

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR UNIVERSITY-ANNABA

-

UNIVERSITE BADJI MOKHTAR-
ANNABA-



جامعة باجي مختار
- عنابة -

Faculté : Sciences de l'Ingénieur

Année 2008

Département : Informatique

MEMOIRE

Présentation en vue de l'obtention du diplôme de magister

COLLAGRID: Environnement Collaboratif à base de GRID

Option

Texte Parole et Image

Par

Hafed ZARZOUR

DIRECTEUR DE MEMOIRE: M. SELLAMI

PR

Univ. Annaba

DEVANT LE JURY

PRESIDENT:

N. DOGHMANE

PR

Univ. ANNABA

EXAMINATEURS:

M.T. KHADIR

M.C

Univ. ANNABA

M.K. KHOLLADI

M.C

Univ. CONSTANTINE

N. FARAH

M.C

Univ. ANNABA

ملخص

مع توسع شبكة الانترنت، أشكال جديدة من التعاون أخذت في الظهور بين منظمات المعرفة. في الواقع، بنية خدمة الشبكة المفتوحة (أوكسا) ظهرت على أساس الانتقال من رؤية موجهة نحو النظام إلى رؤية موجهة نحو الخدمة و مبنية على معايير مفتوحة وعلى نحو أدق خدمات ذات حالة. هذه المذكرة تهدف إلى عرض بنية عامة لبيئة تعاونية مرنة على الويب، تقدم إلى أعضاء مجتمع افتراضي حلول متكاملة للعمل على تحقيق أهداف مشتركة بطريقتين لتعاون متزامن و غير متزامن. مستنديين إلى التوجهات الجديدة لبنية خدمة الشبكة (أوكسا) إلى جانب حل مكتب مشترك. مفهوم المكتب المشترك أدخل كبعد جديد للتفاعل بين المشاركين من مختلف المجتمعات الافتراضية للدورات التعاونية وخدمات الشبكة. الشبكة تهدف إلى تحقيق تقاسم مرّن، أمن و منسق للموارد الموزعة جغرافيا وذلك بجعلها متجانسة ومنسقة كما تهدف أيضا لحل تعاوني للمشكلات وهذا ضمن المنظمة الافتراضية. طريقتنا تم تجربتها وفقا لسيناريو خاص بإنشاء تفاعلي لانتولوجيا التوربينات الغازية لاستخدامها في المجال الصناعي.

كلمات مفتاحيه : خدمة ويب؛ خدمة الشبكة؛ أوكسا، بيئة تعاونية على الانترنت، المكتب المشترك، الانتولوجيا المشتركة.

ABSTRACT

With the expansion of the internet network, new forms of collaboration are emerging between the knowledge communities. In fact, Open Grid Service Architecture (OGSA) was born by passing through an orientated vision system towards an orientated vision service based on open standard and more precisely statefull service. This thesis aims to present a generic architecture of a flexible collaborative environment on the Web offering to the members of virtual community solutions of integrated work to achieve common goals in two modes of collaboration: synchronous and asynchronous mode, while being based on the new orientations of Open Grid Service Architecture coupled to a solution of shared desktop. The shared desktop concept is introduced as a new dimension of interaction between the members of different virtual communities of the collaborative sessions and the GRID services instances. GRID aims to enable flexible, secure, coordinated resources distributed geographically while making them homogeneous and coordinated problem cooperative solving in virtual organization. Our approach was validated according to an interactive construction of a shared ontology scenario of the gas turbines in industrial field.

Keywords : Web Service, GRID Service, OGSA, Collaborative environment on the Web, Shared desktop, Shared ontology.

RESUME

Avec l'expansion de l'Internet de nouvelles formes de collaboration tendent à se développer entre communautés du savoir. De fait, l'architecture de service GRID OGSA (Open Grid Service Architecture) est née sur la base d'un passage d'une vision orientée système vers une vision orientée service fondée sur des standards ouverts et plus précisément des services à état. Ce mémoire a pour but de présenter une architecture générique d'un environnement collaboratif flexible sur le Web offrant aux membres d'une communauté virtuelle des solutions de travail intégré pour atteindre des objectifs communs en deux modes de collaboration à savoir le mode synchrone et asynchrone, en s'appuyant sur les nouvelles orientations de l'architecture de service GRID OGSA couplée à une solution de Bureau partagé. Le concept de bureau partagé est introduit comme une nouvelle dimension d'interaction entre les participants de différentes communautés virtuelles des sessions collaboratives et les instances des services GRID. GRID vise à réaliser le partage flexible, sûr, et coordonné des ressources distribuées géographiquement en les rendant homogènes, ainsi que la résolution coopérative des problèmes au sein d'une organisation virtuelle. Notre approche a été validée selon un scénario de construction interactive d'une ontologie partagée sur les turbines à gaz pour l'exploitation en milieu industriel.

Mots clés : Service Web, Service GRID, OGSA, Environnement collaboratif sur le Web, Bureau partagé, Ontologie partagée.

DEDICACES

REMERCIEMENTS

Je tiens tout d'abord à exprimer toute ma reconnaissance et mon profond respect au Prof SELLAMI Mokhtar pour son encadrement, sa confiance, son soutien, et l'opportunité qu'il m'a offerte de réaliser ce travail en collaboration avec le Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, LIRMM. Son soutien et ses précieux conseils m'ont permis de mener ce travail dans de très bonnes conditions

Je voudrais remercier Prof DOGHMANE de m'avoir fait l'honneur de présider mon jury. Je veux également exprimer toute ma gratitude à Dr KHADIR, Dr KHOLLADI et Dr FARAH pour avoir accepté d'être examinateurs de ce mémoire.

J'adresse également mes plus sincères remerciements au Prof CERRI Stefano de m'avoir accueilli en stage avec beaucoup de gentillesse au sein du laboratoire d'informatique, de robotique et de microélectronique de Montpellier. Tout au long de ce stage, j'ai pu bénéficier de ses conseils et j'ai pu apprécier sa disponibilité.

Je tiens à exprimer également mes plus chaleureux remerciements au Dr GOUAICH A. Elkader pour son accueil et la patience qu'il m'a accordé durant le mois de stage au LIRMM

Je tiens à exprimer ma sincère gratitude au Dr KHADIR Mohamed .Tarek pour avoir lu et commenté avec précision la première version de ce mémoire. Ces commentaires m'ont été précieux.

Merci à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

LISTE DES FIGURES

Figure 2.1- Les grands domaines du Travail collaboratif	23
Figure 2.2- Aperçu de l'interface TIXEO	28
Figure 2.3- Aperçu de l'interface WebEx	33
Figure 2.4- Aperçu de l'interface WebCT	34
Figure 2.5- Architecture générale de Sakai	35
Figure 2.6- Présentation de l'interface Moodle.....	36
Figure 3.1-Cycle de vie de service Web.....	49
Figure 3.2- Concepts clés de GRID	53
Figure 3.3- Les couches de l'architecture GRID.....	61
Figure 3.4- Services, APIs, SDKs et Protocoles de la couche collective et ressource.....	66
Figure 3.5- API et Protocole de l'infrastructure GRID.....	67
Figure 3.6-Cycle de vie de Service GRID.....	71
Figure 3.7- Modèle simplifié de l'architecture Service GRID OGSA	72
Figure 3.8- GRID et Service Web : convergence.....	74
Figure 4.1-Architecture Générique de COLLAGRID.....	87
Figure 4.2-Processus de partage des ressources.....	89
Figure 4.3- Modèle d'instanciation dynamique de services à état.....	90
Figure 4.4-Instanciation des Conteneurs de Services Collaboratifs Synchrones et Asynchrones	98
Figure 4.5-Instanciation du bureau partagé pour la communauté virtuelle.....	99
Figure 4.6 - Bureau partagé.....	101
Figure 4.7- Modèle intégré de COLLAGRID	103
Figure 5.1 -Diagramme de fonctionnement de la turbine à gaz à deux arbres.....	108
Figure 5.2- Déroulement du scénario de collaboration pour construire une ontologie partagée	115
Figure 5.3 - Exemple de modélisation de discussion autour de la classe Commutateur par Compendium	118
Figure 5.4- Exemple de définition de la classe Commutateur générée par Compendium de genre HTML.....	119
Figure 5.5- Exemple de représentation de la classe Commutateur en Protégé2000	120

Figure 5.6- Architecture générale de notre système collaboratif	121
Figure 5.7- L'administrateur apprend aux membres de la CVOnto comment utiliser le service Compendium	122
Figure 5.8- Discussion autour de la classe Instrument par le biais du service Compendium partagé	122
Figure 5.9- Exemple de description de la sous-classe Commutateur en Compendium partagé	123
Figure 5.10- L'un des participants invite les membres de la CVOnto à consulter un site des turbines en important un Service URL dans le bureau partagé.....	123
Figure 5.11- Génération de la version HTML de la classe Instrument	124
Figure 5.12- Description de la classe Indicateur en HTML généré par Compendium	124
Figure 5.13- Transformation et Visualisation de l'Ontologie des turbines à gaz en Protégé2000.....	125
Figure 5.14- Exemple de visualisation et validation de la classe Commutateur par les membres de la CVonto	125

LISTE DE TABLEAUX

Tableaux 2.1- Synthèse des principales caractéristiques des environnements collaboratifs....	40
Tableau 3.1- Les acteurs du GRID.....	76
Tableau 3.2 - Cinq classes majeures d'application GRID	78
Tableau 4.1- Représentation des ressources de traitement et de stockage	88
Tableau 4.2- Représentation de la Virtualisation, Réification et des Ressources virtualisées..	89
Tableau 4.3- Représentation des éléments clés de la couche Services collaboratifs	94
Tableau 4.4- Représentation de différentes communautés virtuelles collaboratives	95
Tableau 5.1- Illustration des nœuds utilisés pour modéliser le discours des membres de la CV dans la description de la turbine à gaz.....	117

TABLE DES MATIERES

ملخص.....	ii
ABSTRACT	iii
RESUME.....	iv
DEDICACES	v
REMERCIEMENTS.....	vi
LISTE DES FIGURES	vii
TABLE DES MATIERES	x
Chapitre 1 : Introduction Générale.....	14
1.1. Introduction	15
1.2. Potentialités de GRID.....	15
1.3. Problématique.....	16
1.4. Structure du mémoire	17
Première Partie : Etat de l'art	19
Chapitre 2 : Environnements collaboratifs sur le Web	20
2.1. Introduction	21
2.2. Travail collaboratif.....	22
2.2.1. Définition	22
2.2.2. Domaine pluridisciplinaire	22
2.2.3. Catégorie des systèmes collaboratifs.....	24
2.2.3.1. Les collecticiels	24
2.2.3.2. Le Workflow	24
2.3. Environnements collaboratifs sur le Web	24
2.3.1 Environnements collaboratifs simples	25
2.3.1.1 Colab	25
2.3.1.2 Wiki Wiki Web	25
2.3.1.3 CALiF.....	26
2.3.1.4 Platine.....	26
2.3.2 Environnements collaboratifs intégrés	27
2.3.2.1 TIXEO	27
2.3.2.2 NetDIVE.....	29
2.3.2.3. PlaceWare.....	30
2.3.2.4. WebEx	31

2.3.2.5. Sametime	33
2.3.2.6. WebCT	33
2.3.2.7. Sakai	35
2.3.2.8. Moodle	36
2.4. Bilan et positionnement de notre proposition	39
2.5. Conclusion.....	42
Chapitre 3 : Principes fondamentaux des services Web & GRID.....	43
3.1. Introduction	44
3.2. Architecture Service Web	44
3.2.1. Informatique orientée service	44
3.2.2. Les Service Web.....	45
3.2.3. Architecture des services Web	46
3.2.3.1. Principaux concepts de l'architecture.....	46
3.2.4. Technologies Standards des services Web.....	47
3.2.5. Cycle de vie des services Web	49
3.2.6. Avantages des Services Web.....	50
3.2.7. Limites des services Web	50
3.3. GRID	52
3.3.1. Concepts clés du GRID	52
3.3.1.1. Partage des ressources	52
3.3.1.2. Accès sécurisé	53
3.3.1.3. Utilisation des ressources	54
3.3.1.4. Abolition de la distance.....	55
3.3.1.5. Normes ouvertes.....	56
3.3.2. Qu'est-ce qu'un GRID informatique ?.....	57
3.3.2.1. Couches de GRID.....	59
3.3.3. L'anatomie de GRID	60
3.3.3.1. La couche FABRIC	60
3.3.3.2. La couche connectivité.....	61
3.3.3.3. La couche ressource	62
3.3.3.4. La couche collectivité.....	63
3.3.3.5. La couche application.....	66
3.3.4. Boîte à outils Globus	67
3.3.5. Les services GRID	69
3.3.5.1. Cycle de vie de service GRID	69
3.3.6. L'architecture Service GRID OGSA.....	72
3.3.6.1. Evolution d'OGSA	73
3.3.7. Une méthodologie de mise en place.....	75

3.3.8. Projets et applications GRID	77
3.3.8.1. Applications	77
3.3.8.2. Projets.....	78
3.3.9. Evolution de GRID.....	81
3.3.9.1. GRID sémantique.....	81
3.3.9.2. GRID de connaissances.....	81
3.3.9.3. GRID learning	82
3.4. Conclusion.....	83
Deuxième partie : Contributions	84
Chapitre 4 : Architecture proposée.....	85
4.1. Introduction	86
4.2. Architecture proposée.....	87
4.2.1. Couche Ressources Distribuées	87
4.2.2. Couche Virtualisation.....	88
4.2.2.1. Partage des ressources	89
4.2.3. Couche Services Collaboratifs Synchrones & Asynchrones.....	90
4.2.3.1. Service à état	90
4.2.3.2. Service Collaboratif Synchrone	91
4.2.3.3. Service Collaboratif Asynchrone	91
4.2.4. Couche Services Collaboratifs	92
4.2.4.1. Eléments clés.....	92
4.4 .5. La couche Bureaux partagés.....	99
4.3. Modèle intégré.....	101
4.4. Conclusion.....	105
Chapitre 5 : Expérimentation	106
5.1. Introduction	107
5.2. Turbine à Gaz	107
5.2.1. Description	107
5.2.2. Principe de fonctionnement.....	108
5.3. Ontologie.....	109
5.3.1. Notion.....	109
5.3.2. Contenu d'une ontologie	110
5.3.3. Pourquoi développer une ontologie ?.....	111
5.3.4. Classification des ontologies	112
5.4. Outils utilisés.....	113
5.4.1. Protégé2000.....	113
5.4.2. Compendium	113
5.5. Le Scénario.....	114
5.5.1. Etape Préliminaire	115

5.5.2. Etape de construction de l'ontologie	115
5.6. Conclusion.....	126
Conclusion Générale & Perspectives	127
6.1. Conclusion.....	128
6.2. Perspectives.....	129
Références Bibliographiques.....	131

Chapitre 1

Introduction Générale

1.1. Introduction

L'architecture de service GRID (OGSA) représente une évolution vers une architecture de système GRID basée sur les technologies et les concepts des services Web. Elle est née d'un changement de paradigme dans le monde du génie logiciel, c'est-à-dire d'un passage d'une vision orientée système vers une vision orientée service et plus précisément de service à état. En permettant une utilisation rationnelle des ressources informatiques et en fournissant des capacités de stockage et des puissances de calcul en terme de service à la demande de façon sécurisée. Cette technologie couplée à une solution de bureau partagé ouvre de nouvelles voies au développement des technologies de la collaboration et de la communication.

L'usage de GRID était à ses débuts l'apanage du calcul intensif mais son usage est désormais étendu à n'importe quel type d'application notamment la collaboration entre des participants distants utilisant des services en mode synchrone et asynchrone pour atteindre des buts communs. L'idée du GRID est de proposer un environnement informatique qui interagit avec les membres d'organisation virtuelle pour définir leurs besoins en service, puis pour satisfaire leurs besoins par le biais d'un environnement hétérogène constitué de ressources virtualisées à savoir : bases de données, puissance de traitement, installations spécifiques, etc, pour que le participant à une session de collaboration ait une impression d'environnement collaboratif homogène. L'intérêt croissant que suscite GRID est en partie lié à son ouverture technologique de par la normalisation des services. Ceci a permis à la communauté du logiciel libre d'oeuvrer efficacement à stabiliser un intergiciel de référence Globus, très répandu aujourd'hui pour déployer des services GRID, permettant de mutualiser des ressources informatiques hétérogènes. La propriété d'état dans les services GRID ouvre à OGSA des perspectives nouvelles vers des services collaboratifs comme ceux décrits par le concept GRID sémantique [1].

1.2. Potentialités de GRID

Des recherches récentes [2, 3, 4] favorisent l'usage de l'approche orientée service de la technologie GRID dans tous les contextes, notamment quand la collaboration humaine est exigée (e-learning, e-government, e-health, e-business, e-science). En fait, GRID a beaucoup d'aspects intéressants pour le développement des environnements collaboratifs et peut être motivé par de nombreux facteurs, parmi les quels:

- L'infrastructure GRID offre un environnement sûr, fiable et robuste dans lequel les utilisateurs peuvent importer de nouveaux services et introduire de nouveaux utilisateurs selon leur besoin.
- GRID apporte une nouvelle dimension dans la conception des applications. A travers les réseaux à très haut débit comme Geant au niveau européen, il devient possible d'envisager des applications capables d'exploiter de nombreuses ressources dispersées géographiquement (supercalculateurs, bases de données, instrument scientifiques).
- Usage efficace et bien réparti des ressources physiques en fonction de leur disponibilité, il s'agit de virtualiser les ressources et les restituer sous forme de services afin de pouvoir les assembler et les désassembler en fonction des besoins.
- Collaboration flexible en deux modes synchrone et asynchrone grâce aux services à état.
- 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.
- Création dynamique des environnements d'exécution des services où ils disposent de leurs propres ressources.
- Services accessibles sur le WEB à travers des interfaces et des protocoles normalisés conformes aux standards des services Web.

1.3. Problématique

Dans le contexte actuel, où l'Internet est en pleine expansion, de nouvelles formes de collaboration entre intervenants distants deviennent possibles, ceci a donné naissance à de nouveaux sujets de recherche dans le domaine du travail coopératif assisté par ordinateur (Computer Supported Cooperative Work). Les services développés dans ce contexte ont pour but principal de proposer de nouveaux paradigmes de travail en équipe et de développer de

nouveaux environnements collaboratifs qui s'appuient sur des réseaux informatiques, avec en particulier l'Internet. La nouvelle génération du Web (Web 2.0), lancée en 2004 [5], s'oriente sur le développement d'outils collaboratifs sur le Web. Le Web 2.0 combine, d'un côté, une amélioration des interfaces utilisateurs et de l'autre, des architectures plus flexibles, des protocoles de communication plus ouverts (Services Web), une interopérabilité plus poussée. En fait, le point fort du Web 2.0 n'est pas une réalisation technologique mais repose sur le succès de solutions collaboratives ou pair-à-pair dont Wikipédia symbolise la réussite. [6] prédit qu'avec le Web 2.0 nous assistons à l'explosion des contenus générés par les utilisateurs avec au premier chef les environnements collaboratifs.

Partant de cette constatation, et vu que GRID dispose d'énormes potentialités pour supporter la collaboration sur le Web selon une approche orienté service, il semble important de s'appuyer sur l'architecture des services GRID OGSA pour développer de nouveaux environnements collaboratifs qui permettraient à plusieurs utilisateurs distants de collaborer en ligne, et en particulier de partager des services dont ils auront besoins.

Notre objectif, dans ce travail de magister, est de proposer une architecture générique d'un environnement collaboratif flexible sur le Web offrant aux membres d'une communauté virtuelle des solutions de travail intégré en deux modes de collaboration à savoir le mode synchrone et asynchrone, afin de leur permettre la résolution coopérative des problèmes, en s'appuyant sur les nouvelles orientations de l'architecture de service GRID OGSA couplée à une solution de Bureau partagé accessible à travers un terminal léger sans aucune installation, configuration ou mise à jour préalable d'application.

1.4. Structure du mémoire

Nous avons structuré ce mémoire en deux parties, une introduction et une conclusion générale. La première partie consacrée à l'état de l'art est composée de deux chapitres. Le premier sur les environnements collaboratifs existants sur le Web pour situer et comprendre ce concept et le second sur l'infrastructure GRID afin de mettre en évidence l'apport des propriétés de GRID:

Le chapitre 2 intitulé « Environnements collaboratifs sur le Web », explore le concept du travail collaboratif, et présente une analyse de différentes propositions des environnements

offrant des solutions collaboratives sur le Web. Nous cherchons à identifier les limites des solutions actuelles afin de pouvoir positionner et mettre en évidence notre approche.

Le chapitre 3 intitulé « Principes Fondamentaux des services Web & GRID », dresse un état de l'art relatif au passage technologique de l'architecture des services Web vers les services GRID en mettant l'accent sur l'anatomie de GRID, ainsi que sur les motivations de s'appuyer sur les potentialités offertes par le modèle de l'architecture OGSA dans le contexte des environnements collaboratifs sur le Web. Une dernière section est consacrée aux grands projets et à l'évolution de GRID.

La deuxième partie présentant nos contributions est divisée en deux chapitres :

Le chapitre 4 intitulé « Architecture proposée CollaGRID », présente l'architecture de notre proposition centrée sur un environnement collaboratif orienté GRID destiné à la résolutions coopérative des problèmes sur le Web, exprimé en terme de couche dont les aspects conceptuels reposent sur les concepts et les mécanismes clés de l'architecture de Service GRID OGSA (Service GRID, Conteneur de services, Organisation virtuelle, Virtualisation et Réification, etc.) couplée à une puissance d'interaction fournie par le concept du bureau partagé pour permettre aux membres d'une communauté virtuelle d'accéder à travers un terminal léger aux instances des services collaboratifs et de visualiser et/ou diffuser leurs bureaux virtuels selon leur besoin.

