante infrastructure de calcul au monde. L'infrastructure délivre actuellement 250 *TeraFlops* (TF)² de puissance de calcul, 30 *PetaOctets* (PO) de stockage et envisage de fournir des environnements de visualisation pour faciliter l'accès aux ressources [TeraGridurl]. Réparties sur cinq sites³, les ressources sont interconnectées par un réseau à 40 *Gbps*, et peut traiter un volume de données de l'ordre d'un PO .

En Europe, l'activité GRID pour le calcul intensif se mobilise essentiellement autour de deux projets. Le *Large Hadron Collider* (*LHC*) va prochainement être mis en service au *Centre Européen en Recherche Nucléaire* (*CERN*). Le projet *LHC Computing Grid* (*LCG*) [LCGurl] vise à traiter les millions d'interactions par secondes du *LHC*. Celles-ci sont équivalentes à un flux de données de l'ordre de 15 PO par an. *LHC* se base sur l'expérience acquise par *Data Grid* [DataGridurl]. L'autre projet est *Enabling Grid for eScience* (*EGEE*) [Egeeur], un projet d'infrastructure sur 150 sites répartis dans 32 pays, pour des applications en géophysique, énergie, bio-informatique et astrophysique.

FLOPS

Floating point Operations Per Second (*FLOPS*) est l'unité de référence servant à mesurer la puissance de traitement informatique. Un ordinateur de bureau de base comme un Pentium 4 opérant à une fréquence d'horloge de 2 *GHz* fournit une puissance de l'ordre de quelques *GigaFlops* (*GF*). Pour se fixer un ordre de grandeur, la puissance du premier *superordinateur* le *Cray 1* était en mesure de fournir une puissance bien inférieure à 1 *GF* en 1976. Aujourd'hui, *Blue Gene/L* (voir photo 3.2), le calculateur le plus puissant du monde, délivre une puissance de 280,6 TF . Il est composé de 64 groupes de 1024 nœuds équivalents chacun à deux *PowerPC 440* à 770 *MHz* (2,8 *GF* chacun). Les concepteurs de *Blue Gene/P*, la nouvelle génération de ce calculateur, annoncent avoir franchi le mur théorique du *PetaFlop* en 2007. Dans le cas des systèmes distribués, il faut distinguer la puissance maximum théorique (somme des puissances des différents éléments distribués) et la puissance effective qui dépend du couplage entre les différents éléments. Un exemple de système distribué de 40 TF est développé sous *Linux* avec 4 500 processeurs et de 128 000 *Go* de stockage [Linux40TFurl].

Infrastructures à ressources volatiles

Ces initiatives, sont des infrastructures constituées d'ordinateurs de bureau⁴. Elles consistent à mutualiser la ressource dormante d'un très grand nombre d'ordinateurs de bureau sur la base du volontariat des utilisateurs. Cette approche vise à rationaliser une infrastructure informatique déjà existante. Toutefois, concevoir efficacement un tel système représente un défi majeur. En effet, l'environnement d'exécution, composé d'ordinateurs personnels répartis géographiquement, est très hétérogène à cause de la variation importante des performances d'accès aux ressources (traitement, stockage et réseau). De plus, il s'agit d'un système avec des couplage entre des nœuds instables, voire indisponibles. Jusqu'à présent, ces systèmes ont principalement été destinés aux applications de type calcul intensif. Deux projets très médiatiques de ce type se sont distingués : *seti@home* [SetiAtHomeurl] pour l'analyse de signaux interstellaires, ou plus récemment le *Décrypton* [Décryphonurl] pour la recherche en génomique et la

²un TeraFlop pour 10^{12} Flops et un PetaOctet pour 10^{15} octets.

³Sites de TeraGrid : National Center for Supercomputing Applications (NCSA-UIUC), San Diego Supercomputer Center (SDSC), Argonne National Laboratory, Center for Advanced Computing Research (CACR-CalTech), Pittsburgh Supercomputing Center (PSC).

⁴Traduit de l'anglais Desktop Grid.



FIG. 3.2 – Blue Gene/L

lutte contre les myopathies. Avec le support des quelques 500 000 ordinateurs mis à disposition par la *World Community Grid*, coordonnée par *IBM*, des puissances de plusieurs dizaines de *TF* ont ainsi pu être atteintes [WCGurl].

Plus récemment, des projets comme *XtremWeb*⁵, *DSLlab* ou encore *XtremTube*⁶, proposent également des plateformes à ressources volatiles. *DSLlab* vise à déployer une plateforme expérimentale de systèmes distribués et interconnectés par une liaison *ADSL*. *XtremTube* vise à développer une application *Grid* dédiée au traitement vidéo à grande échelle et fournir à une organisation virtuelle une facilité de stockage vidéo et de post-traitement à la demande, permettant la conversion de vidéo dans le format vidéo de leur choix et l'application de filtres de traitement d'image.

La problématique commune à tous ces projets est d'organiser un système distribué sur des ressources volatiles qui soit efficace du point de vue du stockage des données, du calcul et de la diffusion vers un très grand nombre de nœuds.

Fouille de données

La fouille de données consiste à extraire de la connaissance à partir de larges corpus de données. Depuis peu, l'usage de *GRID* fait l'objet de nombreuses expérimentations probantes à grande échelle. Nous pouvons citer le projet *EuroGrid* [EuroGridurl] dans le domaine des bio-technologies et en météorologie, ou encore les projets *GriPhyN* [GridPhyNurl] et *Particle Physics Data Grid (PPDG)* [PPDGurl] en physique, ou encore *International Virtual Data Grid Laboratory (iVDGL)* [iVDGLurl] en astrophysique.

⁵XtremWeb (<http://www.xtremweb.net>) est développé au LRI (Laboratoire de Recherche en Informatique) de l'Université de Paris XI.

⁶DSLlab (<http://dslab.org>) et XtremTube sont deux projets développés à l'INRIA.

Activité GRID en Europe

En Europe, le projet *CoreGrid* [CoreGridurl] est un réseau d'excellence (projet européen du 6^e PCRD⁷) qui vise à déployer une infrastructure reconfigurable dont la vocation est d'établir durablement une communauté scientifique rassemblant les meilleurs chercheurs européens dans le domaine GRID. En France, le projet *Grid5000* [Grid5000url], est une infrastructure expérimentale déployées sur neuf sites⁸ pour tester des mécanismes et des principes expérimentaux GRID. Ce projet vise à délivrer des moyens de calcul scientifique avec une souplesse de configuration [Cappello *et al.*2006]. Avec le recul des différentes expérimentations réalisées, il en ressort une convergence technologique entre GRID, les systèmes *pair-à-pair* et, plus généralement, les infrastructures de services.

Visualisation

La visualisation à distance est un principe formalisé dans les années 1990, dont la description la plus connue [Haber & McNabb1990] consiste en un flux de données brutes correspondant à des images qui sont transmises à travers trois niveaux de traitement : (i) le filtrage, (ii) la reconstitution, (iii) le rendu. La visualisation à distance est devenue une nécessité pour le *TCAO*. L'idée de s'appuyer sur une infrastructure GRID n'est pas neuve [Shalf & Bethel2004] et une proposition d'architecture baptisée *Network-Oriented Visualization Architecture (NOVA)* a été proposée par la communauté GRID [Foster *et al.*1999]. Depuis lors, de nombreux projets ont été répertoriés de façon exhaustive dans une étude récente [Ken W. Brodlie & Wood2004]. Parmi ces projets, figurent ceux de la catégorie de la *visualisation collaborative distribuée*. Le projet *COVISE*⁹ développe une plateforme pour la visualisation distribuée de simulations. Le projet *Cactus* s'appuie sur *Globus* pour gérer la sécurité.

AccessGrid un projet développé par l' *Argonne National Laboratory (ANL)* qui combine les notions de collaboration et de visualisation sur GRID, est, de loin, le plus avancé de cette catégorie [AccessGridurl]. Ce projet présente de nombreux atouts, tels que la richesse de l'environnement avec du transfert vidéo d'excellente qualité, des accès multiplexés vers des applications et bénéficie d'un soutien important de la part d'une bonne partie de la communauté GRID. Dans sa version actuelle (version 3), cet environnement est un ensemble de ressources incluant un environnement de visualisation interactif basé sur des interfaces à des services GRID. La sécurité est basée sur l'échange de certificats *X.509* pour les hôtes mais pas au niveau des utilisateurs. L'architecture comporte un ensemble de ressources de calcul, d'interfaces à GRID via l'intergiciel *Globus* (voir section 3.1.3) et d'un environnement de visualisation, *Collabvis* [Collabvisurl], qui utilise le protocole *Remote Frame Buffer (RFB)* [Richardson2005]. La téléprésence et la recherche de compétences est une idée force du modèle. Le système de sécurité adopte les principes de GRID (voir section 3.2.4). Sur la photo 3.3, *Collabvis* a été développé par *Western Canada Research Grid*. *AccessGrid*, aujourd'hui déployée sur 201 nœuds dans 24 pays et mis au point en étroite collaboration avec la communauté *Globus*, est certainement la solution la plus proche de notre vision d'*espace collaboratif ubiquitaire* avec plusieurs caractéristiques propres : outre l'ubiquité, elle présente une *IHM* multimodale supportant des services de communication synchrones et asynchrones. Toutefois, elle ne répond pas à la caractéristique d'immanence car l'organisation des communautés gravite autour d'une gestion technique centralisée et relativement complexe à mettre en œuvre. En dépit d'un effort de documentation bien fournie et accessible en ligne, la configuration d'un environnement est complexe sans une forte compétence en ingénierie système.

⁷PCRD : Programme Cadre de Recherche et de Développement

⁸Les sites de *GRID5000* : Orsay, Bordeaux (Labri), Lyon (Lip), Rennes (Idris), Nancy (Loria), Toulouse (Irit), Lille (Lifl), Grenoble (Lig) et Sophia-Antipolis sont interconnectés par la fibre noire du réseau Renater (10 Gbps).

⁹ *Collaborative Visualization and Simulation Environment (COVISE)* développé au centre de calcul *High Performance Computing Center (HLRS)* de Stuttgart.



FIG. 3.3 – *Collabvis*, visualisation collaborative sur *AccessGrid*

Collaboration et apprentissage

Plusieurs projets s'intéressent de près à la mise en place d'environnements pour le *TCAO* (cf. section 2.1) sur GRID. Le projet *DOE SciDAC* [Johnston2004] s'appuie sur des services GRID persistants comme moyen de fournir des fonctions permettant de constituer un système sécurisé pour la planification de tâches. L'environnement de collaboration consiste ici à partager en temps réel des résultats d'analyse obtenus à partir d'une instrumentation distante dans le domaine de la physique, chimie et climatologie. Ce projet repose sur des travaux théoriques antérieurs dont les principales contributions proviennent du *Pervasive Collaborative Computing Environment (PCCE)* [Agarwal2001] et de la notion de *serveur de visualisation* [Parvin et al.1997].

ELeGI : Plus récemment, le paradigme de la collaboration a touché le domaine de l'apprentissage, avec le projet *European Learning Grid Infrastructure (ELeGI)* [ELeGIurl], dans lequel le *Lirmm* est impliqué, qui propose de mettre en perspective une infrastructure GRID pour l'apprentissage collaboratif. L'objectif principal de ce projet est d'appliquer des pratiques d'apprentissage à travers un environnement Grid. Des scénarii pour l'exploitation de services *Service Elicitation and Exploitation Scenarios (SEES)* permettent d'évaluer la qualité des services dans un contexte d'apprentissage formel ou informel. Parmi les résultats de ce projet, le *Lirmm* a développé un démonstrateur s'appuyant sur des concepts, tels que la distributivité des services, pour modéliser l'apprentissage collaboratif avec GRID.

3.1.3 Les solutions GRID

L'intérêt croissant que suscite GRID est, certes, lié à des principes novateurs que nous décrivons plus loin (cf. section 3.2) mais aussi grâce à son ouverture technologique accompagnée d'un effort de normalisation soutenu avec *Open Grid Service Architecture (OGSA)* et *Web Services Resource Framework (WSRF)*. Ceci a permis à la communauté du logiciel libre d'œuvrer efficacement à proposer des solutions comme l'intergiciel *Globus* [Globusurl] qui permet de mutualiser des ressources informatiques hétéro-

gènes ou encore *XtreemOS* [XtreemOSurl, Morin2007] qui propose un noyau totalement ouvert sous *Linux* pour exploiter des ressources distribuées.

Aspects normatifs

L'architecture GRID est principalement décrite par deux normes complémentaires.

- La norme *OGSA* a été adoptée par le *Global Grid Forum (GGF)* ¹⁰ lors de la conférence de Toronto en 2002. *OGSA* définit un modèle d'architecture *orientée-services* qui vise à virtualiser les ressources et à les restituer sous forme de services afin de pouvoir les assembler et les désassembler en fonction des besoins [Foster *et al.*2002]. Le principe fondamental du modèle *OGSA* provient du processus de génération dynamique des services au moyen du couple *générateur-instance* de service. Ceci implique l'existence de mécanismes de réservation et de libération des ressources. L'autre point fort de cette norme concerne la sécurité. Ce sont les principaux aspects novateurs d'*OGSA* par rapport aux *services Web*.
- La norme *WSRF* est apparue en 2004 en remplacement de la norme *Open Grid Service Infrastructure (OGSI)* en vue de rapprocher les concepts GRID avec les *services Web* [Czajkowski *et al.*2004b]. Cela a nécessité une harmonisation dans la définition des interfaces et des protocoles d'échanges entre services. La base des *services WEB* a finalement été adoptée [Foster *et al.*2005]. Ceci a débouché sur une série de spécifications qui reprend les concepts *OGSI* [Czajkowski *et al.*2004a] (voir table 3.1). Ces spécifications définissent les interfaces en *Web Services Description Language (WSDL)* et les protocoles en *Service Oriented Application Protocol (SOAP)* [Booth *et al.*2004] permettant la création et la libération de ressources et la gestion des notifications.

Ces normes sont stabilisées en vue de maintenir un bon niveau d'inter-opérabilité entre services et intergiciels. Nous nous intéressons plus particulièrement, dans le cadre de cette thèse, aux principes généraux décrits par ces normes. Ces principes sont développés dans la section 3.2.

Globus : l'intergiciel OGSA

Lancé en 1998, *Globus* est l'intergiciel de référence très répandu aujourd'hui pour déployer des services GRID [Foster & Kesselman1999a]. Son succès tient du dynamisme de la communauté *Globus Alliance* [Globusurl] qui maintient un code source ouvert et bien documenté. Cet intergiciel se charge d'orchestrer les différents éléments de GRID tels que les ressources, les services et les communautés d'utilisateurs. *Globus Toolkit version 4 (GT4)* disponible depuis 2005 est la version la plus récente conforme aux normes *OGSA* et *WSRF*. Parmi les composants de base de *GT4*, les deux plus importants adressent l'allocation dynamique des ressources et la sécurité :

- *Globus Resource Allocation Manager (GRAM)* [Czajkowski *et al.*1998, Foster & Kesselman1999b] qui facilite le lancement et le contrôle des processus à distance, sous forme de requêtes d'exécution à la façon d'une *Remote Procedure Call (RPC)* dans une architecture *client-serveur*. *GRAM* gère la ressource (réservation, libération) nécessaire aux instances de services produites par les générateurs de services (factory). Les instances de services sont accessibles à travers un lien dynamique (handle) géré par un registre *GRAM*.
- *Grid Security Infrastructure (GSI)* [Foster *et al.*1998] qui réalise les fonctions de sécurité élémentaires (authenticité mutuelle par certificats, trust).

gLite : L'integiciel du projet EGEE

gLite se présente comme une pile avec un cœur de bas niveau et une gamme de services de plus haut niveau. *gLite* hérite de toute l'ingénierie du projet *DataGrid* et intègre des composants provenant

¹⁰La structure du GGF a été revue en 2005 pour intégrer l'*Enterprise Grid Alliance (EGA)* qui forment maintenant ensemble l'*Open Grid Forum (OGF)* <http://www.ogf.org>.

TAB. 3.1 – Correspondance entre les concepts OGSi et WSRF

CONCEPT	DÉFINITION OGSi	DÉFINITION WSRF
Représentation du service Propriété du service	<i>Grid service</i> findServiceData setServiceData	WS-Resource WS-Resource Properties
Durée de vie du service	requestTerminationAfter requestTerminationBefore Destroy	WS-Resource Lifetime
Instantiation du service	<i>Factory</i> createService	WS-Resource factory
Identification du service Adresse du service	<i>HandleResolver</i> findByHandle GSH (Grid Service Handler) GSR (Grid Service Reference) WS-Addressing	WS-Renewable References endPoint reference
Service de notifications	<i>NotificationSource, NotificationSink</i> subscribe deliverNotification	WS-Notification
Gestion des groupes	<i>ServiceGroup</i>	WS-Service Group
Gestion des exceptions	Base fault type	WS-Base Faults

de projets tels *Condor* et *Globus*, ainsi que les composants développés pour le projet *LCG* (sécurité, accès aux ressources, gestion des données, gestion de la charge, surveillance, gestion du réseau, etc.). Le déploiement de *gLite* est soutenu par le programme de formation du projet *EGEE*.

XtreemOS : un système d'exploitation distribué

Le projet *XtreemOS* [XtreemOSurl] propose l'utilisation de *Linux* pour supporter un système d'exploitation distribué complètement ouvert. L'idée est d'exporter des interfaces pour résoudre l'inconvénient de *Globus*. En effet, *Globus* se retrouve très vite limité dès lors que la combinaison des services devient trop importante car l'absence d'une interface de programmation commune et d'un modèle d'interaction unifié pose d'énormes difficultés lors de la programmation de nouveaux services et dégrade l'aisance d'utilisation. *XtreemOS* a choisi *Linux* pour des raisons pragmatiques (disponibilité, stabilité, ouverture) et valide ses développements à travers des cas d'utilisation concrets dans des domaines aussi variés que l'aéronautique, l'énergie, les télécommunications, la finance, etc.

Autres solutions

Une solution parfois rencontrée pour la mise en place de grappes de serveur de calcul est la solution *Message Passing Interface (MPI)*¹¹. En pratique, cette solution est réalisée avec des connexions

¹¹*MPI* est une bibliothèque de fonctions *C* pour des programmes massivement parallèles sur des systèmes distribués en grappes [MPIurl]

sécurisées de type *Secure SHell (SSH)* entre les serveurs sur lesquels est installée une bibliothèque de fonctions *MPI* dont l'une des implantation est *MPICH*¹². Cette solution présente quelques difficultés pour l'organisation des clés publiques. Afin d'obtenir des connexions sécurisées et automatisées, le protocole *SSH* requiert l'usage d'agents qui gèrent les couples de clés publiques et privées. Ces clés doivent être préalablement générées et sauvegardées dans une arborescence de fichiers connue et accessible par les agents *SSH*. Les problèmes parfois rencontrés se situent au niveau de la communication inter-grappes et sont liés à la façon dont les clés sont gérées par *MPICH*. Cette solution n'est pas appréciée des administrateurs de réseaux car elle exige d'autoriser des connexions sur tous les ports, ce qui est incompatible avec les principes de sécurité des réseaux. Ceci montre, malgré une complexité apparente, l'intérêt de la gestion de la sécurité dans *Globus*.

Il existe, par ailleurs, de nombreuses technologies pour virtualiser les ressources qui se présentent comme des solutions GRID. L'une d'entre elles s'appuie sur l'intergiciel *VMWare* [VMWareurl] mais repose toutefois sur un produit propriétaire. Une autre solution, sous licence *GPL*, portée sur le traitement de données sur un large système de fichier virtualisé est *Hadoop* [Hadoopurl].

3.2 Principes

La norme *OGSA* est bâtie sur des principes liés aux *architectures distribuées*. Cette section décrit ce qu'est une *architecture distribuée orientée-service* et comment s'organise l'accès aux ressources réparties de façon sécurisée par des communautés d'utilisateurs.

3.2.1 Généralités sur les architectures distribuées

L'architecture d'un système informatisé est dite distribuée quand les ressources sont réparties géographiquement. La topologie d'une telle architecture est un réseau dont les nœuds sont des ressources informatiques qui s'échangent des messages pour exécuter des programmes. Chaque nœud est identifiable par une adresse propre. Deux catégories d'architectures distribuées coexistent sur le réseau *internet* :

- les architectures *orientée-systèmes* dont les nœuds sont des terminaisons physiques ;
- les architectures *orientée-services* dont les nœuds sont des terminaisons logiques.

Pour saisir l'intérêt d'une architecture *orientée-services*, il est utile de partir de l'idée d'une architecture *orientée-systèmes* de type *client-serveur* et d'en transposer les principes. Dans une architecture *orientée-systèmes*, chaque nœud constitue une terminaison physique, c'est-à-dire qu'il représente l'adresse d'un serveur ou d'une grappe de serveurs. Sur le réseau *internet* ces terminaisons correspondent à un couple adresse *Internet Protocol (IP)* et *port* d'une machine. L'interface de communication entre deux nœuds, appelée une *socket* correspond donc à une paire d'adresses *IP* et de *ports*. Dans l'architecture *client-serveur* (voir 3.4) chacune des deux extrémités d'une *socket* correspond soit au *client* soit au *serveur*. Le *client* envoie des requêtes et le *serveur* traite ces requêtes. L'accès aux ressources du serveur s'effectue le plus souvent à travers des *Remote Procedure Call (RPC)* qui permettent l'appel de procédures sur une machine distante. Dans une interaction de type *client-serveur* le rôle des parties (le client ou le serveur) est attribué une bonne fois pour toutes. De ce point de vue, les systèmes *pair-à-pair* sont plus souples que les systèmes *client-serveur* mais sont eux aussi contraints par les limites de l'architecture *orientée-système* de par l'utilisation de terminaisons *IP* liées à la structure physique du réseau de transport. Bien qu'utilisant un mode de programmation *orienté-objet*, les systèmes comme *Common Object Request Broker Architecture (CORBA)* ou *Remote Method Invocation (RMI)* présentent les mêmes caractéristiques que les architectures *orienté-systèmes* car l'identification des objets est liée à la localisation de ces objets dans le réseau physique. Ces architectures se sont fortement répandues car

¹²Des tests de la solution *MPICH* avec *SSH* ont été effectués à l'*INRIA* courant 2003. Ces tests ont mis en évidence les inconvénients de cette solution et l'ont abandonnée au profit de *Globus*.

elles s'appuient sur des technologies qui ont fait leurs preuves. Toutefois, leur gros inconvénient réside dans leur manque d'évolutivité.

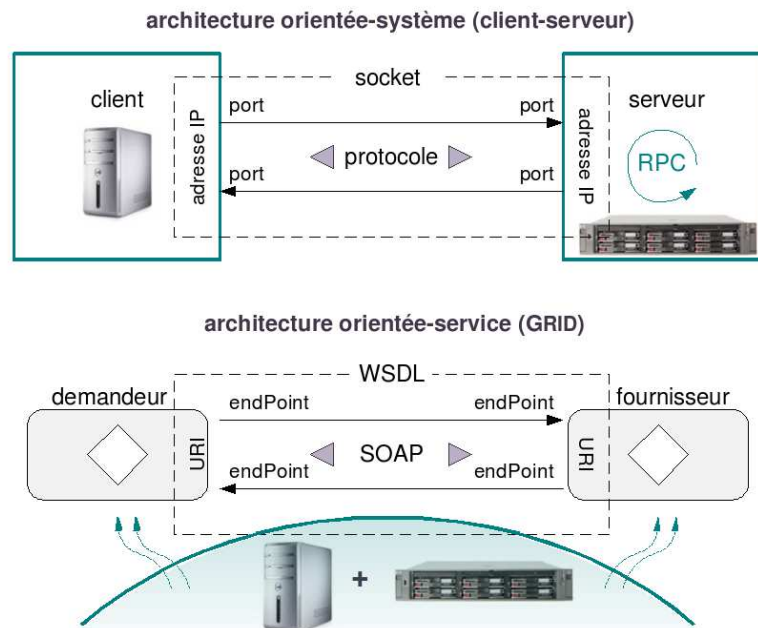


FIG. 3.4 – Comparaison d'architectures *client-serveur* et *orientée-services*

3.2.2 Architectures *orientées-services*

Les travaux réalisés par la communauté *Service Oriented Computing (SOC)* présentent de solides bases pour décrire les principes d'une architecture *orientée-services* [Singh & Huhns2005]. Dans une architecture *orientée-services* les nœuds ne correspondent plus à des serveurs ou à des clients mais à des terminaisons logiques. Ces nœuds se posent tantôt en fournisseur tantôt en demandeur de services (cf. 3.4). Demandeur ou fournisseur de service est un rôle qui n'est pas figé une bonne fois pour toutes mais qui est attribué à titre temporaire, le temps de l'exécution d'une tâche. Un système d'adressage logique, indépendant de la structure physique du réseau, permet l'identification de ces nœuds, de manière à pouvoir localiser les services correspondants. Ainsi, chaque service est adressable de façon unique par son *Unique Registry Identifier (URI)*. En pratique, cet identifiant est généré dynamiquement et assigné au service lors de son instantiation. Les services communiquent grâce au protocole *SOAP* à travers une interface *WSDL*. La flexibilité est certes un avantage des architectures *orientée-services* sur les architectures *orientée-systèmes* mais le désavantage des premières est lié à de piètres performances par rapport aux secondes et à un manque de maturité technologique.

SERVICE

Le terme service, dans une architecture-orientée services, désigne une entité logicielle, conçue à des fins d'interopérabilité entre systèmes hétérogènes par des messages en format *XML* travers le protocole de transport *HTTP*. Un service se présente avec une interface normalisée (spécifiquement en *WSDL*) et communique avec celle-ci par des messages dans un format également normalisé (spécifiquement *SOAP*). Un service se caractérise par le fait (i) qu'il se définit par un langage de description, (ii) qu'il est publié dans un registre de services, (iii) qu'il peut être découvert dans ce registre et (iv) qu'il peut être composé avec d'autres services.

Taxinomie des services

Dans une architecture *orientée-services*, le service constitue l'entité opérationnelle de base. Ces services se distinguent par leur fonction et leurs propriétés. Afin de les classifier, cette taxinomie s'inspire de celle proposée par IAN FOSTER [Foster *et al.*2004b] :

- Une *instance de service* est une entité logicielle identifiable par une adresse unique, dont le rôle est d'exécuter des opérations dans un environnement d'exécution propre et de communiquer avec d'autres instances de service à travers une interface. Idéalement, cette interface et le protocole de communication entre services sont normalisés. Les services GRID se basent sur les normes du *W3C*, *WSDL* comme format d'interface et *SOAP* comme protocole.
- Un *service sans état* est une instance qui accomplit un traitement séquentiel de messages sans utiliser des informations qui ne sont pas contenues dans le message lui-même. L'état des résultats produits par ces opérations n'est pas conservé. Le contexte d'exécution de ces services ne nécessite pas de ressource de mémorisation. Ces services peuvent être assimilés à des fonctions. Un service de compression-décompression de données est un exemple typique de *service sans état*.
- Un *service conversationnel* est une instance qui accomplit un traitement séquentiel de tel sorte que le résultat d'une opération agit sur une opération suivante. Le comportement d'une opération est produit par une séquence logique de messages. De nombreux sites sur le WEB utilisent cette technique à travers l'utilisation de *cookies*.
- Un *service à état* est une instance de service qui peut manipuler des données extérieures au service en réponse à des sollicitations, et qui conserve l'état des résultats produits par ces opérations en vue d'utilisations ultérieures. Le contexte d'exécution de ces services nécessite de la ressource de mémorisation propre.
- Un *générateur de services* ou *service factory* est une instance de service d'un type particulier car sa seule fonction est de générer d'autres instances de service. Cette forme de génération de service constitue l'un des principes fondamentaux de *Service Oriented Computing (SOC)* [Singh & Huhns2005] qui est à la base du modèle *OGSA*.
- Un *service persistant* est instancié lors de l'initialisation de son environnement d'exécution (conteneur de services) et cette instance perdure indéfiniment. Ce service peut être éventuellement détruit par un processus externe (*out-of-band process*) mais reste normalement actif aussi longtemps que son environnement d'exécution le lui permet.
- Un *service éphémère* est instancié par *générateur de service*. Cette instance de service a une durée de vie limitée dans le temps ou plus précisément un état final naturel (*soft-state*). Ces services sont d'une importance capitale dans l'architecture GRID car ils permettent d'allouer ou de libérer de la ressource de façon dynamique et d'être composable de différentes façon. Ainsi plusieurs services élémentaires peuvent être *orchestrés* pour composer un service de plus haut niveau.

SERVICE À ÉTAT

Le terme état est vague dans la mesure où il peut couvrir de nombreux aspects d'un système informatisé. Le sens admis par la communauté GRID est qu'un service à état dispose d'une ressource dédiée de telle sorte que l'état de cette ressource peut s'exprimer selon une syntaxe univoque (*i.e* document *XML*), a un cycle de vie bien défini dans le temps et peut interagir avec d'autres services (avec ou sans état). De nombreux exemples, tels qu'une structure de fichiers dans un arbre ou des entrées dans une base de données, ou encore une combinaison de plusieurs de ces objets, peuvent être modélisés par des services à état (FOSTER [Foster *et al.*2004b]).

Propriétés du service à état

Les propriétés du service à état découlent du modèle **générateur de services** → **instance de service** tel que spécifié par *OGSA* qui s'inspire du modèle *orienté-objet* **classe d'objet** → **instance d'objet**. Puisqu'un *service à état* est caractérisé par sa capacité à mémoriser le résultat d'actions antérieures, il acquiert de cette façon une forme d'autonomie. Il peut gérer ses propres opérations, temporisées ou non, indépendamment des connexions qu'il reçoit. Dans la terminologie *OGSA*, un service est dit *persistant* si celui-ci est instancié lors de l'initialisation de son environnement d'exécution (*i.e* un conteneur de services) et actif en permanence dans ce conteneur.

3.2.3 Gestion des ressources réparties

Le principal défi de GRID est de coordonner de façon dynamique des ressources partagées entre diverses institutions. Le partage ne concerne pas seulement des fichiers mais aussi des accès à toutes sortes d'équipements informatisés. Une infrastructure GRID est constituée d'un ensemble de ressources informatiques réparties géographiquement. L'usage de ces ressources est dynamique, ce qui implique une gestion qui prévoit des mécanismes de réservation de ressources pour des tâches planifiées et des mécanismes de libération de ces ressources une fois ces tâches accomplies. Ce principe se décompose en trois étapes élémentaires de la façon suivante :

1. **Mutualisation** de ressources informatiques hétérogènes distribuées ;
2. **Virtualisation** de cet ensemble de ressources en un unique *super-ordinateur* ;
3. **Réification**¹³ de parties de cet ensemble ainsi virtualisé. Cette étape permet d'allouer dynamiquement de la ressource aux services selon leur besoin.

Cette ressource ainsi rationalisée se traduit par de la capacité informatique quantifiable qui se présente sous trois formes :

- capacité de **traitement**
- capacité de **stockage**
- capacité d'**échange**

Nous appelons **hôte** une unité élémentaire de ressource comprenant ces trois formes de capacité. La figure 3.5 est une représentation schématique d'un *hôte* selon l'architecture VON NEUMANN avec les trois formes de capacité décrites ci-dessus. Ces capacités sont toujours présentes dans des proportions extrêmement variées selon les *hôtes*. Elles sont couplées en interne par un bus.

Pour qu'un *hôte* devienne partie intégrante d'une infrastructure GRID, il doit satisfaire les deux conditions suivantes :

- Avoir un intergiciel GRID installé et convenablement configuré (par exemple *Globus*).

¹³En informatique, la **réification** consiste à transformer un concept abstrait en un objet concret. Ici, la ressource virtualisée est vue comme un concept abstrait : une forme de super-ordinateur virtuel. Pour être utilisée, cette ressource est restituée par le biais de conteneurs de services. Ainsi, ces conteneurs sont des objets qui réifient une portion de ressource.

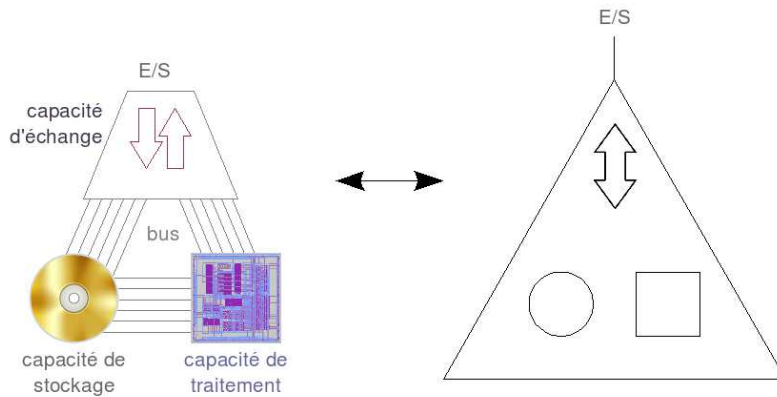


FIG. 3.5 – Représentation schématique d'un *hôte* selon l'architecture VON NEUMANN

- Être reconnu par l'infrastructure (par exemple un certificat d'hôte valide l'appartenance de l'hôte à l'infrastructure, comme décrit dans la section 3.2.4 traitant de la sécurité).

Les *hôtes* sont interconnectés par un réseau. Ce réseau est généralement *internet* mais ce peut être un réseau spécialisé dans le cas notamment des interconnexions locales. Pour simplifier, nous considérons qu'un *hôte* est identifiable par une adresse fixe et unique. Cette adresse est, dans la vaste majorité des cas, une adresse *Internet Protocol version 4 (IPv4)*.

Dans notre formalisme graphique initié dans [Dugénie2005] puis spécifié en détail dans [Jonquet *et al.*2007], nous avons prévu de décrire comment s'organise la gestion des ressources depuis les éléments de capacité élémentaires jusqu'à l'étape de réification des ressources virtualisées. La figure 3.6 décrit ces différentes étapes de gestion des ressources.

Les étapes de mutualisation puis de virtualisation des *hôtes* GRID mènent à la constitution d'une large capacité de ressources élémentaires qui se mesure en *PetaOctets (PO)* pour le stockage et en *TeraFlops (TF)* pour la capacité de traitement. Par contre la capacité d'échanges va dépendre de la localisation des ressources et du couplage entre ces ressources. Si toutes ces ressources sont interdépendantes entre elles, nous voyons qu'il existe des solutions optimales pour les disposer dans une topologie adéquate. L'optimisation des flux et la localisation des ressources sont des problèmes du ressort de l'ingénierie des systèmes en réseaux et de la recherche opérationnelle. Ces questions ne sont pas abordées dans le cadre de cette thèse.