Le chapitre 5 intitulé « Expérimentation », présente des résultats expérimentaux obtenus à partir d'une mise en oeuvre d'un prototype de notre environnement COLLAGRID selon un scénario de construction collaborative d'ontologie partagée des turbines à gaz au sein d'une communauté virtuelle. Dans un premier temps, nous présentons la description fonctionnelle de la turbine à gaz, ainsi que la notion d'ontologie. Dans un deuxième temps, nous décrivons les étapes de déroulement du scénario autour de deux services collaboratifs Compendium et Protégé 2000.

Première Partie

Etat de l'art

Chapitre 2

Environnements collaboratifs sur le Web

2.1. Introduction

Suite à l'extension des services offerts par Internet, et plus particulièrement par le Web, différents utilisateurs ont commencé à participer au développement de cet immense espace d'information et à l'enrichir par différents types de données, et ce afin d'échanger et de partager tout genre d'information. Nous constatons ainsi une évolution du mode de travail d'une autonomie à une collaboration qui passe d'un état centré sur l'individu à un état plus orienté vers une communauté. Le travail est donc devenu collaboratif, ce qui justifie la notion d'environnement collaboratif et d'interopérabilité de manière efficace des systèmes dispersés et distants. De ce besoin, naît ainsi un domaine d'application très développé qui s'intéresse à la mise en place d'un support informatique permettant à un groupe de personnes de travailler ensemble dans le cadre de projets communs. On parle de Travail Collaboratif Assisté par Ordinateur (Computer Supported Cooperative Work CSCW)). Les systèmes conçus pour ce type de travail s'appellent groupware ou collecticiels [7]. Ce mode de travail cherche à profiter de toutes les possibilités offertes par l'informatique et l'environnement réseau, et d'établir ainsi des nouveaux rapports et processus de travail. Les environnements collaboratifs permettent donc la collaboration en temps réel ou différé de personnes reliées par un ensemble de ressources informatiques.

Dans ce chapitre nous présentons un état de l'art sur les environnements collaboratifs sur le Web, ainsi que les approches à mettre en œuvre pour supporter l'engagement de plusieurs personnes dans un projet commun.

Dans un premier temps, nous présenterons de manière détaillée la notion du travail collaboratif et les grandes catégories des systèmes coopératifs, ainsi que le classement de différentes propositions que nous avons trouvées dans la littérature, et qui, d'une manière ou d'une autre, offrent des solutions pour élargir les possibilités actuelles de collaboration sur le Web.

Dans un deuxième temps, nous présenterons un tableau récapitulatif contenant une synthèse des principales caractéristiques des environnements collaboratifs où nous mettrons en évidence le positionnement de notre proposition.

2.2. Travail collaboratif

2.2.1. Définition

Dans la littérature scientifique consacrée à ce domaine de recherche. Nous avons noté quelques nuances entre « travail coopératif » et « travail collaboratif » : « Le travail coopératif implique une division du travail entre les participants, chaque participant étant responsable d'une partie du problème à résoudre. Dans la collaboration, les participants s'engagent tous dans les mêmes tâches, en se coordonnant, afin de résoudre le problème ensemble». Cette distinction n'est pas toujours pertinente, il n'est pas toujours facile de différencier les deux [8]. Les dictionnaires ne nous aident pas vraiment dans cette tâche : une des définitions pour “collaborer” étant “coopérer”. Nous prenons en conséquence le parti de considérer les termes de coopération et de collaboration comme étant synonymes.

Le travail collaboratif [9], appelé à l'origine travail coopératif (ou Computer Supported Cooperative Work), correspond à l'utilisation d'ordinateurs en réseau pour le travail de groupes d'utilisateurs. Pour clarifier nos propos, deux citations très générales définissent bien le domaine des systèmes collaboratifs: la première est donnée par [10] :

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

Et la deuxième est donnée par [11] :

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

Donc un système collaboratif est un système informatique qui regroupe plusieurs utilisateurs répartis au sein de groupes différents. Ce système facilite la prise en charge d'activités commune aux membres d'un groupe et fournit une interface pour un environnement partagé.

2.2.2. Domaine pluridisciplinaire

Le travail collaboratif est par essence pluridisciplinaire, il est apparu dans les années 90, il est issu principalement de quatre grands domaines, figure 2.1 :

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

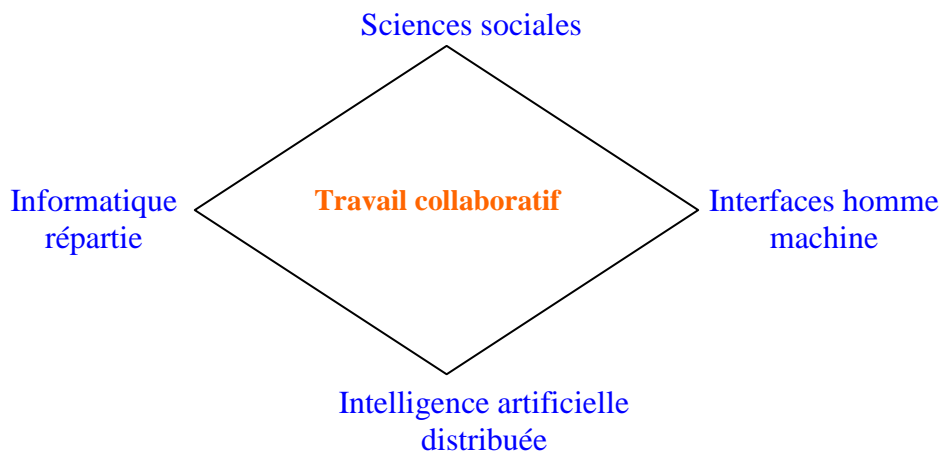


Figure 2.1- Les grands domaines du Travail collaboratif

Le travail collaboratif et les activités de groupe sont organisés en sessions. Une session, unité essentielle du travail en groupe, se compose d'un ensemble d'utilisateurs qui partagent des intérêts communs [12]. Ces utilisateurs se connectent de divers emplacements et manipulent des données, partagées à l'aide des outils support de la collaboration [13].

Une session peut être synchrone ou asynchrone selon la présence ou l'absence des membres du groupe durant la collaboration. Les membres distribués géographiquement peuvent interagir en même temps et en direct (les actions d'un participant sont immédiatement transmises aux autres) ou en temps différé, ce qui définit la collaboration synchrone ou asynchrone.

2.2.3. Catégorie des systèmes collaboratifs

Les systèmes collaboratifs actuels sont principalement répartis suivant deux catégories : les collecticiels et les systèmes de Workflows [7].

2.2.3.1. Les collecticiels

Le collecticiel (ou groupware) désigne tous les produits logiciels, outils, services ou plates-formes, conçus pour des groupes d'utilisateurs. Le collecticiel représente la technologie qui met en oeuvre cette coopération en exploitant les progrès marqués par les technologies d'information dans le domaine des réseaux, mais aussi dans les domaines du matériel et du logiciel. Toutes les technologies qui mettent en oeuvre la communication et le partage d'information ont aussi une place importante dans cette technologie (groupware) qui peut être définie comme étant tout système aidant les individus à coopérer au sein d'un groupe en vue d'atteindre des objectifs communs [14].

2.2.3.2. Le Workflow

Le workflow s'attache à la définition des rôles et à l'exécution des tâches inhérentes aux activités du processus. Au-delà, le workflow s'étend à la gestion des mécanismes de coordination et de coopération entre les différents acteurs d'un processus. Dans toutes les organisations, la division du travail et le mécanisme de coordination qui lui sont sous-jacents, reposent sur de nombreux procédés structurés et prédéfinis. A cet effet, le workflow recouvre tous les processus de travail prédéfinis pour lesquels il est possible de formaliser des règles de gestion et de coordination. Ces règles sont ensuite implémentées dans un système de gestion de workflow, système qui permet d'améliorer les performances en qualité des produits et services fournis, aussi en réduction des coûts et de délais au niveau de nombreux processus.

2.3. Environnements collaboratifs sur le Web

Nous avons effectué une recherche bibliographique dans le but de prendre connaissance de différentes approches, qui proposent d'une manière ou d'une autre, d'étendre les capacités du Web pour qu'il puisse être utilisé en tant qu'environnement collaboratif. Dans le cadre de cette recherche nous avons identifié plusieurs propositions qui abordent ce problème en fonction de différents points de vue. Nous les avons classées en deux grandes catégories : (i) des environnements offrant des solutions collaborative spécifiques dits Environnements collaboratifs simples et (ii) des environnements offrant des solutions de travail intégrés dits

Environnements collaboratifs intégrés, et qui permettent à des utilisateurs de collaborer par différents outils sur le Web.

2.3.1 Environnements collaboratifs simples

Nous présentons dans cette section, quelques uns des projets qui ont été les plus significatifs dans cette catégorie en fonction des solutions bien spécifiques. Chaque environnement est constitué généralement d'un seul outil collaboratif à savoir le co-navigateur, le tableau partagé, etc. Nous expliquons aussi leurs caractéristiques et fonctionnalités principales.

2.3.1.1 Colab

Le navigateur collaboratif Colab, développé par G. Hoyos [15], est un environnement prototype basé sur le Web. Le principal intérêt de cet environnement prototype est de proposer une extension paramétrable et plus souple, de la façon dont un groupe d'utilisateurs peut naviguer en commun sur des pages Web. En effet, le mode de co-navigation, implanté par les environnements tels que WebEx [16], force une présentation identique des pages Web chez tous les membres du groupe. Cette présentation est pilotée par un membre privilégié.

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

L'architecture de Colab se compose (i) d'un serveur Web qui joue le rôle de Proxy et (ii) d'un moteur de collaboration. Le serveur Web Proxy se charge de synchroniser l'activité de navigation collaborative. Le moteur de collaboration définit dynamiquement les droits d'accès aux ressources Web, selon des politiques définies sous forme de règles. Les clients de ce navigateur collaboratif sont des applets JAVA qui sont synchronisées par le serveur Web pour visualiser les pages Web chez chacun des participants.

2.3.1.2 Wiki Wiki Web

Ce projet [17] met en place un système coopératif pour créer des pages sur le Web. Cela signifie que n'importe quel utilisateur enregistré peut, depuis n'importe quel endroit du monde, et à n'importe quel moment, ajouter des nouvelles pages Web au système, et il peut

également, s'il dispose des privilèges nécessaires, modifier et éliminer des pages existantes. Ce projet ouvre des possibilités très intéressantes puisqu'il permet de créer de manière coopérative des bases d'information qui sont mis à jour régulièrement et qui sont consultables sur le Web. Ce projet a donné naissance à la Wikipédia [18] qui est une encyclopédie créée dynamiquement par ses propres utilisateurs.

2.3.1.3 CAiF

La plateforme collaborative CAiF Multimédia [19] permet de concevoir des applications coopératives multimédia pour différents domaines sans avoir à se soucier des problèmes liés aux communications des membres coopérants, à la gestion de la cohérence des données et à la diffusion des média continus (vidéo, voix ...). Cette plate-forme repose sur un bus CORBA [20] qui gère les problèmes d'hétérogénéité, d'interopérabilité, de portabilité et d'accès aux ressources. Elle est composée de trois services principaux [21]:

(i) le service Communication de groupe permet aux membres d'une application coopérative de diffuser les informations (uniquement sous forme de média discrets : texte, dessin...) qu'ils produisent. Tous les objets générés et modifiés par un utilisateur sont transportés vers les autres membres du groupe .Il doit permettre d'assurer la cohérence de la topologie du groupe ainsi que l'intégrité et l'ordre des messages. (ii) Le service Cohérence permet de maintenir la cohérence de la mémoire partagée distribuée, c'est à dire d'assurer que tous les objets de l'application coopérative, qui sont répliqués sur tous les sites soient identiques chez chaque utilisateur. (ii) Le service Audio/Vidéo permet de transporter des flux continus (audio, vidéo). Les bus compatibles à la norme CORBA 2 ne gèrent pas efficacement la diffusion de ces média continus. Pour fournir cette fonctionnalité, il a été développé le service Audio/Vidéo de CAiF Multimédia pour la standardisation du transport des flux.

Cette plateforme a été utilisée pour la télé-maintenance (projet PROTEUS) et les applications de télé-médecine (projet de TENECI).

2.3.1.4 Platine

Le tableau blanc partagé de l'environnement prototype Platine [22] est entièrement développé en JAVA. Plusieurs utilisateurs peuvent travailler ensemble au travers de cet outil. Un membre autorisé du groupe affiche des transparents au format GIF ou JPEG. Il peut ajouter des annotations sur le transparent affiché. Les autres membres du groupe, s'ils ont l'autorisation requise, peuvent aussi ajouter d'autres annotations qui sont visibles par

l'ensemble des participants connectés. Les annotations sont échangées sous forme d'événements graphiques qui consomment peu de bande passante pour leur transmission.

2.3.2 Environnements collaboratifs intégrés

Nous présentons maintenant des environnements proposant des solutions intégrées pour gérer la collaboration entre plusieurs utilisateurs raccordés sur le Web. Les deux caractéristiques communes de ces systèmes sont : (i) ils sont constitués de plusieurs composants de communication permettant aux utilisateurs d'interagir entre eux, (ii) ils sont accessibles depuis un simple navigateur Web. Parmi les systèmes les plus représentatifs que nous avons identifiés dans cette catégorie nous pouvons citer TIXEO, NetDive, PlaceWare, WebEx, Sametime, WebCt, Moodle et Sakai, qui sont des produits commerciaux et libres.

2.3.2.1 TIXEO

La société TIXEO [23] est spécialisée dans les solutions informatiques pour la communication, la coopération et l'interaction entre personnes. Elle développe en collaboration avec des centres de recherche en informatique, les logiciels WorkSpace3D et meeting3D qui fournissent des espaces de travail partagés et apporte des solutions performantes sur le marché de la Web conférence et vidéo conférence. WorkSpace3D et meeting3D permettent de communiquer par la voix et la vidéo (sur IP), échanger, visualiser, partager des applications, naviguer sur Internet à plusieurs, modifier des documents, dessiner sur un tableau blanc, etc. très simplement et efficacement. Les solutions de vidéo conférences proposent ces fonctions au travers d'environnements virtuels en 3D entièrement personnalisables. Tous les participants durant la Web conférence sont représentés par un personnage virtuel, voient par les yeux leurs personnages et interagissent avec des applications en 3D. Un aperçu de l'interface de ce système est présenté dans la Figure 2.2.



Figure 2.2- Aperçu de l'interface TIXEO

Les applications de travail coopératif adapté à la Web conférence:

- Présentation de diapositives : Cette application permet d'effectuer des présentations PowerPoint à tous les utilisateurs connectés à la réunion en ligne.
- Partage de fichiers : Cette application permet l'échange et la gestion de tous les documents de l'espace de travail.
- Partage de bureau / Partage d'applications : Il permettent de partager et de rendre accessible n'importe quelle application sur un ordinateur à tous les utilisateurs de l'espace de travail.
- Tableau blanc partagé : Visualiser une image, éditer un schéma est possible grâce au tableau blanc.
- Surfer sur Internet à plusieurs : c'est un navigateur Internet collaboratif qui permet d'effectuer simultanément et à plusieurs des recherches d'informations.
- Éditeur de texte coopératif : Éditer en équipe un même document texte et cela simultanément, permet un gain de productivité important.
- Visualisateur d'objets 3D : Permet de charger un modèle 3D et de le voir sous tous ses angles.

La 3D permet également de créer des salles de réunions virtuelles conviviales et facilement accessibles. Intuitivement on sait « qui » fait « quoi », « où » et « comment », ce qui n'est pas le cas avec les autres logiciels de vidéo conférence. La Web conférence est plus productive et efficace, même si les collaborateurs sont éloignés géographiquement. En général les solutions de réunions par Internet Tixeo sont conseillées pour : les Web conférences, les formations à distance, les travaux collaboratifs, les réunions de travail, les relations clientèle et partenaires et les communications.

2.3.2.2 NetDIVE

Ce système [24] est un environnement collaboratif intégré, destiné à des activités commerciales, qui est composé de quatre applications principales :

- WeMeeting : une messagerie instantanée permettant à des utilisateurs de rester en contact les uns avec les autres, tout en restant vigilant sur l'état de présence (notion exprimée par le terme awareness en anglais).
- eAuditorium : un outil permettant d'établir des facilités de téléconférence via le Web avec jusqu'à 3000 participants. Il permet l'organisation de réunions s'appuyant sur le Web, les utilisateurs étant catégorisés en modérateurs, audience et orateurs en fonction de leur nom et mot de passe. Les échanges entre participants peuvent être réalisés en VoIP, par des messages textuels, par un outil de partage d'applications (navigation Web synchronisée ou présentation synchronisée de transparents PowerPoint) et un outil de tableau blanc partagé. Il est également possible de mettre en place des votes et des enchères.
- CallSite : un système s'appuyant sur le Web capable de permettre à des utilisateurs de communiquer entre eux tout en visitant un site Web. Le principe de fonctionnement est simple : un utilisateur ayant besoin de contacter quelqu'un de l'entreprise dont la page Web est actuellement affichée dans son navigateur, peut cliquer sur un bouton particulier, action par laquelle il entre en contact direct avec la personne désirée.
- SiteSticky : un serveur de chat capable de gérer des communications vocales, des discussions immersives par l'utilisation d'avatars, etc.

2.3.2.3. PlaceWare

Ce système a été récemment racheté par Microsoft et intégré dans l'environnement de travail Office, sous le nom de Microsoft Office Live Meeting [25] (il fait partie intégrante de la version 2003 d'Office). Il s'agit d'un applet qui contient un environnement collaboratif dans lequel il est possible de partager des documents natifs Office, et qui comprend un certain nombre d'outils de communication, tel qu'un tableau blanc partagé et un chat. Quelques unes des caractéristiques principales de cet environnement sont présentées ci-dessous :

- Visionneuse PowerPoint : elle permet à plusieurs utilisateurs de visualiser de manière synchronisée des présentations PowerPoint, y compris les animations qui peuvent être contenues dans ces présentations. L'affichage peut se réaliser en plein écran ou dans une fenêtre dédiée.
- Visionneuse de documents : elle permet à plusieurs utilisateurs de visualiser de manière synchronisée tout type de document imprimable.
- Partage de bureau : il permet de contrôler l'exécution d'applications à distance. Le contrôle du bureau partagé peut être passé dynamiquement d'un utilisateur à l'autre.
- Outils interactifs : des outils tel qu'un gestionnaire de questions permettant à des utilisateurs de visualiser et de répondre à des questions de façon individuelle ou collective. D'autres outils interactifs permettent de gérer des votes en temps réel, des indicateurs d'humeur, des chats, des annotations, des tableaux blancs partagés, etc.
- Intégration avec Outlook et Messenger : Microsoft Office Live Meeting permet la planification de réunions à partir de différents outils Microsoft dont Outlook, Word, Excel, PowerPoint, Project, et Visio, ou par l'intermédiaire d'outils de communication instantanée, tels que Messenger et MSN Messenger ;
- Intégration avec Lotus Notes : il est possible de planifier des réunions par utilisation de Lotus Notes, en vérifiant la disponibilité des participants et en réalisant un suivi des traces de leurs confirmations ;

- Diffusion audio sur Internet : c'est une alternative à l'audio conférence qui permet aux utilisateurs d'envoyer des flux audio sur l'Internet ;
- APIs de Live Meeting : cette API permet d'intégrer Live Meeting à d'autres applications ; Meeting Lobby : c'est un lieu ne nécessitant pas de réservation où il est possible de rencontrer d'autres utilisateurs ;
- Live Meeting Replay : c'est une facilité qui permet d'enregistrer une présentation faite au moyen de Live Meeting pour pouvoir la reproduire ultérieurement ;
- Politiques de sécurité: le chiffrement des mots de passe est une caractéristique primordiale dans ce type de système. Il est également possible de définir des dates d'expiration pour certaines des ressources disponibles.

2.3.2.4. WebEx

WebEx [26] est un outil pour la réalisation de réunions virtuelles sur le Web, qui offre les fonctionnalités suivantes :

- le partage de présentations effectuées dans n'importe quel format ;
- l'échange de fichiers et de documents ;
- le partage de bureau et la prise de contrôle à distance des applications s'exécutant sur un serveur de partage d'applications ;
- des possibilités de communication audio.

Un aperçu de l'interface de WebEx est présenté dans la figure 2.3.

L'architecture de WebEx est construite autour d'un composant principal, auquel il est possible d'ajouter de nouvelles fonctionnalités sous la forme d'extensions. Les composants à la base de WebEx sont ainsi :

- Meeting Center : c'est le composant central de l'architecture de WebEx qui offre, à partir d'un navigateur Web, des facilités de communication audio et vidéo pour de petits groupes de travail ;

- Event Center : c'est le module qui permet l'organisation de présentations pour de plus grandes audiences; il comprend la planification, la promotion, la diffusion de ces présentations et tout ce qui concerne le traitement post-présentation. Parmi les facilités proposées, nous trouvons le partage d'applications, la diffusion de présentations à base de transparents en format PowerPoint, l'intégration d'outils de téléphonie classique, des mécanismes de votes, d'enregistrement et de reproduction de présentations, etc. ;
- Training Center : c'est le composant de WebEx permettant la diffusion « live » ou sur demande de formations. Il dispose des outils nécessaires pour générer une ambiance pédagogique propice à un environnement d'apprentissage, tels que des outils pour des interactions de type question-réponse, intégration avec Outlook et PowerPoint, outils de votes, etc. ;
- Support Center et SMARTtech : c'est un composant qui permet à un représentant du service technique de voir, diagnostiquer et résoudre des problèmes « on-line ». Les fichiers de l'utilisateur à l'origine de la requête de service peuvent être transmis pour une analyse post-mortem ; avec l'autorisation du client, le personnel de service technique peut exécuter des logiciels de diagnostic et installer les corrections nécessaires ;
- Sales Center : c'est le composant de WebEx orientée à la vente directe. Il offre des outils de communication de base, des outils spécialisés, tel qu'un indicateur d'attention du client, et la possibilité d'inviter en ligne des spécialistes pour faire des démonstrations spécifiques.

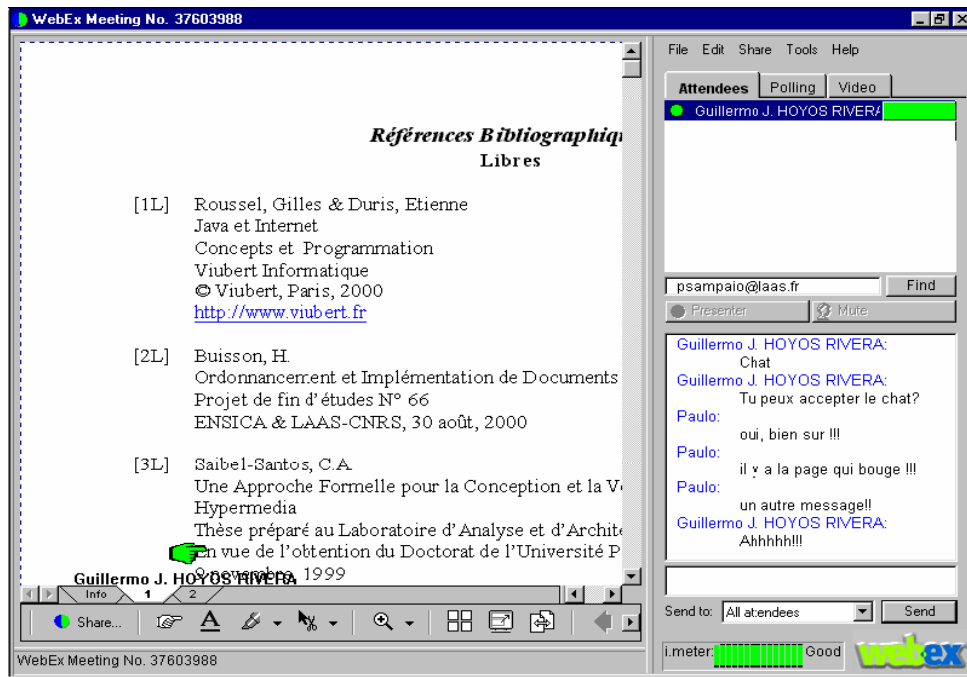


Figure 2.3- Aperçu de l'interface WebEx

2.3.2.5. Sametime

Appelé également IBM Lotus Instant Messaging and Web Conferencing, il permet d'organiser une réunion pour des groupes géographiquement distribués. Il dispose d'une grande diversité de méthodes de collaboration, parmi lesquelles [27] :

- transfert de fichiers et envoi de messages spontanés en temps réel ;
- communication à l'aide des fonctions audio et vidéo de l'ordinateur ;
- modification de fichiers partagés et enregistrement de modifications ;
- envoi de pages Web et de sondages aux participants à une réunion ;
- enregistrement de réunion de façon à ce qu'elle puisse être visionnée ultérieurement par d'autres personnes.

2.3.2.6. WebCT

La plateforme WebCT [28, 29] est un système basé sur Internet qui permet le développement, la gestion, puis la diffusion asynchrone de modules de formation. Le cœur de WebCT s'appuie sur un serveur Web. Le mode auteur, accessible à distance, assiste le concepteur et lui permet d'organiser le contenu de ses cours en pages et en modules. Le contenu de ces modules peut être réalisé par d'autres outils de présentation, comme

Macromedia Dreamweaver, Microsoft FrontPage ou Microsoft Powerpoint. WebCT peut récupérer tout document converti au format HTML.

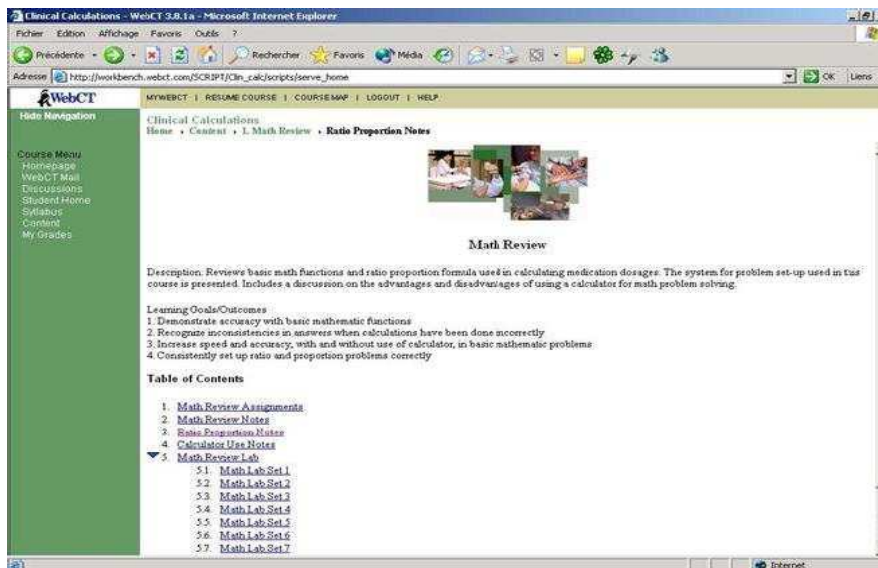


Figure 2.4- Aperçu de l'interface WebCT

Les cours réalisés peuvent contenir :

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

Les utilisateurs distants n'ont pas besoin d'installer de logiciel spécifique pour accéder aux cours : ils utilisent un navigateur standard. Les étudiants enregistrés à certains cours délivrés par WebCT communiquent avec leur instructeur en utilisant la messagerie électronique, des groupes de discussion ou un dialogue textuel. L'évaluation des étudiants se fait au travers de Questionnaires à Choix Multiples et de Devoirs. Les enseignants peuvent aussi recevoir des données qui mesurent l'impact de leur cours. WebCT contient en dernier lieu des fonctions d'administration intégrées, pour le suivi et la gestion des apprenants. L'outil de syllabus décrit la structure de l'enseignement délivré. Un calendrier définit les dates de suivi des cours ainsi que les dates d'examen et de rendu des devoirs. En dernier lieu des outils de gestion de notes et de suivi de l'avancée des étudiants sont aussi disponibles. Les outils de

communication disponible dans WebCT sont : Email, forum de discussion, chat et tableau blanc.

2.3.2.7. Sakai

Sakai [30] est environnement collaboratif Open Source lancé en 2004 par quatre universités américaines avec pour objectif de consolider leurs développements en matière de plateforme d'apprentissage. Chacune de ces universités, soit Indiana University, Massachusetts Institute of Technology (MIT), Stanford University et University of Michigan utilisaient des systèmes de gestion de cours différents, souvent développé à l'interne. Au groupe s'est joints des membres de uPortal ainsi du projet OKI (Open Knowledge Initiative).

Sakai est développé en Java et utilise plusieurs technologies issues de J2EE. Il est construit sur des frameworks et bibliothèques libres tels que Spring, Hibernate, log4j et Jakarta Common. Sakai permet la mise en place d'une architecture orientée service (SOA) en offrant la possibilité d'interconnecter les différents composants à l'aide des services Web.

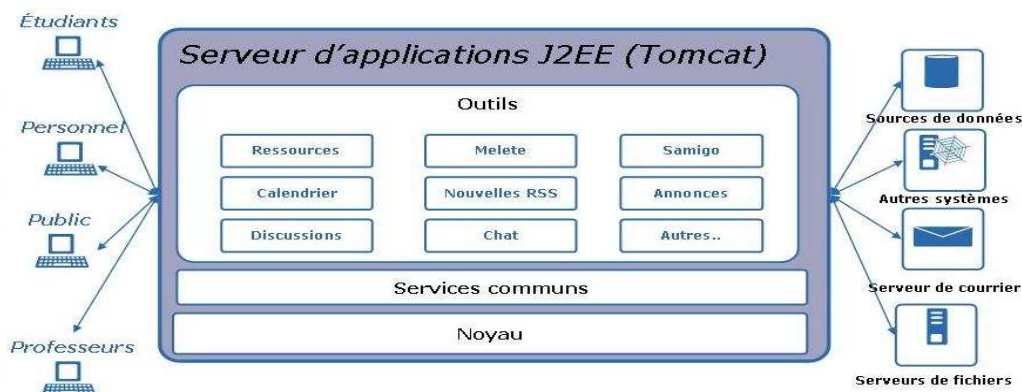


Figure 2.5- Architecture générale de Sakai

Sakai se décompose en trois composants principaux : les outils, les services et le Framework. Le Framework agit comme le point central de l'application. Son rôle est de gérer et de référencer les services et outils. Les services regroupent l'ensemble des opérations relatives à la plateforme. Il s'agit, entre autres, de l'authentification, de la gestion des utilisateurs, de la gestion des contenus, de l'accès aux données, de la gestion des groupes, etc. Les outils correspondent aux fonctionnalités présentées aux utilisateurs (barres de navigation, texte de présentation, forum de discussion, calendrier,...). Seuls les outils génèrent du contenu graphique. Les contenus qu'ils présentent sont accédés via les services. Sakai propose des

outils de collaboration synchrones, tel que les salles de discussion (également appelées chat room), ainsi que des outils de collaboration asynchrones comme par exemple des forums de discussion. Les salles de discussion permettent aux élèves inscrits dans les même cours de discuter entre eux en direct. Le professeur peut également organiser des périodes de consultation pendant lesquelles il sera connecté à la salle de discussion afin répondre aux questions des étudiants. Ces échanges ne sont pas confidentiels et il est possible de consulter les derniers messages sur la page d'accueil du site. Quant aux forums de discussion, ils permettent aux élèves et enseignants d'échanger des messages qui sont consultables par tous les membres du site.

2.3.2.8. Moodle

Moodle [31] est une plateforme d'e-learning sous licence open source, utilise les fonctionnalités de base d'une plateforme de cours en ligne, il constitue un produit simple mais efficace. L'administrateur de la plateforme peut s'occuper de tout : création des comptes utilisateurs, mise en ligne des cours et affectation des professeurs, modération des forums, analyse des logs contenant l'intégralité des actions effectuées au sein de la plateforme. La communauté Moodle particulièrement active, le produit subit beaucoup d'améliorations au fil des versions.



Figure 2.6- Présentation de l'interface Moodle

Voici une sélection des fonctions importantes de Moodle:

Gestion du site :

- Le site est géré par un administrateur, définit durant l'installation.
- Un plug-in "thèmes" permet à l'administrateur de personnaliser les couleurs du site, les polices de caractère, la mise en page, etc. pour répondre aux besoins locaux.
- Un plug-in de modules d'activités peut être ajouté à l'installation existante de Moodle.
- Un plug-in de langages permet une adaptation complète à n'importe quel langage. Ces langages peuvent être édités en utilisant un éditeur Web inclus. Actuellement il existe 28 packs de langage.
- Le code PHP est clairement écrit sous la licence GPL (General Public Licence, logiciel libre).
- Module de forum :
- Différents types de forums sont disponibles, comme par exemple un forum pour professeurs uniquement, les informations sur les cours, ouverts à tous, et un article par utilisateur.
- Tous les postes peuvent avoir la photo de l'auteur attachée.
- Les discussions peuvent être imbriquées, à plat ou en fil, de la plus ancienne à la plus récente ou inversement.
- Chaque personne peut souscrire à des forums individuels et ainsi des copies sont transmises par email, ou le professeur peut forcer la souscription pour tous.
- Le professeur peut choisir de ne pas permettre les réponses (ex : pour un forum d'annonces uniquement).
- Les articles de discussion peuvent être facilement déplacés entre les forums par l'enseignant.

Module de journal :

- Les professeurs peuvent définir une base de données de questions pour une réutilisation dans différents quiz.
- Les questions peuvent être stockées dans des catégories pour un accès simplifié, et ces catégories peuvent être publiées pour être accessibles depuis n'importe quel cours sur le site.
- Les quiz sont automatiquement notés, et peuvent être re-notés si les questions sont modifiées.
- Les quiz peuvent avoir un temps limité d'affichage, au-delà ils ne sont plus accessibles.
- Les quiz peuvent être remplis plusieurs fois, et peuvent montrer des commentaires et/ou les bonnes réponses.
- Les questions peuvent être mélangées au hasard.
- Les questions permettent d'insérer du HTML et des images.
- Les questions peuvent être importées depuis des fichiers textes.
- Les questions à choix multiples peuvent avoir : des réponses à coche, une réponse courte, vrai - faux

Module de ressources :

- Supporte l'affichage de plusieurs logiciels : Word, PowerPoint, Flash, Vidéo, Sons, etc.
- Les fichiers peuvent être téléchargés et gérés sur le serveur ou créés à la volée en utilisant des formulaires Web.
- Un lien extérieur Web peut être inclus.
- Des applications extérieures du Web peuvent être liées avec des paramètres transmis.

Module d'enquête :

- Des rapports de sondages en ligne sont disponibles, incluant des sorties graphiques. Les données sont téléchargeables dans des tableaux Excel ou un fichier format CSV (Comma Separated Values, valeurs séparées par des virgules).
- Les réponses sont fournies aux étudiants comme étant leur résultat comparé à la moyenne de la classe.

D'autres environnements collaboratifs existent que nous ne développerons pas ici comme GatherPlace, BeamYourScreen, Goto Meeting et ViSim, etc.

2.4. Bilan et positionnement de notre proposition

Tous ces environnements que nous venons de présenter ont certains éléments en commun, et d'autres sont spécifiques à la solution du problème posé. Dans le but de donner un aperçu général des capacités techniques de ces environnements et de positionner notre approche, nous présentons, dans le tableau 2.1, une synthèse des principales caractéristiques de chacun de ces éléments selon les principaux critères suivants :

- *Mode de collaboration*: la collaboration s'effectue en mode synchrone (Syn) et/ou asynchrone (Asyn).
- *Outils de collaboration disponibles*: Messagerie instantanée, Tableau blanc, Vidéoconférence, etc.
- *Architecture utilisée*: l'environnement est conçu selon une architecture orientée service (AOSys) ou orientée système (AOS) ;
- *Type de l'environnement* : simple ou intégré;

	Colab	WikiWiki	Calif	Platine	Tixeo	NetDiv	Placeware	WebEx	Sametime	Webct	Moodle	Sakai
Type de l'environnement	Simple	Simple	Simple	Simple	Intégré	Intégré	Intégré	Intégré	Intégré	Intégré	Intégré	Intégré
Mode	Sync	ASyn	Sync	Sync	Sync	Sync	Sync	Sync	Syn/ Asyn	Syn/ Asyn	Syn/ Asyn	Syn/ Asyn
Architecture	AOSys	AOSys	AOSys	AOSys	AOSys	AOSys	AOSys	AOSys	AOSys	AOSys	AOSys	AOS
Vidéoconférence			●		●	●		●	●			
Audioconférence					●	●	●	●	●			
Messagerie instantanée						●	●		●	●	●	●
Tableau blanc partagé				●	●	●	●			●		●
Navigation partagée	●				●	●						
Application partagée					●	●	●	●				
Transfert de fichier					●				●		●	●
WIKI		●										
Messagerie électronique										●		●
forum de discussion										●	●	●
Listes de diffusion												●

Tableaux 2.1- Synthèse des principales caractéristiques des environnements collaboratifs

Ces solutions adoptent plusieurs approches pour offrir trois fonctionnalités principales :

1. Ecran partagé (ou écran diffusé : Desktop Droadcasting) permettant aux utilisateurs de partager leur affichage d'écran sur un réseau pour que d'autres utilisateurs puissent voir exactement le même écran qu'ils voient.
2. Bureau partagé permettant à tous les utilisateurs de prendre contrôle à distance des dispositifs (Souris, Clavier...) à partir de leur bureau. Les fonctionnalités Ecran/Bureau partagé sont traditionnellement utilisées en sessions de travail collaboratif pour des projections de diapositives ou des démonstrations d'application.
3. Environnement commun fournissant à tous les utilisateurs un environnement virtuel qui peut être un ensemble simple de fenêtres ou un espace 3D complexe avec des personnages virtuels pour la représentation des utilisateurs. Les utilisateurs peuvent interagir et travailler ensemble au travers des applications disponibles sur l'environnement commun (Chat, présentation de diapositives, édition de document, navigation partagée, etc.).

Cependant, on constate que, premièrement, aucun de ces environnements ne satisfait l'ensemble des critères définis auparavant. Par exemple : La plupart des environnements fonctionnant uniquement en mode synchrone avec quelques peu d'outils de collaboration insatisfaisants vis-à-vis la flexibilité d'une telle plateforme. Ainsi, ils sont basés sur une architecture client serveur, ce qui limite leurs interactions en simple requête/réponse. Prenons à titre d'exemple l'environnement Sakai, il est conçu sur une architecture orientée service avec quelques outils des deux modes de collaboration. Cependant, des services de conférences synchrone : texte, audio, vidéo et des services de tableau blanc et partage des applications ne sont pas encore disponibles. Deuxièmement, nous pouvons confirmer qu'aucun d'entre eux ne bénéficie de la totalité de potentialités offertes par les Service GRID.

En effet, notre proposition s'intègre dans la catégorie des environnements collaboratifs intégrés en s'appuyant sur l'architecture OGSA couplée à une solution de Bureau partagé, ce qui permet de bénéficier de façon parfaite de l'énorme potentiel de l'infrastructure GRID.

2.5. Conclusion

Parmi les différentes solutions que nous venons d'analyser nous avons identifié différentes approches permettant, d'une manière ou d'une autre, de gérer des interactions entre utilisateurs distants en mode synchrone et/ ou asynchrone sur différentes architectures. L'analyse de l'évolution des grandes classes des environnements collaboratifs nous montre que ces environnements convergent tous vers les technologies basées sur le Web. En effet, leur conception se fait de plus en plus selon une approche orientée service Web. Ils deviennent accessibles soit directement au travers de navigateurs classiques, soit au travers de "plug-ins" qui s'exécutent aussi au travers de navigateurs classiques. Cependant, parmi les solutions existantes, étudiées dans ce chapitre, aucune ne satisfait l'ensemble des contraintes d'utilisation en termes de composition de services, d'autonomie des groupes, et, plus généralement, en termes opérationnels. Au vu de ces limites, nous présenterons également, dans le chapitre suivant, les principes fondamentaux des technologies Service Web/Grid dédié à la collaboration qui vont nous aider à bâtir notre proposition d'environnement.

Chapitre 3

Principes fondamentaux des services Web & GRID

3.1. Introduction

Le terme de “GRID” est apparu au début des années 2000. A l’origine, le GRID était conçu comme un grand nombre d’ordinateurs géographiquement répartis à travers le monde et interconnectés par des réseaux de longue distance où les ressources de calcul et de stockage, qui peuvent être de natures très variées et situées sur des sites différents, étaient fédérées par un intergiciel et partagées en fonction des besoins et à la demande. Récemment, le développement des systèmes distribués s’est vu passer d’architecture orientées systèmes à des Architectures Orientées Services (AOS) facilitant l’interopération et l’intégration de composant distribués tels que des documents, des ressources, des applications et des services. Dernièrement, le GRID a significativement amélioré les AOS en ajoutant au concept de service la gestion de l’état et la dynamique (gestion de la durée de vie) pour une instance de service. Ce sont les services Grid. Le GRID est ainsi la première infrastructure de partage de ressource développée dans une perspective service. En effet, la propriété d’état ouvre à la norme OGSA des perspectives nouvelles vers des services collaboratifs.

Ce chapitre dresse un état de l’art concernant les services Web et GRID et leurs apports au concept de collaboration : tout d’abord, nous précisons la notion de service Web et ses standards, nous discuterons ensuite les motivations de s’appuyer sur les services GRID pour supporter la collaboration au sein d’organisation virtuelle en mettant l’accent sur les concepts clés de l’architecture OGSA. Enfin, nous présenterons les grands projets et l’évolution de GRID.

3.2. Architecture Service Web

3.2.1. Informatique orientée service

La communauté SOC (Service Oriented Computing, en français : Informatique orientée service) propose un ensemble de modèles pour les systèmes informatiques basés sur le concept Service. Ces modèles définissent l’architecture orientée service (AOS). [32] explique les principes et la pratique de Service Oriented Computing, bien que la plupart de ses concepts soient développés dans le contexte des services Web. Il présente les concepts, les architectures, les théories, les techniques, les normes, et l’infrastructure nécessaires pour utiliser des services. Il présente également une vue d’ensemble complète de l’état de l’art des services Web et des disciplines reliées.

3.2.2. Les Service Web

Le terme “services Web” (de l’anglais Web services) est très souvent utilisé à l’heure actuelle et il présente différentes définitions dans la littérature. Une définition plus précise est fournie par W3C (World Wide Web Consortium) [33] qui représente le comité de coordination responsable de l’architecture et de la standardisation des services Web. La définition du W3C est la suivante:

Définition 1: «A Web Service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts, and which supports direct interactions with other software applications using XML-based messages via Internet-based protocols» [33].

Plus informellement, nous pouvons dire que les services Web sont des applications qui définissent un ensemble d’interfaces, s’appuyant sur XML (eXtensible Markup Language [34]), et qui peuvent interagir dynamiquement entre elles et avec d’autres applications grâce à l’échange de messages basés sur XML utilisant les protocoles de transport Internet disponibles.

Définition 2: « Web Services are software applications that can be discovered, described, and accessed based on XML and standard Web protocols over intranets, extranets, and the Internet. »[35]

Définition 3 : « Un service Web est une application accessible à partir du Web. Il utilise les protocoles Internet pour communiquer et utilise un langage standard pour décrire son interface » [36].