3.2.4 Sécurité

La sécurité est un élément fondamental de *Grid* car elle détermine une topologie de reconnaissance mutuelle sur un ensemble d'entités que sont les utilisateurs et les ressources informatiques. La sécurité intègre les principes suivants :

1. La **confidentialité** qui repose sur un chiffrement à clé asymétrique.

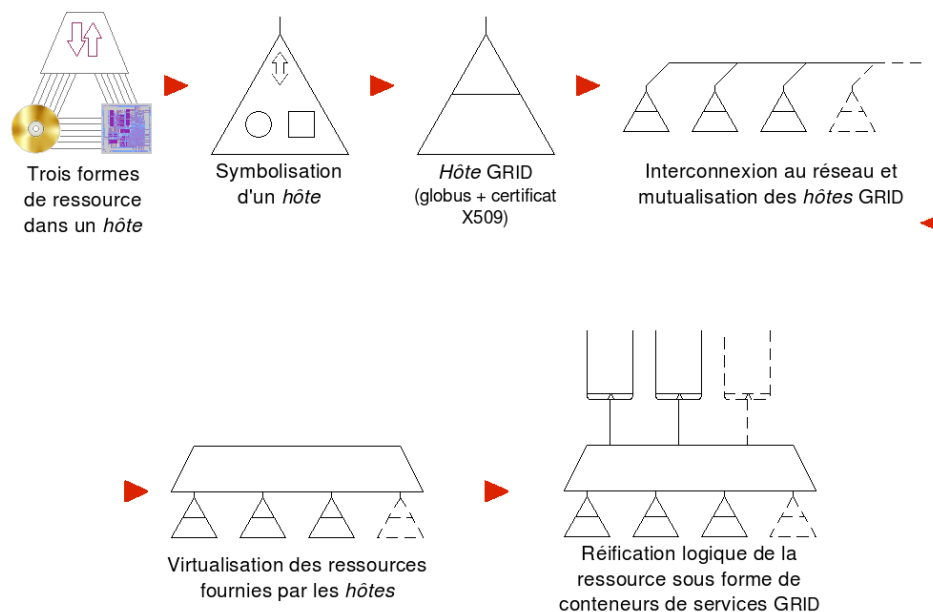


FIG. 3.6 – Différentes étapes pour la gestion des ressources dans GRID

2. L'**intégrité** qui repose sur un *algorithme de hachage*.
3. L'**authenticité** mutuelle qui repose sur l'échange de certificats X509 comportant des signatures électroniques.
4. La **délégation** qui assure la délivrance de certificats X509 valides sur une courte période.
5. L'**autorisation** qui détermine les conditions d'accès aux services d'une *organisation virtuelle*.

Les quatre premiers principes sont décrits par la spécification *Grid Security Infrastructure (GSI)* [Foster *et al.*1998]. Le cinquième principe relève d'une autre spécification décrite par le *Community Authorization Service (CAS)* [Pearlman *et al.*2002].

Sécurité 1/5 : confidentialité

La confidentialité de l'information est assurée dans *GSI* par un chiffrement à clé asymétrique de type *RSA*¹⁴. Cet algorithme consiste à générer deux clés différentes, l'une privée, l'autre publique, pour chiffrer puis déchiffrer les messages. L'émetteur d'une information chiffre ses messages avec la clé publique du destinataire. Ce dernier peut, à son tour, déchiffrer le message avec sa clé privée.

Sécurité 2/5 : intégrité

L'intégrité signifie que le destinataire d'une information est assuré de la conformité de cette information avec l'original. Pour ce faire, l'information est transmise avec une *empreinte* qui a été générée à partir de cette information par une *fonction de hachage*. Ce principe consiste à associer une série d'octets de taille déterminée à une chaîne de caractères de taille quelconque. Si l'information est modifiée durant son transfert, celle-ci n'est plus conforme avec son *empreinte* et le test d'intégrité échoue. Cette

¹⁴ Rivest, Shamir, Adleman (RSA), du nom des trois inventeurs [RSAurl].

transformation n'a pas de fonction réciproque, autrement dit, il n'est pas possible de déduire la chaîne de caractères originale à partir de l'*empreinte*. L'algorithme de hachage utilisé par *GSI* est le *Message Digest 5 (MD5)* [MD5url]. Par exemple, l'expression « *Bonjour de Montpellier* » a pour *empreinte MD5* la chaîne b41162f945e36624b357a2ffc1290f8a. Un autre intérêt des *fonctions de hachage* est de pouvoir mémoriser des mots de passe chiffrés dans une base de données.

Sécurité 3/5 : authenticité

Pour qu'une conversation soit totalement sécurisée, la confidentialité et l'intégrité des messages ne sont pas suffisantes. En effet, un agent mal intentionné pourrait transmettre une information en usurpant l'identité d'un autre agent puisqu'il chiffre ses messages avec la clé publique de son destinataire. Pour éviter ce type d'abus, *GSI* prévoit un mécanisme de *reconnaissance mutuelle d'authenticité* sur la base de certificats *X509* [CCITT1988] qui contiennent un certain nombre d'informations dont une signature chiffrée de l'émetteur. Cette signature est produite à partir de l'*empreinte* des informations sur l'identité du porteur et chiffrée avec la clé privée de l'émetteur. Ainsi, en comparant l'*empreinte* du certificat et la signature déchiffrée avec la clé publique de l'émetteur, le destinataire peut valider l'authenticité du message. Ce mécanisme est basé sur *Transport Layer Security (TLS)* [TLSurl], un dérivé de la norme *Secure Sockets Layer (SSL)* [SSLurl]. *GSI* permet l'assignation de certificats aux *utilisateurs* et aux *hôtes*.

La confiance au sein d'une communauté peut ensuite être instaurée au moyen d'un système de validation des certificats. Un certificat est valide s'il a été signé par une *autorité* habilitée à le signer ou *Certification Authority (CA)* [CAurl]. Cette *autorité* est elle-même porteuse d'un certificat valide. Il apparaît ici une structure hiérarchique qui implique l'existence d'une *autorité* originale. Cette *autorité*, qui a elle-même signé son propre certificat, se place à la racine de cette structure. La figure 3.7 décrit comment s'organise une telle structure avec deux *autorités*, dont l'une est à la racine. Les flèches indiquent la relation « *est signataire du certificat X509 de...* ». Dans la terminologie *GSI*, cette structure arborescente constitue un *trust* (voir encadré) que l'on peut décrire comme un espace à l'intérieur duquel la confiance mutuelle est garantie. En pratique, lors des échanges, les parties disposent d'une copie du certificat de l'*autorité* dont la clé publique est contenue dans ce certificat. Ceci évite la multiplicité des requêtes pour l'obtention de la clé publique de l'*autorité* de certification.

TRUST

Terme anglais de la terminologie GRID qui n'a pas d'équivalent simple en français. Un *trust* est un ensemble d'entités correspondant aux ressources et aux utilisateurs formant un ensemble GRID cohérent. Ces éléments se reconnaissent mutuellement comme faisant partie d'un même *trust*. En pratique, un *trust* est circonscrit par l'origine des signatures des certificats *X509* (cf. figure 3.7). En général, un environnement *Grid* couvre plusieurs domaines ayant leurs propres politiques de sécurité. Le *trust* permet de contrôler les interactions entre domaines en s'adaptant aux différentes politiques de sécurité locales des domaines.

Sécurité 4/5 : délégation

La délégation consiste à permettre à un agent de donner temporairement un pouvoir à un autre agent. Une situation typique se présente lorsqu'un service doit accéder à une ressource indépendamment de l'utilisateur. Pour ce faire, ce service doit disposer des privilèges pour accéder à cette ressource. Ces privilèges sont accordés temporairement par l'utilisateur sous forme d'un certificat signé par cet utilisateur. *GSI* prévoit la création d'une *procuration* appelée *Grid Proxy Certificate (GPC)* dans la terminologie *GSI*. *GPC* est une extension du protocole *TLS*. Ce certificat est une forme de mandat pour réaliser une séquence de tâches bien définie dans le temps afin de réduire le nombre de requêtes d'autorisation.

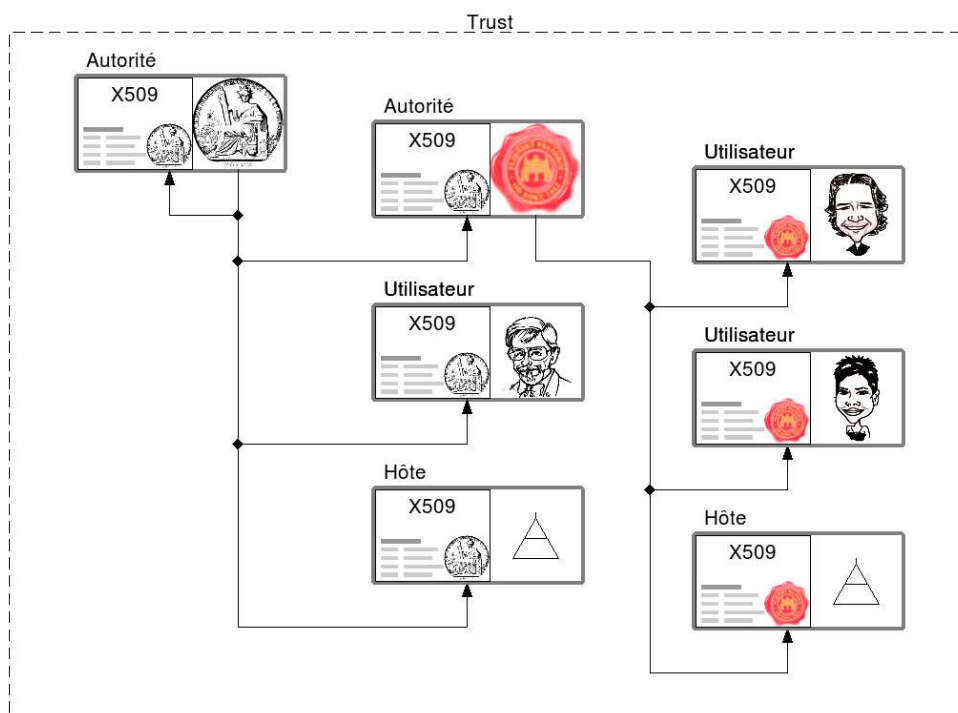


FIG. 3.7 – Structure hiérarchique de validation des certificats X509

Sécurité 5/5 : autorisation

L'autorisation est un principe de sécurité qui spécifie la relation entre les membres et les services d'une *organisation virtuelle*. Ce mécanisme est décrit dans la section 3.2.5.

3.2.5 Organisations virtuelles

Généralités

Dans sa définition originale, une *organisation virtuelle* est une collection dynamique d'individus, d'institutions et de ressources [Foster *et al.*2001]. Cette définition a évolué au fil du temps. Elle peut être de taille extrêmement variable en termes d'individus, de ressources et de durée indéfinie allant de quelques heures à quelques années. Bien que cette définition ne le précise pas de façon explicite, l'article dont elle est extraite sous-entend que chaque *organisation virtuelle* dispose d'un contrôle total sur les règles de sécurité vis-à-vis des ressources qui lui sont allouées. Cet aspect est un apport particulièrement novateur de GRID. Selon IAN FOSTER, les solutions d'architectures distribuées *orientées-systèmes* ne sont pas adaptées à la gestion d'*organisations virtuelles* multiples [Foster *et al.*2001] : il cite notamment CORBA qui ne gère que l'équivalent d'une simple *organisation virtuelle* ; *Distributed Computing Environment (DCE)*, basé sur le principe des *RPC*, qui supporte de la ressource sécurisée mais est trop peu flexible ou encore *Storage Service Provider (SSP)* qui ne fonctionne qu'à travers un *Virtual Private Network (VPN)*. De manière générale, les exemples cités sont composés d'un groupe de serveurs dont les utilisateurs se reconnaissent par le fait qu'ils appartiennent au même domaine. Or, le gros inconvénient du domaine est la rigidité de la relation entre adresses *IP* et domaines *DNS*. De fait, ceci crée une forte dépendance entre la structure des *organisations virtuelles* et la topologie de l'infrastructure. Le *VPN* est une solution qui permet de contourner en partie ce problème d'un point de vue de la répartition géogra-

phique mais n'est pas adapté à des situations très dynamiques lorsque les membres et les ressources des *organisations virtuelles* doivent être souvent réalloués.

Parmi les *organisations virtuelles* existantes, nombreux sont les projets basés aux États-Unis avec des vocations scientifiques diverses : *Sloan Digital Sky Survey* en astronomie ; *Collaboratory for the Multi-scale Chemical Science (CMCS)* en chimie ; *Network for Earthquake Engineering Simulation (NEES)* et *Southern California Earthquake Center (SCEC)* en géologie ; *Earth System Grid* et *Linked Environments for Atmospheric Discovery (LEAD)* en climatologie ; *Long Term Ecological Research (LTER)* en écologie ; *Biomedical Informatics Research Network (BIRN)* en médecine ; *Laboratory for the Ocean Observatory Knowledge INtegration Grid (LOOKING)* en océanographie ou encore *Laser Interferometer Gravitational-Wave Observatory (LIGO)* en physique.

Gestion des autorisations d'une organisation virtuelle

Une solution rudimentaire, initialement proposée par *Globus* pour gérer les autorisations d'une *organisation virtuelle*, consiste à utiliser une table de correspondance entre utilisateurs et services à partir d'un simple fichier (connu sous le nom de fichier `gridmap-file`). Lorsque le nombre d'utilisateurs augmente dans une large mesure, l'utilisation d'un simple fichier devient une solution complexe à administrer. Ainsi, plusieurs solutions sur la base d'un annuaire *Lightweight Directory Access Protocol (LDAP)* ont été proposées, comme celle des projets *EDG* [`DataGridurl`] et *DataTag* [`DataTagurl`]. Une solution de ce type plus élaborée est apparue ensuite avec *Community Authorization Service (CAS)*. *CAS* permet de déléguer la gestion d'un ensemble de ressources aux membres d'une *organisation virtuelle*, en laissant aux membres qui en ont le droit la possibilité d'ajuster les permissions des autres membres vers les services [Pearlman *et al.*2002]. *Virtual Organization Membership Service (VOMS)* est une solution alternative au *CAS* [Alferi2003] qui gère des comptes utilisateurs dans une base de données relationnelle de type *MySQL*. Ce service exploite davantage les concepts *GSI*, tel que la délégation à base de *GPC* ou encore la simple authentification en remplaçant l'utilisation de `grid-proxy-init` par une procédure plus évoluée, `voms-proxy-init` et présente quelques avantages sur *CAS* en agrégeant l'accès à de multiples *organisations virtuelles* simultanément. Toutefois, *VOMS* propose une solution d'administration centralisée des droits d'une *organisation virtuelle* avec l'idée qu'une *organisation virtuelle* est une structure hiérarchique composée de groupes et de sous-groupes. Dans la terminologie *VOMS* un groupe est un sous ensemble d'une *organisation virtuelle* composé de membres. Les groupes sont organisés de façon hiérarchique sous forme arborescente. avec un groupe racine à la base. *VOMS* introduit aussi la notion de groupe primaire pour chaque utilisateur. La notion de rôle permet d'attribuer des droits à un utilisateur pour exécuter des tâches. La distinction entre rôle et privilèges au sein d'un groupe est parfois un peu floue. Un utilisateur occupe un rôle de façon passagère alors que les privilèges sont déterminés au moment où un utilisateur devient membre d'un groupe.

VOMS apporte une dimension relationnelle supplémentaire avec la relation *Groupe-Rôle* dont nous devons tenir compte dans notre démarche de modélisation.

Caractérisation d'une organisation virtuelle

En somme, une *organisation virtuelle* se présente comme un système multi-utilisateurs virtuel dont la taille peut varier dans le temps. C'est une couche d'abstraction très dynamique et indépendante de la structure physique des domaines.

La structure logique d'une *organisation virtuelle* doit être réduite à sa plus simple expression : d'une part ses membres, d'autre part ses services (le service est une couche d'abstraction vers des ressources). Une *organisation virtuelle* dispose d'un conteneur de services unique qui fournit les ressources aux services et dans lequel sont définies les autorisations d'accès aux services pour les membres. D'un point de vue de la sécurité, l'authentification est gérée au niveau du *trust* qui, d'un point de vue hiérarchique est placé au dessus de l'*organisation virtuelle*.

3.3 Synergie entre GRID et les SMA

3.3.1 Un modèle GRID simplifié

Les sections précédentes montrent, d'un côté, l'énorme potentiel de GRID en termes de gestion des ressources et en termes de sécurité, et, de l'autre côté, font apparaître une relative complexité du modèle d'architecture. Certes, ceci tient du fait de la richesse des fonctionnalités mais aussi parce que GRID est un objet très convoité qui attise déjà les intérêts industriels alors qu'il n'a pas encore atteint une pleine maturité conceptuelle. Pour cela nous avons jugé pertinent de faire abstraction des évolutions industrielles propres à une vision particulière en nous focalisant uniquement sur les concepts fondamentaux afin d'en dégager une représentation simplifiée. En nous appuyant sur l'ouvrage de référence [Foster & Kesselman2003] et les publications originales [Foster *et al.*2001, Foster *et al.*2002], une représentation des concepts clés de GRID, sous forme d'un formalisme graphique, a été présentée pour la première fois en 2005 dans l'un de nos articles de vulgarisation [Dugénie2005]. Une synthèse de ces concepts est représentée sur la figure 3.8. La gauche de cette figure représente les différents concepts et leur variantes (service, organisation virtuelle et ressource). La droite de cette figure représente les relations et les opérations dynamiques telle que la relation de membre, l'accès à un service par un membre d'une organisation virtuelle, l'instanciation d'un service par un générateur de service, deux services en interaction ou encore l'exécution d'un service dans son conteneur.

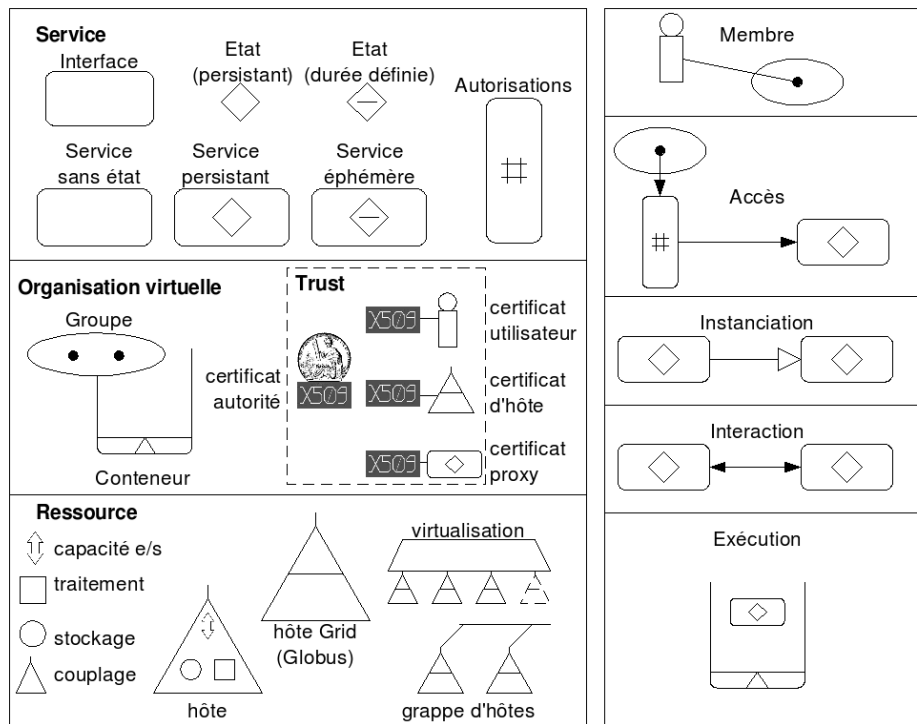


FIG. 3.8 – Synthèse des concepts GRID

Ces concepts, une fois établis, ont été réassemblés pour former un modèle d'architecture GRID simplifié (voir figure 3.9). Le bas de la figure représente l'espace des ressources composé d'hôtes ou de grappes d'hôtes. Des certificats X509 sont attribués aux hôtes et aux utilisateurs pour former un *trust*. Les deux opérations de virtualisation puis de réification de ces ressources permettent de constituer des conteneurs de services pour chaque organisation virtuelle. Le haut de la figure représente la formation des groupes composés de membres qui disposent de droits d'accès aux services de la communauté virtuelle. Ces droits sont définis dans une matrice $M \times S$ qui établit la relation entre chaque membre M avec

chaque service S . Bien que ce ne soit pas visible sur cette figure dans un souci de clarté, cette matrice peut être extensible à une matrice $(M + S) \times S$ si l'on considère qu'au sein d'un même conteneur, tout service peut aussi avoir des droits sur les autres services.

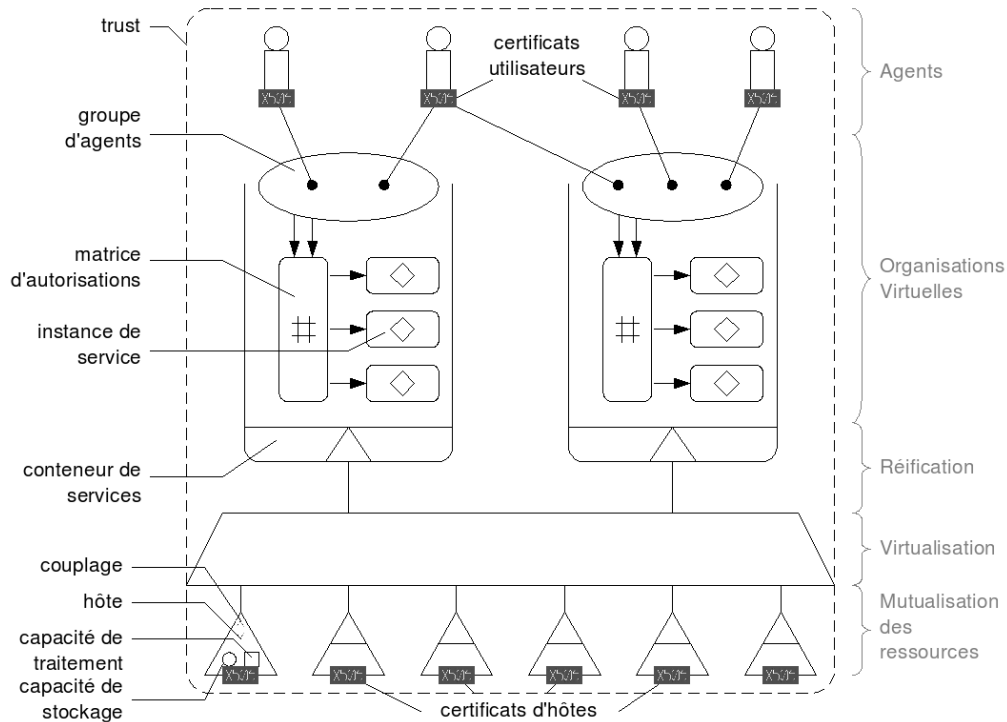


FIG. 3.9 – Architecture GRID simplifiée

Ce formalisme a été appliqué dans la recherche des synergies potentielles entre GRID et les SMA. Ce formalisme a d'abord permis d'effectuer une analyse conjointe des deux domaines pour en déduire une ontologie unifiée [Duvert *et al.*2006]. Ensuite, cette ontologie a permis de développer un langage de description [Jonquet *et al.*2007, Jonquet *et al.*2006].

3.3.2 Analogies

Bien que GRID et les SMA soient deux types de systèmes distribués, leurs motivations sont à la base différentes. GRID vise à définir une infrastructure pour sécuriser et fiabiliser l'accès à des ressources partagées à travers des services tandis que les SMA se concentrent sur les aspects liés aux interactions entre agents, de façon semblable à l'idée d'*intelligence collective* [Ferber1995]. IAN FOSTER a posé les bases d'une complémentarité par la métaphore du corps humain en situant les agents au niveau du cerveau et l'infrastructure GRID en tant que force musculaire [Foster *et al.*2004a]. Depuis lors, de nombreuses analogies ont émergé de la communauté SOC [Singh & Huhns2005]. Nos récents travaux portent sur l'analyse des analogies que présentent les interactions entre agents et les services d'une architecture GRID [Jonquet *et al.*2006, Jonquet *et al.*2007]. Ces travaux viennent compléter ceux réalisés dans le domaine de la *Génération Dynamique de Services (GDS)* qui consiste à construire dynamiquement un service en fonction des conversations entretenues avec les utilisateurs de ce service. Par cette approche, la GDS entrevoit des avancées significatives avec l'intégration de ces deux domaines [Jonquet2006].

Comme le représente la figure 3.10, le concept de service est à l'intersection des domaines GRID et SMA et constitue la clé de voûte de leur intégration. GRID prévoit la gestion de l'état et la durée de vie des services. Les SMA sont composés d'agents autonomes, doués d'une capacité d'intelligence et en

perpétuelle interaction. De plus en plus, les agents sont vus comme des entités capables de fournir des services dynamiques, adaptables, composables et sémantiques [Singh & Huhns2005].

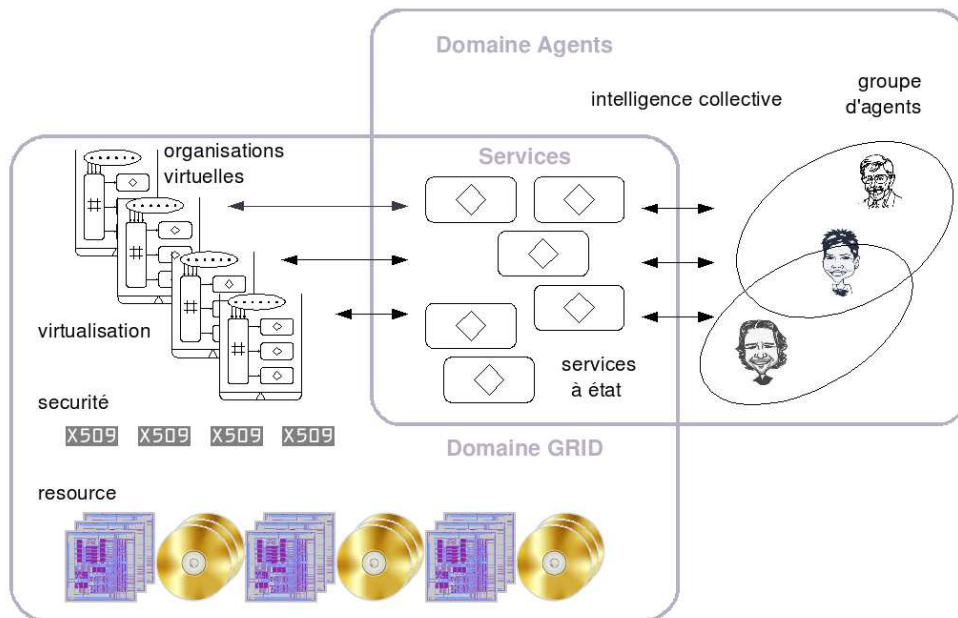


FIG. 3.10 – Le concept de service à l’intersection des domaines GRID et SMA

3.3.3 Modèle d’intégration

Selon les différentes règles précédemment formalisées, ceci a permis de définir un modèle d’intégration GRID et SMA [Jonquet *et al.*2006] dont voici le récapitulatif :

- Les ressources sont délivrées par des hôtes qui regroupent chacun une unité de calcul et une unité de stockage et qui sont couplés entre eux fortement (grappes de serveurs) ou faiblement (sur une longue distance).
- Deux opérations complémentaires sur ces ressources sont la *virtualisation*, puis la *réification* de ces ressources dans des conteneurs de services.
- Ces conteneurs permettent aux services de disposer d’un environnement d’exécution propre.
- Ces conteneurs de services constituent une entité de ressources sécurisées dédiées à un groupe d’agents.
- Une *relation bijective* associe l’ensemble des conteneurs de service et l’ensemble des groupes d’agents. Chacune de ces associations forme une *organisation virtuelle*. Ces conteneurs de services constituent ainsi une entité de ressources virtualisées dédiées à un groupe d’agents.
- Un groupe d’agent est constitué de membres qui peuvent être des agents humains ou des agents artificiels. Un agent peut être membre de plusieurs groupes.
- Les services présents dans un conteneur peuvent être de différente nature : avec ou sans état, persistant ou éphémère.
- Chaque service possède un identifiant unique et une interface normalisée.
- Tout service, qu’il soit persistant ou éphémère, peut instancier un autre service à partir du moment où il en a les droits.

- Des agents humains qui disposent d'un certificat (utilisateur ou mandat) valide peuvent être intégrés à un groupe selon une politique d'ajout de membre propre au groupe en question.
- Dans le cas d'agents artificiels qui disposent d'un certificat valide (mandat), ceux-ci sont autorisés à accéder à des services du groupe ou instancier de nouveaux services dans le conteneur de ce groupe selon une politique propre au groupe en question.
- Un conteneur de services est lui-même une instance de service. A son initialisation, il est constitué d'un ensemble de services de base (gestion des membres, gestion des droits et des services) qui permettent d'amorcer le fonctionnement de l'organisation virtuelle.
- La relation entre les membres d'un groupe et les services de ce groupe est gérée par le service des autorisations. Ce service maintient une relation ternaire entre les agents, les groupes et les services.
- L'interaction entre agents sont des échanges de service. Le service est le concept unificateur du modèle d'intégration GRID et SMA.
- Le modèle relationnel unificateur est le modèle *Agent Groupe Rôle (AGR)* avec l'ajout de la notion de service GRID.

3.4 Projection future

Désormais, avec un réseau en épine dorsale délivrant des débits de plusieurs *GigaBits/s* et des ressources informatiques convenablement réparties sur ce réseau, GRID se présente comme une infrastructure générique qui révolutionne l'approche classique dans la conception des applications. A travers les réseaux à très haut débit comme *Renater* en France, ou *Geant* au niveau européen (voir figure 3.11), il devient possible d'envisager des applications capables d'exploiter de nombreuses ressources dispersées géographiquement (supercalculateurs, bases de données, instruments scientifiques).

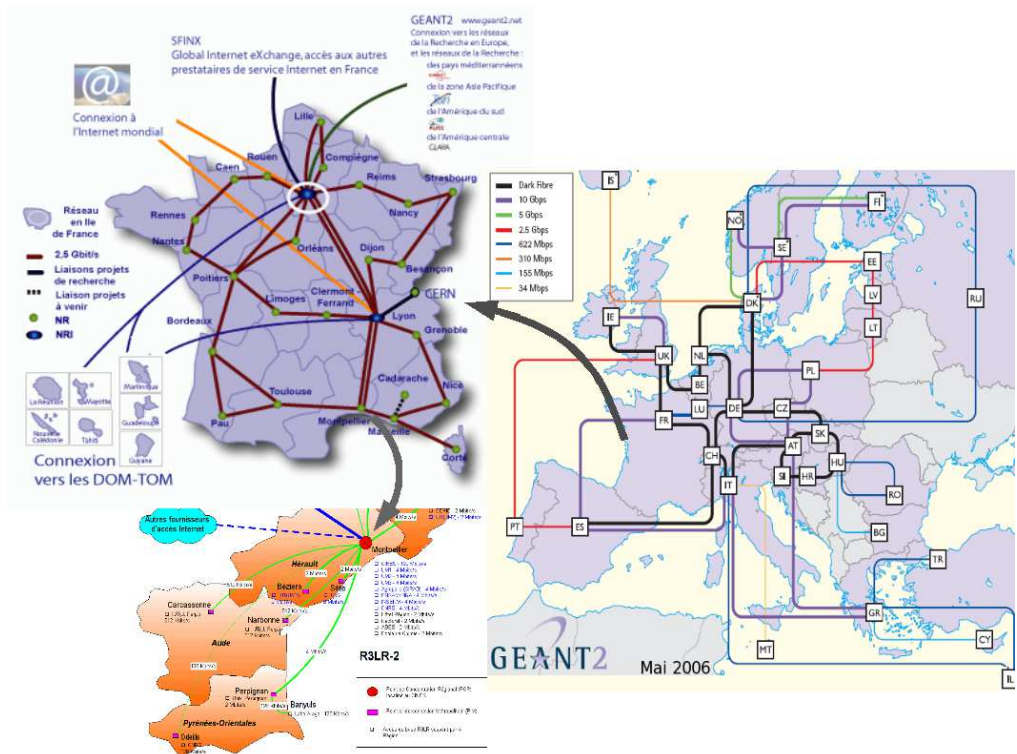


FIG. 3.11 – Réseaux *Geant*, *Renater* et *R3LR4* au niveau de la région LR

3.4.1 Bilan de l'activité de recherche

L'évolution de l'activité de recherche dans le domaine de GRID peut se résumer succinctement par un bilan quantitatif des articles parus dans les revues internationales les plus citées. On observe une tendance à l'accroissement du nombre d'articles qui rapprochent les thématiques autour de la connaissance, de l'ubiquité et de GRID. D'un nombre marginal de publications en 2001 contenant tous ces mots clés, nous en avons dénombré une trentaine en 2005 dans les différentes revues et conférences de l' *International Electrical and Electronics Engineering (IEEE)*. Les journaux les plus actifs dans ces domaines sont : *IEEE Transactions on Knowledge and Data Engineering* et *IEEE Intelligent Systems*. Parmi les conférences, on retiendra surtout *Information Technology Coding and Computing (ITCC)*, *High Performance Distributed Computing (HPDC)*, *Web Intelligence (WI)* et *International Conference on Advanced Learning Technologies (ICALT)*.

La majeure partie de ces articles porte sur l'application de techniques d'ingénierie de la connaissance, comme la fouille de données ou encore la gestion de bases de connaissances.

3.4.2 Ubiquité informatique

A travers notre analyse du contexte technologique, nous constatons que la vocation principale de GRID reste essentiellement tournée vers le calcul intensif ou le traitement de données en masse. En adoptant une vision SMA, cette approche du calcul intensif peut s'apparenter au comportement d'un système où chaque agent correspond à un processus de traitement. Dans ce cas de figure, un agent est, à son niveau, hautement consommateur de ressources. Leur nombre dépend surtout de la capacité de ressources mises à disposition. Une autre façon de considérer GRID avec une vision SMA est de partir d'une population d'agents donnée mais avec une consommation moyenne de ressource par agent limitée par la capacité globale de ressources.