Cette définition nous semble la plus générale et la plus complète puisqu’elle souligne les aspects clés qui donnent à toute application en ligne un accès généralisé et permet une interopérabilité systématique.

Ainsi, comme pour le World Wide Web, l’un des principaux facteurs de succès des services Web réside dans l’utilisation d’une pile simple de protocoles s’appuyant sur des normes ouvertes. De plus, les services Web partagent une architecture commune qui a pour fondement le concept d’architecture orientée service (Service-Oriented Architecture ou SOA [37]). Il s’agit de répartir une application en petites unités fonctionnelles ou services. Une

architecture orientée service doit se concentrer sur la façon dont ces services sont décrits et organisés pour supporter leur découverte, leur utilisation et leur dynamique [37].

3.2.3. Architecture des services Web

L'architecture des services Web comme l'architecture du Web sont des instances d'architectures orientées services (SOA : Service Oriented Architecture) [37]. Elle propose une perspective globale sur le développement, la gestion et le fonctionnement des services Web. Le choix d'une architecture SOA entre dans la perspective de transformer le Web en une énorme plateforme de composants faiblement couplés et automatiquement intégrables.

3.2.3.1. Principaux concepts de l'architecture

La définition de l'architecture services Web consiste à mettre en évidence les concepts, les relations entre ces concepts ainsi qu'un ensemble de contraintes qui assurent l'objectif premier des services Web à savoir l'interopérabilité.

Les concepts : Un concept représente un élément qui doit avoir une correspondance dans n'importe quelle réalisation de l'architecture. Le concept désigne l'élément qui devrait être présent dans le contexte du service Web sans pour autant prescrire sa réalisation. [38].

Les principaux concepts intervenant dans l'architecture des services Web sont :

Le fournisseur du service : d'un point de vue conceptuel, il désigne la personne ou l'organisation responsable juridiquement de l'agent logiciel (i.e. le service). D'un point de vue opérationnel, les fournisseurs de services peuvent également désigner le serveur qui héberge les services déployés.

Le client du service : comme pour le fournisseur, il représente une personne ou une organisation, client potentiel des services de l'agent logiciel comme il désigne également, d'un point de vue opérationnel, l'application cliente qui invoque le service.

Le registre des services : il représente la personne ou l'organisation responsable de l'hébergement du dépôt de publication des services. D'un point de vue fonctionnel, il représente l'entité logicielle qui joue le rôle de l'intermédiaire entre les clients et les fournisseurs de services. Le concept registre ou dépôt de service est essentiel dans l'architecture services Web. Il joue un rôle central dans le processus de localisation des besoins et dans l'interopérabilité, car il est supposé fournir aux clients les informations

techniques et sémantiques sur le fonctionnement du service et ceci dans des langages formels (interprétables par les machines).

Le service : le service est une notion abstraite pour désigner les fonctionnalités d'un agent logiciel qui implémente une fonctionnalité du service. Quand on parle de service, il est essentiel de faire la distinction entre le service vu d'un angle sémantique et sa réalisation qui est l'entité logicielle. Un seul service Web, par exemple, peut avoir plusieurs agents qui l'implémentent (utilisant des langages différents).

La description du service : c'est la spécification du service exprimée dans un langage de description interprétable par les machines. Il existe deux niveaux de description de services. Une description technique dans laquelle le service est vu en terme de messages, de formats, de types, de protocoles de transport et d'une adresse physique. Cette description joue le rôle d'un contrat d'interaction entre le client et le service. La deuxième description concerne la sémantique du service. La description sémantique peut avoir une existence formelle comme elle peut prendre la forme d'un accord entre le fournisseur et le client. L'architecture des services Web impose à chaque entité logicielle accessible sur le Web d'éditer sa description afin d'assurer l'interopérabilité entre le client et le service.

Les messages : le message joue un rôle important dans l'architecture de référence, c'est la plus petite unité d'échange de données entre les clients et les services. Le concept message ne reflète aucunement la sémantique de son contenu, seuls la structure et les mécanismes de transport sont considérés dans l'architecture. La structure des messages qui permettent l'invocation des différentes unités fonctionnelles qui composent le service doit figurer dans la description du service.

La ressource : le concept ressource désigne l'identifiant du service. Il constitue un point important dans l'architecture des services Web. Chaque service doit avoir une identification en terme d'adresse, généralement une URL.

3.2.4. Technologies Standards des services Web

Bien que le Web soit constitué de plates-formes totalement hétérogènes où les intérêts des différents acteurs du marché s'entremêlent, ceci ne l'a pas empêché de se développer et d'être universel. Ce succès est dû essentiellement à l'édiction d'un ensemble de standards ouverts, dont les plus connus sont le protocole HTTP(Hypertext Transfer Protocol) et le format MIME(Multipurpose Internet Mail Extension). Ensemble, ils offrent un mécanisme

d'échange de données de toute sorte quelque soit la nature des plates-formes impliquées. Le développement du modèle services Web a suivi la même approche en mettant en place une compagnie de standardisation qui a touché les aspects les plus importants du développement d'un modèle d'intégration d'applications hétérogènes.

Comme pour les architectures par composants, l'infrastructure offre un certain nombre de services qui permettent de gérer l'interaction entre les composants. Les services offerts par l'infrastructure des services Web concernent essentiellement deux aspects fondamentaux : un service de communication qui permet l'échange de données entre les services Web et un ensemble de services techniques destinés à automatiser le processus de localisation et d'invocation des composants [36]. L'originalité de l'infrastructure des services Web consiste à mettre en place ces services en se basant exclusivement sur les protocoles les plus répandus d'Internet.

Les trois technologies Standard des Services Web :

Aujourd'hui, l'infrastructure services Web s'est concrétisée autour de trois spécifications considérées comme des standards [39].

SOAP (Simple Object Access Protocol [40]) – Les services Web communiquent par messages. Ils supposent que SOAP est employé dans les couches basses de la pile de protocoles en isolant le transfert des messages des détails de la couche transport. SOAP est un protocole qui définit la structure des messages échangés par les applications via Internet, chaque message étant un document XML.

WSDL (Web Service Description Language [41]) – WSDL fournit un mode de description de l'interface d'un service Web. Il permet de décrire un contrat de service qui fournit des informations telles que les méthodes du programme, les paramètres, le format des réponses et l'adresse du service.

UDDI (Universal Description Discovery and Integration [42]) – Un registre UDDI représente un annuaire qui contient des informations détaillées concernant les services Web. UDDI définit un protocole d'interrogation et de mise à jour d'un tel annuaire. En quelque sorte, UDDI propose d'automatiser toute la procédure de recherche et de découverte des services Web.

3.2.5. Cycle de vie des services Web

- Chaque service Web est défini par un fournisseur. Le fournisseur de services déploie et publie la description de son service dans des registres en vue d'être localisé par des clients ;
- Les clients localisent leurs besoins en terme de services en effectuant des recherches sur les registres de services Web ;
- Une fois le service localisé, le client extrait sa description du registre ;
- Sur la base des informations définies dans la description du service, le client entreprend une interaction.

La figure 3.1 montre le cycle de vie de service Web autour d'interaction simple entre application cliente et fournisseur de service Web

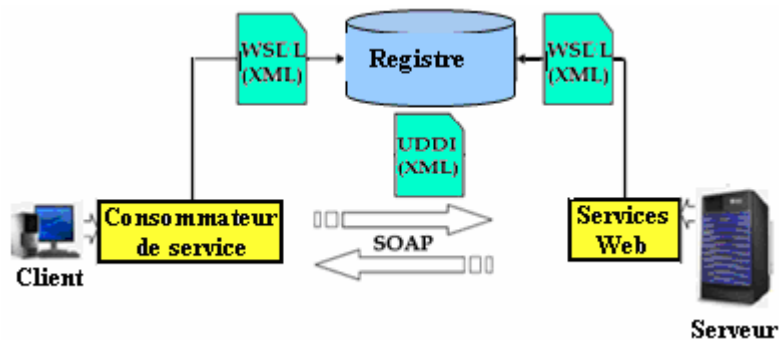


Figure 3.1-Cycle de vie de service Web

- **SOAP** : « assure la communication avec et inter services Web »
- **WSDL** : « offre un schéma formel de description des services Web »
- **UDDI** : « offre une manière uniforme de définir des registres des services Web et aussi un schéma uniformément extensible de descriptions des services Web. »

Actuellement, SOAP, WSDL et UDDI sont les trois standards qui constituent l'architecture Services Web. Ensemble, ils résolvent les problèmes de l'hétérogénéité des systèmes pour l'intégration d'applications en ligne [38].

3.2.6. Avantages des Services Web

Il y a trois caractéristiques qui ont réellement fait la différence et sont en grande partie responsable du succès des services Web [43]:

- Technologie basée sur des standards ouverts (XML) comme décrit dans la section précédente
- Loose Coupling ou le Découplage : Séparation nette entre les interfaces et les implémentations. Une fois qu'un programme est publié en tant que service Web, il est relativement simple de le déplacer grâce au fait que ces services constituent une couche d'abstraction des fonctionnalités logicielles par rapport à l'interface.
- La disponibilité permanente: Utilisation universelle et abondance des outils disponibles. La disponibilité permanente correspond aux capacités des services Web à être actifs et disponibles partout sur un réseau, ou un groupe de réseaux, sans aucuns impacts quand à ses fonctionnalités.

3.2.7. Limites des services Web

Les services Web présentent quelques inconvénients [43, 44]:

- Service sans état (Stateless Service).
- Orientation client/serveur.
- Aucune adaptation à l'utilisateur.
- Aucune mémoire.
- Aucune gestion de vie.
- Aucune conversation (interaction requête/réponse simple).
- Communication synchrone.
- Pas plus que des appels à des procédures à distance passifs.

- Performance critiquable comparée a d'autres approches de systèmes distribués telles que RMI, CORBA ou DCOM. C'est le prix à payer quand on s'appuie sur des fichiers en format texte. Des représentations binaires comme SOAP MTOM promettent d'améliorer l'efficacité des messages en XML.

3.3. GRID

3.3.1. Concepts clés du GRID

Bien sûr, il y a beaucoup de concepts derrière GRID. Et bien sûr, certains étaient dans le vent bien avant que cette appellation n'ait été choisie. Pourtant, si nous examinons ce à quoi les ingénieurs en génie logiciel et les réalisateurs de logiciel consacrent leur temps et leurs efforts, on constate qu'il existe cinq grands domaines, figure 3.2.

3.3.1.1. Partage des ressources

Le premier concepts clé de GRID est le partage des ressources - qui nous permet d'accéder au GRID pour utiliser des ressources distantes, pour faire des choses que nous ne pourrions pas effectuer sur notre propre ordinateur ou au centre informatique avec lequel nous travaillons normalement (si on est, disons, un scientifique faisant des simulations très pointues sur ordinateur). Cette capacité de partage implique plus qu'un simple transfert de fichiers, il requiert un accès direct au logiciel, aux ordinateurs et aux données. Elle peut même permettre d'avoir un accès direct à des capteurs, à des télescopes et à d'autres appareils, distants, qui ne nous appartiennent pas - et de les commander.

Un défi majeur, quant à la mise en œuvre de GRID, est un fait très simple. Les ressources sont la propriété de personnes différentes, ce qui signifie qu'elles relèvent de domaines administratifs différents, qu'elles exécutent des logiciels différents et qu'elles sont régies par des politiques de sécurité et de contrôle d'accès, également différentes. Le système est donc fondé sur la confiance, ainsi que sur des mécanismes de gestion des abus de confiance. C'est le point crucial de la démarche GRID. Pas question d'avoir rien pour rien ou d'offrir au monde des ressources informatique par pure bonté de cœur.

Il s'agit plutôt d'instaurer, au sein des détenteurs de ressources informatiques, une situation dans laquelle tous les acteurs concernés verront un avantage à partager et où des mécanismes seront en place, visant à ce que chaque fournisseur de ressources estime qu'il peut se fier à tout utilisateur à qui un autre fournisseur de ressources fait confiance. Par exemple, quand des personnes responsables d'un centre informatique décident de partager leurs ressources sur GRID, normalement, elles imposent des conditions d'utilisation de ces ressources, spécifiant des limites quant au moment et à l'objet de cette utilisation.

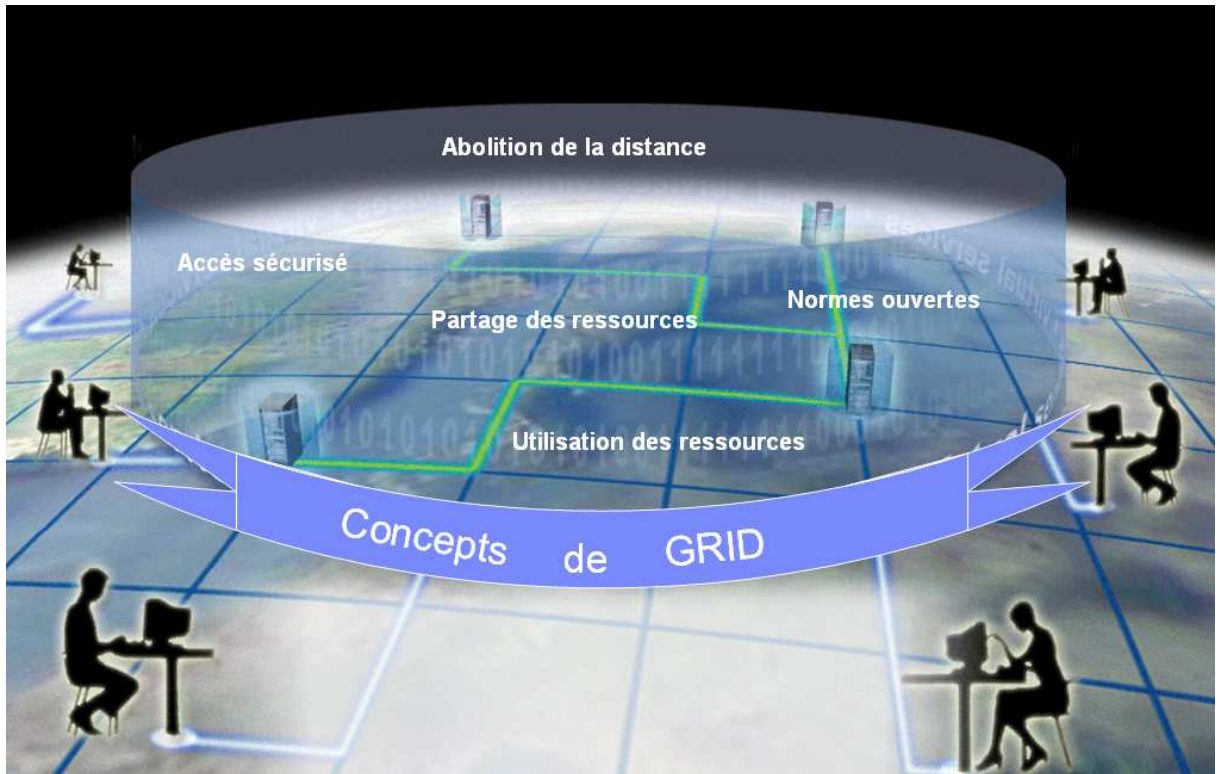


Figure 3.2- Concepts clés de GRID

3.3.1.2. Accès sécurisé

Le deuxième concept clé de GRID est, en bref, l'accès sécurisé - qui est une conséquence directe du premier concept clé. Le partage des ressources se traduit par certains des problèmes les plus difficiles liés à la réalisation de GRID :

La politique concernant l'accès - les fournisseurs et les utilisateurs des ressources doivent définir clairement et soigneusement ce qui est partagé, qui est autorisé à partager et les conditions dans lesquelles le partage a lieu ;

- l'authentification - il faut un mécanisme permettant d'établir l'identité de l'utilisateur d'une ressource ;
- l'autorisation - il faut un mécanisme permettant de déterminer si une opération est conforme aux relations de partage qui ont été définies.

Bien sûr, GRID doit disposer d'un moyen efficace pour garder trace de toutes ces informations. Qui est autorisé à utiliser GRID et quelles sont les ressources que chacune de ces personnes autorisées est autorisée à utiliser ? Qui authentifie un utilisateur donné ? Quelles sont les politiques d'utilisation des différentes ressources ?

Toutes ces choses pouvant changer d'un jour à l'autre, GRID doit être extrêmement souple et disposer d'un mécanisme de comptabilisation. En fin de compte, cette comptabilisation sera utilisée pour décider d'une politique d'établissement des prix d'utilisation de GRID. En sécurité informatique, il est courant de dire que les trois fonctions clés sont l'autorisation, l'authentification et la comptabilisation. C'est certainement vrai pour GRID. La question fondamentale, liée à ces problèmes de confiance, est celle de la sécurité. Nous pouvons nous fier aux autres utilisateurs, mais estimons-nous en toute confiance que nos données et nos applications sont protégées lorsqu'elles circulent sur Internet en traversant d'autres ressources informatiques ou lorsqu'elles sont traitées sur d'autres ordinateurs ?.

Sans une sécurité adéquate, il est aujourd'hui effectivement possible que quelqu'un utilise nos données (confidentielles ou autres) et, éventuellement, les modifie ou modifie leur trajet sur Internet - d'où les mises en garde concernant notre sécurité, chaque fois que nous utilisons notre carte de crédit sur Internet. En outre, sans une sécurité adéquate, il est possible que lorsque nos données résident sur un autre ordinateur de GRID, le propriétaire de cette machine - ou un pirate informatique - réussisse à les lire.

Un gros travail est en cours, pour trouver une solution à tous ces problèmes, qui concernent en réalité tout le spectre des technologies de l'information, et pas seulement GRID. Pour régler la question de la sécurité, par exemple, il est fait appel à des techniques de chiffrement perfectionnées, mises en œuvre lors de la transmission des données, ainsi que lors de leur représentation ou de leur mise en mémoire sur des ressources externes. De nouvelles solutions à beaucoup de problèmes de sécurité, sont mises au point en permanence. Mais c'est une course sans fin, car il faut garder de l'avance sur les malicieux pirates informatiques.

3.3.1.3. Utilisation des ressources

Le troisième concept clé de GRID est que, quand nous aurons réglé toutes les formalités de partage des ressources, il faudra les utiliser de façon efficace. C'est là que GRID commence réellement à être intéressant, même pour les privilégiés qui disposent d'abondantes ressources informatiques. Car, quelle que soit l'abondance de nos ressources, il arrive toujours un moment où se crée une file d'attente d'utilisateurs désireux d'en disposer. S'il existe un mécanisme d'affectation efficace et automatique des travaux à de nombreuses ressources, il permet de réduire les files d'attente.

Sur GRID, en principe, nous disposons de l'information concernant les différents travaux en cours de soumission, et comme tout ce qu'il y a à faire est de les exécuter sur des ordinateurs, nous devrions être capable de calculer l'allocation optimale des ressources. La réalisation de l'intergiciel (en anglais middleware), qui accomplit cette tâche et gère d'une façon générale l'activité sur GRID, est l'objectif principal de beaucoup des projets de GRID en cours partout dans le monde.

3.3.1.4. Abolition de la distance

Le quatrième concept clé de GRID pourrait être appelé l'abolition de la distance. Les connexions à haute vitesse entre ordinateurs rendent possible GRID véritablement mondiale. Il y a dix ans, il eût été stupide d'essayer d'expédier de grandes quantités de données autour du monde pour les faire traiter plus rapidement sur d'autres ressources informatiques, car le délai de transfert de ces données eût annulé l'avantage d'un traitement plus rapide.

Aujourd'hui, le développement impressionnant de la technologie des réseaux rend GRID possible. Ces dernières années, les performances des grands réseaux ont doublé environ tous les neuf mois. Certains grands réseaux fonctionnent maintenant à 155 mégabits par seconde (Mb/s), alors qu'en 1985 les centres de supercalculateurs des États-Unis étaient connectés à 56 kilobits par seconde (Kb/s) - soit une amélioration d'un facteur 3 000 en 15 ans.

Bien sûr, la distance ne sera jamais complètement abolie, parce que quelqu'un aura toujours un problème à traiter sur GRID, pour lequel les connexions les plus rapides sembleront lentes. Par exemple, pour collaborer avec des collègues de différentes parties du monde, à l'analyse de grandes quantités de données, certains scientifiques auront toujours besoin de plus de capacité de connexion à grand débit - allant jusqu'à des dizaines de gigabits par seconde (Gb/s). D'autres demanderont un temps d'attente ultracourt pour leurs applications, en sorte de ne percevoir aucun retard lorsqu'ils travaillent avec des collègues sur GRID.

D'autres, encore, voudront une remise " juste à temps " des données au travers de GRID, afin de pouvoir effectuer des calculs complexes qui nécessitent une communication permanente entre processeurs. Pour éviter les goulets d'étranglement de communication, les réalisateurs de GRID devront aussi trouver des moyens pour compenser toute défaillance se produisant sur celui-ci au cours d'un calcul, qu'il s'agisse d'une erreur de transmission ou d'un blocage d'ordinateur individuel.

Pour satisfaire des exigences aussi critiques, il faut résoudre plusieurs difficultés propres aux réseaux à hautes performances - parmi lesquelles figurent l'optimisation des protocoles de transport et la mise au point de solutions techniques telles que la commutation Ethernet à hautes performances.

3.3.1.5. Normes ouvertes

Le cinquième concept clé de GRID est l'utilisation de normes ouvertes. Il s'agit de convaincre les ingénieurs en génie logiciel qui réalisent actuellement GRID, y compris ceux des principales sociétés d'informatique, d'établir au départ des normes communes pour GRID, de telle sorte que des applications faites pour être exécutées sur GRID puissent l'être sur toutes les autres. Cet objectif peut sembler utopique - après tout beaucoup de sociétés de génie logiciel gagnent de l'argent précisément parce qu'elles ne partagent pas leurs normes avec leurs concurrents. Toutefois, comme de par sa nature GRID vise au partage, il est admis d'une façon générale qu'il est dans l'intérêt de tout un chacun d'établir des normes communes, donc ouvertes. La question difficile est de savoir quelles normes utiliser GRID.

Des dizaines de projets et des centaines de réalisateurs de logiciel travaillent de par le monde à la création de GRID, chacun avec ses propres idées sur ce qu'est une bonne norme. Pendant ce temps, la technologie continue d'évoluer et de produire de nouveaux éléments utiles à intégrer au matériel de GRID, pouvant nécessiter une révision des normes.

Qui est responsable du choix des normes - et qui suggère les révisions?

L'Internet et le Web reposent sur des normes clés telles que TCP/IP (protocole de transport/protocole inter-réseau) et HTTP (protocole de transfert hypertexte), qui ont été essentielles pour l'évolution des milieux concernés. Il s'agissait au début de spécifications établies par des organismes de normalisation dont la création résultait généralement d'initiatives locales, qui ont ensuite évolué vers le statut de norme internationale par un processus de caractère consensuel. L'Internet Engineering Task Force (IETF) est un organisme de normalisation s'occupant de l'Internet, alors que le World Wide Web Consortium (W3C) se consacre au Web.

Des normes spécifiques au GRID sont en cours d'élaboration par une entité de normalisation du même genre, le Global GRID Forum. Fédérant plus de 5 000 chercheurs et praticiens individuels, cet organe représente une force significative en matière d'édition de

normes et d'élaboration d'éléments permettant le travail en commun. Actuellement, une norme connue sous le nom OGSA (Architecture ouverte de services de GRID), toujours en cours de définition, est considérée comme une référence clé pour les projets d'élaboration de GRIDs futures.

Même maintenant, il existe un niveau d'accord extraordinaire quant aux technologies de base - compte tenu de ce que l'informatique en GRID est encore dans l'enfance. Tous les projets importants sont essentiellement fondés sur les protocoles et les services fournis par la boîte à outils Globus, une infrastructure en exploitation libre permettant de disposer des services de base nécessaires pour construire des applications GRID, tels ceux ayant trait à la sécurité, la découverte de ressources, la gestion de ressources et l'accès aux données.

3.3.2. Qu'est-ce qu'un GRID informatique ?

Comme toutes les technologies en pleine évolution, GRID peut être défini de différentes manières selon le contexte. Dans la littérature, GRID [45] est fréquemment décrit de manière informelle en la comparant au réseau électrique (d'où le nom GRID pour les anglophones, qui provient de powerGRID). Par analogie, les ressources informatiques sont aussi facilement accessibles, que les ressources électriques le sont, avec le réseau électrique. D'après la liste des caractéristiques de GRID extraites de la littérature et les différents catégories de support fourni par GRIDs qui ont été identifiés dans [46], GRID peut être défini comme une infrastructure logiciel et matériel géographiquement distribuée à grande échelle composée de ressources hétérogènes gérées en réseau. Ces dernières sont possédées et partagées par plusieurs organisations administratives qui sont coordonnées pour fournir un support informatique transparent, sûr et consistant à une grande gamme d'applications. Ces applications peuvent être utilisés pour l'informatique répartie, l'Informatique sur demande, le traitement intensif de données, l'informatique collaborative ou l'informatique multimédia. [47] définit GRID comme un type de système parallèle et distribué permettant le partage, la sélection, et l'agrégation de ressources autonomes géographiquement distribués de façon dynamique. Ces ressources ont leur propre disponibilité, capacité, performance, coût, et utilisateurs, avec leurs propres contraintes de qualité de service. Parmi les différentes définitions du GRID qui sont proposées, une définition plus précise est fournie par [48, 49] :

« GRID informatique consiste à fédérer des ressources hétérogènes de calcul et de stockage distribuées et mises en commun par différentes organisations indépendantes telles que des laboratoires universitaires ou des entreprises afin d'en agréger la puissance. La plate-

forme ainsi constituée est alors naturellement une constellation de réseaux locaux (internes à chaque laboratoire) connectés entre eux par un réseau à grande échelle le plus souvent à haut débit ».

Donc le GRID a pour but de réaliser le partage flexible, sûr et coordonné de ressources ainsi que la résolution coopérative de problème au sein d'organisations virtuelles.

Plus formellement, GRID doit être flexible, sécurisé, coordonner le partage des ressources à travers une collection d'individus, d'institutions et de ressources, à l'intérieur d'une organisation virtuelle. Dans certains cas, en plus de gérer tous problèmes à l'intérieur de l'organisation, GRID doit fournir une authentification unique, une autorisation et un accès aux ressources. Ces ressources sont potentiellement qualifiées de [50] :

- **Partagées** : elles sont mises à la disposition des différents consommateurs de GRID et éventuellement pour différents usages applicatifs.
- **Distribuées** : elles sont situées dans des lieux géographiques différents.
- **Hétérogènes** : elles sont de toute nature, les ordinateurs peuvent différer par exemple de par leur système d'exploitation.
- **Coordonnées** : les ressources sont arrangées, mises en relation et politisées en vue d'une fin, d'un objectif. Cette tâche est souvent remplie par un ou plusieurs ordonnanceurs.
- **Externalisées** : les ressources sont accessibles à la demande chez un fournisseur externe.
- **Non-contrôlées** : les ressources ne sont pas contrôlées par une unité commune. Contrairement à un cluster, les ressources sont hors de la portée d'un moniteur de contrôle.

Le principe de GRID est aussi simple que génial: il s'agit d'interconnecter, par Internet, un grand nombre d'ordinateurs disséminés dans le monde entier et de les faire fonctionner comme un grand ordinateur virtuel unique. La capacité inutilisée du disque dur, ainsi que la puissance de calcul non utilisée des ordinateurs peuvent alors servir pour répartir automatiquement un déluge de données et des calculs très complexes entre les ordinateurs.

GRID garantit donc des qualités de service non-triviales, c'est-à-dire qu'il se distingue des autres infrastructures dans son aptitude à répondre adéquatement à des exigences (accessibilité, disponibilité, fiabilité, ...) compte tenu de la puissance de calcul ou de stockage qu'il représente. GRID se compose de ressources informatiques : tout élément qui permet l'exécution d'une tâche ou le stockage d'une donnée numérique. Cette définition inclût bien sûr les ordinateurs personnels, mais également les téléphones portables, les calculatrices et tout objet qui présente un composant informatique.

Le terme GRID est aussi une aspiration à utiliser des protocoles standards afin de pouvoir politiser le partage de ressources entre groupements autonomes d'individus/d'institutions appelés organisations virtuelles. GRID exploite la puissance de calcul (processeurs, mémoires, ...) de milliers d'ordinateurs afin de donner l'illusion d'un ordinateur virtuel très puissant. Ce modèle permet de résoudre d'importants problèmes de calcul nécessitant des temps d'exécution très longs. GRID intègre un middleware, l'ensemble des couches réseaux et services logiciels qui permettent le dialogue entre les différents composants d'une application répartie. Le middleware masque la complexité des échanges inter-applications.

3.3.2.1. Couches de GRID

Jeffery [51], Rana et Moreau [52] et De Roure et al. [1] identifient trois couches dans l'infrastructure de GRID :

GRID informatique, la couche la plus basse, est principalement concernée à regrouper à grande échelle des ressources de traitements et de données ; Cette couche est responsable de la façon que les ressources sont allouée, Schedulede et exécutées, et aussi la façon que les données sont utilisées et échangées entre les diverses ressources de traitement ;

GRID d'information, la couche moyenne, permet l'accès uniforme aux ressources d'informations hétérogènes et fournit des services généralement fonctionnant sur les ressources distribuées. Cette couche s'occupe de la représentation, du stockage, de la consultation, du partage et du maintien de l'information;

GRID de connaissances, la couche supérieure, fournit les services spécialisés qui impliquent une agrégation de différents types de services et une intégration des données et des résultats fournis par chaque service. Cette couche est responsable de l'acquisition, de

l'utilisation, de la recherche, et du maintien de la connaissance afin d'aider des scientifiques pour atteindre des buts précis, pour résoudre des problèmes ou pour élaborer des décisions.

3.3.3. L'anatomie de GRID

La description de l'architecture GRID est donnée par [48]. Il propose une énumération tout à fait complète de tous les protocoles et les services spécifiant GRID. L'implémentation de ces protocoles et services est également détaillée dans la boîte à outils Globus [53], l'outil courant principal qui implémente les spécifications GRID. Les couches de l'architecture GRID sont :

3.3.3.1. La couche FABRIC

Le terme fabrique en français désignent l'ensemble de l'infrastructure physique de GRID, incluant les ordinateurs et les réseaux de transmission (en anglais fabrics, littéralement tissus). La couche Fabric de GRID fournit des ressources partagées dont l'accès est contrôlé par des protocoles de GRID : par exemple, les ressources CPUs, systèmes de stockage, ressources réseaux, et des capteurs. Une ressource peut être une entité logique, telle qu'un système de fichiers, un cluster, ou un ensemble d'ordinateurs ; dans ces cas-ci, l'implémentation de ressources peut requérir l'utilisation de protocoles internes, mais ce n'est pas un problème lié à l'architecture de GRID. La figure 3.3 présente les différentes couches de l'architecture GRID.

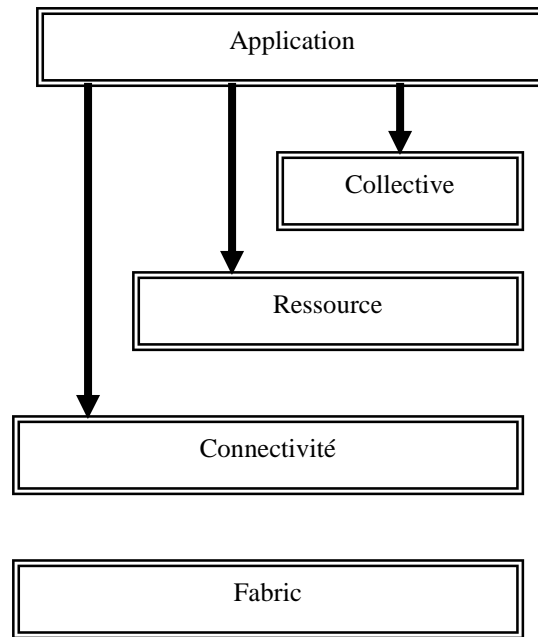


Figure 3.3- Les couches de l'architecture GRID

Les composants de la couche Fabric mettent en application les opérations locales qui se produisent sur les ressources spécifiques (physique ou logique) en raison des partages d'opérations à des niveaux plus élevés. Il y a ainsi une interdépendance entre les fonctions mises en application au niveau de la Fabric, d'une part, et les opérations de partage soutenues, de l'autre. Une fonctionnalité plus riche de la Fabric permet des opérations de partage plus sophistiquées ; en même temps, si nous plaçons peu de demandes sur des éléments de la Fabric, puis déploiement de l'infrastructure GRID est simplifié. Par exemple, le support des niveaux de ressources des réservations avancées permet que les services de plus haut niveau agrègent des ressources de façon intéressantes qui seraient impossibles autrement à réaliser.

3.3.3.2. La couche connectivité

La couche connectivité définit des protocoles de communication et d'authentification exigés pour des transactions réseaux spécifiques au GRID. Les protocoles de communication permettent l'échange des données entre les ressources de couche Fabric. Les protocoles d'authentification basés sur les services de communication fournissent des mécanismes sécurisés de cryptographie pour vérifier l'identité des utilisateurs et des ressources. Les conditions de communication incluent le transport, l'acheminement, et la résolution de

noms. Alors que des solutions de rechanges existent certainement, on suppose ici que ces protocoles sont tirés de la pile de protocole de TCP/IP : Spécifiquement, l'Internet (IP et ICMP), transport (TCP, UDP), et application (DNS, OSPF, etc...) les couches de l'Internet ont fixés l'architecture des protocoles. Ce qui ne veut pas dire que dans le future, les communications de GRIDs n'exigeront pas de nouveaux protocoles qui tiennent compte des types particuliers de la dynamique réseau. En ce qui concerne les aspects de sécurité de la couche de connectivité, nous observons que la complexité du problème de la sécurité rend plus important le fait que toutes les solutions soient basées autant que possible sur des normes existantes. Comme avec la couche communication, plusieurs des normes de sécurité développées dans le cadre de la suite de protocole d'Internet sont applicables ici.

3.3.3.3. La couche ressource

La couche ressource est basée sur la couche connectivité et utilise ses protocoles de communications et d'authentications pour définir des protocoles (APIs et SDKs) pour la négociation, le déclenchement, la surveillance, le contrôle, et l'achèvement d'opérations de partage sur différents types ressources. La couche ressource et ses protocoles appellent des fonctions de la couche Fabric pour contrôler et accéder aux ressources locales.

Deux classes primaires des protocoles de la couche ressource peuvent être distinguées :

- Les protocoles d'information sont utilisés pour obtenir des informations sur la structure et l'état d'une ressource, par exemple, de sa configuration, charge courante, et politique d'utilisation.
- Les protocoles de gestion sont utilisés pour négocier l'accès à une ressource partagée, définissant, par exemple, des ressources requises (réservation avancée et qualité de service) et les opérations à exécuter, comme la création de processus, ou d'accès aux données. Puisque les protocoles de gestion sont responsables de l'instanciation des rapports de partage, ils doivent servir "de point d'application des politiques locales" s'assurant que les opérations demandées aux protocoles soient conformes à la politique sous laquelle la ressource doit être partagée.

Profil des protocoles généralement utilisés:

- Un Protocole d'information des ressources sur GRID (GRIP, actuellement basée sur le Protocole d'accès d'annuaire : LDAP) est utilisé pour définir un protocole standard d'information des ressources et un modèle d'information associée. Un protocole associé d'enregistrement de ressources d'état logiciel, le Protocole d'enregistrement de ressource de GRID (GRRP), est utilisé pour enregistrer des ressources avec des Serveurs d'information d'index de GRID, discutés dans la prochaine section.
- Le protocole HTTP d'accès et de gestion des ressources de GRID (GRAM) est employé pour l'attribution des ressources CPUs, de la surveillance et du contrôle du calcul sur ces ressources.
- Une version améliorée du Protocole de transfert de fichier, le GRIDFTP, est un protocole de gestion pour l'accès aux données ; les améliorations incluent l'utilisation des protocoles de sécurité de couche de connectivité, de l'accès partiel aux fichier, et de la gestion du parallélisme pour les transferts à grande vitesse. Le ftp est utilisé en tant que protocole de transfert de données de base en raison de son support de tiers transferts et parce que ses canaux séparés de contrôle et de données facilitent l'exécution des serveurs sophistiqués.
- Le LDAP est également utilisé comme protocole d'accès de catalogue.

La boîte à outils Globus coté client définit les APIs et SDKs en C et en Java pour chacun de ces protocoles. Les SDKs coté serveur et les serveurs sont aussi fournis pour chaque protocole, pour faciliter l'intégration de diverses ressources (informatiques, stockage, réseau) dans GRID. Par exemple, le service d'information de ressource de GRID (GRIS) active coté serveur les fonctionnalités LDAP, avec des légendes tenant compte de la publication d'information arbitraire des ressources.

3.3.3.4. La couche collectivité

Alors que la couche de ressource se concentre sur des interactions avec une ressource simple, cette couche dans l'architecture contient les protocoles et les services (APIs et le SDKs) qui ne sont pas associés à une ressource spécifique mais plutôt à des interactions avec

de multiples ressources. C'est pour cette raison, qu'il se réfère à la prochaine couche de l'architecture comme couche collective. Puisque les constituants de la couche collective sont basé sur les couches Ressource et connectivité, elles peuvent effectuer un grand nombre de partages sans interférer avec les ressources déjà partagés. Par exemple :

Les services d'annuaire permettent à des participants d'OV de découvrir l'existence et/ou les propriétés des ressources des OV. Un service d'annuaire peut permettre à ses utilisateurs de rechercher des ressources par nom et/ou par attributs tels que le type, la disponibilité, ou la charge. Les protocoles niveau ressource GRRP et GRIP sont employés pour construire des annuaires.

La Co-attribution, programmation, et les services brokering permettent à des participants d'OV de demander l'attribution d'une ou plusieurs ressources dans un but spécifique et l'établissement du programme des tâches sur les ressources appropriées. Les exemples incluent Apple-S, Condor-G, le Nimrod-G, et le DRM broker.

Les services de surveillance et de diagnostics supportent la surveillance des ressources des OV en cas d'erreur, d'attaque directe ("détection d'intrusion"), surcharge, etc...

Les services répliation des données supportent la gestion des ressources de stockage des OV (voir de réseau et de calcul) pour maximiser l'accès aux données en respectant les métriques tels que le temps de réponse, la fiabilité, et le coût.

Les systèmes de programmation de GRID active permettent à des modèles de programmation familiers d'être employés dans des environnements de GRID, en utilisant de divers services de GRID pour gérer la découverte de ressource, de sécurité, d'attribution de ressource entre autres.

Systemes de gestion de charge de travail et cadres de collaboration - également connus sous le nom d'environnements de résolution des problèmes (PSEs)-fournissent pour la description, l'utilisation, et la gestion des déroulements des opérations multi-niveaux, asynchrones, à plusieurs éléments.

Les services de découverte de logiciel recherchent et choisissent la meilleure plateforme d'exécution logiciel basée sur les paramètres du problème à résoudre. Les exemples incluent NetSolve et Ninf.

Les serveurs d'autorisation de Communauté imposent des politiques de communauté régissant l'accès des ressources, permettant qu'un membre de la Communauté peuvent utiliser pour accéder aux serveurs de ressources de la communauté .Ces serveurs fournissent un service global d'application de politique par la construction sur l'information de ressources, et des protocoles de gestion de ressource (dans la couche ressource) et les protocoles de sécurité dans la couche connectivité.

Les services de comptabilité et de paiement de la Communauté recueillent des informations sur l'utilisation des ressources afin de fournir la comptabilité, le paiement, et/ou la limitation de l'utilisation de ressource par membre de la Communauté.

Les services de Collaboration supportent l'échange d'information coordonné dans les communautés d'utilisateur potentiellement grandes, de façon synchrone ou asynchrone.

Ces exemples illustrent la grande variété de protocoles de la couche collectivité et des services qui sont rencontrés dans la pratique. Il est à noter que les protocoles de la couche de ressource doivent être généraux et sont largement déployés, les protocoles de la couche collectives passent de généraux à spécifiques pour les applications ou les domaines, avec dernier exister peut-être seulement dans OV's spécifique.

Des fonctions collectives peuvent être mises en application en tant que services persistants, avec des protocoles associés, ou comme SDKs conçu pour être lié avec des applications. Dans les deux cas, leur exécution peut se faire sur la couche ressource (ou sur une autre couche collectives) protocole et APIs. Par exemple, la figure 3.4 montre une Co-attribution collective API et SDK (la rangée du milieu) utilisant un protocole de gestion de la couche ressource afin d'utiliser les ressources fondamentales.

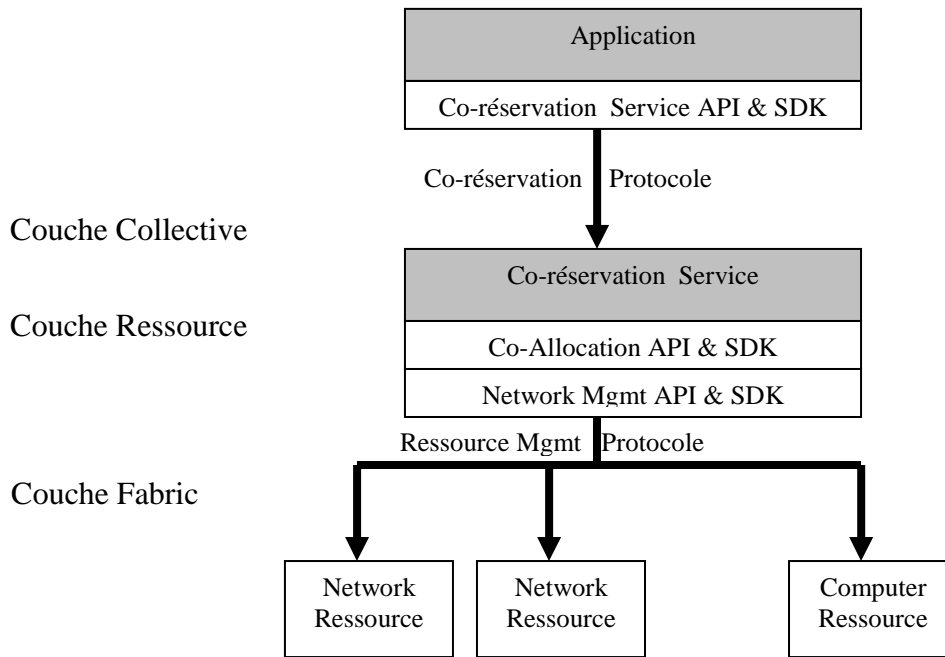


Figure 3.4- Services, APIs, SDKs et Protocoles de la couche collective et ressource

Des composants collectifs peuvent être conçus en fonction des conditions d'une communauté d'utilisateur, d'une OV, ou d'un domaine spécifique d'application, par exemple, un SDK qui utilise un protocole spécifique d'une application, ou un service de Co-réservation pour un ensemble spécifique de ressources réseaux. D'autres composants collectifs peuvent être plus d'usage divers, par exemple, un service de réplique qui contrôle une collection de systèmes de stockage internationale pour de multiples communautés, ou un service d'annuaire conçu pour permettre la découverte des OV. En général, plus la communauté d'utilisateurs ciblés est grande, plus il est important que le protocole et l'API(s) d'un composant collectif soient basés sur des normes. On dit souvent que ces services sont persistants (c.-à-d. ; toujours disponible - opposé à l'éphémère (provisoire))

3.3.3.5. La couche application

La couche finale dans cette architecture de GRID comprend les diverses applications d'utilisateur qui fonctionnent dans un environnement d'OVs. Les applications sont construites en termes de services définis à chaque couche. C'est la couche que verront les utilisateurs de GRID. La figure 3.5 illustre la vue d'un programmeur d'application de l'architecture GRID. Les applications sont construites en appelant les services définis sur n'importe quelles couches. À chaque couche, nous avons des protocoles bien définis qui permettent d'accéder à un service bien précis : la gestion de ressource, l'accès aux données, la découverte de ressource, et ainsi de

suite. À chaque couche, les APIs peuvent définir l'implémentation d'échange de protocole des services approprié à exécuter pour réaliser l'action désirées.