Cette seconde vision est davantage en phase avec les récentes idées d'intégration qui proposent de faire converger les technologies propres à l'*ubiquité informatique* avec les technologies GRID [Denis *et al.* 2003, Sugiuchi *et al.* n.d.]. En effet, l'*ubiquité informatique* telle que décrite, dans la section 2.4.3 nécessite de la ressource accessible en tout lieu et à tout instant. En d'autres termes, il convient de voir qu'il s'agit ici d'une ressource fournie par l'infrastructure pour le besoin immédiat d'un *espace collaboratif*. Or, pour que l'infrastructure puisse délivrer et coordonner de la ressource à la demande, elle doit nécessairement adopter des concepts GRID, sans quoi ce besoin ne pourrait évoluer dans le temps. Par conséquent, un *espace collaboratif* devient réellement *ubiquitaire* lorsqu'il est opérable sur une infrastructure de ce type. Des *espaces collaboratifs* peuvent être conçus dans d'autres cas de figure (infrastructure propriétaire, VPN, etc.) mais ils perdront le caractère d'*ubiquité* dès lors où ils seront déployés sur une infrastructure figée par sa propre topologie physique.

C'est probablement la raison pour laquelle l'idée d'*informatique ubiquitaire*, datant de 1991, soit restée orpheline depuis lors. Nous comprenons ainsi mieux, par cette nouvelle adéquation technologique, pourquoi l'*ubiquité informatique* montre aujourd'hui un intérêt croissant au sein de la communauté GRID [Denis *et al.* 2003]. Nous remarquons aussi l'effort en ce sens de la recherche européenne qui a ciblé certains de ses objectifs sur l'*ubiquité informatique* dans son programme cadre 2005-2006 avec *Ubiquitous Grid Services (UGS)*¹⁵ [IST2006].

Nous pouvons enfin relater une vision d'intégration technologique proposée par un projet qui a attiré notre attention : *Grid Resource Broker (GRB)* [G. Aloisio 2007] conçu sur *Globus* qui fixe un cadre général pour l'accès à des ressources GRID à travers un bureau ubiquitaire. Dans son principe, *GRB* consiste à spécifier une interface générique pour exploiter des ressources GRID à travers un portail accessible par un navigateur.

¹⁵ Information Technologies Society (IST) 5^e Appel à projets

3.4.3 Infrastructure idéale pour *espaces collaboratifs ubiquitaires*

Bien que ce ne soit pas une tâche aisée de convaincre des entités individuelles, organismes publics ou entreprises privées de partager leurs ressources à un niveau collectif, il n'en reste pas moins que une infrastructure à ressources mutualisées présente de nombreux avantages économiques et techniques pour déployer des *espaces collaboratifs ubiquitaires*. Par exemple, dans le domaine industriel, nous constatons un besoin croissant pour une plus grande souplesse dans la gestion des systèmes d'information. Au fil de l'évolution de leurs stratégies (fusions, acquisitions, diversification des offres commerciales, etc.), les entreprises cherchent à *urbaniser* leurs services¹⁶, c'est-à-dire organiser la transformation progressive et continue de leur système d'information afin de le rendre le plus flexible possible vis-à-vis des évolutions stratégiques [Sassoon1998]. Il en va de même pour les organisations scientifiques qui misent sur l'interdisciplinarité et sur la coopération entre laboratoires tant au niveau national qu'international. Pour cela, nous avons identifié quatre critères pour qu'une telle infrastructure puisse être acceptée dans une large mesure :

1. le critère de **confidentialité et d'intégrité** des données et des échanges ;
2. le critère de **haute disponibilité** du service ;
3. le critère d'**indépendance** vis-à-vis des technologies ;
4. le critère d'**autonomie** des organisations virtuelles.

L'état de nos investigations indique que *OGSA* présente de nombreux avantages sur les trois premiers critères. *GSI* apporte une réponse adaptée à la confidentialité des échanges et à l'intégrité des données. *GRAM* permet d'assurer une qualité de service adaptée aux exigences de haute disponibilité et d'équilibre de charge, par replication des instances et par une recherche dynamique de solutions optimales. De plus, *OGSA* prévoit des mécanismes de supervision des défaillances pour accroître la disponibilité des services. Le *service à état* permet de déployer des services ubiquitaires qui se chargent de tous les aspects d'interopérabilité. Ceci garantit une quasi indépendance technologique au niveau du client.

Par contre, nous nous sommes restés dubitatifs face aux approches proposées pour la gestion des *organisations virtuelles*. D'abord le concept d'*organisations virtuelles* est posé de manière assez vague si bien qu'il existe différentes manières de le comprendre. La structure arborescente des relations entre utilisateurs et groupes dans un annuaire *LDAP* proposée par *globus* présente de nombreux inconvénients. Ceci implique notamment l'existence d'un super-utilisateur qui dispose de tous les privilèges d'administration (ajouter/retirer des membres, activer des services, etc.). Le modèle relationnel introduit par *VOMS* résoud le problème de hiérarchie des groupes et des membres mais implique l'existence d'une relation entre les *organisations virtuelles* et des *sous-groupes* qui rend la structure du modèle peu enclin à l'autonomie organisationnelle des groupes.

En explorant les approches des *SMA* le modèle *AGR* nous a paru hautement plus séduisant. Derrière une simplicité conceptuelle composée d'une relation ternaire entre les trois concepts Agent Groupe et Rôle, se dessinent de nombreuses potentialités pour l'autonomie organisationnelle des groupes.

Ces bases ainsi posées, nous ont permis d'aborder, dans la suite, une démarche de modélisation qui tient compte de ces critères dans leur ensemble.

¹⁶Le concepts d'*urbanisation* de l'espace anthropologique du territoire a été appliqué en informatique pour modéliser l'agencement des systèmes d'informations des entreprises.

MODÈLE ET PLATEFORME AGORA

Nous ne raisonnons que sur des modèles.
Paul Valéry

CE CHAPITRE décrit le modèle d'architecture AGORA, un *espace collaboratif ubiquitaire* conçu pour une infrastructure exploitant, de façon originale, le caractère complémentaire et innovant de GRID et des *Systèmes Multi Agents (SMA)*. Selon les conclusions du chapitre 2, les modèles d'*espaces collaboratifs* existants sont, *de facto*, limités par les architectures actuelles. Par ailleurs, le chapitre 3 montre que, grâce au service à état, l'architecture *orientée-services* GRID résout théoriquement une partie des recommandations du chapitre 2 telles que l'ubiquité, la sécurité ou encore le passage à l'échelle. Pour ce qui est de l'auto-organisation des groupes, cette recommandation est résolue avec l'apport des *SMA*.

Table des matières

4.1	Cadre conceptuel	68
4.1.1	Objectifs	68
4.1.2	Caractéristiques	68
4.1.3	Domaines d'application	73
4.2	Description fonctionnelle	75
4.2.1	Modèle conceptuel	75
4.2.2	<i>Services de Base Persistants (SBP)</i>	76
4.2.3	Services collaboratifs	77
4.2.4	Protocole PISA	77
4.3	Développement du prototype AGORA	78
4.3.1	Chronologie	78
4.3.2	Gestion utilisateurs et groupes	79
4.3.3	Stade actuel du développement	80
4.4	Développement du service COLLABUREAU	80
4.4.1	Description formelle	80
4.4.2	Fonctionnement dynamique à sémaphore	83
4.5	Intégration	84

4.1 Cadre conceptuel

Comme nous l'avons vu au chapitre 1, l'*intelligence collective* émerge d'un système complexe composé d'agents qui interagissent selon un schéma de contextualisation *socio-constructiviste*. Nous avons ensuite caractérisé, au chapitre 2, le concept d'*espace collaboratif ubiquitaire* comme moyen de catalyser ces interactions. Enfin, la description de l'infrastructure (c.f chapitre 3) vient maintenant poser le cadre conceptuel du modèle AGORA .

4.1.1 Objectifs

Compte tenu des contraintes opérationnelles rencontrées (disponibilité des technologies, calendrier des expérimentations, etc.), nous avons délibérément choisi d'adopter une approche pragmatique afin de disposer rapidement d'un prototype prêt à être soumis à évaluation par des utilisateurs. Puisque notre but, *in fine*, est de mettre en évidence l'émergence d'une nouvelle forme d'*intelligence collective* par une conceptualisation technologique particulière, nous nous sommes focalisés sur trois objectifs.

Ces objectifs sont calqués sur les caractéristiques des *espaces collaboratifs ubiquitaires* décrits dans la section 2.4 :

1. offrir une **IHM multimodale** la plus riche possible, supportant des modes de communication classiques sous forme textuelle ou vocale, mais, surtout, permettant l'échange de représentations visuelles en s'appuyant sur des services synchrones ;
2. octroyer un maximum d'**autonomie décisionnelle** aux groupes pour qu'ils puissent gérer eux-mêmes leur ressources et se fixer leurs propres règles de sécurité ;
3. accroître l'**indépendance technologique** entre l'utilisateur et l'environnement dès qu'une collaboration doit se mettre en place.

L'association de ces trois objectifs distingue AGORA des modèles existants. Par exemple, la plupart de ces modèles sont contraints d'utiliser la ressource informatique au niveau du poste client pour conserver une mémoire de l'état des sessions. Il est courant de devoir installer des modules logiciels attachés à son navigateur ou de devoir autoriser l'accès en écriture sur son disque local (ex. autoriser les *cookies*¹). Nous avons déjà évoqué les nombreux inconvénients de cette approche (cf. section 2.3). Dans un contexte collaboratif, chaque groupe effectue un choix d'applications commun à l'ensemble de ses membres, mais variable selon les groupes. Cette situation peut très vite devenir insoutenable pour les utilisateurs membres de plusieurs groupes. En effet, ces utilisateurs se retrouvent à chaque fois dans l'obligation de se soumettre à la maîtrise d'un nouvel environnement technologique qu'ils n'ont pas choisi, et ceci pour autant de groupes dont ils sont membres, au lieu de consacrer leur temps à leur domaine de compétence.

4.1.2 Caractéristiques

Sur la base des conclusions du chapitre 2 Les caractéristiques suivantes nous aident à orienter les choix de conception du modèle AGORA . Celles-ci sont accompagnées de conditions à satisfaire *in fine* dans la solution que nous proposons.

1. Une interface **multimodale** renforce, d'une certaine manière, le couplage entre agents en interaction, ce qui permet d'accroître l'efficacité de la collaboration ;
2. L'**auto-organisation** des groupes d'agents peut être facilitée par une approche de modélisation systémique inspirée des *SMA*.

¹Les *cookies* sont utilisés par le protocole *HTTP* pour conserver, temporairement en mémoire, des données utilisateurs afin d'automatiser la connexion vers des serveurs distants.

3. L'**immanence**, comme faculté à circonscrire les décisions à l'intérieur même de l'espace collaboratif du groupe, est réalisée grâce à des services de base persistants instanciés dans le conteneur associé à chaque groupe.
4. L'**ubiquité**, comme faculté de conserver l'état des conversations indépendamment de la localisation des agents, est réalisée grâce aux services à état décrits par les normes *OGSA* et *WSRF*.
5. La **sécurité**, qui garantit la confidentialité, l'intégrité et l'authenticité des échanges, est réalisée grâce à l'infrastructure de sécurité décrite par *GSI*.
6. Le **passage à l'échelle** est facilité lors du déploiement des services avec l'allocation dynamique des ressources distribuées tel que décrit par *GRAM*.

Ces propositions sont le fruit de notre analyse de l'état de l'art technologique du chapitre 3. Les modèles issus de la communauté GRID comme *OGSA* et ceux issus des travaux de la communauté des *SMA* comme *Agent Groupe Rôle (AGR)* [Ferber *et al.* 2003] apportent chacun une solution partielle au problème d'organisation des groupes et des ressources. Nous proposons d'exploiter certains aspects de ces deux modèles pour bâtir une solution globale.

Nous avons reporté sur la figure 4.1 les différentes origines des contributions au modèle *AGORA*. Notre contribution adresse, d'une part, l'intégration technologique des modèles *OGSA* et *AGR* et, d'autre part, porte sur deux des six caractéristiques décrites précédemment, à savoir : la caractéristique de *multimodalité* et le caractère d'*immanence*.

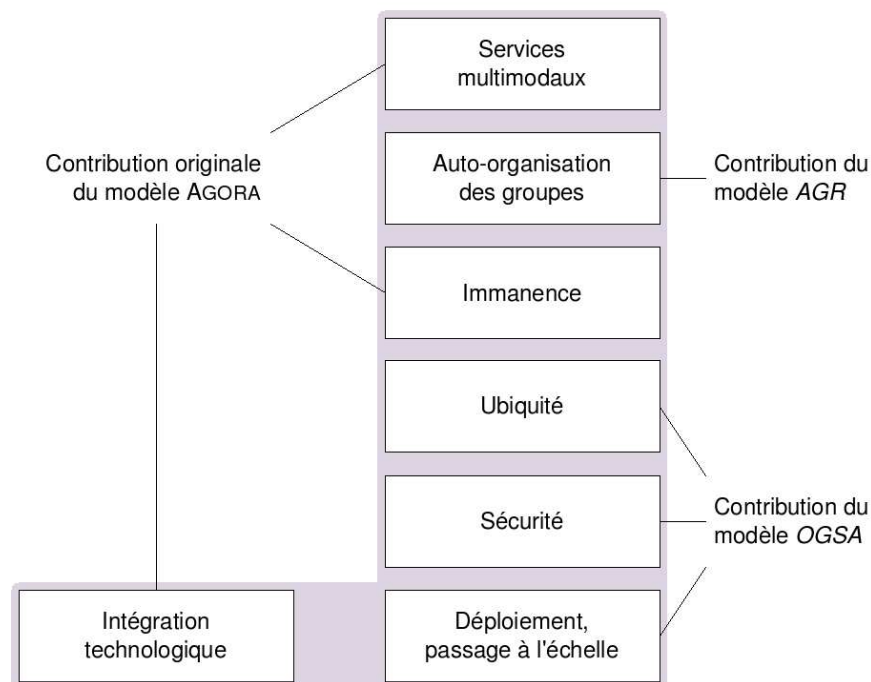


FIG. 4.1 – Contribution du modèle *AGORA*.

Caractéristique 1/6 : interface multimodale → renforcement du couplage

Une interface *multimodale* prévoit l'emploi de modes de communication, combinés (textuel, graphique, vocal, vidéo, *etc.*) pour des services *synchrones* et *asynchrones*. Cette caractéristique primordiale va permettre de renforcer le couplage entre agents en interaction. Le couplage décrit ici une capacité

potentielle de flux d'information entre les différents agents du système. Le renforcement du couplage entre agents permet d'accroître globalement l'efficacité du processus de collaboration². L'efficacité de ce processus va dépendre de deux facteurs :

1. du degré de pertinence des interactions (les interactions qui ne mènent à aucune nouvelle connaissance génèrent du bruit et nuisent par conséquent à l'efficacité du système) ;
2. de l'intensité des échanges (les interactions qui mènent à la construction de nouvelles connaissances doivent être renforcées dans le système).

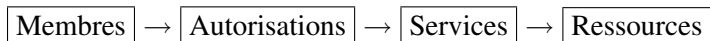
Ainsi, nous avons décidé de renforcer le couplage par une contextualisation visuelle synchrone au moyen d'écrans partagés imbriqués. Idéalement, cette imbrication pourrait être effectuée selon divers critères de pertinence identifiés de manière systématique. Nous nous en tiendrons, dans un premier temps, à considérer la formation d'un groupe comme critère de pertinence unique et accepté comme tel. Nous acceptons l'idée que tout agent est membre d'un groupe, parce qu'il a jugé pertinent d'en faire partie.

Le choix d'une interface *multimodale* a plusieurs conséquences sur la conception du modèle. Les protocoles de communication entre services reposent sur des principes très différents selon le niveau d'interaction : en synchrone, il est nécessaire de gérer les notifications de rendez-vous, l'état de présence des agents, ou encore l'ordonnancement dynamique des sessions graphiques, tandis qu'en asynchrone, il est nécessaire de gérer les accès concurrents aux contenus et le suivi des modifications opérées sur ces contenus.

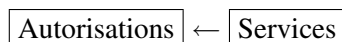
Caractéristique 2/6 : auto-organisation → un modèle systémique

Le comportement d'un groupe étant non-déterministe, un modèle systémique est plus approprié qu'un modèle analytique pour décrire un tel comportement et ainsi favoriser l'auto-organisation des groupes. AGORA cherche à se concentrer sur les interactions en minimisant l'effet des conditions initiales comme, par exemple, des règles pré-établies qui régieraient les modes d'organisation de tous les groupes.

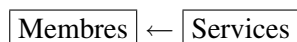
Pour y parvenir, la figure 4.2 décrit comment le modèle AGORA propose de réaliser la condition d'auto-organisation des groupes. Il s'agit d'un modèle systémique, c'est-à-dire un modèle à causalité circulaire comportant des liens directs et des boucles de rétroaction entre certains éléments du système. Les quatre blocs élémentaires, disposés sur cette figure, forment une instance d'*espace collaboratif ubiquitaire* pour un groupe G donné. G est constitué de [Membres], de règles d' [Autorisations]) d'accès à des instances de [Services]) du groupe et de [Ressources] diverses sous forme de documents, de processus ou de périphériques. Les membres de ce groupe ont la possibilité d'accéder aux ressources en mode *synchrone* ou *asynchrone*. Le chemin en pointillés indique que cet accès est indirect. Il dépend des droits spécifiés dans une matrice de permissions et de l'état des services.



Par exemple, dans cette matrice, un membre m_i de G dispose des privilèges $p(m_i, s_j)$ sur l'instance de service s_j . Une première boucle de rétroaction (*a*) indique que les instances de service peuvent interagir entre-elles selon des privilèges définis dans la matrice des autorisations.



Une seconde boucle de rétroaction (*b*) indique que les instances de service peuvent aussi interagir avec les membres par le biais de notifications principalement.



²le terme *awareness*, difficile à traduire, est couramment employé en TCAO pour décrire cette propriété.

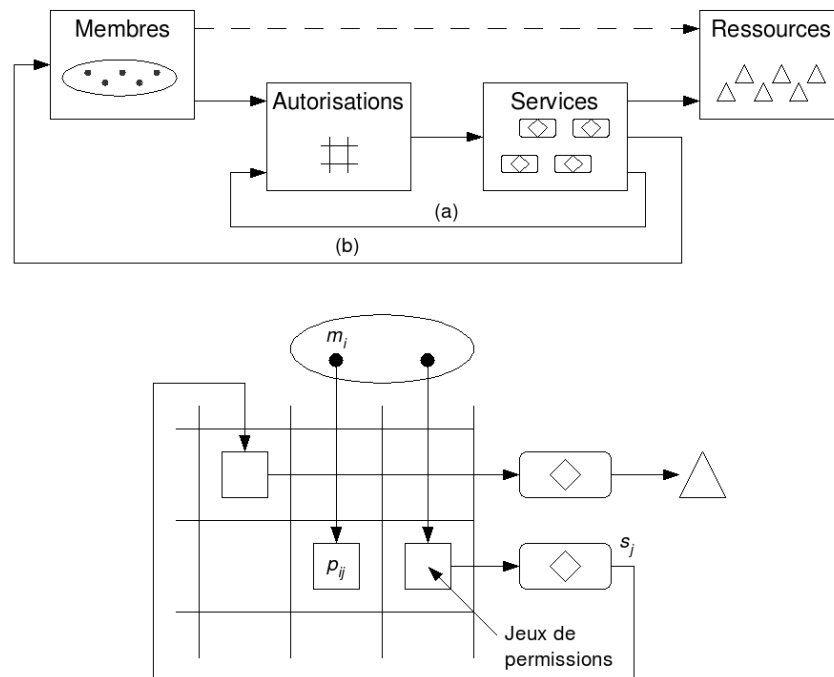


FIG. 4.2 – Un groupe vu comme un système (vue d'ensemble et vue détaillée).

Caractéristique 3/6 : immanence → *Services de Base Persistants (SBP)*

Comme nous l'avons vu dans la section 2.4.2, la condition d'*immanence* est satisfaite si, dès la création d'un groupe, toute décision influant sur le fonctionnement de ce groupe provient du groupe lui-même. Cette caractéristique est essentielle pour faciliter l'autonomie des groupes mais elle implique un minimum de services de base qui doivent être actifs en permanence dans le conteneur de services du groupe. En effet, lors de la création d'un groupe, un certain nombre d'opérations est nécessaire pour amorcer la constitution du groupe (ajout de membres au groupe, attribution des rôles au sein du groupe, etc.). Le conteneur de services associé au groupe comporte, dans sa configuration initiale, plusieurs instances de services persistants qui vont se charger de ces opérations, les *Services de Base Persistants (SBP)*

Ces services sont au nombre de six, comme le représente figure 4.3. Ce nombre est irréductible et ils sont mutuellement dépendants les uns des autres. L'annexe A (les fonctions) décrit de façon détaillée les étapes de constitution d'un groupe et les règles minimales pour permettre au groupe de fonctionner en toute autonomie.

Caractéristique 4/6 : ubiquité → **services à état**

Cette caractéristique est totalement satisfaite si les trois conditions suivantes sont réunies : (i) tout utilisateur peut accéder aux ressources des groupes dont il est membre, (ii) à partir de n'importe quel point d'accès, et ceci, (iii) sans le moindre effort de configuration préalable. Cette dernière condition est d'ordre théorique, car il est difficile d'imaginer en pratique un accès totalement exempt d'une configuration préalable. Toutefois, l'essentiel est que le modèle s'affranchisse de toute dépendance entre la plateforme cliente et les services.

Pour y parvenir, l'état des sessions de collaboration ne doit en aucun cas être géré au niveau du client. Le client doit être un client sans état, il n'a aucune mémoire des opérations réalisées. Il se charge

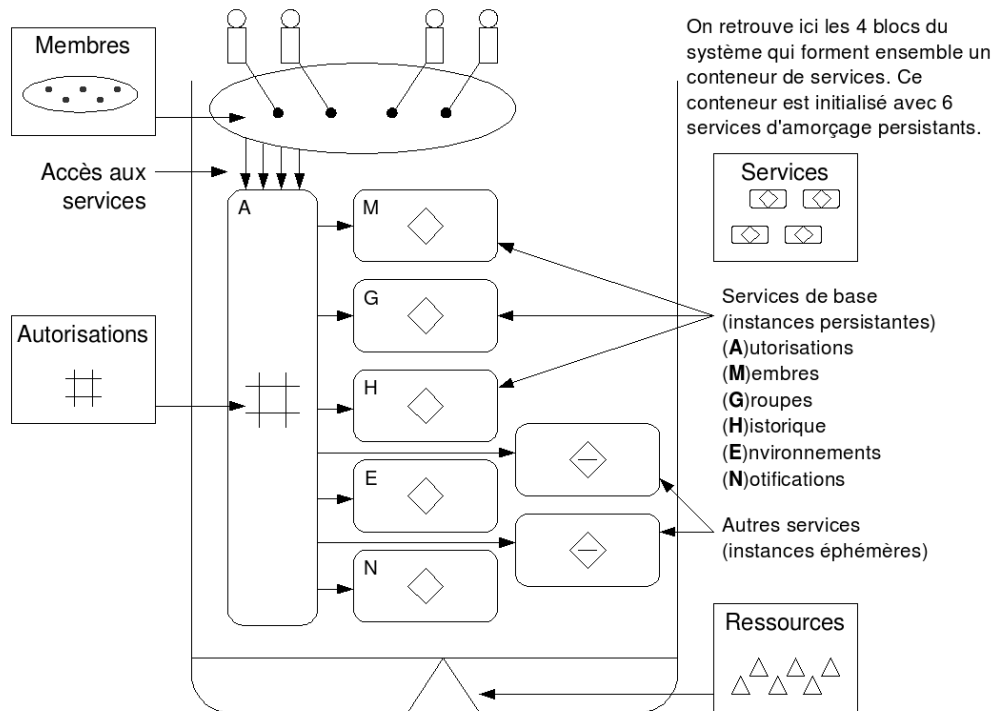


FIG. 4.3 – Les six *Services de Base Persistants (SBP)*

simplement d'échanger des flux d'information réalisant, tout au plus, des opérations d'adaptation de format ou de compression. AGORA prévoit de gérer l'état de la collaboration au niveau des instances de service distantes avec lesquelles le client communique. Comme ces services sont des services à état, ils peuvent ainsi traiter des conversations simultanées et mémoriser les événements au cours des différentes sessions de collaboration.

La figure 4.4 nous montre comment s'établit une communication entre services et le rôle du service à état pour l'ubiquité et la gestion des conversations. L'axe horizontal représente l'espace collaboratif des services S_n et l'axe vertical correspond au temps t . Puisque l'état des conversations est conservé au niveau du service, ces conversations peuvent s'enchaîner de façon logique avec transmission de connaissances d'un service à l'autre. L'ubiquité est réalisée par le fait que le porteur de connaissance est référencé dans ce système de coordonnées $\eta(S_n, t)$.

Caractéristique 5/6 : sécurité → intégrer GSI et améliorer CAS ou VOMS

La *sécurité* est une caractéristique qui peut être satisfaite à deux niveaux. Au niveau global, pour assurer dans tout le système confidentialité, intégrité et reconnaissance mutuelle comme cela est prévu avec l'infrastructure de sécurité GSI. Au niveau local, pour mettre en place des politiques de sécurité particulières pour les organisations virtuelles. AGORA vise à intégrer le modèle de sécurité GSI et à se consacrer à la sécurité au niveau local par des politiques de sécurité flexibles et indépendantes, en proposant une alternative à CAS ou VOMS.

Caractéristique 6/6 : faciliter le passage à l'échelle → intégrer l'approche GRAM

Le passage à l'échelle est une étape délicate qui implique prendre en considération, dès la conception, les aspects de liés aux mécanismes de déploiement des instances de services. Cette opération doit être effectuée en rapport avec le fonctionnement de la gestion des ressources et les contraintes de disponibilité

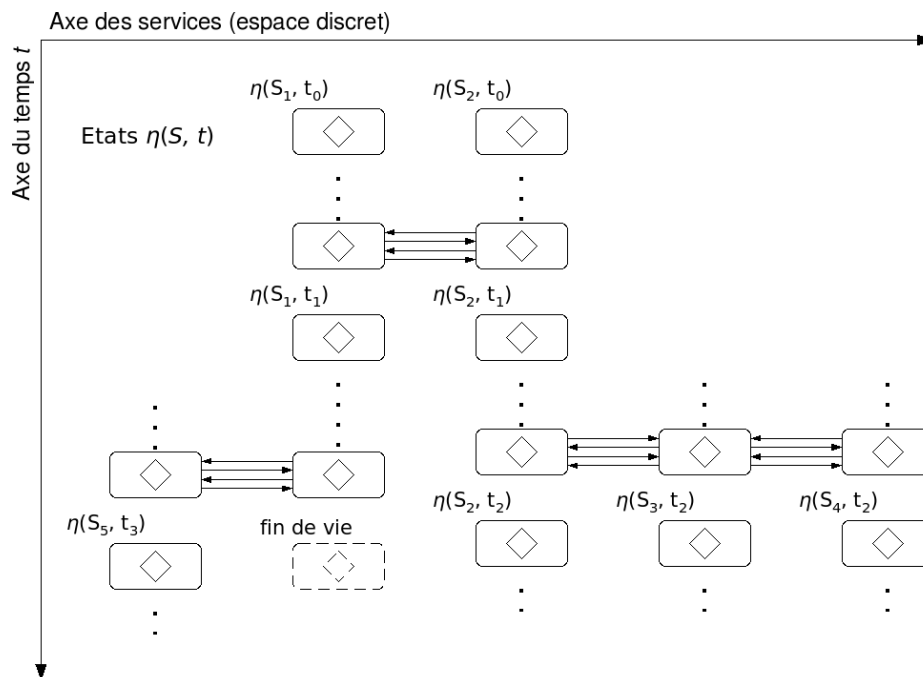


FIG. 4.4 – Rôle du service à état pour l'ubiquité.

des services. AGORA doit être conçu pour exploiter la ressource mise à disposition dans des conteneurs de services par un mécanisme de virtualisation tel que *GRAM*.

4.1.3 Domaines d'application

Le concept d'*espace collaboratif ubiquitaire* ouvre un champ d'application vaste, qu'il est intéressant d'exploiter pour le modèle AGORA. Dialoguer avec des chercheurs du monde entier simultanément, prendre en main les instruments d'une salle blanche depuis une salle de cours sont un échantillon d'exemples représentatifs. Plus généralement, nous avons identifié six domaines d'application :

1. Formation, aide à la maîtrise d'outils complexes
2. Mutualisation de compétences techniques réparties
3. Accès partagé à une instrumentation lourde et coûteuse
4. Visualisation simultanée de diagnostic dans le domaine médical
5. Discussion de résultats scientifiques

Domaine d'application 1 : Formation, aide à la maîtrise d'outils complexes

Des outils informatiques de plus en plus sophistiqués sont développés pour toutes disciplines confondues. Ceci entraîne une demande croissante d'assistance à la prise en main de ces outils. Habituellement les éditeurs de logiciels organisent des sessions de formation qui mobilisent de nombreux intervenants. Ces sessions obligent les intervenants à dispenser en blocs leurs modules de formation pour minimiser les déplacements. Cette méthode s'avère incompatible avec les théories cognitives qui préconisent un apprentissage graduel entrecoupé de périodes de pratique.

AGORA permet de diffuser en continu (direct ou différé) des sessions de formation ou d'assistance spécialisée sans contraindre les intervenants à se déplacer.

Domaine d'application 2 : Mutualisation de compétences techniques réparties

Ingénieurs et techniciens ont quotidiennement un lot de tâches spécialisées à accomplir dans le cadre de leur mission de conception. Le plus souvent ces tâches correspondent à leur domaine de compétence mais ce n'est pas toujours le cas. Ainsi, plusieurs options s'offrent à eux : soit ils tentent d'acquérir eux-mêmes cette nouvelle compétence, soit ils la recherchent ailleurs. C'est la mesure du coût mais aussi du délai de réalisation qui va infléchir sur l'une ou l'autre solution. Quoiqu'il en soit ce délai a une valeur irréductible de l'ordre de quelques jours.

Ainsi, AGORA permet réduire de considérablement les délais d'ingénierie en privilégiant la mise en commun de compétences face à un problème technique complexe. Chacun d'eux obtient une réponse immédiate à la résolution d'un problème ponctuel sans aucune barrière d'espace et de temps. Ainsi, la durée globale d'un projet se retrouve diminuée de l'ensemble des délais d'inertie entre chaque tâche pour mobiliser une compétence particulière.

Plus généralement dans un cadre inter-disciplinaire, AGORA contribue à minimiser les délais intercalaires entre les différentes phases de développement d'un projet.

Domaine d'application 3 : Accès partagé à une instrumentation lourde et coûteuse

La recherche scientifique et l'industrie de pointe disposent souvent d'instruments dispendieux (ex. microscope à effet tunnel, instruments de métrologie à l'échelle nanométrique) centralisés sur un site mais servant à une population distribuée géographiquement : d'une part, les déplacements pour venir utiliser ces instruments ne sont pas toujours compatibles avec les emplois du temps des scientifiques, et, d'autre part, l'accès aux instruments n'est pas toujours possible physiquement.

AGORA permet d'accéder à des ressources distantes, lourdes, coûteuses ou confinées dans des environnements protégés (salles blanches, etc.)

Domaine d'application 4 : Visualisation simultanée de diagnostic dans le domaine médical

Le télédiagnostic médical est une voie prometteuse pour réagir sur des situations urgentes. Toutefois, les solutions informatisées sont peu conviviales (élément portatif lourd et complexe à utiliser) et nuisent à l'efficacité du travail du médecin.

AGORA permet de déployer un service de visualisation instantané avec des terminaux légers entre plusieurs services médicaux.

Domaine d'application 5 : Discussion de résultats scientifiques

Une activité importante en sciences dures empirico-formelles, en particulier en physique, est la modélisation des phénomènes en vue d'expliquer une théorie. Ces modélisations aboutissent à des représentations visuelles qui ont parfois demandé de nombreux efforts de paramétrage de la part des scientifiques et de lourds traitements de simulation informatique. L'interprétation *a posteriori* de ces résultats doit de nouveau faire appel à l'expérience des scientifiques pour confirmer ou infirmer le bien fondé de l'expérimentation (hypothèses de départ, choix des paramètres, approximations, etc.).

Il en va de même pour les sciences herméneutiques (sociologie, art, histoire, droit, etc.) qui sont fondées sur l'interprétation des écrits ou œuvres picturales et leur mise en perspective par rapport à leur contexte (subjectivité de l'auteur, période historique). Ces études demandent aux chercheurs de se retrouver pour confronter leurs points de vue à divers stades de leur recherche.

AGORA permet aux scientifiques localisés aux quatre coins du monde de discuter en temps réel avec des représentations visuelles de leurs résultats, d'exposer leurs points de vue respectifs en diffusant les documents appropriés, et, d'archiver et de retracer l'historique de leurs travaux (individuels et collectifs) en vue d'une consultation périodique.

4.2 Description fonctionnelle

L'architecture du modèle AGORA comporte deux entités fonctionnelles :

- Une entité **organisationnelle** en charge des aspects statiques que nous décrivons par un *modèle conceptuel*, six *Services de Base Persistants (SBP)* et un nombre indéfini de *services collaboratifs*, en particulier le *COLLABUREAU* pour le partage de sessions graphiques en mode synchrone ;
- Une entité **conversationnelle** en charge des aspects dynamiques que nous décrivons par le *Protocole d'Interaction entre Services et Agents (PISA)*.

4.2.1 Modèle conceptuel

La figure 4.5 représente le modèle conceptuel AGORA. Les zones repérées en pointillés indiquent la partie inspirée du modèle AGR à gauche et la partie inspirée du modèle OGSA à droite. La relation ternaire qui lie les concepts *Agent*, *Groupe* et *Service* est une abstraction du concept *Rôle* du modèle AGR. Dans cette relation, le concept de *Rôle* correspond à une relation d'autorisation entre le membre d'un groupe et un service. Le concept d'*Autorisation* présente un niveau de granularité plus fin par rapport à un service que le concept de *Rôle*. Un rôle étant défini non pas vis-à-vis d'un service en particulier mais de façon générale au sein d'un groupe. Le concept d'autorisation correspond ici mieux à l'idée d'assignation de droits sur des services. Un rôle est, par conséquent, vu comme un ensemble de droits sur un ensemble de services, ce qui se traduit par une façon d'autoriser ou non un membre d'un groupe à faire quelque chose.