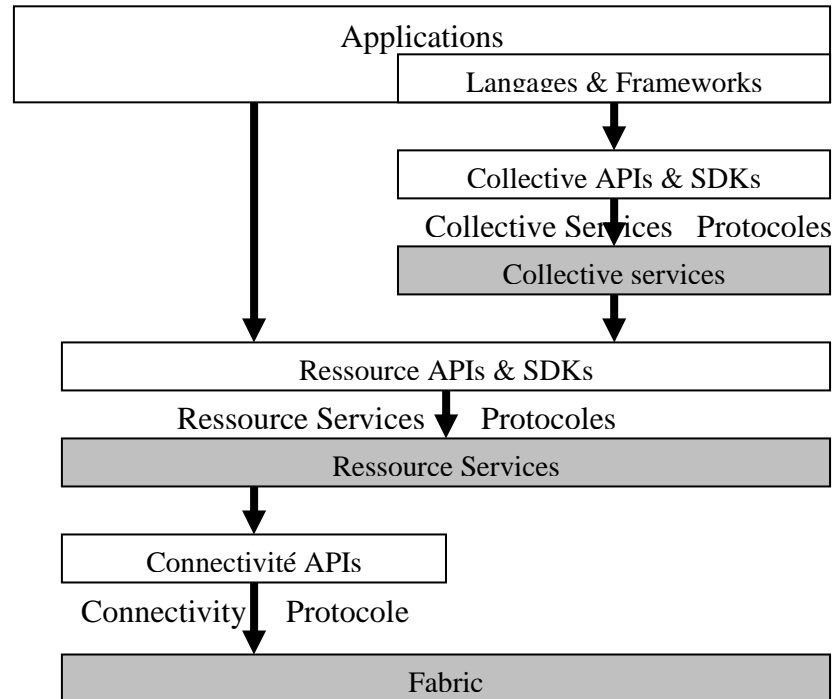


Figure 3.5- API et Protocole de l'infrastructure GRID

3.3.4. Boîte à outils Globus

Pratiquement tous les grands projets de GRID sont fondés sur des protocoles et des services fournis par la boîte à outils Globus [54, 55]- élaborée dans le cadre d'un projet par le consortium Globus Alliance, dont les principaux membres sont l'équipe dirigée par Ian Foster au Laboratoire national d'Argonne (États-Unis) et celle de Carl Kesselman à l'université de Californie du Sud de Los Angeles.

Cette boîte à outil permet de disposer d'un ensemble de moyens de génie logiciel pour la mise en œuvre des services et des capacités de base requis pour construire GRID de calcul, tels que ceux ayant trait à la sécurité, à la localisation des ressources, à la gestion des ressources et aux transmissions.

Globus comprend des programmes tels que :

- **GRAM** (programme Globus de gestion de l'affectation des ressources) - qui détermine comment convertir une demande de ressources en commandes que des ordinateurs locaux peuvent comprendre;
- **GSI** (infrastructure de sécurité de GRID) - qui assure l'authentification de l'utilisateur et détermine les droits d'accès de cette personne ;
- **MDS** (service de surveillance et de découverte) - qui collecte des informations concernant les ressources (capacité de traitement, bande passante, type de mémoire, etc.) ;
- **GRIS** (service d'information sur les ressources de GRID) - qui demande aux ressources quelles sont leurs configuration, capacités et situation, actuelles ;
- **GIIS** (service de répertoire de GRID) - qui coordonne les services GRIS génériques ;
- **GRIDFTP** - qui assure des mécanismes de transfert de données, robustes, sécurisés et à hautes performances ;
- **Le catalogue des répliques** - qui peut être consulté par d'autres composants Globus pour déterminer où peuvent se trouver des répliques d'un ensemble déterminé de données ;
- **Le système de gestion des répliques** - qui relie le catalogue des répliques et le système GRIDFTP, pour permettre à des applications de créer et de gérer des répliques de grands ensembles de données

Beaucoup des protocoles et des fonctions définis par la boîte à outils Globus sont semblables aux protocoles utilisés aujourd'hui sur les réseaux et lors des échanges avec les mémoires - mais ils sont optimisés aux fins de mises en œuvre spécifiques à GRID.

Deux raisons principales expliquent le potentiel et le succès de la boîte à outils Globus : GRID devra prendre en charge des applications très diverses, créées selon différents paradigmes de programmation. Plutôt que de proposer un modèle uniforme de programmation pour les applications GRID, la boîte à outils Globus suit une démarche " orientée objet ", en

fournissant un " panier de services " dans lequel les réalisateurs d'applications spécifiques peuvent choisir les plus appropriés à répondre à leurs besoins particuliers. Les éléments de la boîte à outils peuvent être incorporés un à un à des programmes existants pour les rendre de plus en plus exploitables sur GRID.

À l'instar de la stratégie adoptée pour le Word Wide Web et pour le système d'exploitation Linux, les créateurs de la boîte à outils Globus mettent le logiciel à disposition, au titre d'un contrat de licence en exploitation libre (open source) - ce qui permet à d'autres utilisateurs de s'en servir et d'y incorporer toutes les améliorations qu'ils lui apportent

3.3.5. Les services GRID

Un service GRID est considéré comme un service Web à état respectant des protocoles standards. [56] appelle service GRID, «une instance de service à état qui supporte une invocation fiable et sûre, une gestion de la durée de vie, des notifications, des règles de gestion, une identification et la virtualisation ».

OGSA (Open GRID Services Architecture) [57] spécifie rigoureusement les services GRID. Là où les services Web sont sans état et persistants, les instances de service GRID peuvent être soit avec ou sans état et éphémères ou persistants. Les services sans état sont synchrones et interagissent par de simple requête/réponse. Ils retournent simplement une réponse à une invocation précise. Les services à état peuvent être asynchrones et interagissent par des conversations. Ils permettent plus d'adaptation dans le service fourni. Plus récemment, WSRF (Web Service Resource Framework) a uniformisé les mécanismes pour accéder à des ressources à état avec des services Web/GRID.

3.3.5.1. Cycle de vie de service GRID

Les services GRID peuvent maintenir un état interne durant le cycle de vie du service. L'existence de l'état distingue une instance d'un service d'un autre qui fournit la même interface. Le terme instance de service GRID est utilisé pour référer à une instanciation particulière d'un Service GRID. Une instance de service est créée par un autre service appelé: Générateur de Service GRID (GRID service factory). Les instances de services GRID peuvent être créées et détruites dynamiquement.

Les instances de service GRID sont créées avec une vie spécifiée. La vie initiale peut être prolongée par une période du temps spécifiée par une demande explicite de l'utilisateur

ou un autre service GRID. Si cette période de temps expire sans recevoir une réaffirmation d'intérêt du client, l'instance de service GRID se termine et les ressources associées sont libérées. On constate que cette approche de gestion de vie fournit un service avec une autonomie considérable. L'extension du cycle de vie demandée par le client n'est pas obligatoire: le service peut appliquer ses propres politiques en accordant une telle demande. Un service peut décider à tout moment de prolonger sa vie, en réponse à une demande d'utilisateur de prolongation de vie ou pour n'importe quelle autre raison. L'instance de Service GRID est identifié par GSH(GRID Service Handle) et GSR (GRID Service Reference). Le GSH est un URI globalement unique et qui aide un utilisateur à localiser un service, mais qui ne diffuse pas assez d'information pour qu'un utilisateur communique directement avec L'instance de Service (par exemple, adresse de réseau). C'est le rôle du GSR, un document WSDL, qui contient l'information qu'un client doit communiquer avec le service. Un GSH est valide durant la vie de l'instance de service GRID, alors qu'un GSR pourrait devenir inadmissible. Une instance de service GRID exécute une ou plusieurs interfaces, où chaque interface définit un ensemble d'opérations qui sont invoquées en échangeant un ordre défini des messages. Les informations concernant l'instance de Service GRID (c.-à-d., nom, type, Handle, durée de vie, etc.) sont encapsulées dans un document XML.

Nous montrons le cycle de vie des services GRID autour des relations, d'une part, entre les trois technologies standard des service WEB à savoir WSDL, SOAP et UDDI et le couple Générateur-instance de service et l'utilisateur GRID d'une autre part à l'aide d'un exemple d'interaction simple entre utilisateur et générateur de service GRID dans la figure 3.6.

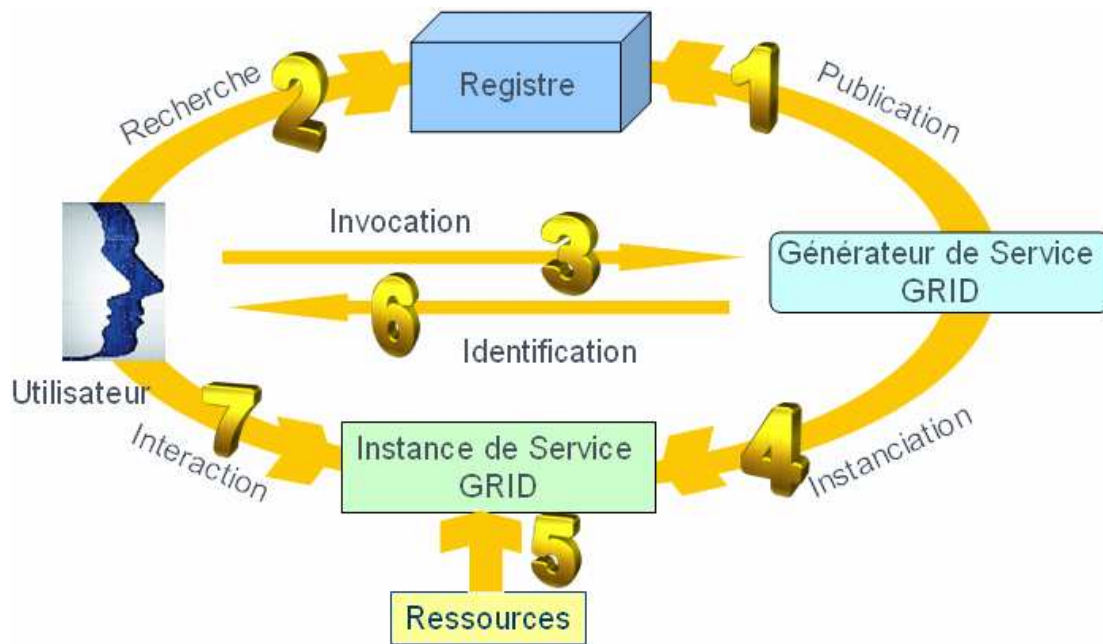


Figure 3.6-Cycle de vie de Service GRID

Un générateur de service GRID (1) doit tout d'abord déployer et publier son service auprès d'un annuaire UDDI pour qu'il puisse être retrouvé. L'utilisateur (2) désirant interagir avec un service GRID doit ainsi le localiser. Cet utilisateur doit donc effectuer une recherche auprès d'un annuaire en envoyant une requête UDDI au registre de service. Une fois le service localisé, l'utilisateur (3) peut avoir accès au document WSDL décrivant comment contacter le générateur de service GRID respectif. L'utilisateur est ainsi capable de construire un message SOAP conforme au format spécifié dans le document WSDL. Après avoir recevoir un message provenant de l'utilisateur, le générateur de service lui crée (4) dynamiquement des instances de service GRID qui utiliseront de la ressource (5) et libéreront cette ressource en fin de vie. Les instances de services GRID disposent des états internes et peuvent être créés et détruits dynamiquement. Le générateur de service GRID (6) ensuite communique à l'utilisateur le GSH (GRID Service Handle) et le GSR (GRID Service Reference). Le GSH est un identifiant unique du service alors que le GSR est un document WSDL contenant l'information qu'un utilisateur doit communiquer avec le service. Finalement l'utilisateur (7) entreprend une interaction directe avec l'instance du service sous le format d'un message SOAP.

3.3.6. L'architecture Service GRID OGSA

Open GRID Service Architecture (OGSA) [57, 58] a été publiée en 2002 pour standardiser [56] une architecture GRID en tant qu'évolution des services Web pour répondre à des besoins plus particuliers des GRIDs. Elle est vue comme une série de spécifications techniques par lesquelles on définit une infrastructure pour intégrer et gérer les services à l'intérieur d'une organisation dynamique, virtuelle et distribuée. Elle est principalement supportée par le Global GRID Forum. La version 3 du Globus Toolkit [53] est une implantation de l'architecture OGSA.

La figure 3.7 est une représentation synthétique des différentes notions essentielles introduites par OGSA. Ce modèle OGSA simplifié [59, 60, 61] décrit un ensemble d'unités de ressource couplées par un réseau TCP/IP. Cet ensemble ainsi virtualisé fournit une capacité de traitement et de stockage globale. Les services sont ensuite instanciés dans des conteneurs dynamiques sur une portion de ressource réifiée.

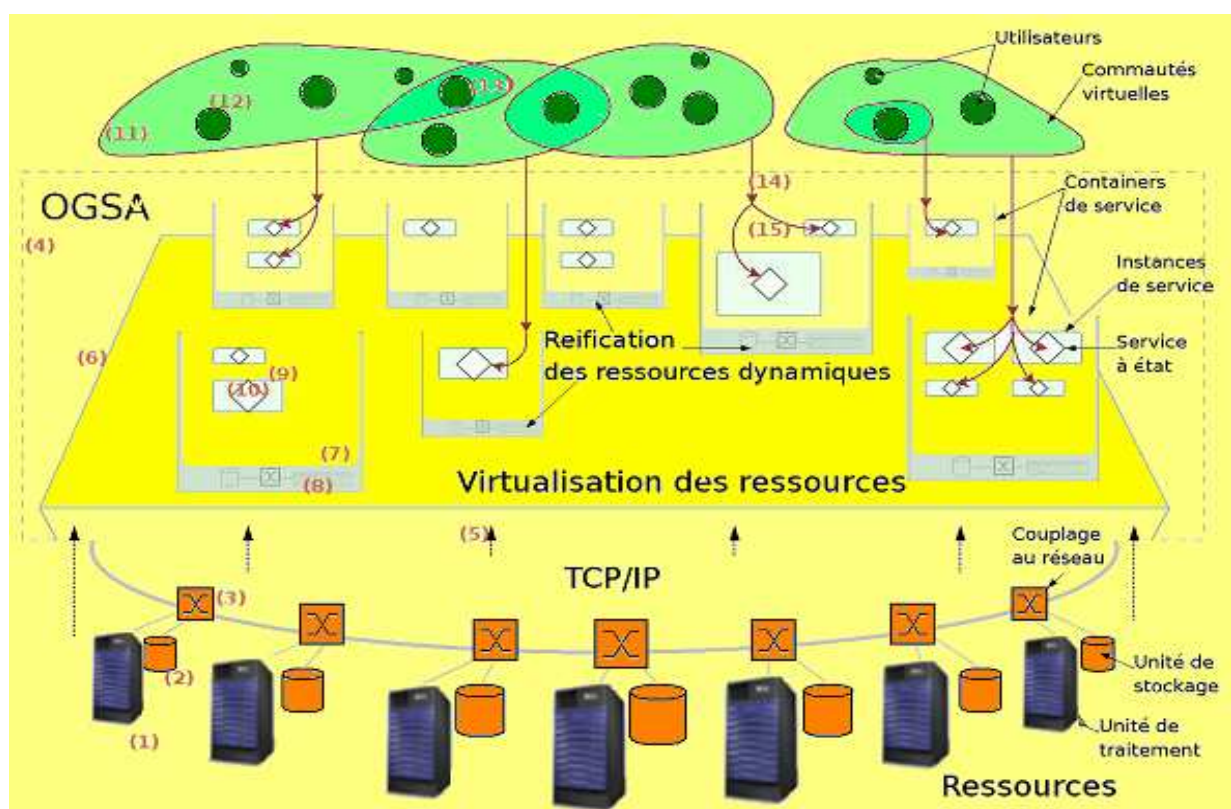


Figure 3.7- Modèle simplifié de l'architecture Service GRID OGSA

Sur le bas du diagramme, les ressources sont représentées sous forme d'unités qui fournissent de la capacité de traitement (1), stockage (2) et qui sont couplées à travers une infrastructure de réseau TCP/ IP. L'intérieur du rectangle en pointillé (4) au milieu délimite le champ d'OGSA. En premier lieu, les ressources physiques sont virtualisées (5). Toutes les unités de ressource sont additionnées pour former une capacité globale de traitement et de stockage (6). Deuxième temps des conteneurs de service (7) sont créés dynamiquement avec une portion de ressource réifiée (8). Les instances de service sont exécutées dans ces conteneurs (9). L'état des services GRID (10) est ainsi géré de cette manière. Une communauté virtuelle (11) est composée d'un ou plusieurs utilisateurs appelé membre (12). Un même utilisateur peut être membre de plusieurs communautés virtuelles (13). Les membres d'une communauté virtuelle peuvent accéder et modifier l'état des services (15) propres au conteneur de leur communauté virtuelle selon les droits qui leurs sont attribués (14). Il est à noter que la terminologie employée ici est tirée de la spécification OGSI (Open GRID Service Infrastructure) [62]. Les concepts (ex : ressources, service, Conteneur de services, etc.) et les mécanismes clés d'OGSA (ex : virtualisation, réification, handling) sont décrits dans le chapitre consacré à notre proposition.

3.3.6.1. Evolution d'OGSA

OGSA représente une évolution vers une architecture de système GRID basée sur les technologies et les concepts de Web services. Depuis la réalisation de la boîte à outils Globus 3.0, le projet Globus offre une collection Open Source de services GRID suivant les principes de l'architecture OGSA. La boîte à outils Globus offre également un environnement de développement pour produire de nouveaux services GRID qui suivent les principes OGSA. [63] a proposé d'insérer le H dans un modèle intégré OGHSa pour enrichir OGSA par la dimension Humaine. Le but de la proposition d'OGHSa est de placer humain en tant qu'agent à l'extrémité de la chaîne de virtualisation et de simplifier l'utilisation côté client. L'infrastructure Open GRID Service Infrastructure (OGSI) [62] lancée en juin 2003 est le standard émergent pour le développement de services de GRID. Elle constitue l'infrastructure de base pour la définition des services de GRID composant l'architecture Open GRID Service Architecture (OGSA) [57], ainsi que, potentiellement, pour tout service de GRID développé en réponse à des besoins spécifiques. La spécification OGSI a pour objectif de normaliser les services composant les GRIDs, afin de garantir l'interopérabilité de systèmes hétérogènes pour le partage et l'accès à des ressources de calcul et de stockage distribuées. Cette

spécification définit ce qu'est service GRID; un service Web respectant un ensemble de conventions adaptées aux contraintes de l'environnement de GRID.

Récemment, les concepts OGSi ont été repris un à un avec une nouvelle terminologie issue du groupe de normalisation WSRF (Web service Resource framework) [64]. Ainsi, la notion d'instance de service correspond à WS-Ressources dans la nouvelle terminologie et la notion d'identifiant de service correspond à WS-Addressing. En effet WSRF est ensemble de spécifications de Service Web développé par l'Organisation OASIS [42]. Ces spécifications décrivent comment implémenter les capacités OGSA en utilisant les services Web, la boîte à outils Globus 4.0 [65] et les versions postérieures fournissent un kit de développement Open source et un ensemble de services WSRF.

La figure 3.8 montre la convergence de GRID et services Web [66]:

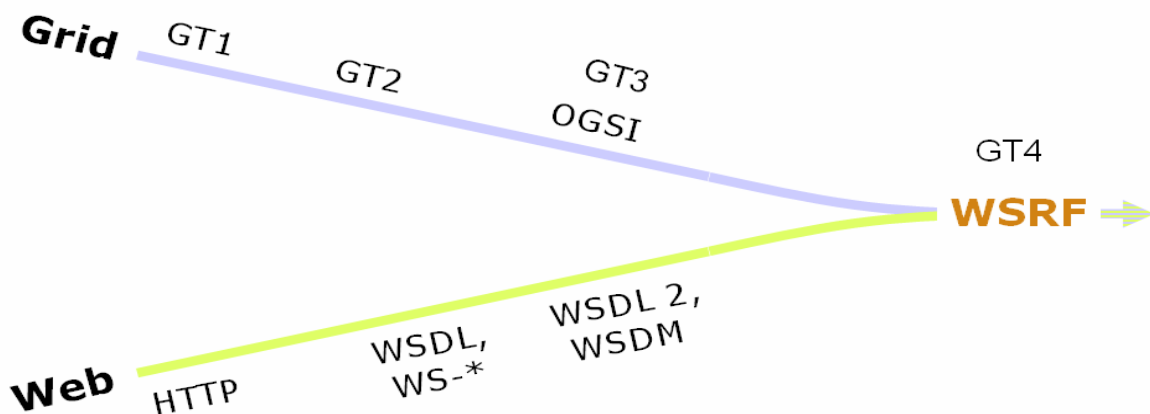


Figure 3.8- GRID et Service Web : convergence

Le point de convergence est identifié par Web Services ressource Framework(WSRF), qui est un mécanisme commun permettant de définir, d'inspecter et de contrôler à distance l'état du service. Il contribue à l'utilisation de GRID à très grande échelle, à sa fiabilité et à l'interopérabilité entre application de GRID et le service GRID. Pour un récent aperçu des concepts des services GRID et leur standardisation voir par exemple [67].