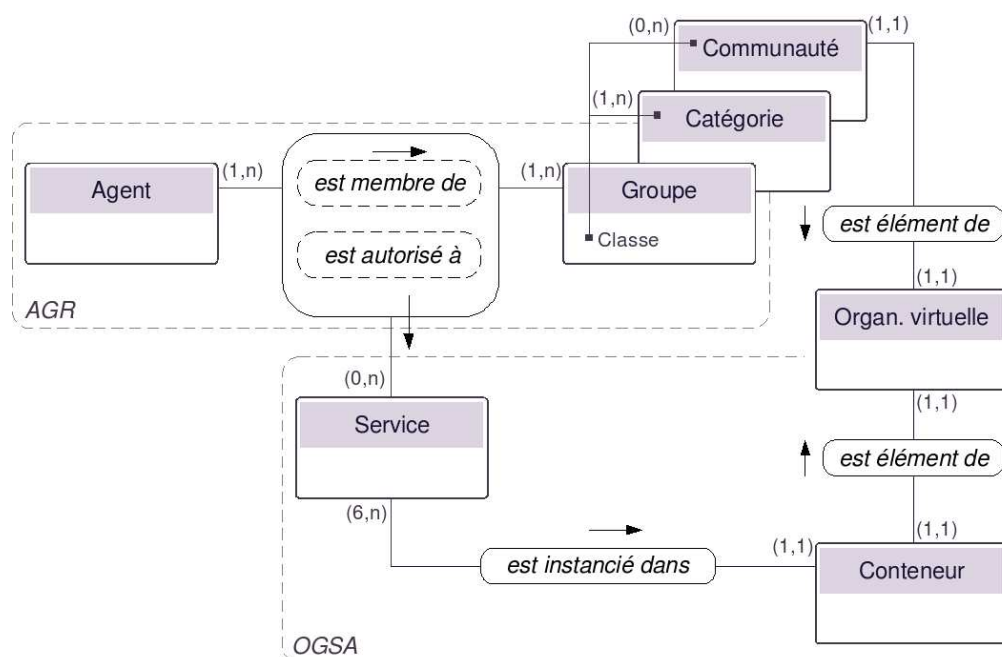


FIG. 4.5 – Modèle conceptuel AGORA

Par ailleurs, un groupe peut se décliner en deux classes :

- une classe qui correspond aux **catégories** d'agents et
- une classe qui correspond aux **communautés** d'agents.

Cette distinction est motivée par deux raisons. La première est due à l'obligation de spécifier au moins un groupe d'appartenance initiale lors de la création des identités d'un nouvel utilisateur. Ce

groupe initial n'a aucune raison d'être une communauté (laquelle choisir par défaut ?) mais plutôt une catégorie qui peut éventuellement regrouper des agents de façon plus ou moins générale. Par exemple le groupe des invités doit être vu comme une catégorie et non pas comme une communauté constituée en vue d'une collaboration. Il est envisageable d'avoir des catégories par compétence (ingénieur, chercheur, etc.) ou par statut (fonctionnaire, etc.).

La seconde raison est d'ordre ontologique. Catégories et communautés d'agents sont des spécialisations différentes du concept de groupe à plusieurs titres.

- Au niveau des cardinalités, tous les utilisateurs doivent appartenir au moins à une catégorie mais ils peuvent n'appartenir à aucune communauté. Pour des raisons de simplification nous avons restreint la cardinalité de la relation des utilisateurs à une seule catégorie possible, quitte à la rendre très générale.
- La relation de membre est une relation fondamentale entre un utilisateur et une communauté mais n'a pas de raison d'être dans le cas d'une catégorie.
- Pour collaborer, une communauté d'utilisateurs a besoin de ressources. A contrario une catégorie n'a pas besoin de ressources, ce n'est qu'un rattachement logique d'un utilisateur à un groupe.
- La quatrième distinction entre communauté et catégorie est qu'une catégorie ne peut être supprimée tant qu'elle n'est pas vide.

Une catégorie peut être vide tant qu'aucun utilisateur n'a été assigné à cette catégorie, ou s'il est prévu la possibilité de transférer un utilisateur d'une catégorie à une autre. La suppression d'une communauté d'utilisateurs peut être décidé selon des règles internes à la gestion du groupe. La suppression d'une communauté d'utilisateurs n'a pas d'incidence sur l'intégrité de la relation utilisateurs-groupes puisque les utilisateurs restent attachés à leur catégorie.

D'autres approches comme *VOMS* se sont heurtés à cette question, et ont apporté une réponse qui tend à restreindre la capacité d'auto-organisation des groupes. En effet, le groupe est vu comme un sous ensemble d'une organisation virtuelle et chaque membre appartient à un groupe principal. C'est une solution classique de gestion des entités groupes-utilisateurs mais qui présente de nombreux inconvénients (ex. qu'advient-il des groupes dépendant d'un organisation virtuelle lorsque celle-ci est détruite).

4.2.2 Services de Base Persistants (SBP)

Un *SBP* est, comme son nom l'indique, un service *persistant* car il est instancié lors de l'initialisation du *conteneur* de services rattaché à un groupe qui forment ensemble une *organisation virtuelle* (cf sections 3.2.2 et 3.2.3). La durée de vie des *SBP* est indéfinie, elle perdure aussi longtemps que la durée de vie du conteneur de services. Ces *SBP*, au nombre de six, indispensables à la gestion des groupes, se décrivent ainsi :

1. **Autorisations** : Les membres d'un groupe n'ont pas tous les mêmes niveaux de permissions sur les services du groupe. Ce service assigne des droits aux membres, y compris les permissions d'utiliser ou non les services de base.
2. **Membres** : Un groupe est composé de membres. Ce service gère la description des membres d'un groupe donné ainsi que l'ajout de nouveaux membres ou la radiation de membres à ce groupe.
3. **Groupes** : Un groupe est caractérisé par ses propriétés. Outre son identifiant, ce sont essentiellement des propriétés descriptives. Par ailleurs, toute opération doit être effectuée dans le contexte d'un groupe comme, par exemple, fonder un nouveau groupe. Ainsi ce service distingue deux types d'opérations : (i) les *endops* ou opérations *intra-groupe* qui permettent d'agir sur les propriétés d'un groupe, dans le contexte de ce même groupe (ex. modification d'informations descriptives du groupe) et (ii) les *exops* ou opérations *extra-groupe* qui permettent d'agir à l'extérieur d'un groupe donné. Par exemple, la création ou la destruction d'un autre groupe sont des *exops*.

4. **Historique** : Les données d'un groupe doivent être sauvegardées et maintenues. Ce service garde la trace des résultats de la collaboration. Il peut s'agir de documents, de la journalisation d'événements ou encore de l'enregistrement de sessions de collaboration.
5. **Environnement** : Un groupe dispose d'un environnement matérialisé par un *conteneur* de services. Les instances de services de ce groupe disposent à leur tour d'un contexte d'exécution délivré par de la ressource de ce *conteneur*. Ce service permet de créer ou de supprimer des instances dans ce *conteneur*.
6. **Notifications** : La communication entre les membres d'un groupe et les services est réalisée sous forme de notifications. Ce service traite le flux des notifications et l'état des messages échangés.

Par la suite, la notation simplifiée **A**, **M**, **G**, **H**, **E**, **N** pourra être aussi utilisée.

4.2.3 Services collaboratifs

Les services collaboratifs d'un groupe sont également instanciés dans le conteneur de services du groupe. Ceux-ci permettent aux membres d'effectuer des sessions collaboratives en utilisant un seul service ou une composition de ces services. Nous avons développé trois services collaboratifs génériques afin de disposer d'une plateforme d'évaluation minimale du modèle AGORA en effectuant une composition de ces services. Il s'agit avant tout d'un service de bureaux partagés :

- le COLLABUREAU qui distribue l'affichage de bureaux virtuels selon l'organisation des groupes. Ce service est complété par deux autres services collaboratifs auxiliaires :
- DEVISU , un service de communication audio-vidéo en full-duplex sur le protocole *Real Time Media Protocol (RTMP)* et
- TRANSFILE , un service *Common Gateway Interface (CGI)* de transfert de fichiers sur *http* développé en langage *perl*.

Le développement du COLLABUREAU est détaillé dans la section 4.4.

Par ailleurs, chaque groupe peut, grâce au service **E**, importer de nouveaux services collaboratifs. Cette opération n'est pas encore possible de façon dynamique. Nous avons toutefois prévu dans le cadre des expérimentations, un moyen statique pour importer des services collaboratifs spécifiques (Protégé, Resyn Assistant *c.f* chapitre 5).

4.2.4 Protocole PISA

Le *Protocole d'Interaction entre Services et Agents (PISA)* décrit un certain nombre de mécanismes d'interaction permettant aux agents d'effectuer des opérations élémentaires sur les services de base. Par exemple, l'ajout d'un membre à un groupe, la modification des droits d'un membre au sein d'un groupe ou encore la création d'un groupe sont décrits comme des procédures de PISA . Chaque service de base exporte à travers son interface un certain nombre de fonctions propres. Une procédure PISA est une composition d'appels de fonctions des différents services de base. Pour chacune de ces procédures, un certain nombre d'opérations est réalisé, comme, par exemple, la demande de passer tels ou tels paramètres, la vérification de ces paramètres et l'indication que l'opération a réussi ou échoué. Les procédures définies dans PISA sont les suivantes :

- Consulter la liste des groupes
- Créer un nouveau groupe (*)
- Changer d'alias au sein d'un groupe
- Changer de mot de passe
- Solliciter un groupe pour en devenir membre (*)
- Ajouter un membre par cooptation
- Ajouter un membre suite à une sollicitation
- Radier un membre

- Solliciter des droits sur un service
- Assigner des droits à un membre sur un service (*)
- Oter les droits d'un membre sur un service

Le diagramme 4.6 présente, à titre indicatif, le déroulement des échanges pour la procédure d'ajout d'un membre par cooptation. La cooptation signifie ici que le membre M_i du groupe G_i est initiateur de l'ajout d'un nouveau membre cible M_c . Pour ce faire M_i doit disposer des droits suffisants sur le service **Membres**. Les procédures suivi d'une astérisque sont décrites à l'annexe B. Dans la version du prototype actuel, PISA est vu par les utilisateurs comme un méta service du nom d'AGORA Services.

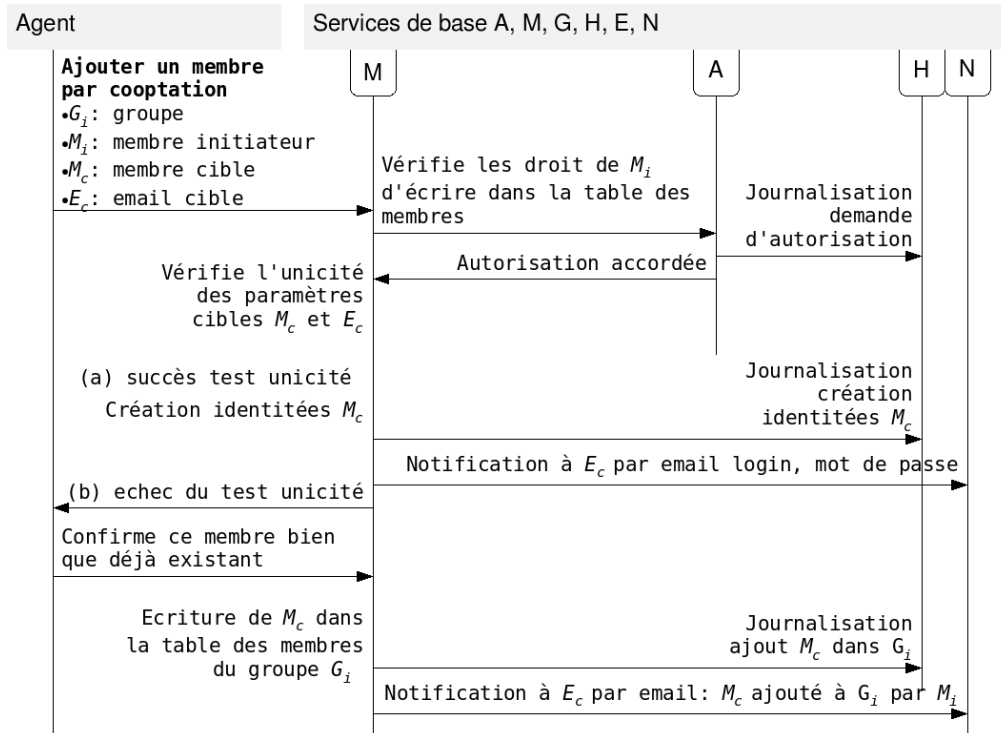


FIG. 4.6 – Procédure PISA , ajout d'un membre par cooptation

4.3 Développement du prototype AGORA

Les choix technologiques adoptés pour le développement du prototype AGORA ont été influencés par deux facteurs antinomiques qui nous ont incité à faire preuve de pragmatisme. Le premier facteur est lié à la disponibilité des briques logicielles. Celles-ci ne sont pas toujours disponibles en temps voulu. Le second facteur est la nécessité de réaliser un grand nombre d'expérimentations avec des utilisateurs. Comme nous avons préféré mettre l'effort sur les expérimentations, le choix a plus souvent porté sur le critère de stabilité des briques logicielles plutôt que sur leur potentiel innovant. Néanmoins, nous avons continuellement surveillé de près les avancées technologiques en cours.

4.3.1 Chronologie

La figure 4.7 présente une chronologie récapitulative en indiquant les étapes clés de ce développement et les possibles orientations que nous envisageons pour la suite.

Dès octobre 2004, l'idée initiale était de développer un service GRID compatible OGSA. D'un autre côté la norme WSRF, qui venait tout juste d'être validée, allait fortement influencer la conception des

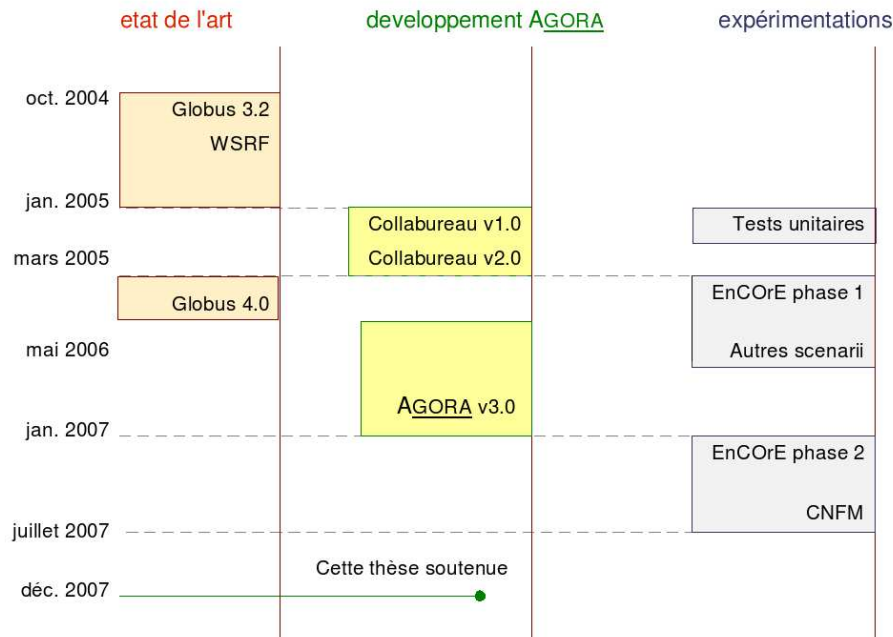


FIG. 4.7 – Chronologie récapitulative du développement

services *OGSA*. Après avoir installé une configuration *Globus Toolkit version 3 (GT3)* sur un serveur dédié à devenir le nœud GRID principal, les premiers tests ont pu commencer dès janvier 2005. Cette période a permis de bien comprendre les concepts GRID, notamment les aspects liés à la *sécurité* et le concept clé de *conteneur* de services. La version *GT3* n'étant pas compatible *WSRF*, il a fallu prendre une décision importante. En attendant une version *GT4* compatible *WSRF*, nous avons engagé l'effort de développement sur la partie fonctionnelle de la première version du *Collabureau* à des fins de tests unitaires. De ce fait, les premiers tests avec les utilisateurs ont pu commencer sur une version 2 du *Collabureau* dès le mois de mars 2005.

Au vu des résultats encourageants de ces premiers tests, nous avons décidé de poursuivre notre effort sur l'amélioration du prototype en augmentant l'autonomie des groupes et en incluant la gestion dynamique des ressources. Les utilisateurs ont été séduits par la simplicité d'accès à l'environnement combinée avec le moyen de collaboration en mode synchrone. La complexité des outils existants ne leur permettait pas de se mettre en situation de collaboration synchrone, ce qui a déclenché un intérêt pour le *Collabureau* bien au-delà de nos espérances. C'est sur cette base que nous avons développé, durant le second semestre 2006, la version 3 du prototype rebaptisé *AGORA* qui étend le concept de bureau virtuel à un *espace collaboratif ubiquitaire* complet sur la base des principes GRID, notamment l'allocation dynamique de la ressource et le service à état.

4.3.2 Gestion utilisateurs et groupes

Certains choix dans la version 2 du prototype ont été influencés au niveau du projet *ELeGI*. Ainsi, la gestion des utilisateurs et des groupes a été initialement réalisée par un annuaire *LDAP* [*IETFurl*]. Toutefois, de par sa structure en arbre, *LDAP* n'est pas adapté pour gérer une relation ternaire entre *agents*, *groupes* et les *autorisations*. Pour résoudre ce problème nous avons opté pour une gestion des utilisateurs avec une base de données relationnelles *MySQL*.

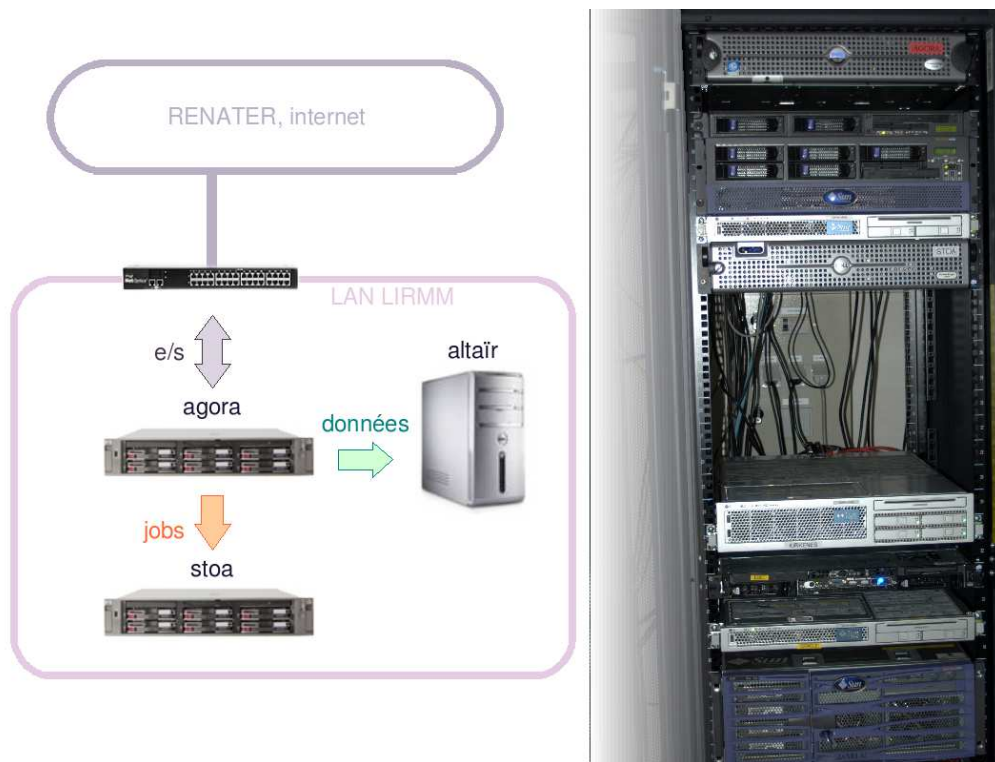


FIG. 4.8 – Plateforme AGORA

4.3.3 Stade actuel du développement

Actuellement, la plateforme matérielle (voir la figure 4.8) est constituée d'un serveur frontal (*agora*) qui communique avec l'extérieur par le réseau Renater et avec une grappe de serveurs localisés sur une plage d'adresses fixes sur le même routeur dans la salle informatique du Lirmm. Le serveur *stoa* soulage la charge de traitement et *altaïr* sert de serveur de sauvegarde.

La version stable la plus récente du prototype est une implémentation partielle du modèle AGORA. Toutefois, l'essentiel des fonctions des services de base et les procédures du protocole PISA nécessaires pour rendre ce prototype opérationnel sont implémentées. Les fonctions pas encore développées sont d'un niveau relativement contingent. La table 4.1 présente un récapitulatif de l'avancement global de ce développement (AGORA version 3.2.3).

4.4 Développement du service COLLABUREAU

Le COLLABUREAU³ est le service collaboratif principal du modèle AGORA. Ce service distribue l'affichage de bureaux virtuels selon l'organisation des groupes. Ainsi, chacun peut visualiser les bureaux partagés des autres membres des groupes auxquels il appartient.

Un bureau se définit comme une instance d'environnement graphique qui permet d'accéder à des services. Ces services font partie de l'*espace collaboratif* d'un groupe donné.

4.4.1 Description formelle

Supposons qu'un physicien fasse partie d'une communauté de chercheurs en physique optique. L'*espace collaboratif* de ce groupe pourra, par exemple, être constitué d'outils de représentation de

³ *Grid Shared Desktop (GSD)* est le nom d'usage dans les publications en anglais.

TAB. 4.1 – Récapitulatif de l'avancement du développement

Service	Fonctionnalités intégrées à AGORA v.3.2.3
AGORA	Installation et désinstallation Supervision du système Suite de tests de déploiement
Autorisation	Création des identités utilisateurs Suppression d'identités Assignment d'un rôles
Membres	Ajout de membres Suppression de membres Modification d'alias
Groupes	Création d'un groupe Suppression d'un groupe Modification des attributs d'un groupe Création de conteneurs de services Destruction de conteneurs de services et libération de la ressource
Historique	Journalisation des événements Synthèse des événements Archivage des événements Système de fichiers partagé
Environnement	Instanciation dynamique des services collaboratifs Destruction d'instances de services et libération de la ressource
PISA	Procédure d'ajout d'un membre par cooptation Procédure d'ajout d'un membre par sollicitation Procédure d'ajout de création d'un groupe
Service	Fonctionnalités planifiées pour des versions ultérieures
Autorisations	Définition de nouveaux rôles Suppression de rôles
Membres	-
Groupes	-
Historique	Enregistrement des sessions Recherche dans les journaux d'événements
Environnement	Import de nouveaux services en cours de session
PISA	Procédure d'assignation d'un rôle

modèles de propagation lumineuse dans des corps cristallins. Ce même physicien peut aussi faire partie d'un autre groupe comme une association de protection de la nature pour qui les outils de propagation lumineuse n'ont pas d'intérêt. Le physicien aura par conséquent accès à deux environnements distincts, un pour chaque groupe contenant des outils différents. Les autres membres de ces groupes (moyennant le fait qu'ils ne sont pas également membres de ces deux groupes) n'ont aucune raison de visualiser le bureau de l'autre groupe de ce physicien.

Plus généralement, cette description se formalise par la théorie des ensembles de la façon suivante.

Soient :

- A un ensemble de référence composé d'une population de I agents $(a_1, \dots, a_i, \dots, a_I)$;
- G l'ensemble de parties de A , correspondant à des communautés d'agents $(g_1, \dots, g_i, \dots, g_J)$;

Soit :

- \mathcal{M} la relation binaire de membre qui lie l'ensemble des agents à l'ensemble des communautés d'agents. Autrement dit, \mathcal{M} est une relation sur $A \times G$ définie par : $\forall a \in A$ et $\forall g \in G$, $a\mathcal{M}g$ si $a \in g$

Le graphe bipartite de la relation \mathcal{M} est l'ensemble des représentants des couples (a, g) tels que $a\mathcal{M}g$.

$$\text{membre} : A \times G \longrightarrow \mathcal{M}$$

Chacune des relations de \mathcal{M} est un élément m d'un ensemble M appelé ensemble de tous les *membres*.

Soit :

- M le graphe bipartite de la relation \mathcal{M} représentés par les couples $(a, g) \in A \times G$ tel que $a\mathcal{M}g$.

Les bureaux partagés sont représentés par des onglets chez les différents membres. Chaque onglet peut se formaliser par une relation binaire de M vers M . Soit donc :

- \mathcal{H} une relation binaire de $M \times M$ définie de la manière suivante : pour tous $m_x, m_y \in M^2$ avec $m_x = (a_x, g_x)$ et $m_y = (a_y, g_y)$, $m_x\mathcal{H}m_y$ si $a_x = a_y$ ou $g_x = g_y$.

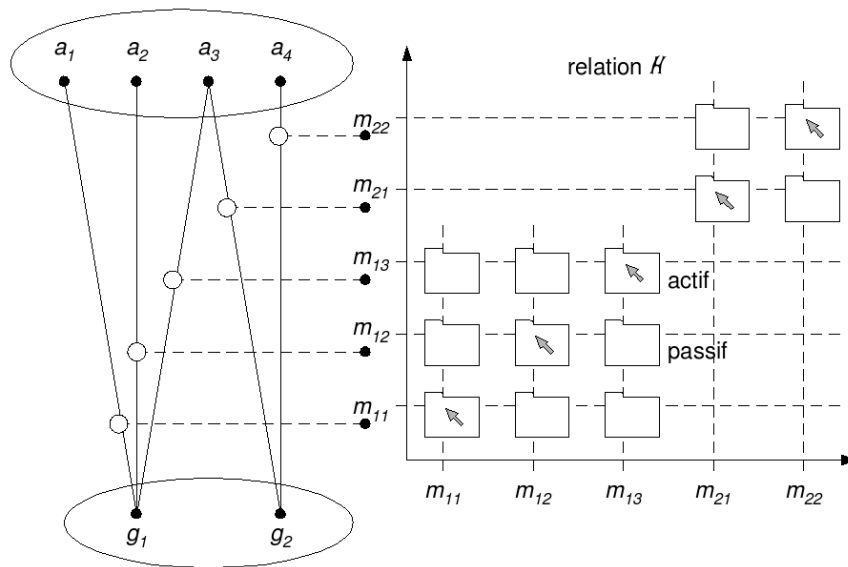
$$\text{onglet} : M \times M \longrightarrow \mathcal{H}$$

A travers ce formalisme, l'onglet se définit ici comme un objet représentant le bureau du membre m_x qui apparaît sur le bureau de tous les membres m_y lorsque a_x est connecté. \mathcal{H} a les propriétés suivantes :

- Réflexive $\forall m_x, m_x \in M$ est en relation \mathcal{H} avec m_x
- Symétrique $\forall (m_x, m_y) \in M^2$, si $m_x\mathcal{H}m_y$ alors $m_y\mathcal{H}m_x$

La propriété de réflexivité indique qu'un agent dispose d'un bureau pour chaque communauté dont il est membre. La propriété de symétrie fait apparaître un risque de duplication des événements si deux agents membres d'un même groupe se détectent simultanément. En effet, les membres connectés sont listés dans une tables de présence. Si la lecture de cette table est effectuée deux fois avant d'avoir été mise à jour, un événement peut causer l'apparition de deux onglets dupliqués chez un agent. Pour pallier à ce risque, un sémaphore est utilisé comme décrit dans la section suivante. Par ailleurs, si $m_x(a_x, g_x)\mathcal{H}m_y(a_y, g_y)$ et $m_y(a_y, g_y)\mathcal{H}m_z(a_z, g_z)$, aucune relation n'est obligatoirement vérifiée entre m_x et m_z . Par exemple, les égalités vraies pourraient ne porter uniquement que sur $a_x = a_y$ et $g_y = g_z$, ce qui, dans ce cas, implique que l'assertion $a_x = a_z$ ou $g_x = g_z$ est fausse. \mathcal{H} n'est donc pas transitive. Ceci explique, dans l'exemple avec le physicien, pourquoi aucun membre d'un groupe ne peut visualiser les onglets des autres groupes s'il n'en fait pas partie.

La figure 4.9 montre comment, de cette façon, les membres d'un groupe peuvent partager un espace de visualisation où chacun dispose d'onglets correspondants à la relation \mathcal{H} . Deux types d'onglets sont représentés. Les onglets de type *passifs* correspondent à une simple visualisation de bureau partagé. Les onglets de type *actifs* correspondent à une visualisation du bureau partagé avec la possibilité d'interaction sur ce bureau.

FIG. 4.9 – La relation \mathcal{H}

4.4.2 Fonctionnement dynamique à sémaphore

Lorsqu'un agent se connecte, il déclenche l'envoi d'une séquence d'événements au système. Le premier événement indique la présence de cet agent à tous les groupes dont il est membre. Dans un deuxième temps deux types d'événements vont se produire consécutivement à la propriété de symétrie de la relation \mathcal{H} . L'un orienté dans le sens du membre vers le groupe et l'autre dans le sens contraire. Le premier est une scrutation de la table des membres déjà présents pour récupérer l'affichage de leur bureau partagé. Le second est un envoi diffusé du bureau partagé de l'agent à tous les membres présents. Ce processus comporte un risque d'accès concurrentiel si ces événements ne sont pas ordonnancés correctement. En effet, lorsqu'un membre va lire la table de présence, les deux types d'événements générés risquent de se retrouver en concurrence avec les événements déclenchés par un autre membre si celui-ci se connecte à un instant relativement proche (inférieur à la durée des processus d'affichage de l'ordre de quelques secondes). Ce problème ayant été rencontré à plusieurs reprises, il a été décidé de développer une fonction sémaphore afin de maîtriser le séquençage des événements (voir figure 4.10).

Le sémaphore est intégré à la version 3 du COLLABUREAU. Dans les versions antérieures, les sessions n'étaient pas dynamiques. Le lancement des sessions de façon statique ne présentait aucun risque d'accès concurrentiel car le séquençage des opérations était maîtrisé. Toutefois les sessions statiques ne sont pas adaptées pour un service déployable à grande échelle. En effet, lorsqu'un utilisateur se connecte, la séquence d'événements suivante va se déclencher :

- parcours de la liste des communautés dont fait partie l'utilisateur,
- lecture de la table de présence de tous les groupes pour déterminer les membres qui sont déjà connectés,
- envoi de l'événement d'affichage du bureau partagé à tous les membres,
- réception de tous les bureaux des membres.

Nous voyons ainsi qu'il existe un risque pour que des utilisateurs qui se connectent à des instants voisins risquent de voir certains affichages absents et d'autres dupliqués. De manière générale, l'accès

aux ressources s'effectue en mode asynchrone. Le COLLABUREAU doit être capable de maîtriser le séquençement des processus lorsque ceux-ci sont déclenchés par des événements indépendants qui vont à leur tour générer de nouveaux événements. Dans certains cas, des événements interdépendants doivent être sérialisés. Le sémaphore se présente comme une file d'attente avec un ordre d'arrivée et d'un canal de traitement à une voie. Le premier événement arrivé, la tâche correspondante passe à travers la file d'attente et va immédiatement occuper le canal de traitement. Cette opération correspond à la prise de sémaphore. Tant que le sémaphore est levé aucune autre tâche ne peut accéder au canal. Alors, les tâches viennent se placer dans la file d'attente selon leur ordre d'arrivée. En fin de traitement, la tâche qui occupe le canal de traitement libère le canal et le sémaphore est lâché. La prise de sémaphore est réalisée par la première tâche de la file d'attente. Nous avons ajouté une temporisation pour s'assurer que tout sémaphore sera relâché dans un temps fini. Ainsi, lorsque le canal est occupé pendant une durée excessivement longue (durée paramétrable), le sémaphore est forcé de se relâcher et le canal est préempté par la première tâche de la liste d'attente.

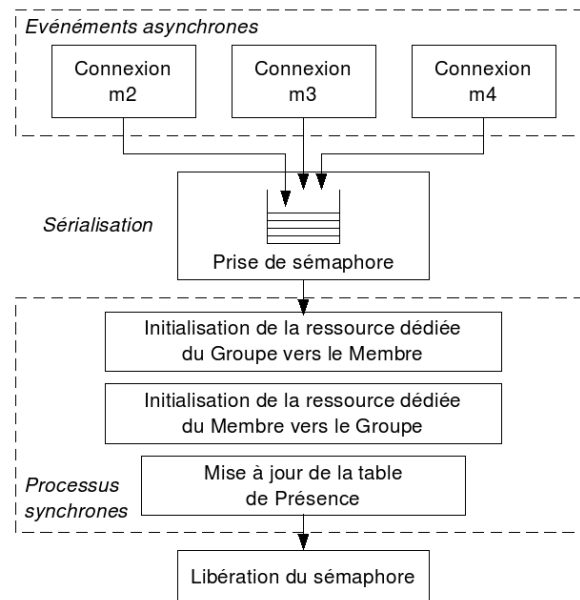


FIG. 4.10 – Un sémaphore pour sérialiser des événements asynchrones

Une remarque qui intervient à ce stade, explique la distinction entre groupe de différentes classes évoquées dans le modèle conceptuel d' AGORA . Les groupes de type *catégorie d'agents* ne donnent pas accès à de nouveaux onglets mais l'une de ces catégories (la catégorie principale) correspond au contexte de groupe dans le bureau privé de l'agent. Les groupes de type *communauté d'agents*, quant à eux, donnent accès à des onglets.

4.5 Intégration

L'intégration technologique a été l'activité la plus exigeante en temps au cours de ces trois années. En effet, les objectifs conceptuels font appel à des technologies qui ne sont pas naturellement employées

ensemble. Pour la réalisation, nous avons exploré diverses technologies sous licences libres⁴, afin de pouvoir les adapter à notre besoin.

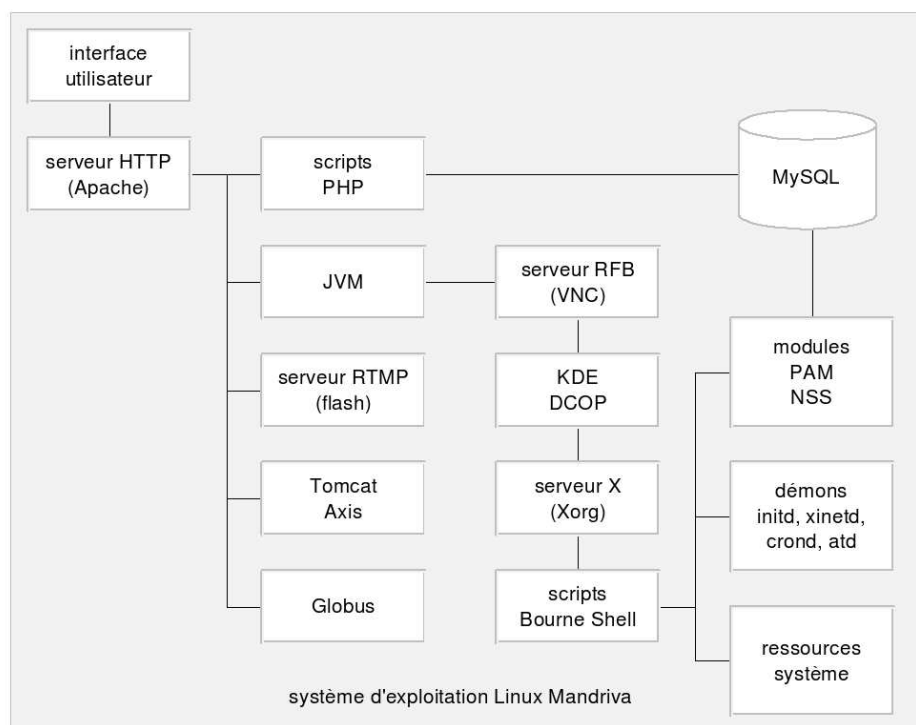


FIG. 4.11 – Technologies pour l'intégration du modèle AGORA

Sur la figure 4.11, *Public Authentication Module (PAM)* est utilisé pour l'authentification et *Name Service Switch (NSS)* pour gérer les droits sur les ressources. *LDAP* a été utilisé durant une courte période avec nos partenaires du projet *ELeGI*. Une base *MySQL* et des scripts en *php* ont remplacé *LDAP* pour administrer les relations entre utilisateurs, groupes et leur assigner des droits. *Apache* comme serveur *HTTP* a été configuré avec l'usage d'hôtes virtuels. La gestion dynamique des ressources a été réalisée avec des démons du système *Linux* : *initd* pour les processus lancé dès la mise sous tension du serveur ; *crond* pour les processus exécutés à intervalles réguliers ; *atd* pour les processus à exécuter à un instant précis et *xinetd* pour les processus à exécuter lors d'une connexion cliente. Les scripts de gestion des processus ont été écrits en *Bourne Shell* ce qui a grandement facilité l'interfaçage entre les services propres au système d'exploitation et les services de haut niveau mis à la disposition des utilisateurs. Pour le bureau virtuel, nous avons utilisé le protocole *RFB* avec la technologie de compression *tight* intégrée au serveur *VNC*. Un serveur *Flash* est utilisé pour la communication audio-vidéo. Cette solution a été préférée à une librairie *Java Media Framework (JMF)*, qui demande une installation complexe sur les postes clients. La gestion des sessions de collaboration en mode statique (*i.e* réservation de la ressource) a initialement été réalisée avec des servlets *Java* sur *Tomcat* et *Axis* pour déployer le service en *WSDL*. Cette solution a toutefois été abandonnée lorsque nous avons opté pour une gestion dynamique des sessions de collaboration. La solution *Globus* est toujours d'actualité mais elle a été mise en attente depuis 2006 pour favoriser les expérimentations du modèle *AGORA*, indépendamment de la façon de le déployer.

⁴Plusieurs versions de *Linux Mandriva* se sont succédées de la version 10.1 à la version actuelle 2007.1

EXPÉRIMENTATIONS

Les méthodes viennent à la fin.
Nietzsche

LES EXPÉRIMENTATIONS ont été réalisées en deux phases avec des utilisateurs volontaires. La première phase, d'une durée de quatre mois a débuté au mois de mars 2005 sur la version 2 du prototype AGORA . La seconde phase a débuté en janvier 2007 sur la version 3 et vient de se terminer au mois de juillet. A ce jour, plus de cinquante utilisateurs à travers le monde se sont cordialement prêtés à cette évaluation. Ces expérimentations ont consisté à amorcer des situations de collaboration puis à laisser les groupes s'auto-organiser. Nous avons ainsi pu identifier les facteurs exerçant une influence positive sur le processus de construction de connaissances tout en cherchant à minimiser rétroactivement les facteurs ayant un effet défavorable sur ce processus.

Table des matières

5.1	Objectif et méthodologie	88
5.2	Cadre expérimental général	88
5.2.1	Plan d'expérimentations	88
5.2.2	Environnement matériel	89
5.3	Scénarii d'expérimentation	90
5.3.1	<i>Encyclopédie de Chimie Organique Electronique (EnCOreE)</i>	90
5.3.2	<i>Coordination Nationale de Formation en Microélectronique (CNFM)</i>	91
5.3.3	Autres scenarii	91
5.4	Résultats majeurs	92
5.4.1	Observations de l'activité du système	92
5.4.2	Résultats d'utilisation pour le scénario <i>EnCOreE</i>	93
5.4.3	Résultats d'utilisation pour le scénario <i>CNFM</i>	94
5.4.4	Autres résultats	95

5.1 Objectif et méthodologie

L'objectif de ces expérimentations est de valider le modèle AGORA dans des situations d'utilisation concrètes. Nous nous sommes intéressés d'une part au comportement des utilisateurs durant les sessions de collaboration, et, d'autre part, à la quantité de ressources utilisées au cours de ces sessions. Au départ, nous nous étions fixés sur la méthode classique d'évaluation des systèmes de communication qui consiste à mesurer des temps de réponse puis à corréliser ces mesures (données quantitatives et objectives) avec des retours utilisateurs (données qualitatives et subjectives). Par exemple, il est possible de déterminer les seuils de qualité de service acceptables pour les utilisateurs en termes de quantité de trafic réseau et d'utilisation de ressources informatiques pour transmettre un flux audio-vidéo et des rafraîchissements d'écrans graphiques [Dugénie *et al.* 2002]. Finalement nous n'avons pas jugé pertinent de chercher de tels critères qui ne restent pas moins arbitraires dans ce contexte stochastique. Nous avons préféré laisser une complète liberté d'usage de l'environnement, puis, établir avec les utilisateurs, un compte rendu final sur l'impression générale qu'ils ont eu. Nous misons sur la spontanéité des comportements en évitant ainsi tout a priori qui risquerait de biaiser nos résultats.

5.2 Cadre expérimental général

Le cadre expérimental général décrit le plan (typologie des utilisateurs, protocole, critères d'évaluation) et l'environnement matériel commun à l'ensemble des scénarii d'expérimentation. Pour des raisons de simplification, le terme *groupe* est utilisé dans ce chapitre pour qualifier une *organisation virtuelle*, c'est-à-dire à la fois la communauté d'utilisateurs et les ressources allouées à cette communauté.

5.2.1 Plan d'expérimentations

Typologie des utilisateurs

Le choix des utilisateurs n'a pas fait l'objet d'une étude sélective particulière. Toutes les personnes qui se sont montrées motivées ont eu accès à l'environnement AGORA. L'important étant de leur laisser le plus de liberté possible sur la façon d'appréhender l'utilisation de leur espace collaboratif. La plupart des utilisateurs rencontrés ne maîtrisent pas les environnements informatiques complexes. L'accès au service de façon intuitive via un navigateur et sur notification par email nous a paru être une exigence de départ. La table 5.1 présente un récapitulatif du nombre des utilisateurs qui se sont cordialement prêtés aux expérimentations. Les utilisateurs membres de plusieurs groupes ont été comptabilisés une seule fois dans cette table. Entre 2005 et 2007, nous avons atteint un nombre total de 69 utilisateurs d'AGORA.

Protocole expérimental

Les nouveaux utilisateurs sont notifiés par email avec les informations de connexion et le groupe dont ils sont membres. Dès qu'ils se connectent avec leur navigateur, ils se retrouvent dans l'espace collaboratif du groupe d'où ils peuvent commencer à interagir avec les autres membres déjà connectés. Dès sa création, un groupe dispose d'une complète autonomie d'évolution. Le fondateur du groupe, qui dispose initialement des droits absolus sur tous les *Services de Base Persistants (SBP)*, peut déléguer ses droits sur les services. Par exemple, après avoir intégré de nouveaux membres, il peut autoriser un ou plusieurs d'entre eux à coopter de nouveaux membres, de fonder un nouveau groupe, voire d'ajuster les droits des membres du groupe.

Critères d'évaluation

Les critères d'évaluation portent sur deux niveaux qui sont traités séparément :

- Le niveau **système** qui porte essentiellement sur la quantité de ressources consommées (mémoire, CPU, bande passante réseau).
- Le niveau **environnement** relatif à l'ergonomie de l'environnement et des outils, ainsi qu'à leurs effets sur le comportement de l'utilisateur dans l'*espace collaboratif*.

5.2.2 Environnement matériel

L'environnement matériel à partir duquel se sont déroulées les expérimentations s'appuie sur l'infrastructure du projet *ELeGI*. Cette infrastructure est représentée sur la figure 5.1. Les services sont déployés sur plusieurs sites des divers partenaires du projet *ELeGI*. Le *Lirmm* apporte la plateforme AGORA, accessible sur le WEB par un serveur frontal (nommé sur la figure *GSD front-end*). L'*Université de Stuttgart* apporte un serveur de fichiers *WebDav* et un serveur *LDAP* pour fédérer les identités utilisateurs. Nous avons utilisé ce serveur dans la première phase d'expérimentation mais, par la suite, nous avons préféré utiliser un serveur *MySQL*¹. *Knowledge Media Institute (KMI)* a apporté *Flashmeeting*, un service de communication audio-vidéo multiplexé entre tous les sites et *BuddySpace*, un service de communication textuelle et de présence augmentée [Eisenstadt & Dzbor2002]. Les utilisateurs sont essentiellement répartis en Europe (Montpellier, Bayonne, Namur), à l'exception de l'*Université de Stanford* en Californie qui a apporté l'outil *Protégé* utilisé dans l'élaboration d'ontologies partagées.

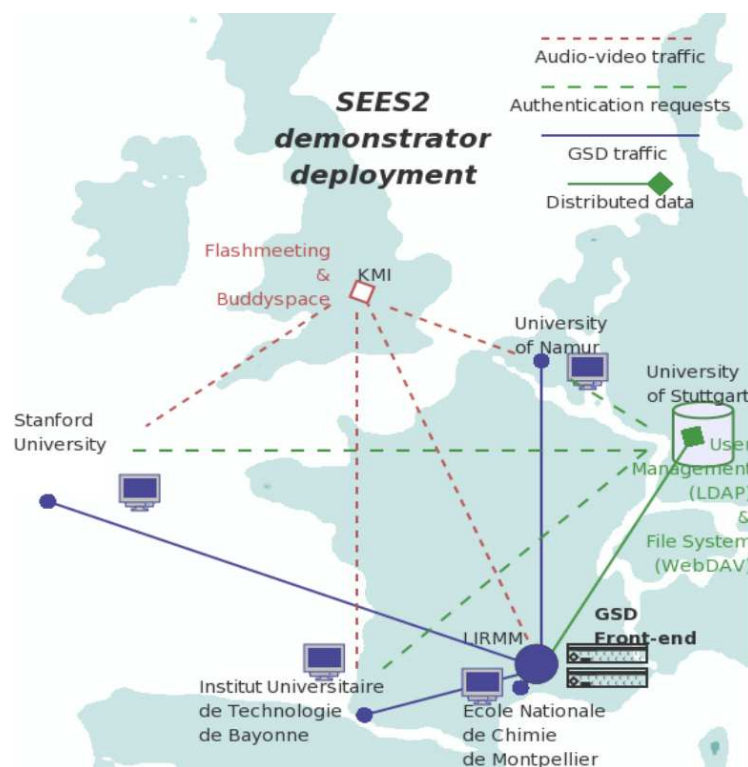


FIG. 5.1 – Infrastructure expérimentale du projet *ELeGI*

¹Les services de l'infrastructure *ELeGI* n'étant pas déployés de façon redondante, il a été souhaitable de minimiser les risques de défaillance pour les expérimentations en regroupant les identités des utilisateurs sur le même site que le service fourni à partir de ce site. L'option du serveur *MySQL* a été motivée par la nécessité de pouvoir gérer les autorisations de membres sur les services de manière relationnelle.

Groupe	Profil	Nombre
encore	Chimistes organiciens	4
encore-dico	Fondé par trois chimistes du groupe encore	-
cnfm	Microélectroniciens	5
elegi	Chercheurs expertise eScience	14
master2007	Etudiants	5
sagora	Etudiants	17
colibri	Physiciens	3
kayou	Chercheurs au Lirmm expertise IA	3
lirmm	Autre personnel du Lirmm	7
lagrid	Ingénieurs expertise Grid	3
eportfolio	Chercheurs-Ingénieurs expertise BD	3
4m	Ingénieurs informatique en italie	2
larzac	Conservation patrimoine en pays cœur d'Hérault	3
Total		69

TAB. 5.1 – Nombre d'utilisateurs (non-dupliqué) par groupe

5.3 Scénarii d'expérimentation

Conjointement à une évaluation objective des performances, le banc d'essais du modèle AGORA comporte des scénarii d'expérimentation subjectifs. Chaque groupe a déroulé ses propres scénarii selon ses problématiques et au fil des situations qui se sont présentées entre 2005 et 2007. Cette section présente de façon plus détaillée deux scénarii qui ont fait l'objet d'un nombre important de sessions de collaboration :

- *Encyclopédie de Chimie Organique Electronique (EnCOre)*
- *Coordination Nationale de Formation en Microélectronique (CNFM)*

5.3.1 Encyclopédie de Chimie Organique Electronique (EnCOre)

Ce scénario comporte initialement deux objectifs [Lemoisson & Cerri2005] :

1. Comprendre les principes de construction d'ontologies à l'aide d'outils spécialisés
2. Exploiter un nouvel outil de synthèse de molécules

Construction d'une ontologie pour l'encyclopédie de chimie organique

En chimie organique, comme dans beaucoup d'autres disciplines, le sens des termes spécifiques au domaine ne font pas toujours consensus. En réalité, la raison n'est pas tant le fait de désaccords sur des principes fondamentaux mais plutôt sur des ambiguïtés dans l'acceptation du sens ou la compréhension des concepts de base. C'est pourquoi, la construction d'une connaissance consensuelle dans un domaine aussi complexe et mouvant que la chimie organique requiert des règles de validation par les experts de ce domaine. Ainsi le projet *EnCOre* vise à élaborer une encyclopédie de chimie organique basée sur des corpus théoriques et expérimentaux. Sur les conseils d'experts en sciences cognitives, la clé pour mener à bien la construction de cette encyclopédie est de constituer un noyau ontologique du domaine dont le but est de définir consensuellement les concepts et les relations entre ces concepts. Toutefois, l'élaboration d'une ontologie requiert des méthodes et la maîtrise d'outils appropriés qui n'est pas une compétence naturelle pour des chimistes. Ces outils sont sophistiqués et hautement spécialisés. Les experts en ontologies leur trouveront tel ou tel avantage (support multiutilisateurs, gestion de relations ternaires, etc.). Ces avantages ne sont à l'évidence pas intrinsèquement liés à la problématique de la chimie organique.

Ainsi, sur la base de recommandations d'experts en ontologies, nous avons choisi d'adopter l'outil *Protégé* [Protégéurl], largement utilisé, car il permet l'élaboration d'ontologies de façon collaborative. Bien que cet outil se limite à la construction d'ontologies à relations binaires, cette limite ne nous paraît pas être un frein dans le cadre de notre expérimentation. Quel que soit l'outil utilisé, les chimistes requièrent l'intervention d'un spécialiste de l'outil et, dans le cas de *Protégé*, ils requièrent aussi une aide sur les principes de construction d'ontologies. Nous voyons ainsi, qu'un groupe d'experts en chimie organique doit acquérir un minimum de connaissance sur les ontologies et sur l'outil qu'ils vont employer. Ainsi, cette expérimentation met en commun des compétences complémentaires. Le groupe est formé de chimistes et d'experts en sciences cognitives avec une spécialité en construction d'une ontologie.

Apprentissage d'un outil de synthèse de molécules

La synthèse de molécules est une activité pratiquée régulièrement en chimie organique. La description des principes de liaisons n'est pas toujours simple à expliquer sans avoir recours à des moyens visuels. Pour cela, l'*Ecole Nationale Supérieure de Chimie de Montpellier (ENSCM)* a mis au point, depuis une dizaine d'années, un outil de synthèse de molécules appelé *RESYN Assistant* [Berasaluce *et al.*2004]. Pour bien expliquer les principes de synthèse avec des experts d'autres universités, un chimiste de l'*ENSCM*, CLAUDE LAURENÇO, a utilisé le *COLLABUREAU* pour diffuser une représentation de molécules et présenter les résultats de synthèses réalisées par l'outil *RESYN Assistant*. Les chimistes de l'*Université de Namur* ont pu assister à cette démonstration.

5.3.2 Coordination Nationale de Formation en Microélectronique (CNFM)

La formation en microélectronique fait appel à des équipements lourds et coûteux. Le *CNFM* dispose à Montpellier d'un nouveau testeur de circuits microélectroniques (voir figure 5.2). Sur ce synoptique sont représentés les différentes fonctions nécessaires pour accéder à ce testeur et pour diffuser les écrans de contrôle aux différentes salles de formation. Plusieurs sites en France peuvent accéder à cette ressource et effectuer en temps réel des cours sur ce testeur avec des formateurs expérimentés [Latorre *et al.*2006].

5.3.3 Autres scénarii

Larzac et pays cœur d'Hérault

Un patrimoine environnemental requiert une organisation minutieuse de sa connaissance car les sources d'information sont multiples (géographiques, historiques, administratives, etc.). L'organisation de la connaissance du patrimoine environnemental du cœur d'Hérault (région Lodève-Larzac) est actuellement réalisée dans une base de données riche en contenu mais qui a atteint ses limites du fait d'une structure peu évolutive. A l'origine, cette base de données n'a pas été conçue par des informaticiens.

L'objectif de collaboration ici est de revoir la structure et réaliser le portage de cette base de données en faisant intervenir des compétences complémentaires.

ePortfolio

De même que le scénario *Larzac*, ce scénario vise à mettre en commun des compétences d'ingénierie réparties. L'objectif est de déployer un service interactif de collaboration avec gestion d'une base de données *SQL*.

Colibri

Colibri est un projet interdisciplinaire *STIC* entre une équipe de chercheurs en physique et une équipe de chercheurs en informatique du *LIRMM*. Au départ, il s'agissait d'une collaboration, de plus

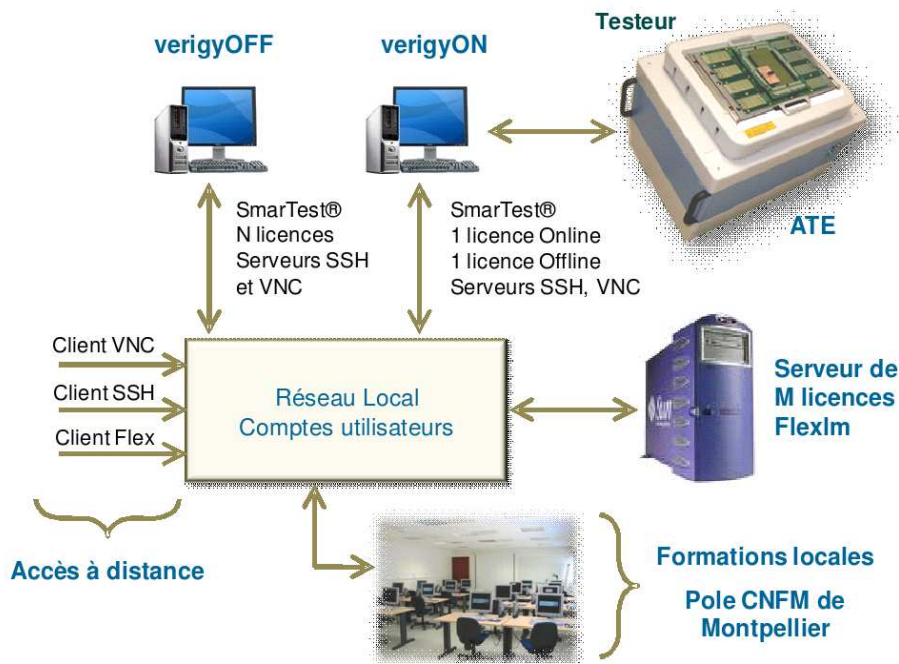


FIG. 5.2 – Synoptique du scénario CNFM

de dix ans, entre deux physiciens et un informaticien du LIRMM sur le thème du bruit des lasers avec plusieurs publications sur le sujet [Arnaud & Esteban1990, Arnaud2002, Chusseau & Arnaud2002, Chusseau *et al.*2003]. L'objectif est de créer un *espace collaboratif* avec des services de modélisation (*Matematica*) en vue d'élaborer des représentations de modèles de fluctuations des systèmes représentables par des équations différentielles linéaires.

5.4 Résultats majeurs

Ces résultats rendent compte d'observations de l'activité et des performances du système, ainsi que les réactions recueillies auprès des utilisateurs.

5.4.1 Observations de l'activité du système

Les mesures de consommation de ressources relevées durant les sessions de collaboration, nous renseignent sur la capacité du système à supporter la charge moyenne estimée. Sur les graphiques 5.3 et 5.4 sont reportées les mesures de consommation de mémoire, la charge des microprocesseurs (CPU) et le volume de trafic échangé sur le réseau. L'échelle des temps en abscisse couvre une durée de sept minutes.

Le graphique 5.3 a été relevé avec quatre connexions espacées d'une minute. La première connexion concerne un utilisateur appartenant à trois groupes. La seconde connexion concerne un utilisateur appartenant à deux groupes. Les deux autres connexions concernent des utilisateurs appartenant à un seul groupe. Nous observons que les pics de consommation CPU et réseau apparaissent surtout lors d'une connexion. Les pointes sont amorties en quelques secondes. La mémoire consommée par le système est initialement de 30% des 8 GO de mémoire totale. Sur le graphique 5.3, nous pouvons observer une progression de l'ordre de 10% de mémoire utilisée pour l'ensemble des utilisateurs pendant les six premières minutes, puis un retour rapide à 30% à la sixième minute. Sur les deux graphiques, la libération

des ressources mémoire est bien réalisée. La consommation de capacité réseau croît graduellement entre les pointes mais pas de manière excessive. Sur le graphique 5.4 les mêmes connexions ont été réalisées de façon quasi-simultanée. Nous observons avec l'étalement des pics de données transférées que le sémaphore a bien séquencé les événements de lancement des bureaux. Le peu d'activité réseau au cours de la période qui suit les connexions montre l'étonnante efficacité du protocole de rafraîchissement graphique.

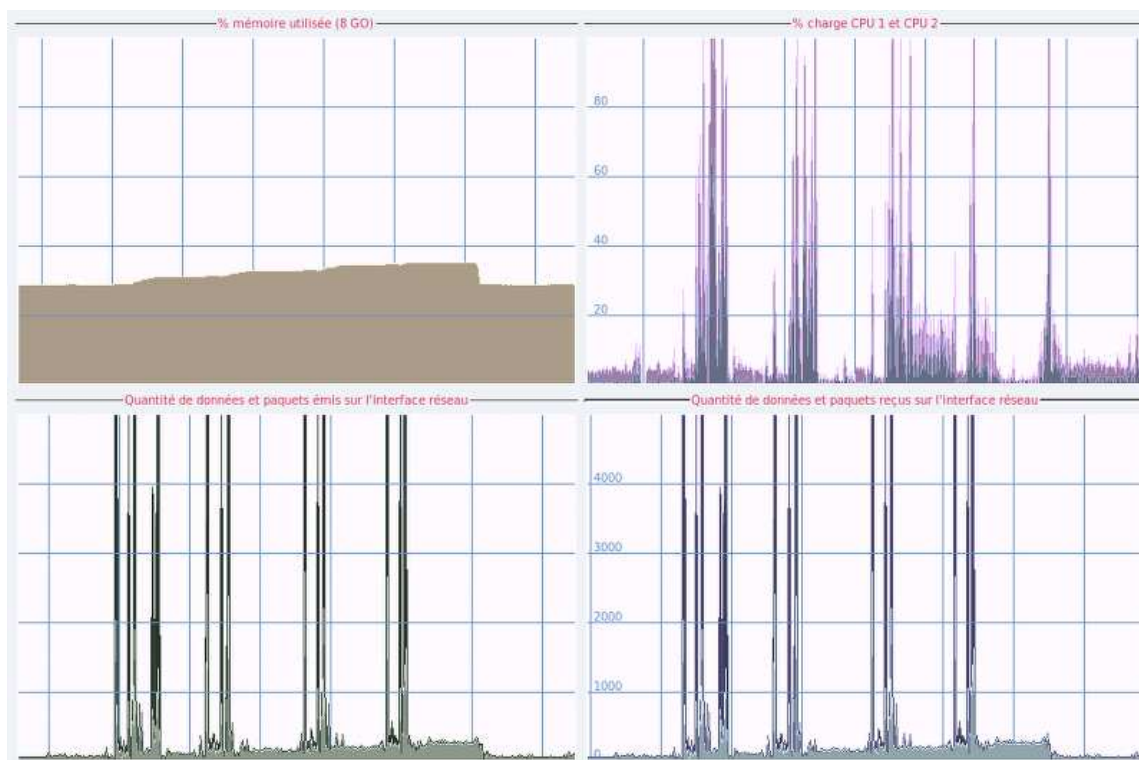


FIG. 5.3 – Consommation de ressources système avec connexions séquentielles

5.4.2 Résultats d'utilisation pour le scénario *EnCOre*

Parmi les résultats attendus, l'environnement AGORA a permis une prise en main de l'outil *Protégé* par les chimistes après cinq sessions de collaboration, avec une bonne compréhension de ce qu'est une ontologie. L'apport du nouvel outil de synthèse de molécules *Resyn Assistant* n'était pas programmé au départ mais s'est avéré être un facteur déterminant dans la construction collaborative des représentations symboliques des molécules. Par exemple, comme le montre la figure 5.5 la description de la synthèse de la molécule *carpanone* a été expliquée à un novice (moi-même) par trois chimistes à partir d'outils différents. L'interdisciplinarité entre sciences cognitives et chimie a été l'une des réussites de ce scénario. Le retour des utilisateurs sur l'ergonomie de l'environnement AGORA est excellent : « *Comme en local* » (CLAUDE LAURENÇO). De plus, cette mise en relation régulière des chimistes a eu pour effet de stimuler leur motivation autour de leurs problématiques. Un autre résultat également très satisfaisant de ce scénario est la création d'un nouveau groupe *encore-dico* en vue d'engager le développement d'un thésaurus. Grâce aux *SBP*, l'un des membres du groupe *encore*, JULIE HENRY a pu créer un nouveau groupe disposant de son propre conteneur de services afin de travailler de manière plus précise sur l'élaboration du dictionnaire. Il est important de noter que Julie n'a aucune notion particulière de la façon d'administrer un serveur d'application, ni même d'installer ces applications sur une machine.

Ces résultats illustrent une situation classique lorsqu'il s'agit de concevoir une solution à un problème posé en termes purement techniques. Le cas du scénario avec les chimistes est un exemple typique.

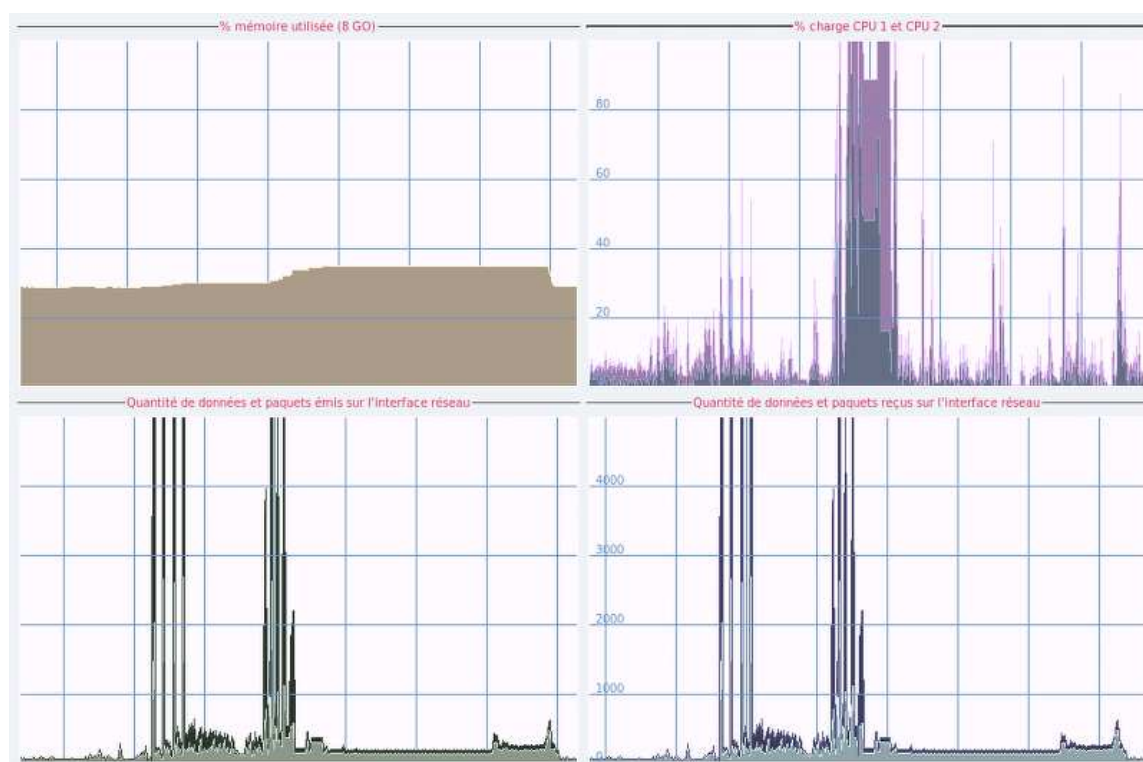


FIG. 5.4 – Consommation de ressources système avec connexions simultanées

L'idée initiale du projet *ELeGI* était d'attendre des utilisateurs qu'ils décrivent leurs besoins sur une base imposée par les termes techniques de l'informatique. La conséquence de ceci fût que les chimistes pensaient avoir besoin d'une ontologie pour décrire leurs concepts. Il s'est ensuite avéré que leur besoin réel était de multiplier leurs périodes de concertation pour clarifier leur terminologie.

AGORA a permis de décrire les représentations visuelles des modèles avec leurs propres outils et de se concerter sur les définitions. Ils ont, par la suite, opté pour l'élaboration d'un thésaurus et abandonné l'idée d'ontologie. Cette idée est certes nécessaire à terme mais à ce stade son usage s'est avéré prématuré. Ce comportement de groupe ne pouvait être prévisible à l'avance. La flexibilité du modèle AGORA a permis de laisser au groupe le libre arbitre sur ses choix d'organisation, sur ses méthodes et sur ses outils.

Lors de la première phase d'expérimentation, les cinq sessions programmées avec les chimistes visaient à expliquer comment utiliser un outil de structuration d'une ontologie partagée. En parallèle, ils pouvaient déjà commencer à utiliser des outils de synthèse de molécules qu'ils connaissaient. De ce fait, ils se sont très vite rendu compte de l'intérêt d'utiliser le *COLLABUREAU* de manière plus générique comme environnement de visualisation partagée.

5.4.3 Résultats d'utilisation pour le scénario *CNFM*

L'expérimentation en microélectronique a donné lieu à trois sessions de collaboration. Les deux premières sessions se sont déroulées avec les formateurs du testeur *Verigy* basés au *LIRMM*. Ces formateurs se sont ainsi familiarisés avec l'environnement AGORA et ont immédiatement senti l'intérêt du concept d'*espace collaboratif multimodal* pour leur besoin. Ils ont aussi donné quelques retours sur les améliorations possibles, à savoir :

- L'agrandissement dynamique des écrans pour visualiser simultanément les circuits et les signaux.

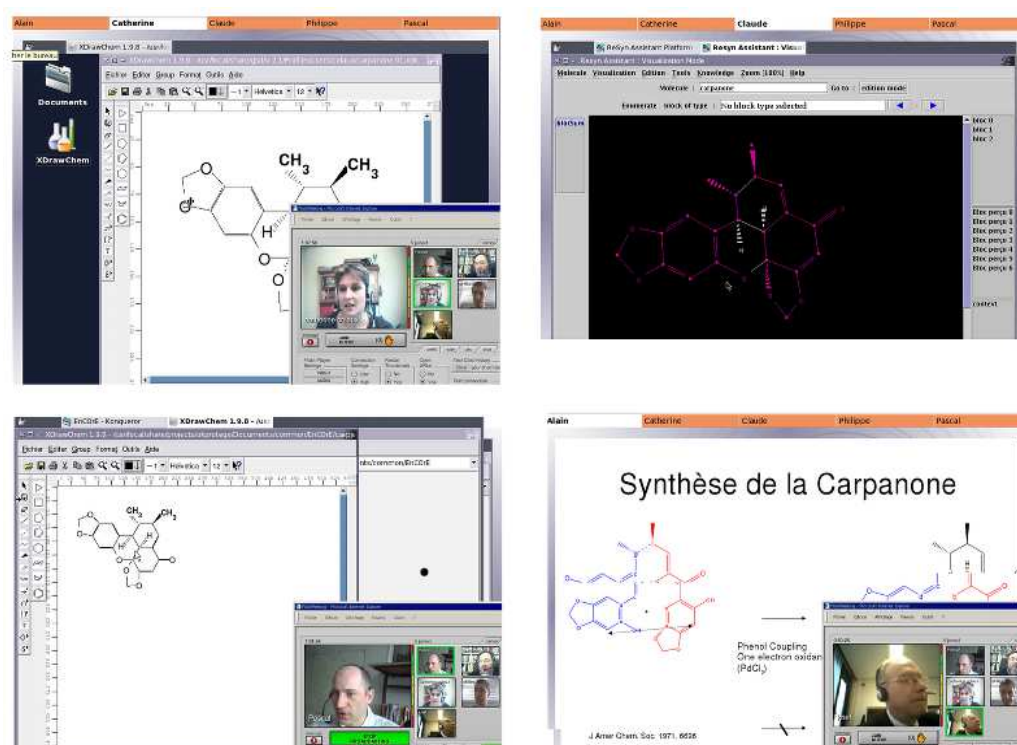


FIG. 5.5 – Discussion autour de la synthèse d'une molécule.

- Offrir la possibilité à un formateur d'effectuer des sessions non-symétriques, c'est à dire que seul le formateur peut voir les écrans de tous les étudiants.

La troisième session a été réalisée avec un formateur du *CNFM* basé à Strasbourg et qui a pu durant cette session effectuer des séquences de tests sur le testeur *Verigy* grâce aux recommandations des formateurs de Montpellier.

5.4.4 Autres résultats

Mesures de performances du protocole *RFB*

Grâce aux performances très satisfaisants du protocole *RFB* et particulièrement de l'algorithme de compression *tight* nous avons pu effectuer des transferts graphiques sur de longues distance avec une très faible sensation de latence. Cet algorithme présente un facteur compression de l'ordre de 137 :1. La consommation de mémoire par session graphique est d'environ 100 Moctets. La table 5.2 donne une synthèse des mesures de performances effectuées dans diverses conditions. Il est important de noter aussi que la latence maximale à ne pas dépasser est de l'ordre de 200 ms. Au delà de cette valeur, l'interaction avec le pointeur de la souris commence à être perceptible. Sur l'ensemble des mesures relevées à Stuttgart (de), Bruxelles, Namur (be), Milton Keynes (uk), Salerne (it), Stanford (us), aucune n'excède 100 ms. La bande passante moyenne se situe entre 120 kbps avec peu de graphiques et 460 kbps avec des graphiques riches [Dugénie & Cerri2006].