3.3.7. Une méthodologie de mise en place

Globus est une boîte à outils open source pour la mise en place de GRID ou la construction d'outils de GRID. Il ne s'agit nullement d'un produit prêt à l'emploi. Mais des éditeurs spécialisés fournissent déjà depuis quelques années des outils qui s'apparentent à des schedulers pour des GRID encore essentiellement intradépartements ou intra-entreprises. C'est le cas de Platform Computing, fondé en 1992, avec LSF ou Activecluster, des outils de réservation de ressources et de répartition de charge en environnement hétérogène (Unix, Linux, Windows, Apple, etc.) ; ou de Data Synapse avec Livecluster ; ou encore d'Avaki avec Data GRID, Computational GRID et Comprehensive GRID - répartition de données, de calculs, ou des deux. Par ailleurs, Platform distribue le Globus Toolkit en France. Installé dans l'Hexagone depuis 2003, Entropia se concentre, lui, sur les PC sous Windows. Son DC GRID répartit les tâches sur une centaine de PC environ. Comme beaucoup d'autres, il rend un service encore plus essentiel sous Windows : isoler l'environnement de travail du poste distant et le « job » exécuté. Sun propose également un gestionnaire de tâches distribuées, appelé GRIDengine. De nouveaux fournisseurs sont récemment apparus, comme le Français GRIDXpert, l'Espagnol GRIDsystems ou l'Italien Nice, qui donnent une couche d'accès et d'optimisation de GRID de type portail au-dessus de ces ordonnanceurs. Tous ces acteurs, montrés dans le tableau 3.1, garantissent la conformité de la prochaine version de leurs outils avec OGSA dès sa sortie.

Société	Activité GRID
Société activité GRID	éditeur spécialisé de GRIDs :GRID de données,de calcul, ou les deux sur tout type de plateforme
Cray, Dell, Intel, Oracle, SGI	De plus en plus de grands acteurs se tournent vers le GRID : en intégrant la boîte à outils Globus sur leurs machines, en participant à OGSA, ou en travaillant sur les applications (Oracle RAC)
Data synapse	éditeur spécialisé. Liveclustur est un outil de répartition de tâches sur machines hétérogènes
Entropia	Installé en France depuis mars 2003. DC GRID répartit la charge applicative sur des PC sous Windows
GRIDSystem, GRIDXpert, Nice	Start up européennes du GRID. L'Espagnol GRIDSystems. le Français GRIDXpert et l'italien Nice proposent un portail évolué de construction et de gestion de GRID à partir des logiciels du marché
HP	Impliqué dans le développement d'OGSA. a monté en janvier 2003 une task force autour du GRID. Constituée essentiellement de responsables de l'entité service de HP. www.hp.com/techservers/GRID
IBM	Participe activement au développement d'OGSA, a équipé tous ses serveurs non Windows de la boîte à outils globus. travaille à des logiciels d'administration de GRID dans Tivoli ou des outils de développement au sein de Websphere. Commercialise un ensemble de package verticaux avec les éditeurs spécialisés. www.ibm.com.GRID
Platform Computing	LSF pour la réservation de ressources et répartition de charges en environnement hétérogène Unix, Linux, Windows, Apple, Distributeur de la boîte à outils Globus en France
Sun	GRIDEngine. Outil libre et gratuit de scheduling de GRID. est impliqué dans le développement d'OGSA. GRIDEngine Entreprise Edition www.sun.com/GRID
United Devices	Distribution de calcul sur un très grand nombre de machines sur Internet (type Seti@home)
Global GRID Forum	Entité indépendante. Dans laquelle les acteurs du GRID collaborent au sein de groupes de travail à l'avancement des recherches sur les différents éléments du GRID. Dans un esprit d'ouverture et des standardisations. www.globalGRIDforum.org
Globus	Projet de recherche d'outils de base pour une architecture GRID développé au laboratoire d'Argonne (Illinois). Fait référence. C'est le standard du domaine. La boîte à outil Globus est la première implémentation du standard OGSA www.globus.org
Unicore	Projet pour une architecture GRID menée par une association de fournisseurs. www.unicore.org/unicore.html

Tableau 3.1- Les acteurs du GRID

Par ailleurs, outre les questions purement techniques, certains fournisseurs s'attachent à ce qui se passe avant et après la mise en place de l'architecture. Depuis novembre 2002, Platform Computing propose un outil de tableau de bord pour « piloter » GRID au sein du système d'information. C'est aussi le cas de GRIDXpert. Platform Computing et IBM formalisent également, chacun, une méthodologie de mise en place de GRID - dimensionnement, expansion, rentabilité, etc. Récemment, les plus grands fournisseurs du marché soutiennent activement le développement du GRID. Ainsi, HP a monté en janvier 2003 une « task force » autour du GRID et interfacé son Utility Data Center avec la boue à outils Globus. Mais c'est IBM qui mène la danse. Il a non seulement équipé tous ses serveurs (y compris bientôt ses zSeries sous Linux) du Globus Toolkit, mais a aussi participé à la mise au point d'OGSA. Son entité Tivoli travaille à l'administration de GRID, et Websphere intégrera l'infrastructure de base d'OGSA, OGSi (Open GRID Services Infrastructure), pour favoriser le développement d'applications. Le géant d'Armonk commercialise même, depuis février 2003, des packages regroupant services et outils à destination de marchés verticaux tels que la finance. Il teste aussi les logiciels du marché avec ses clients dans son centre européen de Montpellier.

3.3.8. Projets et applications GRID

3.3.8.1. Applications

Pour quels types d'applications GRID va être utiliser ? En premier temps, [68] a identifié cinq classes d'applications majeures pour GRID résumées dans le tableau 3.2:

Catégories	Utilisations de GRID pour	Caractéristiques
Supercalculateurs distribués (Distributed Supercomputing)	Agréger les ressources de calculs substantielles afin d'aborder les problèmes qui ne peuvent pas être résolus sur un système simple	Beaucoup de problèmes ont besoin d'un nombre important de CPUs, de mémoires, etc.
Traitement de haut débit (High throughput)	Scheduler un grand nombre de tâches couplées ou indépendantes, avec le but de mettre les cycles de processeur inutilisés au travail	Beaucoup de ressources inactives pour augmenter le débit global
Informatique à la demande (on demand)	Satisfaire les besoins de ressource de calculs	Ressources éloignées intégrées avec le calcul local
Traitement de données intensives (Data-Intensive Computing)	Coupler les ressources de données géographiquement distribuées	Synthèse de nouvelles informations à partir de beaucoup ou grandes sources de données
Informatique collaborative (Collaborative computing)	Connecter les gens	Support la communication ou le travail collaboratif entre les participants

Tableau 3.2 - Cinq classes majeures d'application GRID

3.3.8.2. Projets

L'usage de GRID était à ses débuts l'apanage du calcul intensif mais son usage est désormais étendu à n'importe quel type d'application car il est un excellent moyen de partager des données ou une puissance de calcul. En effet, beaucoup de projets ont vu le jour en se basant sur ce principe. Dans ce qui suit nous allons décrire quelques fameux projets de GRID.

EGEE : [69] Le projet EGEE (Enabling GRIDs for E-sciencE) construit une infrastructure mondiale avec le soutien de la Commission européenne et l'aide d'une nonantaine de partenaires dans le cadre du projet européen DATAGRID (<http://eudataGRIDweb.cern.ch/eu-dataGRID>), porté dans l'intergiciel LCG (<http://lcg.web.cern.ch/LCG/>) qui constitue le coeur d'un intergiciel en gLite [70] orienté services web, et dont les premières versions sont disponibles. EGEE est déjà opérationnel dans 27 pays, interconnectant plus de 180 centres de calcul scientifiques européens et mondiaux, avec une connexion vers des initiatives de GRID aux États-Unis, au Japon, en Chine et en Corée. EGEE entend permettre aux chercheurs des milieux universitaires et

industriels, quel que soit l'endroit où ils opèrent, d'accéder en permanence à une importante puissance de calcul et à une capacité de stockage élevée.,

EUMEDGRID : L'initiative d'EUMEDGRID [71] vise à développer dans le bassin Méditerranéen une infrastructure de GRID pour la Recherche, qui peut devenir une partie d'EGEE et être intégrée avec des initiatives analogues dans les Balkans, l'Europe du nord, l'Amérique Latine et le Loin-Est Asiatique. Un autre but d'EUMEDGRID est d'accroître la conscience et les compétences relatives aux GRIDs parmi les chercheurs qui travaillent dans la Région Méditerranéenne, pour les rendre capables de bénéficier de ce nouvel et puissant instrument, pour stimuler la collaboration avec des projets Européens et dans le monde entier et pour soutenir le développement scientifique et industriel dans la région.

SETI@home [72] : Le projet a été initié par l'Université de Berkeley. C'est une expérience scientifique qui utilise des ordinateurs connectés à Internet pour la recherche d'intelligence extraterrestre. Plus de 3 millions d'utilisateurs ont permis à ce projet de devenir le plus long calcul de toute l'histoire du GRID computing. En effet, près de 600 000 années de calcul temps-processeur ont été consacrés à la recherche extraterrestre.

Genom@home [73]: Développé par l'Université de Stanford, ce projet étudie les interactions entre le génome et les protéines. Près de 90 000 membres actifs proposent leurs machines personnelles pour ces calculs de recherche.

World Community GRID [74]: La communauté de calcul a été élaborée par la société IBM. Elle a pour but de créer un lien entre des laboratoires de recherche qui ont besoin d'une énorme puissance de calcul, et des volontaires soucieux de participer à la recherche scientifique. Ce système permet à l'utilisateur de choisir sa participation sur un des projets lancés. Le repliement du protéome humain, la comparaison du génome, ou encore FightAIDS@HOME (projet permettant l'identification de médicaments candidats pour bloquer la protéase du VIH) ont commencé leurs calculs dans un but précis: le combat contre la maladie. Plus de 250 000 personnes se sont inscrites, pour un total de 80 000 années temps-processeur.

Programme DECRYPTHON: Le Décryphon [75] est une opération lancée en 2001 afin de réaliser la première cartographie du protéome. Près de 75 000 internautes ont participé à sa première version. Le système a tourné pendant 2 mois. Si une seule machine avait tourné

pour ces calculs, il aurait fallu plus de 1100 années. Une seconde version du décrypton permet l'étude des interactions protéine-protéine de 4000 protéines.

GRID'5000[76] : est le nom d'un programme français , d'envergure nationale, vise à promouvoir la mise en place de GRID expérimentale de recherche constituée d'une dizaine de centres équipés de fermes d'ordinateurs personnels. L'objectif d'une telle plateforme est de fournir aux scientifiques les moyens expérimentaux pour mener à bien des recherches dans le domaine des GRIDs.

Data GRID [77]: est un projet européen pour la mise en place de GRID de calcul et de traitement de données pour l'analyse de données issues de grandes expériences scientifiques. Le projet, issu d'une initiative du CERN, réunit 6 partenaires principaux (CERN, CNRS, ESRIN, INFN, NIKHEF, PPARC) et 15 partenaires associés dont trois industriels (CS Communication et Systèmes, DATAMAT et IBM-UK) ; Le projet a pour principaux objectifs : de développer un middleware open-source fondé sur l'outil GLOBUS, de déployer des testbeds à grande échelle, de valider le concept de GRID sur différents démonstrateurs. Trois grandes applications ont été retenues pour valider le projet : Physique des particules, Bio-informatique et Observation de la terre.

EUROGRID [78]: est un projet européen pour le développement des technologies GRID autour du calcul haute performance. Le projet regroupe 6 centres de calcul haute performance dont le centre IDRIS du CNRS et deux grands utilisateurs (GIE, EADS, CCR et Deutsche Wetterdienst). Il s'appuie sur le Middleware UNICORE de la société PALLAS. Les principales applications mises en oeuvre sont : la simulation en recherche biomoléculaire, les modèles de prédiction météorologique notamment les modèles atmosphériques régionaux, le couplage de codes d'IAO notamment pour des applications dans le domaine de l'aéronautique et la simulation multiphysique.

Au niveau européen il convient également de signaler l'initiative GRIDSTART dont l'objectif est de consolider les avancées européennes dans les technologies GRID et de stimuler le développement des GRIDs dans tous les domaines scientifiques, industriels et grand public. Ils associent la plupart des projets européens (AVO, CROSSGRID, DAMIEN, DATAGRID, DATAT0AG, EGSO, EUROGRID, GRIA, GRIDLAB, GRIP).

Pour plus d'informations sur les applications de GRID, les projets, les caractéristiques, etc. la littérature est très riche, par exemple [68, 79, 80]. Pour un aperçu des modèles de programmation de GRID voir [81].

3.3.9. Evolution de GRID

3.3.9.1. GRID sémantique

Récemment, le même niveau sémantique ajouté aux services Web s'est produit également dans le GRID, C'est le concept de GRID Sémantique [1, 82]. Le concept du GRID sémantique a été proposé en 2001 par David DeRoure [1] par analogie au Web sémantique. Cette proposition repose sur l'idée que la gestion de l'état dans les services GRID apporte une dimension nouvelle pour les services collaboratifs. Le GRID sémantique est responsable de la façon que les connaissances sont acquises et la façon qu'elles sont utilisées dans un environnement donné. Il est considéré comme une extension de GRID actuel au même titre qu'est le Web sémantique pour le Web actuel dans laquelle un sens précis est donné aux informations et aux services, pour permettre aux ordinateurs et aux gens de travailler mieux en mode de coopération. D'une façon générale, GRID sémantique est vue comme une application des technologies du Web sémantique au niveau de GRID. Il repose donc sur la mise en place de métadonnées (Il s'agit d'ajouter des informations descriptives aux données et aux ressources du domaine d'application du GRID actuel) pour les ressources du GRID afin de les rendre interprétables et faciles à gérer par des machines (agents logiciels). Pour un bon aperçu du Web Sémantique, GRID et GRID Sémantique voir par exemple [83].

3.3.9.2. GRID de connaissances

[84] propose une architecture à base de GRID pour supporter des processus d'acquisition de connaissances tel que Datamining, analyse de données, etc. Il présente GRID comme meilleure approche pour supporter la mise en oeuvre de plateformes d'acquisition de connaissances parallèle et distribuées supportant des services distribués de haute performance. L'intégration des systèmes à base de connaissances et GRID est détaillée dans [85] qui décrit le concept de GRID de Connaissances comme un environnement d'interconnexion intelligent et soutenable permettant à des personnes et à des machines d'acquérir, de publier, de partager et de manager les ressources de connaissances.

3.3.9.3. GRID learning

Le choix de la technologie GRID pour apprendre est motivé par la recherche récente sur ce sujet [86, 87, 88, 89] qui fournit une contribution significative en comprenant les potentialités réelles de GRID reliées à la collaboration distante, et cela favorise le développement de l'utilisation orientée service de cette technologie dans tous les contextes et particulièrement quand la collaboration humaine est exigée (e-learning, e-gouvernement, e-santé, e-business, e-science). Le paradigme du GRID Learning vise plus à exploiter l'intelligence collective et les usages personnalisés dans les services GRID plutôt qu'à tirer profit des capacités calculatoires et/ou des effets démultiplicateurs du connexionisme. Les services GRID Learning permettront aux utilisateurs individuels ou aux groupes d'utilisateurs de gérer ou partager leur travail par une carte, la GRID-e-Card qui supporte l'assemblage de services (à la carte, donc) tout en supportant la navigation entre les différents dialogues entre communautés virtuelles dans un système de formation [90].

3.4. Conclusion

Au delà d'une solution pour les calculs intensifs, un concept fondamental introduit par GRID , pour les plateformes collaboratives, est l'Organisation virtuelle qui permet de regrouper des communautés d'utilisateurs partageant des ressources globalisées de traitement, de stockage, des instruments et des services largement distribués, quelles qu'en soient les organisations propriétaires. Le GRID est ainsi la première infrastructure de partage de ressource développée dans une perspective service. Les services GRID semblent être la prochaine évolution notoire des infrastructures distribuées grâce à l'essor des architectures orientées services et leurs services Web.

Dans les chapitres suivants où nous allons développer notre approche selon différents aspects comprenant principalement la description de l'architecture et l'expérimentation de notre environnement, nous proposons une architecture générique permettant de fournir des services collaboratifs suivant les concepts, les mécanismes, les standards et les potentialités de l'architecture service GRID OGSA, et ce pour supporter la collaboration en deux modes, synchrone et asynchrone entre des utilisateurs distants dont l'objectif commun.

Deuxième partie

Contributions

Chapitre 4

Architecture proposée

4.1. Introduction

Ce chapitre est consacré à l'architecture de COLLAGRID. COLLAGRID est l'acronyme de : Environnement collaboratif à base de GRID, qui a pour but d'offrir à une communauté virtuelle toute une gamme de services collaboratifs basés sur les concepts d'OGSA afin d'atteindre des objectifs communs. En fait, le but de notre proposition est de présenter de manière détaillée une architecture générique exprimée en terme de couche pour un espace de collaboration sûr et flexible offrant des solutions de travail intégré sur le Web, en matière de service, en deux modes (synchrone et asynchrone) à la puissance de l'architecture de service GRID OGSA couplée à une solution de bureau partagé vue comme une interface d'interaction typique.

Nous allons tout d'abord présenter l'architecture globale de COLLAGRID, en illustrant initialement la couche des ressources physiques hétérogènes et distribuées. Par la suite, nous présenterons la couche de virtualisation des ressources afin de rendre les ressources de différentes natures homogènes et réifiées dans des conteneurs de services. Puis, nous présenterons la couche de générateurs de services GRID collaboratifs synchrones et asynchrones, ainsi que celle de leurs instances, en détaillant quelques éléments clés. Nous décrirons ensuite la dernière couche qui correspond au bureau partagé vu comme un espace de collaboration interactif. Enfin, Nous mettrons le point sur le modèle intégré de COLLAGRID.

4.2. Architecture proposée

L'architecture proposée de notre environnement COLLAGRID est exprimée en terme de couche, Figure 4.1, dont chacune assure une fonction spécifique. D'une façon générale les couches hautes sont centrées sur l'utilisateur, tandis que les couches basses sont plus centrées sur le matériel (orientées vers les ordinateurs et les réseaux). Notre architecture générique, commentée de bas en haut, est composée de cinq couches :

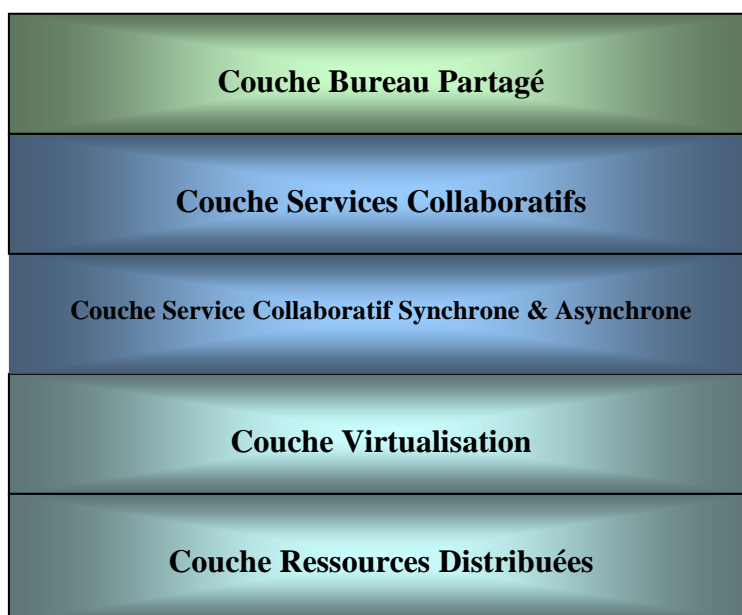


Figure 4.1-Architecture Générique de COLLAGRID

4.2.1. Couche Ressources Distribuées

La notion de ressource est centrale à notre architecture de façon générale. Les ressources correspondent aux capacités de stockage et de traitement. Ces ressources agrégées gérées de façon coordonnée sont à même de satisfaire les besoins sans cesse croissants des applications. Besoins en puissance de calcul, quantité de mémoire, capacité de disque.

Type d'application est souvent utilisé pour qualifier le type de ressources :

Les ressources de traitement: Ressources de traitement sont destinées à des applications gourmandes en calcul, c'est-à-dire dont le goulot d'étranglement est situé au niveau du processeur.

Les ressources de stockage : Ressources de stockage sont destinées à des applications gourmandes en stockage sur disque.

En effet, les deux types de ressources élémentaires partagés dans GRID sont les ressources de traitements et de stockages, qui correspondent respectivement aux unités de traitements et de stockage. Ces unités peuvent être couplées dans un même HOST qui représente une association physique entre ces dernières.

Pour mieux illustrer notre proposition nous allons utiliser des représentations graphiques pour chaque concept utilisé au niveau des couches de notre architecture.


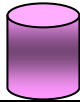
Ressource	Représentation
Ressource de traitement	
Ressource de stockage	

Tableau 4.1- Représentation des ressources de traitement et de stockage

Les ressources physiques incluent les serveurs, les sources de stockages, et les réseaux. Nous parlons de ressource locale ou ressource privée dans les cas d'agrégation de ressources sur un même site.