Mutualisation de compétences réparties

D'un point de vue des performances en termes de temps d'ingénierie, le groupe *eportfolio* a pu résoudre des problèmes complexes en mettant en commun des compétences complémentaires en matière

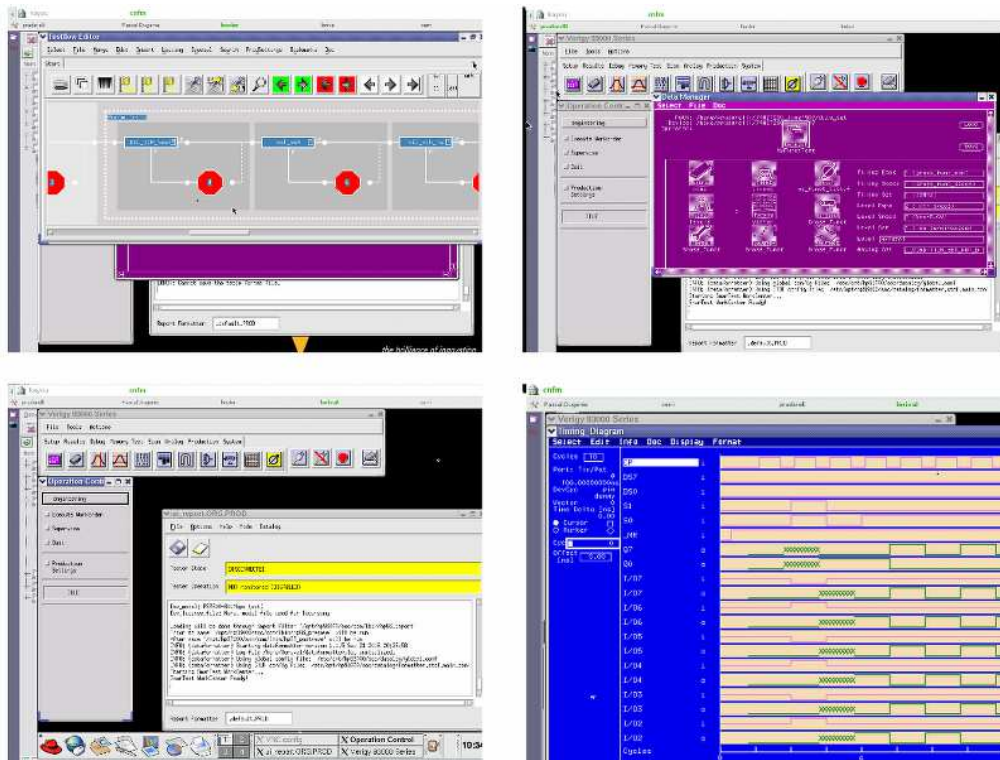


FIG. 5.6 – Utilisation collective du testeur Verigy

d’administration des systèmes et en conception de bases de données. Un problème concret qui demanderait trois ou quatre jours à un ingénieur a ainsi pu être résolu à deux ingénieurs en deux heures seulement.

Immanence et ubiquité

Dans l’ensemble, les scénarii d’expérimentation ont montré une grande satisfaction au vu des aspects d’*ubiquité* indépendamment de la plateforme cliente. Le principe d’immanence a été démontré à plusieurs reprises lorsque les utilisateurs ont pu commencer à déléguer des droits sur des services de base. Le groupe des étudiants *master2007* a largement exploité cette potentialité pour créer de nouveaux groupes de travail, dont le groupe *sagora*. Les groupes se sont mis le plus souvent en situation de collaboration de façon spontanée. L’attribution des rôles s’est en général passée de façon informelle.

Problèmes rencontrés du côté de la plateforme cliente

Deux problèmes liés aux versions des bibliothèques Java côté client ont été observés. Le premier concerne le support des touches META avec la JVM Sun 1.5 sous *MS Explorer* et le second concerne le rafraîchissement graphique incomplet avec le navigateur *Mozilla* sous *MacOS*.

TAB. 5.2 – Synthèse des mesures *RFB*

Paramètres globaux							
Date de la mesure	05/09/06						
Compression RFB	tight (no jpeg)						
Version du serveur RFB	1.2.9						
Taille du canevas (pixels)	1024x768						
Nom du serveur	agora.lirmm.fr						
Nom du client	altair.lirmm.fr						
Point de collecte des mesures	interface eth0 (agora.lirmm.fr)						
Signification des colonnes							
Col. A	Vitesse moyenne de transmission (en kBits/s)						
Col. B	Volume utile de données échangées (en kOctets)						
Col. C	Nombre total de paquets						
Col. D	Taux de paquets de taille inférieure à 79 octets						
Col. E	Taux de paquets de taille entre 80 et 1279 octets						
Col. F	Taux de paquets de taille supérieure à 1280 octets						
Col. G	Appréciation						
Navigation	peu active						
Contenu des fenêtres	en majorité du texte						
-	A	B	C	D	E	F	G
Rx	101	778	1792	36	48	16	Excellent
Tx	11	81	1133	100	0	0	Excellent
Navigation	très active						
Contenu des fenêtres	en majorité du texte						
-	A	B	C	D	E	F	G
Rx	1174	8793	9681	19	35	46	Excellent
Tx	53	402	5834	100	0	0	Excellent
Navigation	très active						
Contenu des fenêtres	en majorité des graphiques						
-	A	B	C	D	E	F	G
Rx	1961	14832	13283	10	22	68	TB
Tx	64	484	7101	100	0	0	TB

SYNTHÈSE

L'ordinateur est à l'informatique, ce que le télescope est à l'astronomie.
Edsger Dijkstra

L'OBJECTIF IDÉAL que nous nous étions fixé au départ était de réaliser le prototype d'un modèle d'*espace collaboratif ubiquitaire* opérant sur une infrastructure à ressources distribuées. Compte tenu du manque de maturité des technologies liées à ce type d'infrastructures et pour répondre à notre problématique, nous avons dû faire preuve de pragmatisme en nous focalisant davantage sur une définition plus fine du concept d'*espace collaboratif ubiquitaire*. Ceci nous a permis de dégager trois caractéristiques décrivant ce concept. La disponibilité des utilisateurs pour les expérimentations a été une chance à saisir dans la période consacrée à cette thèse, ce qui a ajouté une valeur significative au prototype AGORA. Les résultats obtenus au terme de cette thèse sont encourageants d'un point de vue de la faisabilité conceptuelle du modèle AGORA. Concernant l'infrastructure, nous avons exploré le potentiel théorique de GRID d'un point de vue strictement opérationnel. Une architecture *orientée-services* est adaptée pour la gestion des ressources et pour la sécurité mais comporte quelques insuffisances en termes de performance et ne propose pas de solution satisfaisante pour la gestion des *organisations virtuelles*. Nous avons tenté d'y répondre, en combinant GRID avec le modèle *Agent-Groupe-Rôle*. Ce chapitre récapitule nos contributions en rapport avec l'état de l'art et suggère quelques pistes à explorer pour la suite.

Table des matières

6.1	Bilan	100
	6.1.1 Contributions	100
	6.1.2 Enseignements	100
6.2	Discussion	101
	6.2.1 Potentialités d' AGORA	101
	6.2.2 Aspect à approfondir	102
	6.2.3 Un scénario pour l'avenir	102

6.1 Bilan

6.1.1 Contributions

Parmi les contributions de cette thèse au monde de la recherche, les aspects les plus novateurs peuvent se résumer ainsi :

- Le chapitre 2 contribue à caractériser le concept d'*espace collaboratif ubiquitaire* en rapport avec la problématique de l'*intelligence collective* et la mise en relation de compétences réparties.
- Le chapitre 3 contribue à introduire un formalisme graphique pour décrire l'infrastructure théorique idéale pour déployer des *espaces collaboratifs ubiquitaires* à grande échelle. Ce chapitre propose également un modèle d'intégration tirant parti des avantages de deux modèles solidement établis que sont GRID et le modèle *Agent-Groupe-Rôle*.
- Le chapitre 4 contribue à modéliser l'*espace collaboratif ubiquitaire* et propose une réalisation partielle des services de base et des services collaboratifs qui ont, d'ores et déjà, fait l'objet de nombreuses démonstrations. Nous proposons notamment une utilisation originale d'un protocole de distribution graphique (*ie. RFB*) en imbriquant de multiples bureaux partagés sur un nœud fortement couplé au réseau. Ces bureaux sont en relation les uns avec les autres selon l'appartenance des agents aux divers groupes. Les agents sont ainsi toujours en interaction dans le contexte d'un groupe.
- Le chapitre 5 contribue à introduire une méthodologie expérimentale originale selon une approche systémique, en assimilant l'organisation virtuelle à un système complexe, afin de produire des résultats en cohérence avec les contraintes des utilisateurs et le besoin en ressources.

En somme, la valeur ajoutée de ce travail porte essentiellement sur une exploitation potentielle du modèle AGORA¹ en tant que système générique dont l'utilité est de servir tout aussi bien l'enseignement à distance que la collaboration interdisciplinaire.

6.1.2 Enseignements

Au vu de l'émergence encore embryonnaire de la question traitée dans le cadre de cette thèse, il a semblé plus pertinent de décomposer l'étude afin de traiter ensuite la question de départ dans sa globalité. Ainsi, nous avons dressé deux états de l'art séparés auprès de leur communauté respective. Le premier état de l'art, relatif aux *espaces collaboratifs ubiquitaires*, se focalise sur les aspects du côté de l'utilisateur. Le second état de l'art, relatif aux *infrastructures à ressources distribuées*, se positionne du côté de l'opérateur de service. Une fois réunis, ces deux états de l'art nous apportent de précieux enseignements. Tout d'abord, nous remarquons que la propriété d'ubiquité rend ces deux concepts indissociables. GRID réalise la condition d'ubiquité grâce au service à état et élargit le potentiel sécurité grâce à une infrastructure de sécurité normalisée. Toutefois, sans une réflexion globale autour des concepts centrés-utilisateurs et des concepts d'infrastructure, la définition d'un *espace collaboratif* se réduira toujours à des solutions *ad-hoc* peu évolutives. Ceci tend à restreindre l'autonomie des groupes et de ce fait ne tient pas compte des principes essentiels d'immanence et d'auto-organisation que nous préconisons. Ensuite, de par la taille et le dynamisme de sa communauté, GRID est un cadre conceptuel solide. Plusieurs facteurs indiquent que la tendance va se poursuivre en faveur de ce changement de paradigme :

- l'accroissement rapide de la densification des réseaux ;
- la généralisation de l'approche orientée-services à état et sécurisés ;
- l'amélioration, la stabilisation des normes et l'ouverture des logiciels ;
- la chorégraphie et l'orchestration des services pour faciliter le passage à l'échelle (lorsqu'un service est défaillant, d'autres peuvent prendre le relais) ;
- la découverte des service à la fois sur le plan syntaxique et sur le plan sémantique.

¹Le prototype de la plateforme AGORA est l'un des projets *Source Forge* : sourceforge.net/projects/agora. Ce service est actuellement déployé sur un nœud du *Lirmm* : agora.lirmm.fr

6.2 Discussion

Nous avons délibérément choisi d'adopter des protocoles existants sur des critères de légèreté côté client (*i.e* *RFB* pour les sessions graphiques et *RTMP* pour les communications audio et vidéo). En contrepartie, ce choix implique des traitements plus importants à l'autre extrémité du réseau. En s'en tenant à une infrastructure classique centrée-serveur, cette approche montrerait rapidement ses limites. Par contre, sur une infrastructure à ressources distribuées, la mutualisation de ces ressources permet de répartir la charge de manière plus efficace pour offrir une meilleure qualité de service avec une bonne tolérance aux pannes. Ainsi, ce système détient la capacité de traiter l'ensemble des opérations réalisées dans les différents *espaces collaboratif ubiquitaires* en garantissant une disponibilité accrue. La persistance de l'état de la communication est totalement pris en charge par le service qui délivre des flux multimédia au client. Le caractère d'ubiquité est alors assuré par le fait que les utilisateurs peuvent accéder au service depuis n'importe quel poste client qu'il soit fixe ou nomade.

6.2.1 Potentialités d' AGORA

Comparaison entre AGORA et *AccessGrid*

Puisque *AccessGrid* est le seul environnement qui explore aujourd'hui les potentialités de l'architecture GRID que nous avons véritablement recensé, une comparaison avec AGORA peut sembler pertinente à ce stade. Nous avons essayé d'identifier des critères de comparaison reportés dans la table 6.1 à titre indicatif. Les spécifications d'*AccessGrid* ont été recueillies dans le rapport technique de l'ANL [Olson2001]. Force est de constater que, finalement, tel ou tel choix technologique adopté par l'une ou l'autre solution n'a pas véritablement de valeur comparative. Même si la solution, *Access Grid* est fort séduisante et, à certains égards, bien meilleure techniquement qu' AGORA , elle est encore loin, selon nous, de satisfaire des conditions essentielles que nous avons posées initialement. Que peut faire un utilisateur face à *Access Grid* s'il n'a pas suivi au préalable une formation en informatique ? La réponse est qu'il passe à autre chose. Pour mettre ceci en évidence, il serait intéressant de mesurer le temps nécessaire pour configurer un nœud *Access Grid* pour un non-expert du domaine.

Notre approche se démarque radicalement de toutes les solutions que nous avons rencontrées par le simple fait que nous cherchons à apporter une solution centrée-utilisateur, de manière à faire sortir GRID du cénacle du calcul intensif pour le rendre accessible à tous en quelques clics de souris. L'utilisateur, qu'il soit chimiste, microélectronicien ou physicien et qui souhaite bénéficier des avantages de travailler à distance avec des services à très forte valeur ajoutée tels que nous les avons décrits avec l'*espace collaboratif ubiquitaire* n'a même pas besoin de savoir qu'il travaille sur une infrastructure GRID. Les choix techniques liés à l'infrastructure n'intéressent *in fine* que l'opérateur du service, si ce dernier a le soucis d'optimiser ses coûts.

Facilité d'accès aux services

La facilité d'accès est une caractéristique singulière du modèle AGORA . Si nous avons adopté la procédure couramment utilisée pour accéder aux service GRID, nous aurions eu trop peu d'utilisateurs intéressés par nos expérimentations, autres que quelques informaticiens acharnés. Pour illustrer ceci, le mode opératoire ci-dessous, récapitule les instructions demandées aux biologistes qui souhaitent utiliser *BioinfoGRID* [Gridurl] sur l'infrastructure EGEE :

1. Obtenir un certificat délivré par une des autorités reconnues par l'*organisation virtuelle, Biomed VO*.
2. Installer un certificat sur le navigateur délivré par une des autorités reconnues par *Biomed VO*.

Spécifications	<i>AccessGrid</i>	AGORA
Nombre de serveurs par nœud	4	2
Système d'exploitation	Linux	Linux
Protocole Audio-Vidéo	H.261 via RTP	Flash RTMP
Annulation d'écho	oui	non
Visualisation partagée	<i>Collavis</i>	COLLABUREAU
Authentification	X509	PAM
Capacité réseau minimale	10Mbps	1 Mbps
Enregistrement des sessions	oui	non
Niveau d'intégration	faible	moyen
Infrastructure	201 nœuds, 24 pays	2 nœuds, 1 pays

TAB. 6.1 – Comparaison entre AGORA et *Access Grid*

3. Autoriser les *cookies*².
4. Créer un compte pour accéder au téléchargement de l'interface utilisateur *gLite*.
5. Installer l'interface utilisateur *gLite*.
6. Installer ce certificat sur le domaine de l'interface utilisateur où le compte a été créé.
7. S'inscrire à *Biomed VO* en utilisant *Virtual Organization Membership Service (VOMS)*.
8. Suivre une liste de plusieurs pages d'instructions pour s'inscrire à *Biomed VO*.
9. Attendre que la requête soit évaluée par le *Biomed VO manager*.

Cette procédure, relativement longue, explique pourquoi elle n'a pas été traduite dans plusieurs langues. Il serait intéressant de connaître le nombre et la typologie des membres de *Biomed VO*. Un constat plus général est que l'interface utilisateur pour accéder à des systèmes complexes exige de la part des modèles existants, une implication technique excessive pour une grande majorité des utilisateurs.

6.2.2 Aspect à approfondir

Le modèle AGORA a cherché à respecter fidèlement le concept d'*espace collaboratif ubiquitaire* en incluant le caractère d'immanence nécessaire à l'auto-organisation des groupes. GRID qui fournit de la ressource dans un espace sécurisé, est, pour sa part, contraint de respecter des normes pour l'interopérabilité des services. Pour ce faire, la norme *WSDL* fournit les éléments utiles pour la définition des interfaces des services.

Ainsi, il serait souhaitable d'approfondir la question de la définition d'une interface *WSDL* qui contienne l'ensemble des méthodes pour gérer les différentes entités des organisations virtuelles : membres, services, autorisations et ressources. Cette interface serait une contribution utile en normalisation.

Par ailleurs, le prototype de la plateforme requiert un certain nombre d'améliorations ou de parties à développer. Pour l'essentiel, il s'agit d'améliorer l'ergonomie des processus PISA et de développer le SBP (Environnement) nécessaire à rendre totalement dynamique l'*espace collaboratif ubiquitaire*.

6.2.3 Un scénario pour l'avenir

Imaginons le scénario pour lequel un opérateur envisage d'exploiter le service d'*espace collaboratif ubiquitaire*. Dans un soucis de rentabilité, cet opérateur va se fixer deux objectifs. Le premier vise à couvrir un champ d'application le plus large possible pour atteindre le plus grand nombre d'utilisateurs.

²Cette opération est en contradiction avec l'idée que GRID fournit de la ressource à état et ne devrait, par conséquent, pas nécessiter que le client gère l'état de la session.

Le second cherche à rationaliser au mieux l'utilisation de l'infrastructure tout en garantissant un niveau élevé de qualité de service.

Le premier objectif implique une grande souplesse conceptuelle du service déployé. De ce fait, cet opérateur ne pourrait pas s'engager sur le développement d'une solution figée répondant à un besoin précis. En effet, nous avons observé que ce besoin n'est pas exprimé dans des termes qui se traduisent par des solutions d'ingénierie, mais il évolue, voire émerge, par la découverte des champs du possible offert par la technologie.

Le second objectif implique la nécessité d'exploiter des ressources informatiques en maximisant le taux d'utilisation de la capacité globale offerte par la réunion des ressources mises à disposition dans l'infrastructure. Il apparaît clair qu'une infrastructure *ad-hoc* composée de serveurs distribués géographiquement et assignés à une tâche précise ne permet pas d'équilibrer une charge non prévisible au niveau de chaque serveur. Ainsi, la mutualisation des serveurs et la virtualisation de cette capacité informatique globale, à la manière de GRID, permettent de rationaliser l'usage de ces ressources en équilibrant la charge.

Un troisième aspect, plus subjectif mais tout aussi important, est le rapport de confiance que les utilisateurs vont avoir avec cet environnement technologique qui leur est proposé. Cette confiance s'établit ou s'étirole avec un effet boule de neige selon des critères de disponibilité et de sécurité du service mais aussi selon le fait que l'accès au service ne soit pas exigeant en termes de procédure. Dans l'idéal, ce service construit continuellement les relations potentielles entre les membres et l'espace collaboratif qui les accueille. Toute interaction est alors traitée comme un élément constructif de cet ensemble dynamique.

CONCLUSION ET PERSPECTIVES

Nous nous sommes proposés de conjuguer deux concepts innovants dans le but de susciter l'émergence d'une *intelligence collective*. Le premier est l'*espace collaboratif ubiquitaire* et le second est une *infrastructure à ressources distribuées* matérialisée par GRID. Afin de mettre en évidence l'intérêt d'associer ces deux concepts, nous avons développé le modèle AGORA qui vise à favoriser la mise en relation de compétences humaines et à dynamiser l'interaction avec des processus artificiels. Au cours des deux années qui viennent de s'écouler, nous avons réalisé des expérimentations avec plusieurs dizaines d'utilisateurs pour valider ce modèle de façon itérative.

En multipliant les scénarii et en variant les situations de collaboration, nous avons observé des cas qui indiquent l'émergence d'une *intelligence collective*. L'exemple le plus révélateur est la conversation à distance entre deux chimistes face à un modèle de molécule dont la synthèse a été réalisée sous leurs yeux, avec des ressources distantes. Même s'ils se connaissaient de par leur renommée, cette situation leur a permis de mieux comprendre ce qui les différencie sur la manière d'analyser un problème concret. Nous constatons que le renforcement des modalités d'interaction par l'usage d'une interface de communication synchrone a favorisé cette émergence. Les discussions en temps réel, couplées avec des représentations visuelles sur des bureaux partagés, a permis aux intervenants d'accroître l'efficacité du processus de collaboration.

Par ailleurs, le caractère *ubiquitaire* de l'espace collaboratif leur a permis de réaliser un gain de temps non négligeable sur la gestion de leur outil informatique. Ceci a engendré un effet bénéfique sur leur motivation par voie de conséquence. Nous avons ainsi mis en évidence l'hypothèse de départ de PIERRE LEVY selon laquelle les dispositifs techniques, sociaux et sémiotiques sont fortement imbriqués. Il nous semble donc pertinent de les étudier dans leur globalité avant d'élaborer des solutions complètes.

Nous avons également observé l'importance des *SBP* sur la troisième caractéristique de l'*espace collaboratif ubiquitaire* posée au départ : l'*immanence* pour assurer l'*auto-organisation* des groupes. Cette caractéristique permet aux groupes de conserver une maîtrise totale de leur environnement technologique en sélectionnant leurs propres services et en choisissant leurs propres règles de sécurité. Toutefois, la réalisation de l'*immanence* est fortement liée au choix technologique. L'état de l'art nous indique que GRID se retrouve comme une infrastructure idoine car elle permet de coordonner les ressources pour besoins variables selon les communautés d'utilisateurs.

Au stade de son développement actuel, nous pouvons affirmer objectivement la faisabilité technique du modèle AGORA. Bien qu'il ait été évalué sur une plateforme à nœud unique, la méthode expérimentale employée pour mesurer les performances statiques et dynamiques du modèle AGORA apporte des éléments valables pour qualifier le prototype développé. Toutefois, nous n'avons pas engagé de campagne d'expérimentation quantitative pour estimer avec suffisamment de précision le dimensionnement et les conditions requises pour le passage à l'échelle. Cet aspect est important dans la mesure où nous justifions l'utilisation d'une infrastructure GRID pour répondre à cette nécessité. Notre analyse théorique du modèle de déploiement indique que GRID convient pour la coordination des ressources et la sécurité mais le modèle d'*organisations virtuelles* est insatisfaisant. Pour cela nous proposons de lui associer le modèle relationnel *Agent-Groupe-Rôle*. Pour compléter cette analyse théorique il serait souhaitable d'engager une étude de faisabilité de déploiement du modèle AGORA à plus grande échelle comme l'une des perspectives à moyen terme. Cette infrastructure se doit d'être tolérante aux pannes et déployable à moindre effort en ingénierie. A court terme, nous allons poursuivre notre effort de validation du modèle avec une communauté d'utilisateurs du *Coordination Nationale de Formation en Microélectronique*

(*CNFM*). Nous espérons ainsi enrichir ce capital expérimental et, pourquoi pas, finaliser les spécifications du système dans un livre blanc à destination de la recherche et de l'enseignement supérieur.

Mutualiser des ressources informatiques distribuées est une étape intermédiaire qui mène à une autre forme de mutualisation : celle de l'expertise de chacun d'entre nous. C'est lorsque les biologistes et les informaticiens spécialistes du calcul intensif collaborent que nous observons les plus grandes avancées dans le domaine des biotechnologies.

ANNEXE A : FONCTIONS AGORA

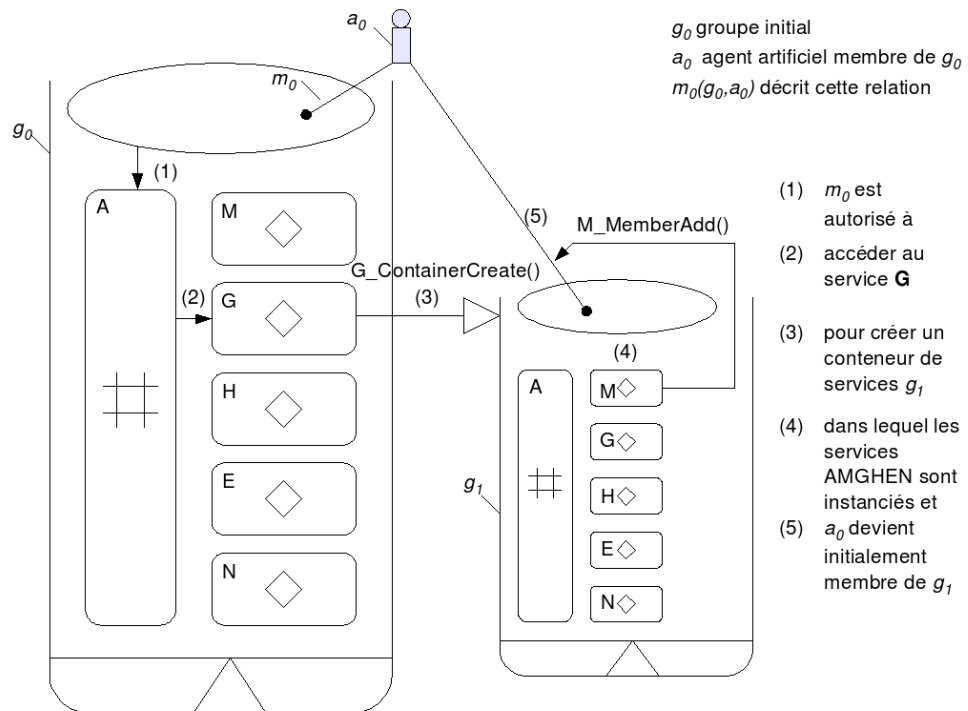
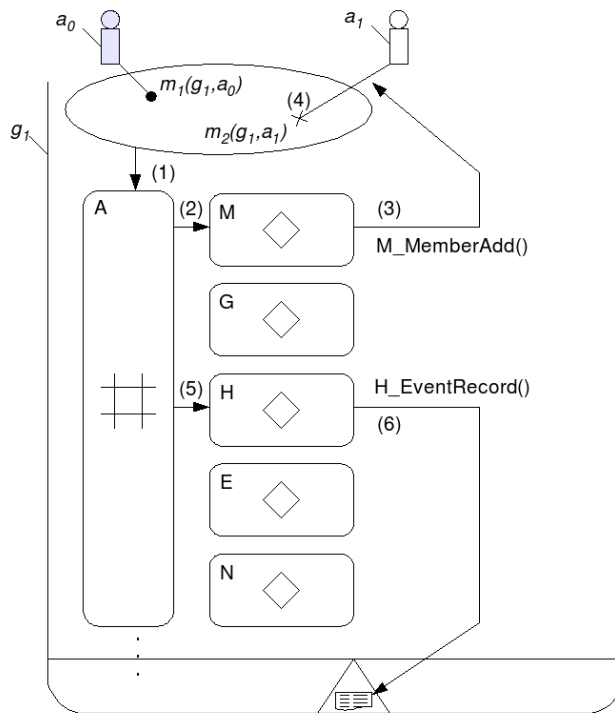
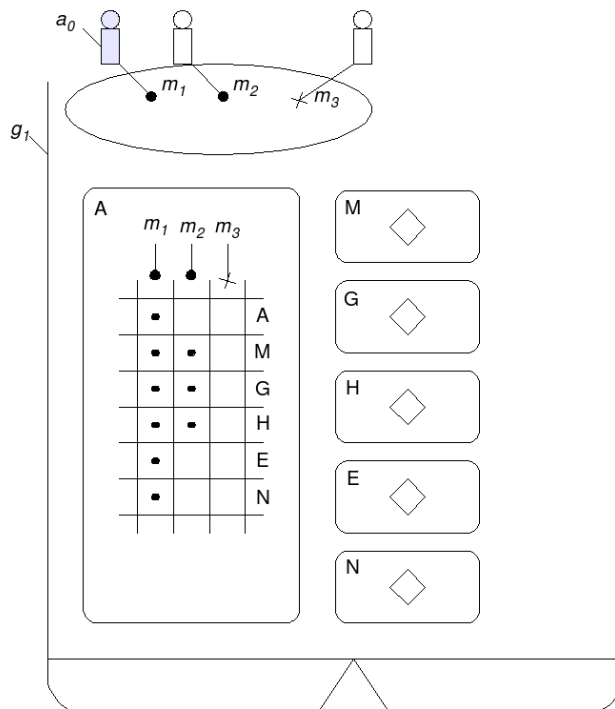


FIG. 1 – Conteneur racine et création d'un premier conteneur



- a_1 agent humain
- (1) $m_1(g_1, a_0)$ est autorisé à
 - (2) accéder au service **M**
 - (3) pour ajouter un nouveau membre $m_2(g_1, a_1)$ au groupe g_1 .
 - (4) La croix indique que m_2 ne dispose que de privilèges limités par défaut (ex. la consultation).
 - (5) Toute opération sur le groupe passe par une invocation du service **H**.
 - (6) L'opération d'ajout du membre est consignée dans le registre des événements du groupe g_1 .

FIG. 2 – Ajout d'un membre dans le premier conteneur

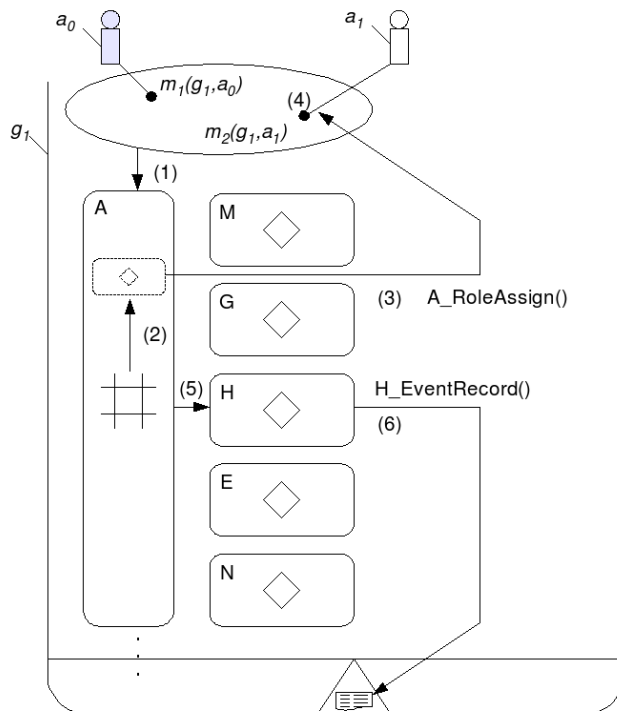


$m_1 m_2 m_3$, membres de g_1 ont chacun une entrée dans la matrice d'autorisations. m_1 en tant que fondateur du groupe dispose initialement de tous les privilèges sur g_1

A chaque rôle correspond un jeu de permissions sur un ensemble de services.

Par exemple, le rôle de responsable des membres correspond à des permissions sur le service **M** (ex. ajout ou suppression de membres).

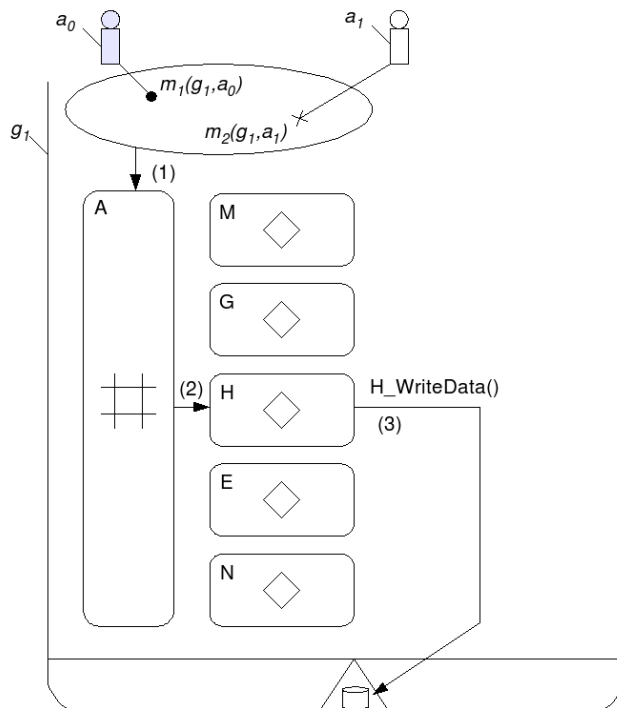
FIG. 3 – Matrice des autorisations



A est aussi une instance de service (représenté ici en pointillé).

- (1) m_1 est autorisé à
- (2) accéder au service **A**
- (3) pour assigner un rôle à m_2 qui donne à ce dernier des privilèges sur les services du groupe g_1 .
- (4) Le point noir indique que m_2 dispose de certains droits sur les services de g_1 .
- (5) Toute opération sur le groupe passe par une invocation du service **H**.
- (6) L'opération de changement des permissions de m_2 est consignée dans le registre des événements du groupe g_1 .

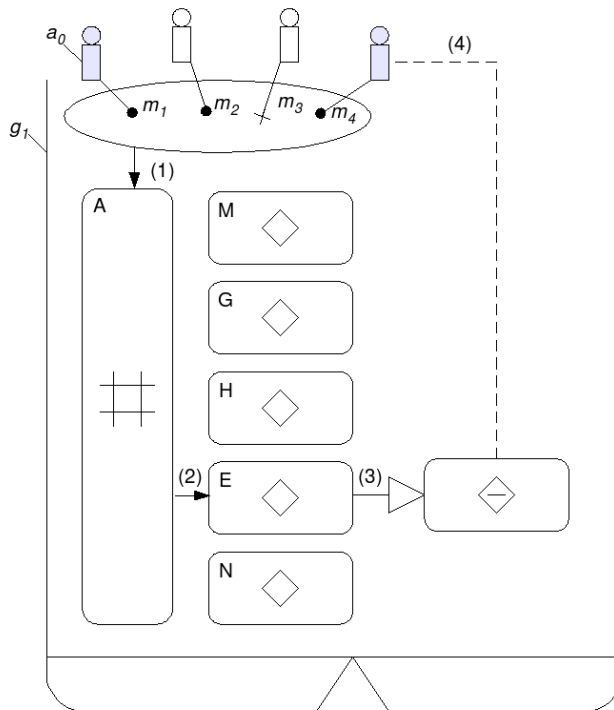
FIG. 4 – Assignment des rôles



L'opération d'écriture dans l'espace des contenus est aussi gérée par le service **H**.

- (1) m_1 est autorisé à
- (2) accéder au service **H**
- (3) pour écrire dans l'espace des contenus du groupe g_1 .

FIG. 5 – Ecriture de données d'un groupe

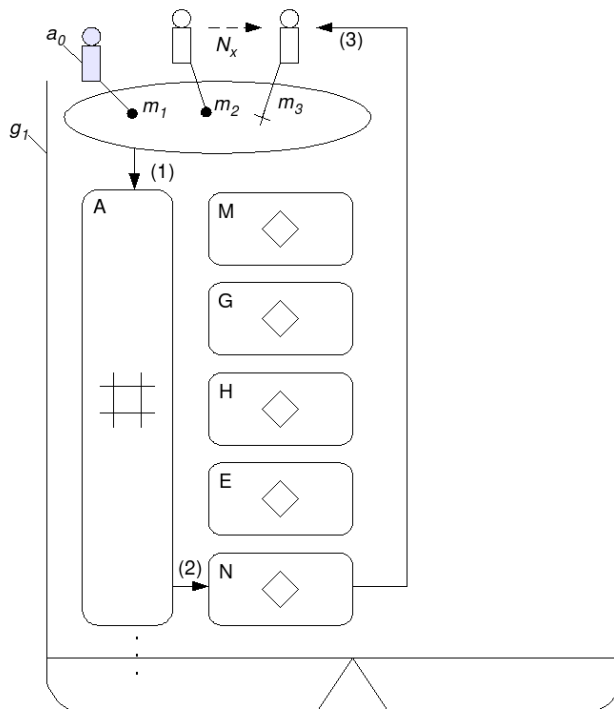


Tout ajout d'un nouveau service pour le groupe s'effectue par le service E.

- (1) m_2 est autorisé à
- (2) accéder au service E
- (3) pour instancier un nouveau service dans le conteneur de g_1 .
- (4) Cette nouvelle instance de service peut rétroactivement être elle-même membre de g_1 (en tant qu'agent artificiel) pour disposer ainsi d'une entrée dans la matrice des autorisations.

Bien que ce ne soit pas représenté ici, cette opération est, elle aussi, consignée dans le registre des événements du groupe g_1 .

FIG. 6 – Instanciation de services

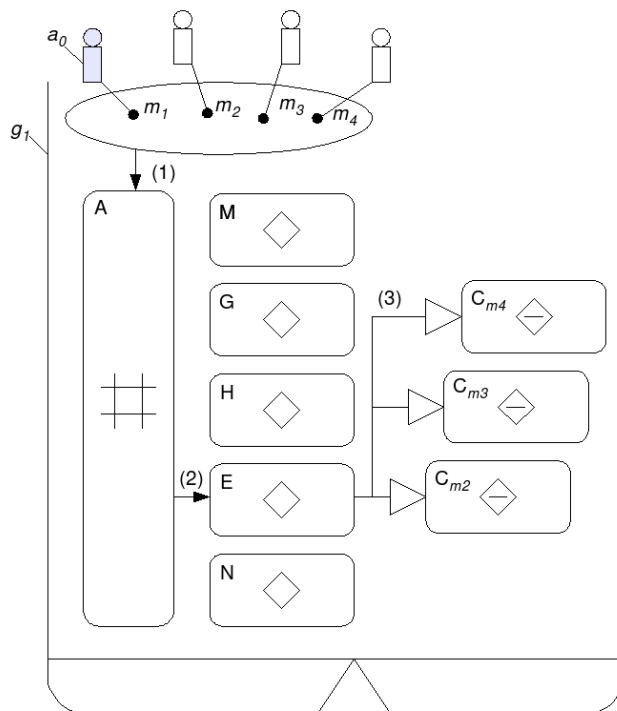


Les notifications transitent via le service N.

- (1) m_2 est autorisé à
- (2) envoyer un type pré-défini N_x de notification
- (3) à l'intention m_2

Bien que ce ne soit pas représenté ici, cette opération est, elle aussi, consignée dans le registre des événements du groupe g_1 .

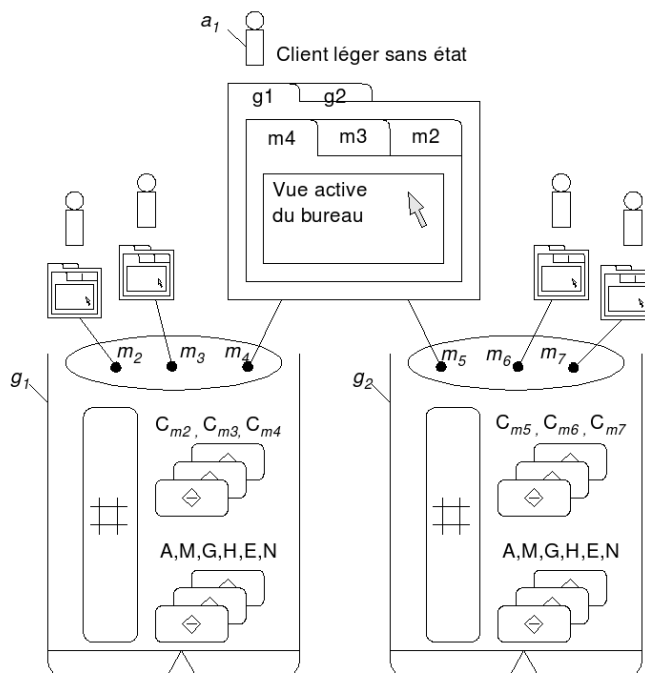
FIG. 7 – Envoi de notifications



Le service **E** se charge de générer une instance du **Collabureau** pour chaque membre connecté.

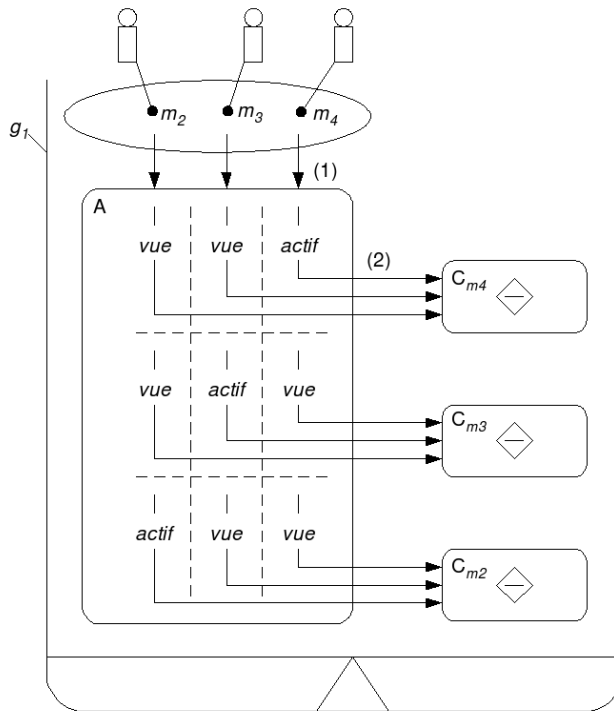
- (1) m_1 est autorisé à
- (2) invoquer $E_ServiceInstanciate()$
- (3) et instancie $C_{m_2}, C_{m_3}, C_{m_4}$

FIG. 8 – Instanciation du Collabureau



Chaque client communique avec les services **Collabureau** associés au membre.
 a_1 visualise deux onglets de groupe puisque a_1 est membre de g_1 et de g_2 .

FIG. 9 – Accès par client léger



Les instances du service **Collabureau** sont accessibles à travers la matrice d'autorisation

- (1) m_2 , m_3 et m_4 sont autorisés à accéder aux instances C_{m2} , C_{m3} et C_{m4}
- (2) soit en mode *vue* pour tous les bureaux des autres membres soit en mode *actif* pour le bureau appartenant au membre lui-même.

FIG. 10 – Couplage des bureaux

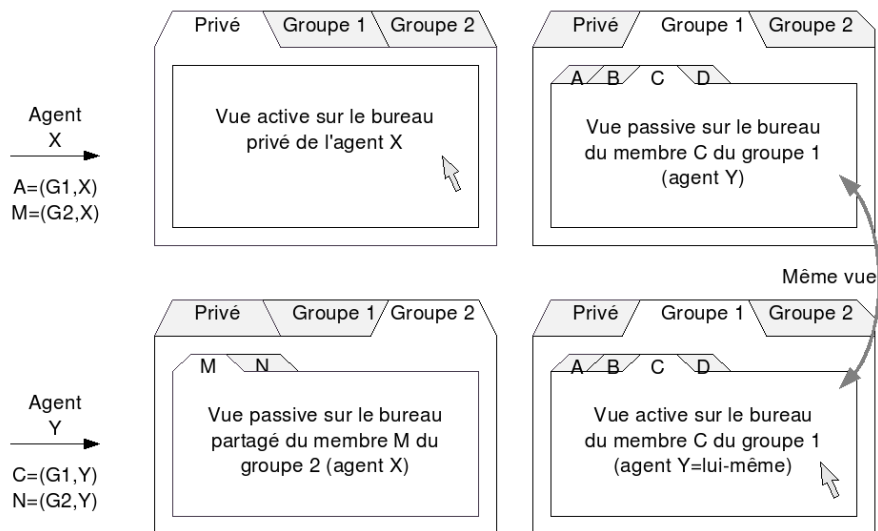


FIG. 11 – Visualisation par onglets

ANNEXE B : PROCÉDURES PISA

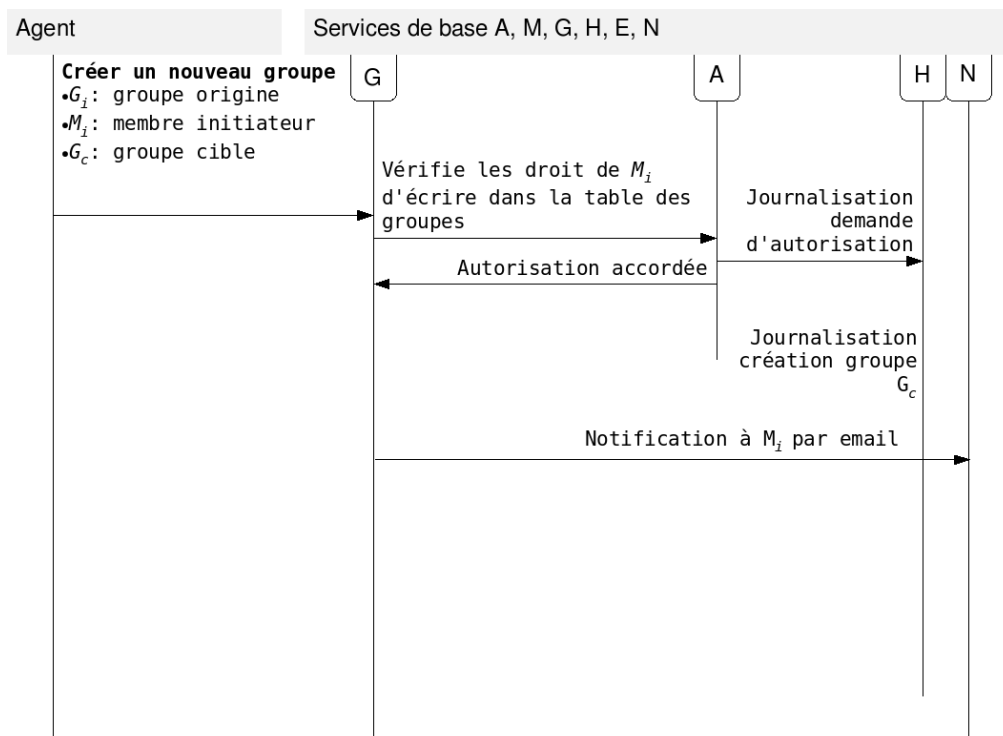


FIG. 1 – Procédure PISA : Créer un nouveau groupe

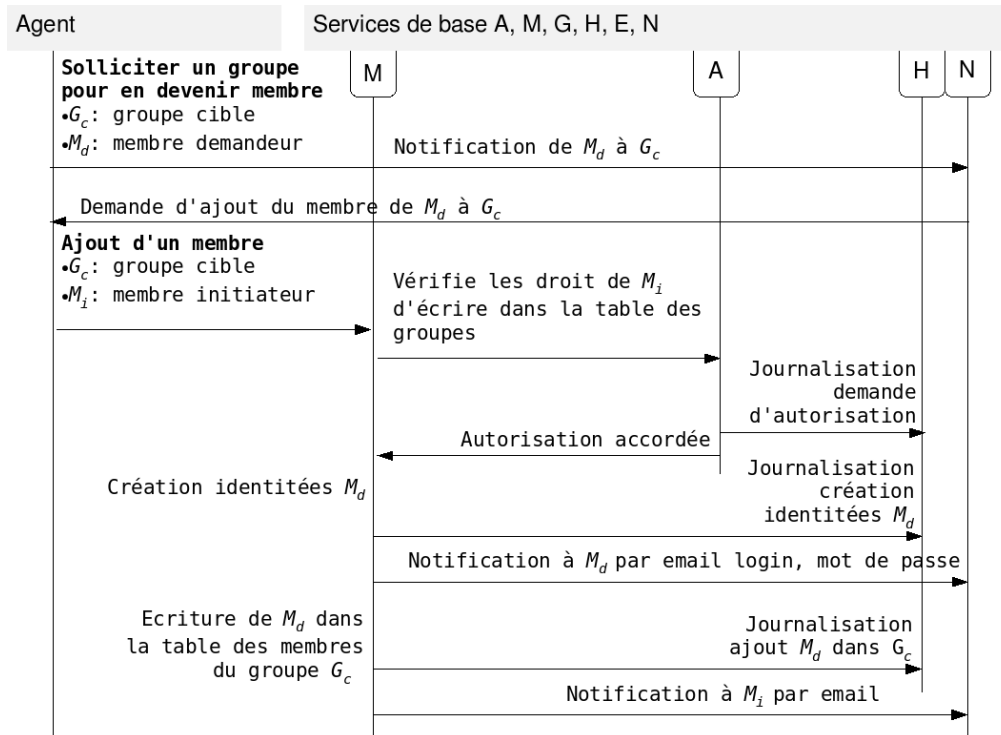


FIG. 2 – Procédure PISA : Solliciter un groupe pour en devenir membre

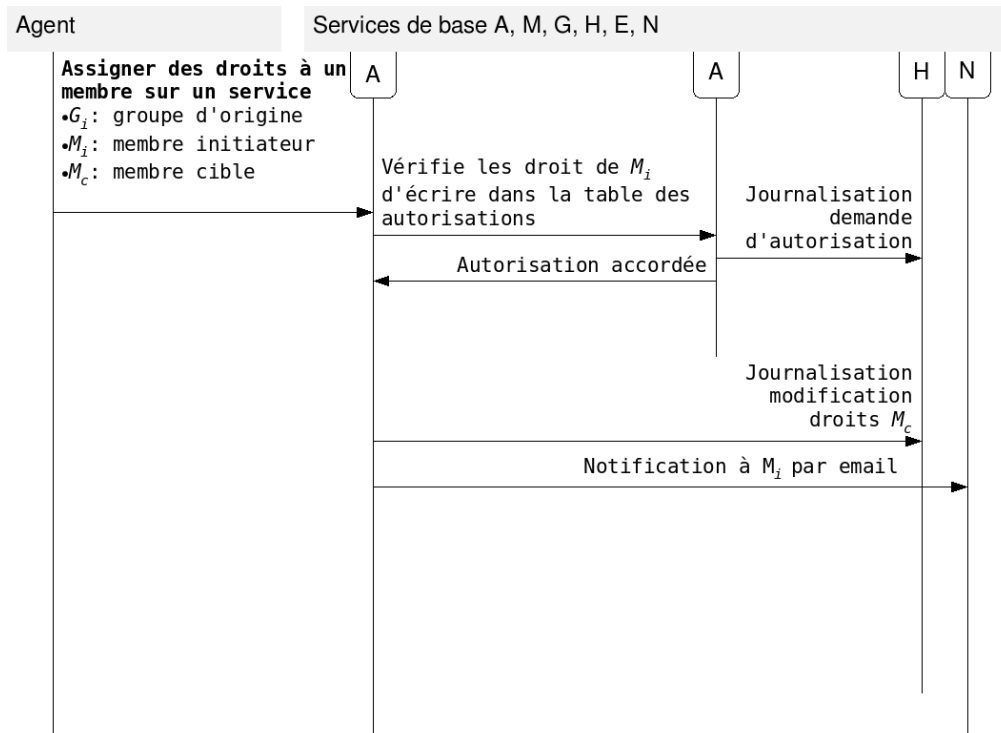


FIG. 3 – Procédure PISA : Assigner des droits à un membre sur un service

TABLE DES FIGURES

1	Buts et moyens fixés dans le cadre de cette thèse	14
1.1	<i>L'intelligence collective</i> dans l' <i>espace anthropologique du savoir</i>	19
1.2	Déroulement d'un processus de collaboration	21
1.3	Interaction de type question-réponse entre deux agents	22
1.4	Contextualisation	25
1.5	Mise en relation de compétences	26
2.1	Métaphore pour comparer <i>espace collaboratif</i> et <i>ENT</i>	32
2.2	Structure des services d'un <i>espace collaboratif ubiquitaire</i>	34
2.3	Exemple caractéristique d' <i>espace collaboratif ubiquitaire</i>	35
2.4	Comparaison de deux approches pour la visualisation partagée	36
2.5	Accès aux contenus par un service synchrone et asynchrone	40
3.1	Trois familles de topologies.	45
3.2	Blue Gene/L	47
3.3	<i>Collabvis</i> , visualisation collaborative sur <i>AccessGrid</i>	49
3.4	Comparaison d'architectures <i>client-serveur</i> et <i>orientée-services</i>	53
3.5	Représentation schématique d'un <i>hôte</i> selon l'architecture VON NEUMANN	56
3.6	Différentes étapes pour la gestion des ressources dans GRID	57
3.7	Structure hiérarchique de validation des certificats X509	59
3.8	Synthèse des concepts GRID	61
3.9	Architecture GRID simplifiée	62
3.10	Le concept de service à l'intersection des domaines GRID et <i>SMA</i>	63
3.11	Réseaux <i>Geant</i> , <i>Renater</i> et <i>R3LR4</i> au niveau de la région LR	64
4.1	Contribution du modèle AGORA	69
4.2	Un groupe vu comme un système (vue d'ensemble et vue détaillée).	71
4.3	Les six <i>Services de Base Persistants (SBP)</i>	72
4.4	Rôle du service à état pour l'ubiquité.	73
4.5	Modèle conceptuel AGORA	75
4.6	Procédure PISA , ajout d'un membre par cooptation	78
4.7	Chronologie récapitulative du développement	79
4.8	Plateforme AGORA	80
4.9	La relation \mathcal{H}	83
4.10	Un sémaphore pour sérialiser des événements asynchrones	84
4.11	Technologies pour l'intégration du modèle AGORA	85
5.1	Infrastructure expérimentale du projet <i>ELeGI</i>	89
5.2	Synoptique du scénario <i>CNFM</i>	92
5.3	Consommation de ressources système avec connexions séquentielles	93
5.4	Consommation de ressources système avec connexions simultanées	94

Table des figures

5.5	Discussion autour de la synthèse d'une molécule.	95
5.6	Utilisation collective du testeur <i>Verigy</i>	96
1	Conteneur racine et création d'un premier conteneur	107
2	Ajout d'un membre dans le premier conteneur	108
3	Matrice des autorisations	108
4	Assignation des rôles	109
5	Ecriture de données d'un groupe	109
6	Instanciation de services	110
7	Envoi de notifications	110
8	Instanciation du Collabureau	111
9	Accès par client léger	111
10	Couplage des bureaux	112
11	Visualisation par onglets	112
1	Procédure PISA : Créer un nouveau groupe	113
2	Procédure PISA : Solliciter un groupe pour en devenir membre	114
3	Procédure PISA : Assigner des droits à un membre sur un service	114

LISTE DES TABLEAUX

1	Segmentation des évolutions conceptuelles autour des paradigmes WEB et GRID	13
3.1	Correspondance entre les concepts OGSi et WSRF	51
4.1	Récapitulatif de l'avancement du développement	81
5.1	Nombre d'utilisateurs (non-dupliqué) par groupe	90
5.2	Synthèse des mesures <i>RFB</i>	97
6.1	Comparaison entre AGORA et <i>Access Grid</i>	102

DEFINITIONS

Espace collaboratif ou collectif

composition de plusieurs services dont l'objet est de gérer des entités (groupes, membres, droits, services, contenus) ainsi que les interactions et les flux d'information (visualisations, parole, notifications) entre ces entités. [page 32]

Service

Le terme service, dans une architecture-orientée services, désigne une entité logicielle, conçue à des fins d'interopérabilité entre systèmes hétérogènes par des messages en format *XML* travers le protocole de transport *HTTP*. Un service se présente avec une interface normalisée (spécifiquement en *WSDL*) et communique avec celle-ci par des messages dans un format également normalisé (spécifiquement *SOAP*). Un service se caractérise par le fait (i) qu'il se définit par un langage de description, (ii) qu'il est publié dans un registre de services, (iii) qu'il peut être découvert dans ce registre et (iv) qu'il peut être composé avec d'autres services. [page 53]

Service à état

Le terme état est vague dans la mesure où il peut couvrir de nombreux aspects d'un système informatisé. Le sens admis par la communauté GRID est qu'un service à état dispose d'une ressource dédiée de telle sorte que l'état de cette ressource peut s'exprimer selon une syntaxe univoque (*i.e* document *XML*), a un cycle de vie bien défini dans le temps et peut interagir avec d'autres services (avec ou sans état). De nombreux exemples, tels qu'une structure de fichiers dans un arbre ou des entrées dans une base de données, ou encore une combinaison de plusieurs de ces objets, peuvent être modélisés par des services à état (FOSTER [Foster *et al.*2004b]). [page 54]

Système

Un système se définit comme un ensemble d'objets et de relations entre ces objets et leurs attributs ; les objets sont les composants ou éléments du système, les attributs sont les propriétés des objets, et les relations sont ce qui fait tenir le système. Si les objets sont des êtres humains, les attributs qui permettent de les identifier dans le système sont leurs comportements de communication (WATZLAWICK, *Une logique de la communication*, p. 120). [page 23]

Trust

Terme anglais de la terminologie GRID qui n'a pas d'équivalent simple en français. Un *trust* est un ensemble d'entités correspondant aux ressources et aux utilisateurs formant un ensemble GRID cohérent. Ces éléments se reconnaissent mutuellement comme faisant partie d'un même *trust*. En pratique, un *trust* est circonscrit par l'origine des signatures des certificats *X509* (*cf.* figure 3.7). En général, un environnement *Grid* couvre plusieurs domaines ayant leurs propres politiques de sécurité. Le *trust* permet de contrôler les interactions entre domaines en s'adaptant aux différentes politiques de sécurité locales des domaines. [page 58]

ABRÉVIATIONS

AGR	Agent Groupe Rôle, [page 64, 69]
ANL	Argonne National Laboratory, [page 48]
BIRN	Biomedical Informatics Research Network, [page 59]
CA	Certification Authority, [page 58]
CAS	Community Authorization Service, [page 56, 60]
CERN	Centre Européen en Recherche Nucléaire, [page 46]
CGI	Common Gateway Interface, [page 77]
CMCS	Collaboratory for the Multi-scale Chemical Science, [page 59]
CNFM	Coordination Nationale de Formation en Microélectronique, [page 87, 90, 91, 105]
CNR	Consiglio Nazionale delle Ricerche de Pisa, [page 5]
CORBA	Common Object Request Broker Architecture, [page 52]
COVISE	COllaborative VItualization and Simulation Environment, [page 48]
CSCW	Computer Supported Cooperative Work, [page 19]
CSIESR	Comité des Services Informatiques de l'Enseignement Supérieur et de la Recherche, [page 44]
CVS	Concurrent Versions System, [page 36]
DCE	Distributed Computing Environment, [page 59]
DNS	Domain Name Service, [page 44]
EGA	Enterprise Grid Alliance, [page 50]
EGEE	Enabling Grid for eScience, [page 46]
ELeGI	European Learning Grid Infrastructure, [page 5, 31, 49]
EnCOre	Encyclopédie de Chimie Organique Electronique, [page 87, 90]
ENSCM	Ecole Nationale Supérieure de Chimie de Montpellier, [page 91]
ENT	Environnement Numérique de Travail, [page 29, 30]
FLOPS	FLoating point Operations Per Second, [page 46]
GDS	Génération Dynamique de Services, [page 62]
GGF	Global Grid Forum, [page 50]
GPC	Grid Proxy Certificate, [page 58]
GRAM	Globus Resource Allocation Manager, [page 50]
GRB	Grid Resource Broker, [page 65]

GSD	Grid Shared Desktop, [page 80]
GSI	Grid Security Infrastructure, [page 50, 56]
GT3	Globus Toolkit version 3, [page 79]
GT4	Globus Toolkit version 4, [page 50]
HLRS	High Performance Computing Center, [page 48]
HPDC	High Performance Distributed Computing, [page 65]
HTC	High Throughput Computing, [page 45]
HTTP	Hyper Text Transfer Protocol, [page 44]
IA	Intelligence Artificielle, [page 11]
ICALT	International Conference on Advanced Learning Technologies, [page 65]
IDATE	Institut de l'Audiovisuel et des Télécommunications, [page 30]
IEEE	International Electrical and Electronics Engineering, [page 64]
IEML	Information Economy Meta Language, [page 12]
IHM	Interface Homme-Machine, [page 31, 39]
IP	Internet Protocol, [page 52]
IPv4	Internet Protocol version 4, [page 55]
IRISA	Institut de Recherche en Informatique et Systèmes Aléatoires, [page 5]
IST	Information Society Technologies, [page 5]
IST	Information Technologies Society, [page 65]
ISTI	Istituto di Scienza e Technologie dell'Informazione, [page 5]
ITCC	Information Technology Coding and Computing, [page 65]
iVDGL	International Virtual Data Grid Laboratory, [page 47]
JMF	Java Media Framework, [page 85]
KMI	Knowledge Media Institute, [page 89]
LCG	LHC Computing Grid, [page 46]
LDAP	Lightweight Directory Access Protocol, [page 60]
LEAD	Linked Environments for Atmospheric Discovery, [page 59]
LHC	Large Hadron Collider, [page 46]
LIESP	Laboratoire d'Informatique pour l'Entreprise et les Systèmes de Production, [page 5]
LIGO	Laser Interferometer Gravitational-Wave Observatory, [page 59]
LOOKING	Laboratory for the Ocean Observatory Knowledge INtegration Grid, [page 59]
LTER	Long Term Ecological Research, [page 59]
MD5	Message Digest 5, [page 57]
MPI	Message Passing Interface, [page 51]
MRI	Mental Research Institute de Palo Alto, [page 24]

NEES	Network for Earthquake Engineering Simulation, [page 59]
NOVA	Network-Oriented Visualization Architecture, [page 48]
NSS	Name Service Switch, [page 85]
OGF	Open Grid Forum, [page 50]
OGSA	Open Grid Service Architecture, [page 49]
OGSI	Open Grid Service Infrastructure, [page 50]
OWL	Ontology Web Language, [page 12]
PAM	Public Authentication Module, [page 85]
PCCE	Pervasive Collaborative Computing Environment, [page 49]
PCRD	Programme Cadre de Recherche et de Développement, [page 5]
PISA	Protocole d'Interaction entre Services et Agents, [page 75, 77]
PO	PetaOctets, [page 46, 56]
PPDG	Particle Physics Data Grid, [page 47]
RCS	Revision Control System, [page 36]
RDF	Resource Description Framework, [page 12]
RFB	Remote Frame Buffer, [page 48]
RMI	Remote Method Invocation, [page 52]
RPC	Remote Procedure Call, [page 50, 52]
RSA	Rivest, Shamir, Adleman, [page 57]
RTMP	Real Time Media Protocol, [page 77]
SBP	Services de Base Persistants, [page 67, 71, 72, 75, 76, 88, 115]
SCEC	Southern California Earthquake Center, [page 59]
SEES	Service Elicitation and Exploitation Scenarios, [page 49]
SMA	Systèmes Multi Agents, [page 12, 14, 18, 43, 67]
SOAP	Service Oriented Application Protocol, [page 50]
SOC	Service Oriented Computing, [page 53, 54]
SSH	Secure SHell, [page 51]
SSL	Secure Sockets Layer, [page 58]
SSP	Storage Service Provider, [page 59]
STROBE	STream OBjects Environment, [page 21]
SVN	SubVersioN, [page 36]
TCAO	Travail Coopératif Assisté par Ordinateur, [page 19, 30]
TF	TeraFlops, [page 46, 56]
TLS	Transport Layer Security, [page 58]
UGS	Ubiquitous Grid Services, [page 65]
UNR	Universités Numériques en Région, [page 30]
URI	Unique Registry Identifier, [page 53]
VOMS	Virtual Organization Membership Service, [page 60, 102]
VPN	Virtual Private Network, [page 59]
WI	Web Intelligence, [page 65]

WSDL	Web Services Description Language, [page 50]
WSRF	Web Services Resource Framework, [page 49]
XML	eXtended Markup Language, [page 12]

BIBLIOGRAPHIE

- [AccessGridurl] AccessGrid. url. *Projet Access Grid*. www.accessgrid.org. 48
- [Agarwal2001] Agarwal, D. A. 2001. *Pervasive Collaborative Computing Environment*. Tech. rept. Lawrence Berkeley National Laboratory. 49
- [Alfieri2003] Alfieri. 2003 (fév.). VOMS : an Authorization System for Virtual Organizations. *In : 1st European Across Grids Conference*. 60
- [Allison et al.2004] Allison, Colin, Cerri, Stefano A., Gaeta, Matteo, & Ritrovato, Pierluigi. 2004 (September). Services, semantics and standards : elements of a learning Grid infrastructure. *Pages 13–23 of : 1st International Workshop on Grid Learning Services, GLS'04*. 13
- [Allison et al.2005] Allison, Colin, Cerri, Stefano A., Ritrovato, Pierluigi, Gaeta, Angelo, & Gaeta, Matteo. 2005. Services, semantics and standards : elements of a learning Grid infrastructure. *Applied Artificial Intelligence, Special issue on Learning Grid Services*, **19**(9-10), 861–879. 13
- [Applegate1991] Applegate, L. M. 1991. Cooperative Work : A Framework for Studying Introduction and Assimilation in Organizations. *Journal Organizational Computing*, **1**(3), 11–39. 30
- [Arnaud2002] Arnaud, J. 2002. Rate equation theory of sub-Poissonian laser light. *Pages 393–410 of : Opt. Quantum Electron.*, vol. 34. 92
- [Arnaud & Esteban1990] Arnaud, J., & Esteban, M. 1990. Circuit theory of laser diode modulation and noise. *Pages 55–63 of : IEE Proc. J.*, vol. 137. 92
- [Authier & Lévy1992] Authier, Michel, & Lévy, Pierre. 1992. *Les arbres de connaissances*. Points Science. Paris : La Découverte. 26
- [Bateson1977] Bateson, Gregory. 1977. *Vers une écologie de l'esprit*. Seuil. 23
- [Beaudouin-Lafon2006] Beaudouin-Lafon, Michel. 2006. Human-computer interaction. *In : Goldin, Smolka, Wegner eds. (ed), Interactive computation, the new paradigm*. Springer. 18
- [Berasaluce et al.2004] Berasaluce, Sandra, Laurenço, Claude, Napoli, Amedeo, & Niel, Gilles. 2004. *Resyn Assistant, An Experiment on Knowledge Discovery in Chemical Databases*. Vol. 3202. Springer Berlin-Heidelberg. Pages 39–51. 91
- [Berger & Luckmann1986] Berger, P., & Luckmann, T. 1986. *La construction sociale de la réalité*. Paris : Méridiens Klingsieck. 25
- [Berners-Lee2001] Berners-Lee, Tim. 2001. The Semantic Web Revisited. 12
- [Berners-Lee et al.2001] Berners-Lee, Tim, Hendler, James, & Lassila, Ora. 2001. The Semantic Web. *Scientific American*, May. 12
- [Bonabeau1994] Bonabeau, E. 1994. *L'intelligence collective*. Paris : Hermes Sciences Publicat. Collection Systèmes complexes. 18
- [Bonabeau et al.2000] Bonabeau, E., Dorigo, M., & Theraulaz, G. 2000. Inspiration for optimization from social insect behaviour. *Nature*, **406**(juillet), 39–42. 18
- [Booth et al.2004] Booth, David, Haas, Hugo, McCabe, Francis, Newcomer, Eric, Champion, Michael, Ferris, Chris, & Orchard, David. 2004 (February). *Web services architecture*. W3C working group note NOTE-ws-arch-20040211. World Wide Web Consortium. www.w3.org/TR/2004/NOTE-ws-arch-20040211/. 12, 50

- [Borghoff & Schlichter2000] Borghoff, Uwe M., & Schlichter, Johann H. 2000. *Computer-Supported Cooperative Work : Introduction to Distributed Applications*. Springer-Verlag New York, Inc. 30
- [Bote-Lorenzo et al.2004] Bote-Lorenzo, M. L., Hernández-Leo, D., Dimitriadis, Y. A., Asensio-Pérez, J. I., Gómez-Sánchez, E., Vega-Gorgojo, G., & Vaquero-González, L. M. 2004. Towards reusability and tailorability in collaborative learning systems using IMS-LD and Grid services. *Advanced Technology for Learning*, **1**(3), 129–138. 13
- [Breuker et al.2005] Breuker, Joost, Cerri, Stefano, Dugénie, Pascal, Eisenstadt, Marc, & Lemoisson, Philippe. 2005. *Conceptual and Organisational Framework for Conversational and Collaboration Processes (D20)*. Tech. rept. Projet ELeGI. 31
- [Buhler & Vidal2003] Buhler, Paul A., & Vidal, José M. 2003. Semantic Web services as agent behaviors. *Pages 25–31 of : Burg, B., Dale, J., Finin, T., Nakashima, H., Padgham, L., Sierra, C., & Willmott, S. (eds), 2nd International Agentcities Workshop on Challenges in Open Agent Environments*. Lecture Note in Artificial Intelligence. Melbourne, Australia : Springer-Verlag. 12
- [Buhler & Vidal2004] Buhler, Paul A., & Vidal, José M. 2004 (July). Integrating agent services into BPEL4WS defined workflows. *In : 4th International Workshop on Web-Oriented Software Technologies, IWWOST'04*. 12
- [Buhler et al.2003] Buhler, Paul A., Vidal, José M., & Verhagen, Harko. 2003. Adaptive workflow = Web services + agents. *Pages 131–137 of : 1st International Conference on Web Services, ICWS'03*. Las Vegas, NV, USA : CSREA Press. 12
- [CAurl] CA, Certification Authority. url. *Autorité de Certification Grid Canada*. www.gridcanada.ca/ca. 58
- [Cabral et al.2004] Cabral, Liliana, Domingue, John, Motta, Enrico, Paynec, Terry, & Hakimpour, Farshad. 2004 (May). Approaches to Semantic Web services : an overview and comparisons. *In : 1st European Semantic Web Symposium, ESWS'04*. 12
- [Calderoni2002] Calderoni, Stéphane. 2002. *Ethologie artificielle et contrôle auto-adaptatif dans les systèmes d'agents réactifs*. Ph.D. thesis, Université de La Réunion, Ecole Doctorale de Mathématiques et d'Informatique de Marseille. 23
- [Cannataro & Talia2003] Cannataro, Mario, & Talia, Domenico. 2003. The Knowledge Grid. *Communications of the ACM*, **46**(1), 89–93. 13
- [Cannon1932] Cannon, Walter B. 1932. *The wisdom of the body*. New York : Norton Co. 23
- [Cappello et al.2006] Cappello, Franck, Caron, Eddy, Daydé, Michel, Desprez, Frédéric, Jeannot, Emmanuel, Jégou, Yvon, Lantéri, Stephane, Leduc, Julien, Melab, Noredine, Mornet, Guillaume, Namyst, Raymond, Primet, Pascale, Quetier, Benjamin, Richard, Olivier, Talbi, El-Ghazali, Iréa, Touche, & Bolze, Raphaël. 2006. Grid'5000 : a large scale and highly reconfigurable experimental Grid test-bed. *International Journal of High Performance Computing Applications*, **20**(4), 481–494. 48
- [CCITT1988] CCITT. 1988. *Recommandation X509*. The directory authentication framework. 58
- [Cerri1996] Cerri, Stefano A. 1996 (October). Cognitive Environments in the STROBE model. *Pages 254–260 of : Brna, P., Paiva, A., & Self, J. (eds), European Conference in Artificial Intelligence and Education, EuroAIED'96*. 20
- [Cerri1999] Cerri, Stefano A. 1999. Shifting the focus from control to communication : the STReam Objects Environments model of communicating agents. *Pages 74–101 of : Padget, J.A. (ed), Collaboration between Human and Artificial Societies, Coordination and Agent-Based Distributed Computing*. Lecture Note in Artificial Intelligence, vol. 1624. Berlin, Germany : Springer-Verlag. 20
- [Cerri2003] Cerri, Stefano A. 2003. Open learning service scenarios on Grids. *In : 3rd LeGE-WG International Workshop : Grid Infrastructure to Support Future Technology Enhanced Learning*. Berlin, Germany : Electronic Workshops in Computing. 13

-
- [Cerri *et al.*2006] Cerri, Stefano A., Crubézy, Monica, Dugénie, Pascal, Jonquet, Clément, & Lemoisson, Philippe. 2006. The Grid Shared Desktop for CSCL. *Pages 1493–1499 of : eChallenges 2006*. Barcelona : IOS-Press. hal-lirmm.ccsd.cnrs.fr/lirmm-00128314. 15
- [Chusseau & Arnaud2002] Chusseau, L., & Arnaud, J. 2002. Monte-Carlo simulation of laser diodes sub-poissonian light generation. *Pages 1007–1023 of : Opt. Quant. Electron.*, vol. 34. 92
- [Chusseau *et al.*2003] Chusseau, L., Arnaud, J., & Philippe, F. 2003. Rate-equation approach to laser light statistics. *Pages 746–754 of : Opt. Spectroscopy*, vol. 94. 92
- [Collabvisurl] Collabvis. url. *Projet Collabvis*. www.westgrid.ca/search/node/collabvis. 48
- [Condorurl] Condor. url. *Projet Condor*. www.cs.wisc.edu/condor/. 45
- [CoreGridurl] CoreGrid. url. *Projet Core Grid*. www.coregrid.net. 48
- [CSIESR2001] CSIESR. 2001. Livre Blanc de l’informatique opérationnelle. *Comité des Services Informatiques de l’Enseignement Supérieur et de la Recherche, Conseil de l’Enseignement Supérieur et de la Recherche*. 44
- [Czajkowski *et al.*1998] Czajkowski, Karl, Foster, Ian, Karonis, Nicholas, Kesselman, Carl, Martin, Stuart, Smith, Warren, & Tuecke, Steven. 1998. A resource management architecture for metacomputing systems. *Pages 62–82 of : 4th Workshop on Job Scheduling Strategies for Parallel Processing, IPPS/SPDP’98*. Lecture Notes In Computer Science, vol. 1459. Orlando, FL, USA : Springer-Verlag. 50
- [Czajkowski *et al.*2004a] Czajkowski, Karl, Ferguson, Donald F., Foster, Ian, Frey, Jeffrey, Graham, Steve, Maguire, Tom, Snelling, David, & Tuecke, Steve. 2004a (May). *From Open Grid Services Infrastructure to WS-Resource Framework : refactoring and evolution*. Whitepaper Ver. 1.0. The Globus Alliance. 50
- [Czajkowski *et al.*2004b] Czajkowski, Karl, Ferguson, Donald F., Foster, Ian, Frey, Jeffrey, Graham, Steve, Sedukhin, Igor, Snelling, David, Tuecke, Steve, & Vambenepe, William. 2004b (May). *The WS-Resource Framework*. Whitepaper Ver. 1.0. The Globus Alliance. 50
- [DataGridurl] DataGrid. url. *Projet EDG, the European Data Grid*. eu-datagrid.web.cern.ch. 46, 60
- [DataTagurl] DataTag. url. *Projet DataTag*. datatag.web.cern.ch. 60
- [David2001] David, Bertrand. 2001. Interface Homme-Machine pour les collecticiels. *Réseaux et Systèmes Répartis*, **13**(novembre), 169–206. 32
- [David & Chalon2005] David, Bertrand, & Chalon, René. 2005 (juillet). Model-Driven Development of Collaborative Tools. *Pages 22–27 of : 11th International Conference on Human-Computer Interaction (HCI 2005)*. 21
- [David *et al.*2003a] David, Bertrand, Chalon, René, Vaisman, & Delotte. 2003a. Capillary CSCW. *Human-Computer Interaction Theory and Practice*, **2**, 879–883. 31
- [David *et al.*2003b] David, Bertrand, Chalon, René, Delotte, Ros, & Boutros. 2003b (octobre). Travail coopératif capillaire en dépannage, maintenance et interventions de crise. *In : Stephanidis C., Jacko J. (ed), 5ème Congrès International de Génie Industriel*. 31
- [de Rosnay1975] de Rosnay, Joël. 1975. *Le Macroscopie*. Seuil. 12
- [de Rosnay1988] de Rosnay, Joël. 1988. *Le Cerveau Planétaire*. Seuil. 12
- [de Rosnay2005] de Rosnay, Joël. 2005. *Joël de Rosnay, entretien de décembre 2005*. csiweb2.cite-sciences.fr/derosnay/articles/Chat Internaute 1206.html. 12
- [De Roure *et al.*2001] De Roure, David, Jennings, Nicholas R., & Shadbolt, Nigel. 2001 (June). *Research agenda for the Semantic Grid : a future e-science infrastructure*. Tech. rept. University of Southampton, UK. Report commissioned for EPSRC/DTI Core e-Science Programme. 13

- [De Roure *et al.*2005] De Roure, David, Jennings, Nicholas R., & Shadbolt, Nigel R. 2005. The Semantic Grid : past, present, and future. *Proceedings of the IEEE*, **93**(3), 669–681. 13
- [Denis *et al.*2003] Denis, Alexandre, Pérez, Christian, Priol, Thierry, & Ribes, André. 2003. *Process Coordination and Ubiquitous Computing*. CRC Press. Chap. Programming the Grid with Distributed Objects, pages 133–148. 65
- [Descartes1996] Descartes, René. 1996. *Discours de la méthode*. Paris : Classique Bordas. 22
- [Dugénie2005] Dugénie, Pascal. 2005. Orientation et usage de l'architecture de services Grille OGSA. *Pages 283–290 of : Vautier, Alexandre, & Saget, Sylvie (eds), Manifestation des Jeunes Chercheurs francophones dans les domaines des STIC, MajecSTIC 2005*. IRISA - IETR - LTSI. hal.inria.fr/inria-00000680. 14, 56, 61
- [Dugénie & Cerri2006] Dugénie, Pascal, & Cerri, Stefano A. 2006. Le Collabureau : un espace de collaboration à la puissance Grid. *In : Technologies de l'Information et de la Communication pour l'Education*. TICE'06. hal-lirmm.ccsd.cnrs.fr/lirmm-00113818. 15, 95
- [Dugénie & Lemoisson2005] Dugénie, Pascal, & Lemoisson, Philippe. 2005. A bootstrapping scenario for eliciting CSCL services within a GRID virtual community. *In : 1st International ELeGI Conference on Advanced Technology for Enhanced Learning*. www.bcs.org/server.php?show=ConWebDoc.3856. 14, 31
- [Dugénie *et al.*2002] Dugénie, Pascal, Munro, Alistair T., & Barton, Mike H. 2002. Toward Assessing Subjective Quality of Service of Conversational Mobile Multimedia Applications Delivered Over the Internet : A Methodology Study. *IEEE Transactions on Multimedia*, **(4)**, 59–67. 88
- [Dugénie *et al.*2006] Dugénie, Pascal, Lemoisson, Philippe, Jonquet, Clément, Crubézy, Monica, & Laurenço, Claude. 2006. The Grid Shared Desktop : a bootstrapping environment for collaboration. *Pages 241–249 of : Advanced Technology for Learning (ATL), Special issue on Collaborative Learning*, vol. 3. 15
- [Duvert *et al.*2006] Duvert, Frédéric, Jonquet, Clément, Dugénie, Pascal, & Cerri, Stefano A. 2006 (novembre). Agent-Grid Integration Ontology. *Pages 136–146 of : Meersman, R., Tari, Z., & Herrero, P. (eds), International Workshop on Agents, Web Services and Ontologies Merging, AWeSOME'06*. Lecture Notes in Computer Science, vol. 4277, no. 1. hal-lirmm.ccsd.cnrs.fr/lirmm-00110569. 62
- [Décryphonurl] Décryphon. url. *Projet Décryphon*. www.decryphon.fr. 46
- [Egeeur] Egee. url. *Projet Egee*. www.eu-egee.org. 46
- [Eisenstadt & Dzbor2002] Eisenstadt, Marc, & Dzbor, Martin. 2002 (June). BuddySpace : Enhanced Presence management for collaborative learning, working, gaming and beyond. *In : JabberConf Europe*. 89
- [ELeGIurl] ELeGI. url. *The European Learning Grid Infrastructure (EleGI)*. www.elegi.org. 49
- [Ellis1995] Ellis, Clarence A. 1995 (mars). A Workflow Architecture to Support Dynamic Change. *Pages 23–30 of : Workshop on Distributed Systems, Multimedia, and Infrastructure*. 19
- [Encoraurl] Encora. url. *Guide des Environnement Numériques de Travail*. www.univ-lyon2.fr. 30
- [Enticeurl] Entice. url. *Projet Entice*. recherche.univ-montp3.fr/mambo/entice. 30
- [EuroGridurl] EuroGrid. url. *Projet Eurogrid*. www.eurogrid.org. 47
- [Ferber1994] Ferber, Jacques. 1994. la Kénétique : des systèmes multi-agents à une science de l'interaction. *Revue internationale de systémique*, **8**(1), 13–27. 18
- [Ferber1995] Ferber, Jacques. 1995. *Les Systemes Multi-Agents, vers une intelligence collective*. Paris, France : InterEditions. 18, 62

-
- [Ferber *et al.*2003] Ferber, Jacques, Gutknecht, Olivier, & Michel, Fabien. 2003. From agents to organizations : an organizational view of multi-agent systems. *Pages 214–230 of : Giorgini, P., Müller, J. P., & Odell, J. (eds), 4th International Workshop on Agent-Oriented Software Engineering, AOSE'03. Lecture Notes in Computer Science, vol. 2935. Melbourne, Australia : Springer-Verlag. 69*
- [FIPA2002] FIPA. 2002 (December). *FIPA contract net interaction protocol specification*. FIPA specifications SC00029H. Foundation for Intelligent Physical Agents. www.fipa.org/specs/fipa00029. 12
- [Foster2002] Foster, Ian. 2002. What is the Grid ? A Three Point Checklist. *GRID Today*. 44
- [Foster & Kesselman1999a] Foster, Ian, & Kesselman, Carl. 1999a. The Globus project : a status report. *Future Generation Computer Systems*, **15**(5-6), 607–621. 50
- [Foster & Kesselman1999b] Foster, Ian, & Kesselman, Carl (eds). 1999b. *The Grid : blueprint for a new computing infrastructure*. San Francisco, CA, USA : Morgan Kaufmann. 12, 50
- [Foster & Kesselman2003] Foster, Ian, & Kesselman, Carl (eds). 2003. *The Grid 2 : blueprint for a new computing infrastructure*. San Francisco, CA, USA : Morgan Kaufmann. 61
- [Foster *et al.*1998] Foster, Ian, Kesselman, Carl, Tsudik, Gene, & Tuecke, Steven. 1998. A security architecture for computational Grids. *Pages 83–92 of : 5th ACM conference on Computer and communications security*. San Francisco, CA, USA : ACM Press. 50, 57
- [Foster *et al.*1999] Foster, Ian, Insley, Joseph, Kesselman, Carl, von Laszewski, Gregor, & Thiebaux, Marcus. 1999. Distance Visualization : Data Exploration on the Grid. *IEEE Computer Magazine*, **32**(12), 36–43. 48
- [Foster *et al.*2001] Foster, Ian, Kesselman, Carl, & Tuecke, Steve. 2001. The anatomy of the Grid : enabling scalable virtual organizations. *Supercomputer Applications*, **15**(3), 200–222. 59, 61
- [Foster *et al.*2002] Foster, Ian, Kesselman, Carl, Nick, Jeff, & Tuecke, Steve. 2002 (June). The physiology of the Grid : an Open Grid Services Architecture for distributed systems integration. *In : Open Grid Service Infrastructure WG, Global Grid Forum*. The Globus Alliance. 50, 61
- [Foster *et al.*2004a] Foster, Ian, Jennings, Nicholas R., & Kesselman, Carl. 2004a (July). Brain meets brawn : why Grid and agents need each other. *Pages 8–15 of : 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS'04, vol. 1. 62*
- [Foster *et al.*2004b] Foster, Ian, Frey, Jeffrey, Graham, Steve, Tuecke, Steve, Czajkowski, Karl, Ferguson, Donald F., Leymann, Frank, Nally, Martin, Sedukhin, Igor, Snelling, David, Storey, Tony, Vambenepe, William, & Weerawarana, Sanjiva. 2004b (May). *Modeling stateful resources with Web services*. Whitepaper Ver. 1.1. The Globus Alliance. 54, 55, 119
- [Foster *et al.*2005] Foster, Ian, Maguire, T., & Snelling, D. 2005. *OGSA-WG recommendation : OGSA WSRF basic profile*. Recommendation 1.0. Open Grid Forum. 50
- [Franck Tarpin-Bernard1999] Franck Tarpin-Bernard, Bertrand T. David. 1999 (Octobre). Ergonomie du Travail coopératif en conception. *In : ERGO-IA'96*. 39
- [G. Aloisio2007] G. Aloisio, M. Cafaro, G. Carteni I. Epicoco S. Fiore D. Lezzi M. Mirto S. Mocarovero. 2007. GRB : The Grid Resource Broker portal. *Concurrency and Computation : Practice and Experience*, **19**(12). dx.doi.org/10.1002/cpe.1131. 65
- [Geldof2004] Geldof, Marije. 2004 (June). *The Semantic Grid : will Semantic Web and Grid go hand in hand ?* Tech. rept. European Commission DG Information Society Unit 'Grid technologies'. 13
- [Globusurl] Globus. url. *Projet Globus*. www.globus.org. 49, 50
- [Goble & De Roure2004] Goble, Carole, & De Roure, David. 2004. The Semantic Grid : myth busting and bridge building. *Pages 1129–1135 of : de Mántaras, R. López, & Saitta, L. (eds), 16th European Conference on Artificial Intelligence, ECAI'04. Valencia, Spain : IOS Press. 13*

- [Goldin *et al.*2002] Goldin, Dina Q., Smolka, Scott A., & Wegner, Peter. 2002. Turing machines, transition systems, and interaction. *Electronic Notes in Theoretical Computer Science*, **52**(1), 120–136. 18
- [Gouaich & Cerri2004] Gouaich, Abdelkader, & Cerri, Stefano A. 2004. Movement and interaction in semantic Grids : dynamic service generation for agents in the MIC* deployment environment. *In : 4th LeGE-WG International Workshop : Towards a European Learning Grid Infrastructure*. Stuttgart, Germany : Electronic Workshops in Computing. 13
- [Gridurl] Grid, Bioinfo. url. *Projet Bioinfo Grid*. www.bioinfogrid.eu. 101
- [Grid5000url] Grid5000. url. *Projet Grid 5000*. www.grid5000.fr. 48
- [GridPhyNurl] GridPhyN. url. *Projet Griphyn*. www.griphyn.org. 47
- [Haber & McNabb1990] Haber, R. B., & McNabb, D. A. 1990. Visualization idioms : A conceptual model for scientific visualization systems. *Visualization in Scientific Computing*, 74–93. 48
- [Hadoopurl] Hadoop. url. *Projet Hadoop Distributed File System*. lucene.apache.org/hadoop. 52
- [IDATE2003] IDATE. 2003. *Environnements Numériques de Travail en Europe (Rapport de synthèse)*. www.dent.caissedesdepots.fr/commun/pdf/publications/Etudes/cartableNum/Synthèse D Dk NL BL.pdf. 30
- [IEMLurl] IEML. url. *Information Economy Meta Language*. www.ieml.org. 12
- [IETFurl] IETF. url. *LDAP RFC 4510-4521*. tools.ietf.org/html/rfc4510. 79
- [IST2006] IST. 2006. *5e Appel à projets, Advanced Grid Technologies, Systems and Services*. cor-dis.europa.eu/ist/workprogramme/wp0506_en/2_5_4.htm. 65
- [iVDGLurl] iVDGL. url. *Projet International Virtual Data Grid Laboratory*. www.ivdgl.org. 47
- [Jeffery1999] Jeffery, Keith. 1999 (September). *Knowledge, Information and Data*. Research report. Central Laboratory of the Research Councils, UK. 13
- [Johnston2004] Johnston, William E. 2004. *DOE Science Grid : Enabling and Deploying the SciDAC Collaboratory Software Environment*. Tech. rept. U.S. Department of Energy Office of Science. 49
- [Jonquet *et al.*2006] Jonquet, Clément, Dugénie, Pascal, & Cerri, Stefano A. 2006. Intégration orientée service des modèles Grid et Multi-Agents. *Pages 271–274 of : Chevrier, V., & Huget, M-P. (eds), 14èmes Journées Francophones sur les Systèmes Multi-Agents, JFSMA'06*. Annecy, France : Hermès. hal-lirmm.ccsd.cnrs.fr/lirmm-00107500. 14, 62, 63
- [Jonquet2006] Jonquet, Clément. 2006. *Dynamic Service Generation : Agent interaction for service exchange on the GRID*. Ph.D. thesis, Université de Montpellier II, Montpellier. 62
- [Jonquet *et al.*2007] Jonquet, Clément, Dugénie, Pascal, & Cerri, Stefano A. 2007. Agent-Grid Integration Language. *Multiagent and Grid Systems, (à paraître)*. hal-lirmm.ccsd.cnrs.fr/lirmm-00139691. 14, 56, 62
- [Ken W. Brodlie & Wood2004] Ken W. Brodlie, David A. Duce, Julian R. Gallop Jeremy P.R.B. Walton, & Wood, Jason D. 2004. Distributed and Collaborative Visualization. *Computer Graphics*, **23**(2), 223–251. 48
- [Latorre *et al.*2006] Latorre, Laurent, Nouet, Pascal, Dugénie, Pascal, & Cerri, Stefano A. 2006. Remote Access to Manufacturing Test Facilities : a Reality in Integrated Circuits, a dream in MEMS. *MST New*, **6**(06), 39–41. hal-lirmm.ccsd.cnrs.fr/lirmm-00128278. 15, 91
- [LCGurl] LCG. url. *Projet Large Hadron Collider Grid*. lcg.web.cern.ch/lcg/. 46
- [Le Moigne1990a] Le Moigne, Jean-Louis. 1990a. *La modélisation des systèmes complexes*. Paris : Dunod. 22
- [Le Moigne1990b] Le Moigne, Jean-Louis. 1990b. *La théorie du système général*. Paris : PUF. 22

-
- [Legionurl] Legion. url. *Projet Legion*. www.cs.virginia.edu/~legion/. 45
- [Lemoisson & Cerri2005] Lemoisson, Philippe, & Cerri, Stefano A. 2005. Interactive Construction of EnCOre (Encyclopédie de Chimie Organique Electronique). *Applied Artificial Intelligence Journal Special issue on Learning Grid Services*, **19**, 933–966. 90
- [Lévy1990] Lévy, Pierre. 1990. *Technologies de l'Intelligence*. Points Science. Paris : La Découverte. 11, 40
- [Lévy1994] Lévy, Pierre. 1994. *L'intelligence collective. Pour une anthropologie du cyberspace*. Paris : La Découverte. 11, 18
- [Levy2007] Levy, Pierre. 2007. L'intelligence possible du XXIe siècle. *Le Monde Diplomatique*, août. 12
- [Linux40TFurl] Linux40TF. url. *Linux va tourner à 40 teraFlops*. www.pcinpact.com/actu/news/Europe_Linux_va_tourner_a_40_teraflops.htm. 46
- [Machado1936] Machado, Antonio. 1936. *Les champs de Castille*. Gallimard. 17
- [Martin et al.2004] Martin, David L., Paolucci, Massimo, McIlraith, Sheila A., Burstein, Mark H., McDermott, Drew V., McGuinness, Deborah L., Parsia, Bijan, Payne, Terry R., Sabou, Marta, Solanki, Monika, & andKatia P. Sycara, Naveen Srinivasan. 2004. Bringing semantics to Web services : The OWL-S approach. *Pages 26–42 of : Cardoso, J., & Sheth, A. P. (eds), 1st International Workshop on Semantic Web Services and Web Process Composition, SWSWPC'04*. Lecture Notes in Computer Science, vol. 3387. San Diego, CA, USA : Springer-Verlag. 12
- [Maturana & Varela1984] Maturana, M.R., & Varela, F.J. 1984. *Autopoiesis and Cognition*. Reidel, Dordrecht. 24
- [McIlraith et al.2001] McIlraith, Sheila, Son, Tran C., & Zeng, Honglei. 2001. Semantic Web services. *Intelligent Systems*, **16**(2), 46–53. 12
- [MD5url] MD5. url. *L'algorithm de hachage Message Digest*. fr.wikipedia.org/wiki/Md5. 58
- [Minsky1989] Minsky, Marvin Lee. 1989. *La société de l'esprit*. Traduction de the society of mind (1985) edn. 11
- [Minsky1991] Minsky, Marvin Lee. 1991. Society of Mind : A Response to Four Reviews. *Artif. Intell.*, **48**(3), 371–396. 11
- [Morin2007] Morin, Christine. 2007 (mai). XtremOS : a Grid Operating System Making your Computer Ready for Participating in Virtual Organizations. *In : IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC)*. 50
- [Morin1991] Morin, Edgar. 1991. *La Méthode, 4 tomes*. Paris : Seuil. Première parution en 1977. 24
- [Morin2005] Morin, Edgar. 2005. *Introduction à la pensée complexe*. Paris : Seuil. 24
- [MPIurl] MPI. url. *Message Passing Interface (MPI)*. www-unix.mcs.anl.gov/mpi. 51
- [Mucchielli & Noy2005] Mucchielli, Alex, & Noy, Claire. 2005. *Approches constructivistes*. Paris : Armand Colin. 26
- [Nkambou et al.2005] Nkambou, Roger, Gouardères, Guy, & Woolf, Beverly P. 2005. Toward learning Grid infrastructures : an overview of research on Grid learning services. *Applied Artificial Intelligence, Special issue on Learning Grid Services*, **19**(9-10), 811–824. 13
- [Olson2001] Olson, Robert. 2001. *Access Grid Hardware Specification*. Tech. rept. Argonne National Laboratory. 101
- [O'reilly2004] O'reilly, Tim. 2004. *What is Web 2.0*. www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. 12
- [Parvin et al.1997] Parvin, B., Taylor, J., Callahan, D. E., Johnston, W., & Dahmen, U. 1997. Visual Servoing for Online Facilities. *IEEE Computer*. 49

- [Pascal1963] Pascal, Blaise. 1963. *Pensées (Oeuvres complètes)*. Paris : Seuil. 22
- [Pearlman et al.2002] Pearlman, Laura, Welch, Von, Foster, Ian, Kesselman, Carl, & Tuecke, Steven. 2002. A Community Authorization Service for group collaboration. *Pages 50–59 of : 3rd International Workshop on Policies for Distributed Systems and Networks, POLICY'02*. Monterey, CA, USA : IEEE Computer Society. 57, 60
- [Piaget1967] Piaget, Jean. 1967. *Logique et connaissance scientifique*. Encyclopédie Pléiade. 24
- [PPDGurl] PPDG. url. *Projet Particle Physics Data Grid (PPDG)*. www.ppdg.org. 47
- [Protégéurl] Protégé. url. *Projet Protégé*. protege.stanford.edu. 91
- [Richardson2005] Richardson, Tristan. 2005 (July). *Remote Frame Buffer protocol*. Tech. rept. Version 3.8. RealVNC Ltd. 48
- [Ritrovato et al.2005] Ritrovato, Pierluigi, Allison, Colin, Cerri, Stefano A., Dimitrakos, Theo, Gaeta, Mateo, & Salerno, Saverio (eds). 2005. *Towards the learning Grid : advances in human learning services*. Frontiers in Artificial Intelligence and Applications, vol. 127. IOS Press. 13
- [RSAurl] RSA. url. *L'algorithme de chiffage de Rivest, Shamir et Adleman*. www.dimgt.com.au/rsa_alg.html. 57
- [S3Ia2002] S3Ia. 2002. *Avant-propos du Schéma Stratégique des Systèmes d'Information et de Télécommunication 2002-2004*. www.education.gouv.fr/syst/schema/default.htm. 30
- [S3Ib2002] S3Ib. 2002. *Schéma Stratégique des Systèmes d'Information et de Télécommunication 2002-2004*. www.education.gouv.fr/syst/schema/s3it.pdf. 30
- [Sassoon1998] Sassoon, Jacques. 1998. *Urbanisation des systèmes d'information*. Paris : Hermes Sciences. 66
- [SDEurl] SDE. url. *Schéma Directeur des ENT*. www.education.gouv.fr/publication/SDET.pdf. 30
- [SDI2002] SDI. 2002. *Schéma Directeur des Infrastructures*. www.education.gouv.fr/publication/sdi.pdf. 30
- [Serres1994] Serres, Michel. 1994. *Citation extraite page 18*. voir référence, Lévy, l'Intelligence collective. 43
- [SetiAtHomeurl] SetiAtHome. url. *Projet seti@home*. www.setiathome.ssl.berkeley.edu. 46
- [Shalf & Bethel2004] Shalf, J., & Bethel, E. W. 2004. The Grid and Future Visualization System Architectures. *IEEE Computer Graphics and Applications*, **23**(2), 6–9. 48
- [Singh & Huhns1999] Singh, Munindar P., & Huhns, Michael N. 1999. Multiagent systems for workflow. *Intelligent Systems in Accounting, Finance and Management*, **8**(2), 105–117. 12
- [Singh & Huhns2005] Singh, Munindar P., & Huhns, Michael N. 2005. *Service-Oriented Computing, Semantics, processes, agents*. John Wiley & Sons. 53, 54, 62, 63
- [SSLurl] SSL. url. *Secure Sockets Layer*. www.openssl.org. 58
- [Sugiuchi et al.n.d.] Sugiuchi, H., Koita, T., & Sato, K. Consideration on the Integration of Ubiquitous and Grid Technologies. In : *Internet and Multimedia Systems and Applications*. Actapress. 65
- [Talia2002] Talia, Domenico. 2002. The open Grid services architecture - where the Grid meets the Web. *Internet Computing*, **6**(6), 67–71. 44
- [TeraGridurl] TeraGrid. url. *Projet TeraGrid*. www.teragrid.org. 46
- [Thevenin1999] Thevenin, Coutaz. 1999. Plasticity of User Interfaces. In : *7th IFIP Interact'99 Human-Computer interaction*. 31
- [TLSurl] TLS. url. *Transport Layer Security*. www.ietf.org/html.charters/tls-charter.html. 58
- [Unicoreurl] Unicore. url. *Projet Unicore*. www.unicore.eu. 45

-
- [Vidal *et al.*2004] Vidal, José M., Buhler, Paul, & Humboldt, Christian Stahl. 2004. Multiagent systems with workflows. *Internet Computing*, **8**(1), 76–82. 12
- [VMWareurl] VMWare. url. *Projet VMWare*. www.vmware.com. 52
- [von Bertalanffy1973] von Bertalanffy, Ludwig. 1973. *Théorie générale des systèmes*. Paris : Bordas. Traduit de l'anglais par J.B Chabrol. 23
- [Vygotsky1978] Vygotsky, L.S. 1978. *Mind in Society*. Harvard University Press, Cambridge. 24, 25
- [W3C2004] W3C. 2004 (February). *OWL Web Ontology Language use cases and requirements*. W3C recommendation. World Wide Web Consortium. 12
- [Watzlawick *et al.*1972] Watzlawick, Paul, Helmick Beavin, Janet, & Jackson, Don. 1972. *Une logique de la communication*. Paris, France : Seuil. 23, 25, 37
- [WCGurl] WCG. url. *World Community Grid*. www.worldcommunitygrid.org. 47
- [Wegner1996] Wegner, Peter. 1996 (October). *The paradigm shift from algorithms to interaction*. Final Draft. 18
- [Wegner1997] Wegner, Peter. 1997. Why interaction is more powerful than algorithms. *Communications of the ACM*, **40**(5), 80–91. 18
- [Wegner & Goldin1999] Wegner, Peter, & Goldin, Dina. 1999 (May). *Interaction, computability, and Church's thesis*. Draft paper. Accepted to the British Computer Journal. 18
- [Wegner & Goldin2003] Wegner, Peter, & Goldin, Dina. 2003. Computation beyond turing machines. *Communications of the ACM*, **46**(4), 100–102. 18
- [Weiser1991] Weiser, Marc. 1991. *The Computer for the 21st Century*. 41
- [Weiser & Brown1995] Weiser, Marc, & Brown, John Seely. 1995. *Designing Calm Technology*. 41
- [Wiener1948] Wiener, Norbert. 1948. *Cybernetics*. Paris : Hermann. 24
- [XMLurl] XML. url. *eXtended Markup Language*. www.w3.org/XML. 12
- [XtreemOSurl] XtreemOS. url. *Projet XtreemOS*. www.xtreemos.org. 50, 51
- [Zhuge2004] Zhuge, Hai. 2004. *The Knowledge Grid*. Singapore : World Scientific. 13

Résumé.

Les réflexions menées autour de l'idée d'*intelligence collective* (ou collaborative) postulent que celle-ci émerge d'un système en interaction dont les dispositifs techniques, sociaux et sémiotiques sont fortement imbriqués (PIERRE LEVY, 1994). Pour relever ce défi, cette thèse propose de conjuguer deux concepts innovants. D'abord celui d'*espace collaboratif ubiquitaire* pour catalyser les interactions entre groupes d'agents et services. Ensuite celui d'*infrastructure à ressources distribuées* pour coordonner, de façon dynamique, les ressources et la sécurité en fonction de critères de qualité de service souhaités par les groupes d'agents. Un *espace collaboratif ubiquitaire* intègre des processus artificiels pour réaliser du traitement intensif (simulations, fouille de données, etc.) et des interfaces multimodales, incluant la visualisation partagée en mode synchrone ou asynchrone, pour les interactions entre humains. En termes d'infrastructure, cette thèse explique pourquoi le WEB n'apporte pas une réponse adaptée à la problématique de l'*intelligence collective* et pourquoi nous nous sommes tournés vers GRID (GRILLE DE RESSOURCES INFORMATIQUES DISTRIBUÉES). Notamment, une propriété singulière de GRID est le service à état qui permet de coordonner des ressources distribuées de façon dynamique et sécurisée. Pour mettre ceci en évidence, nous avons développé le modèle d'architecture AGORA qui propose d'utiliser GRID pour déployer des *espaces collaboratifs ubiquitaires*. De cette façon, les groupes d'agents humains et artificiels s'auto-organisent dans un espace immanent où les ressources sont délivrées dynamiquement par l'infrastructure. Validé par une méthode empirique, ce modèle a été l'objet d'une élaboration itérative fondée sur des retours d'expérimentation avec des communautés d'utilisateurs.

Au vu de l'avancement de nos investigations, une architecture GRID couplée avec des concepts propres aux *systèmes multi-agents* présente les caractéristiques nécessaires pour déployer AGORA à grande échelle avec une utilisation rationnelle des ressources, tout en offrant des garanties de sécurité et de haute disponibilité.

Mots-clés: Grille, architecture orientée-services, ressources distribuées, OGSA, espaces collaboratifs, informatique ubiquitaire, systèmes multi-agents, SMA, intelligence collective

Abstract.

The reflexions lead around the idea of collective (or collaborative) intelligence suggest, that those emerge within a system in interaction of which technical, social and semiotic factors are strongly imbricated (PIERRE LEVY, 1994). In order to leverage this challenge, this thesis propose to mix two innovative concepts. Firstly, the collaborative ubiquitous space is used to catalyse the interactions between groups of agents and services. Secondly, an infrastructure with distributed resources is used to dynamically coordinate the resources and the security according to quality of service criteria. A collaborative ubiquitous space integrates artificial processes to achieve massive computation (simulations, data mining, etc.) and multimodal interfaces including shared visualisation for interactions between humans. In terms of infrastructure, this thesis explains why the Web does not respond suitably to the problem of collective intelligence and why we looked at GRID. In particular, a singular property of GRID is the stateful service that allows to coordinate distributed resources in a dynamic and secured manner. To show it, instead of developing an analytical model, we have chosen a systemic model. The architecture model Agora proposes to use GRID to deploy collaborative ubiquitous spaces. Validated by a empirical method, this model has been evaluated iteratively based on returns from experimentations with communities of users. According to the progress of our investigations, a GRID architecture coupled concepts linked to multi-agents systems show characteristics required for scalability with a rational use of resources, while offering guarantees of security and high availability.

Keywords: GRID, distributed architecture, OGSA, collective intelligence, ubiquitous collaborative spaces, MAS