4.2.2. Couche Virtualisation

La seconde couche de notre architecture est celle de Virtualisation des ressources. Dans cette couche, nous considérons la virtualisation des ressources comme moyen pour apporter portabilité et passerelles trans-paradigme de façon transparente à des services. Nous définissons la virtualisation comme suit : «La virtualisation consiste en l'émulation d'une ressource (dite virtuelle) sur une autre ressource (dite réelle). Les services utilisent la ressource virtuelle comme s'ils utilisaient une ressource réelle ; la ressource virtuelle simule le comportement de la ressource réelle». Ceci permet de faire utiliser par des services des ressources pour lesquelles ils ne sont pas conçus. La virtualisation se fait à l'aide d'un package de Services.

4.2.2.1. Partage des ressources

Le partage des ressources s'effectue grâce à deux processus : D'abord réalisé par un processus de Virtualisation pour cumuler globalement en entrée la capacité disponible et fournie par des unités de traitement et de stockage géographiquement distribuées. En sortie de ce processus, il y aura des ressources virtualisées, mais qui ne sont pas directement utilisables par les couches supérieures.



Figure 4.2-Processus de partage des ressources

Un autre processus dit Réification est indispensable. Il s'agit de restituer les ressources ainsi virtualisées dans des conteneurs de services qui en auront besoins. C'est ce qu'on appelle disponibilité des ressources à la demande. Les deux processus du partage des ressources sont accomplis au niveau du noyau GRID.

Le tableau 4.2 présente les représentations graphiques des deux processus de partage des ressources : Virtualisation et Réification, ainsi que les ressources virtualisées.

Processus	Représentation
Virtualisation	
Réification	
Ressources virtualisées	

Tableau 4.2-Représentation de la Virtualisation, Réification et des Ressources virtualisées

4.2.3. Couche Services Collaboratifs Synchrones & Asynchrones

Tout au long de cette section nous décrivons les Services Collaboratifs Synchrones & Asynchrones qui visent à supporter les deux modes de collaboration, ce qui implique un environnement collaboratif flexible. Dans la suite, il est décrit pourquoi la caractéristique de service à état est indispensable pour déployer les deux types de services collaboratifs. Une des principales raisons qui nous a motivé à utiliser les services GRID est la génération dynamique de services.

4.2.3.1. Service à état

Un service à état est caractérisé par un générateur de service et des instances dynamiques de ce service. Ces services sont dits dynamiques dès lors qu'ils disposent de mécanismes qui limitent naturellement leurs instances dans le temps par un état final. Ainsi, un générateur de service peut émettre des instances qui utiliseront de la ressource et qui libéreront cette ressource en fin de vie (figure 4.3). L'avantage immédiat est un usage efficace et bien réparti des ressources physiques en fonction de leur disponibilité. Cet aspect est crucial pour la mise à l'échelle d'applications réparties sur un système distribué.

Un service dynamique est appelé transitoire (transient) dans la terminologie Grid [56] par opposition à service persistant. Un service persistant ne dispose pas de mécanisme limitant la durée de ses instances. Il est à noter qu'un générateur de services dynamiques peut être lui-même vu comme un service de type persistant.

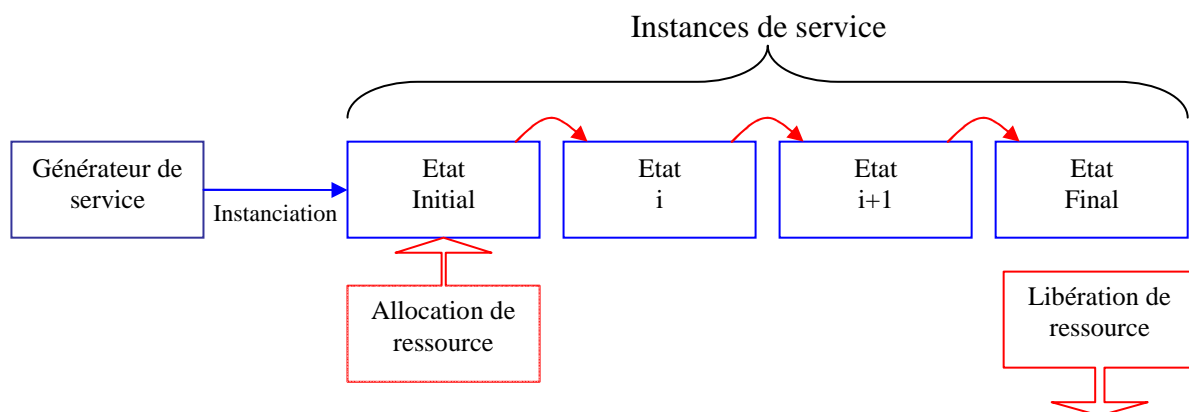


Figure 4.3- Modèle d'instanciation dynamique de services à état

4.2.3.2. Service Collaboratif Synchrones

Service Collaboratif Synchrones est défini comme un service persistant capable de générer des services à état collaboratifs caractérisé par ses capacités à garder en mémoire les résultats d'actions antérieures comme par exemple un indicateur de présence lorsque des personnes se connectent sur un service de messagerie instantanée. Dès lors, il est nécessaire de dédier de la ressource à cet effet.

Les instances de ce type de service s'utilisent en temps réel entre des personnes situées à des endroits différents. La communication à distance entre les interlocuteurs se fait donc un peu comme au téléphone c'est-à-dire en temps réel. Le but de ce type de service est d'échanger des informations le plus rapidement possible pour décider de la suite des événements ou encore de réunir différentes personnes pour les informer en même temps de certaines informations à consigner (Exemple : réunions, cours). Parmi ces types de services collaboratifs synchrones nous citons : la vidéoconférence, la messagerie instantanée, l'utilisation de tableau partagé ou navigateur partagé.

4.2.3.3. Service Collaboratif Asynchrone

Nous appelons Service Collaboratif Asynchrone le service persistant qui peut produire des services à état utilisés à des moments différents dans le temps permettant une communication en continu sans tenir compte des obstacles liés à l'espace et au temps. (ex : La messagerie électronique, le forum de discussion, le transfert de fichiers et le Wiki).

Les services collaboratifs Synchrones et Asynchrones servent donc à :

- fournir des messages aux membres de la communauté pour leur permettre d'être immédiatement informé de la vie de la communauté virtuelle ;
- gérer les membres : ajouter, supprimer ou modifier les profils ;
- gérer de nouveaux services consacrés à la communauté ;
- instancier et gérer dynamiquement de nouveaux services de la couche supérieure.
- gérer les différentes sessions de collaborations;

- tracer l'historique de la collaboration au sein de la communauté.

4.2.4. Couche Services Collaboratifs

Cette couche englobe toutes les instances dynamiques de services collaboratifs fournis par la couche inférieure (Couche Service Collaboratif Synchronique & Asynchronique) à une session collaborative.

4.2.4.1. Éléments clés

Utilisateur de service GRID : Un utilisateur de service GRID est une entité active et communicante qui fournit et utilise des services [60]. L'utilisateur de GRID peut être un membre de plusieurs communautés virtuelles (CV), et peut fournir ou utiliser plusieurs services (dans des CVs différentes). Les utilisateurs se connectent de divers emplacements et manipulent des ressources, partagées à l'aide des services support de la collaboration. Les utilisateurs de service GRID sont appelés également Participants lorsqu'il s'agit d'une session collaborative.

Organisation Virtuelle : Ensemble d'utilisateurs de service GRID (Participants) qui partagent des caractéristiques ou des buts communs. Une organisation virtuelle (OV) est une collection dynamique d'individus, et des ressources empaquetées ensemble afin de partager des ressources et des services en vue d'atteindre des buts communs. Elle devrait être vue comme un groupe de personnes qui partagent des intérêts communs ou participent à une entreprise commune, et qui se réunissent, collaborent, et communiquent de manière éloignée, virtuelle et fiable en utilisant des installations réseau, des outils et des ressources. L'OV peut être totalement décentralisée. Il est nécessaire pour un scientifique ou un laboratoire de posséder un compte utilisateur sur une organisation virtuelle donnée pour pouvoir s'identifier et effectuer des tâches.

Les organisations virtuelles existantes sont des projets à vocations scientifique 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.

Ici, dans notre contexte, on distingue l'OV (organisation virtuelle) de la CV (communauté virtuelle). Une communauté virtuelle désigne l'ensemble des participants d'une session collaborative synchrone ou asynchrone, tendit que l'OV est constituée de plusieurs CVs. On appelle CVS (communauté virtuelle synchrone) le groupe des participants qui travaillent en mode synchrones. On appelle également CVA (communauté virtuelle asynchrone) le groupe des participants qui travaillent en mode asynchrones. La notion de communauté est un élément clé dans les plateformes collaboratives.

Conteneur de services : Le conteneur se définit comme un environnement d'exécution de services où ils disposent de leurs propres ressources. Son rôle est de mettre à disposition, de manière fiable et sûre, une partie des ressources virtualisées de GRID. Un conteneur contient différents types de service GRID. Chaque service possède un identificateur. Un service peut instancier un autre service dans le même conteneur ou dans un autre. Un conteneur de services est lui-même un service GRID instancié soit par un autre service, soit par un mécanisme provenant du noyau de GRID. Un conteneur de services est alloué à (et crée pour) une et une seule CV.

Service des Autorisations de la Communauté : La relation entre une OV et un conteneur de services est réalisée par le Service des Autorisations de la Communauté (CAS) qui formalise les niveaux de droit d'utilisation (permission, restriction, interdiction, etc.) des membres de l'OV sur chaque service. Pour participer à GRID, les unités de traitement et de stockage et les utilisateurs doivent détenir un certificat signé par une autorité de certification. Il joue le rôle de passeport. Toutes les entités avec des certificats sont appelées noeud GRID. Les noeuds GRID forment des réseaux de confiance ou trusts.



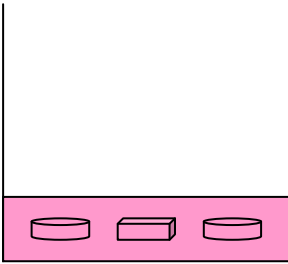


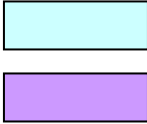


Élément	Représentation
Utilisateur de service GRID	
Organisation virtuelle	
Conteneur de services	
Service des Autorisations de la Communauté	
Générateur de service	
Instance de service	
Instanciation	
GSH : Grid Service Handle	

Tableau 4.3- Représentation des éléments clés de la couche Services collaboratifs

Session : une activité collaborative se déroule dans le contexte d'une Session. Une Session représente les activités collaboratives comprenant plusieurs services collaboratifs et un groupe de participants dit communauté virtuelle synchrone ou asynchrone, tout dépend du

mode de la collaboration. Les participants d'une session travaillent ensemble en vue d'atteindre un but commun. Au départ tous les participants sont des membres de l'Organisation virtuelle. Chaque service est accessible par un identifiant unique. Le plus souvent une URI accessible par le protocole http. Un passeport électronique est attribué à tout participant de la session.

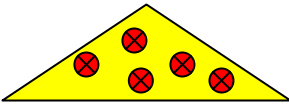
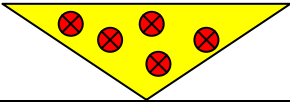
Communauté virtuelles collaboratives	Représentation
Communauté virtuelle Synchrones	
Communauté virtuelle Asynchrone	

Tableau 4.4- Représentation de différentes communautés virtuelles collaboratives

On distingue deux types de sessions :

Session collaborative synchrone comprend, tous les services collaboratifs synchrones actifs où les participants de la CV sont présents en même temps pour réaliser le travail collaboratif. Ils dialoguent de façon interactive et interagissent en temps-réel sur les services et les ressources manipulées dans la session. Un exemple de ce type de session, est une session de Visioconférence. Session collaborative asynchrone Comprend, tous les services collaboratifs Asynchrones actifs où la co-présence des différents participants de la CV n'est pas nécessaire. De ce fait, ne pouvant réaliser de dialogue interactif, les participants d'une session asynchrone ne communiquent que via des médiums asynchrones tel que le service messagerie électronique (email). Par la suite, nous présentons, à titre d'exemple, une liste non exhaustive de services Collaboratifs les plus souvent référencés.

Les instances de services collaboratifs synchrone sont:

Service messagerie instantanée : Ce type de service est dédié à la communication par échange de messages, également appelé Chat, la communication est conçue pour être instantanée, i.e. synchrone.

Service Vidéoconférence : Il permet aux utilisateurs distants de se réunir et communiquer par l'intermédiaire d'un support audio et vidéo en même temps.

Service Audioconférence : Il permet la communication audio entre les participants d'une session collaborative il permet également de diffuser et recevoir des données audio avec une qualité acceptable.

Service Tableau blanc : Le service tableau blanc partagé, comme son nom l'indique, met à disposition une surface de dessin accessible par plusieurs utilisateurs. Il permet ainsi à des utilisateurs de travailler d'une manière synchrone sur des documents 2D. Il est par exemple possible de réaliser des captures d'écran pour les annoter dans le but d'expliquer une idée ou un concept.

Service URL partagé : L'idée de la co-navigation est une extension de la navigation à un groupe d'utilisateurs, pour que les membres du groupe naviguent en commun sur des documents Web. Toute information basée sur le Web peut être partagée. Un participant peut naviguer sur le Web et synchroniser les navigateurs des autres participants.

Service Ecran partagé : Il permet de passer facilement d'un programme à l'autre. Les participants peuvent voir tous les éléments de l'écran partagé, notamment la présentation telle que : PowerPoint.

Service Présentation partagée : Un participant à une session peut spontanément partager une présentation avec d'autres membres. La visualisation peut se faire en mode plein écran.

Les instances de services collaboratifs Asynchrone sont:

Service WIKI : Il est un système de gestion de contenu de site Web qui rend les pages Web librement et également modifiables par tous les participants autorisés

Service Transfert de fichiers : Il est dédié à l'échange informatique de fichiers sur un réseau. Il permet depuis un ordinateur, de copier des fichiers depuis ou vers un autre ordinateur de la communauté virtuelle, de supprimer ou modifier des fichiers sur un ordinateur.

Service messagerie électronique : Ce service représente de loin le moyen de communication asynchrone le plus répandu et le plus utilisé de nos jours. Il désigne le service de transfert de messages envoyés via Internet vers la boîte aux lettres électronique des destinataires choisis par l'émetteur.

Service forum de discussion : Avec ce type de service les membres de la communauté virtuelle peuvent partager leurs connaissances/expériences par des échanges de messages de manière asynchrone.

Service listes de diffusion : Il permet le publipostage (en masse) d'information à un grand nombre d'utilisateurs possédant une adresse courriel. Il est le plus souvent utilisé pour collaborer sur différents sujets ou comme méthode pour diffuser rapidement des informations sur un sujet particulier

La figure ci-dessous montre les services collaboratifs synchrones et asynchrones dans un même conteneur en générant dynamiquement toutes les instances de services nécessaires à la couche supérieure.

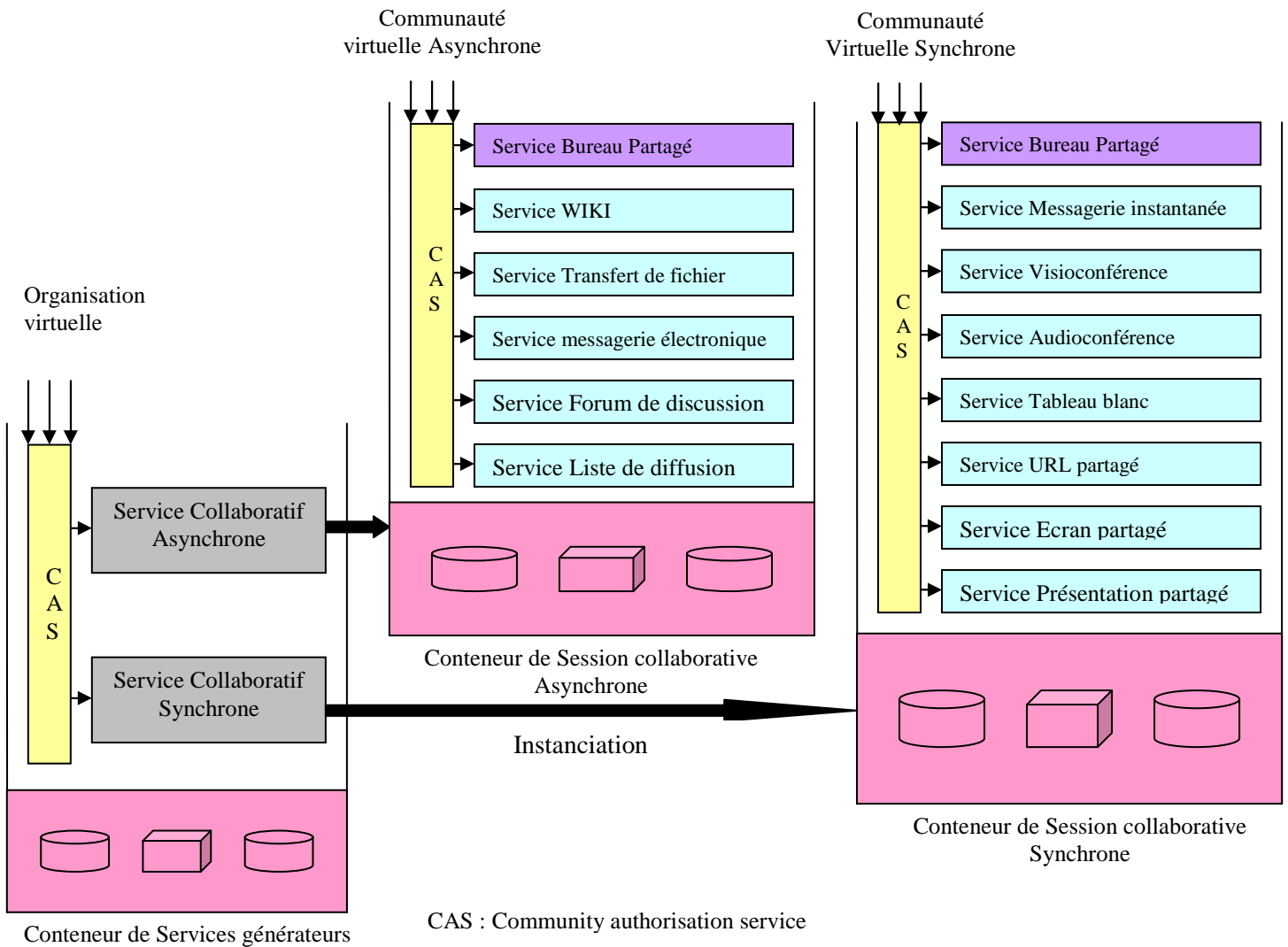


Figure 4.4-Instanciation des Conteneurs de Services Collaboratifs Synchrones et Asynchrones

Chaque conteneur de services instancié correspond à une session, il est essentiel pour, d'une part, créer un environnement d'exécution des services et, d'autre part, assigner des règles de sécurité propres à chaque groupe de participants. Ces règles de sécurité sont consignées dans une matrice d'autorisation dont la gestion est laissée libre à la communauté. Cette matrice d'autorisation lie les droits des Participants vers les différentes instances de service en cours d'exécution dans le conteneur de service. Pour tout Participant d'une session collaborative est attribué un certificat pour créer un espace sécurisé et trust. Un certificat contient des informations sur l'identité du Participant ainsi que les services qu'il peut y accéder.

Il est à noter que dans les conteneurs de services des sessions collaboratives il y a un service de type spécifique appelé : Service Bureau partagé qui vise à instancier les services du

conteneur courant dans un nouvel correspondant à une interface home machine et s'occupe également de la mise à jour des bureaux de La CV concernée, figure 4.5.

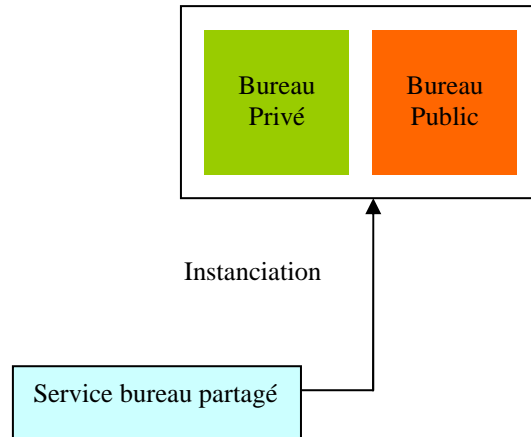


Figure 4.5-Instanciation du bureau partagé pour la communauté virtuelle

4.4 .5. La couche Bureaux partagés

A la lumière du bilan des interactions identifiées lors d'une collaboration, il nous est rapidement apparu naturel d'opter pour le modèle du bureau comme interface homme-machine et d'étendre son usage à la manière d'une interface multimodale entre humains par une combinaison de bureaux imbriqués couplés à des canaux de services GRID.

Le Bureau partagé tel que nous le définissons ici peut être vu comme un environnement global de collaboration Flexible c-à-d il prend en compte les deux modes de collaboration en permettant à plusieurs utilisateurs (travaillant sur des ordinateurs différents et distants) d'utiliser de façon collaborative un service hébergé sur un seul ordinateur. Le bureau partagé est donc considéré comme un espace de collaboration et d'interaction immanent au sein d'une communauté virtuelle et contenant l'ensemble d'instances dynamiques des services de type collaboratif synchrone et asynchrone accessibles à travers un terminal léger. Le tout est conçu de manière à rendre transparente à l'utilisateur toute la complexité de la technologie sous-jacente.

La première tentative de proposition d'un bureau virtuel de GRID a été faite dans le cadre du projet d'IST-2001 GRIDLAB (Grid Application Toolkit and Testbed) [91]. Ce projet a développé GridSphere [92]. Ceci est basé sur ad hoc GUI et est moins susceptible d'être compatible avec tout genre d'applications. Une autre tentative est Entropia [93], un système

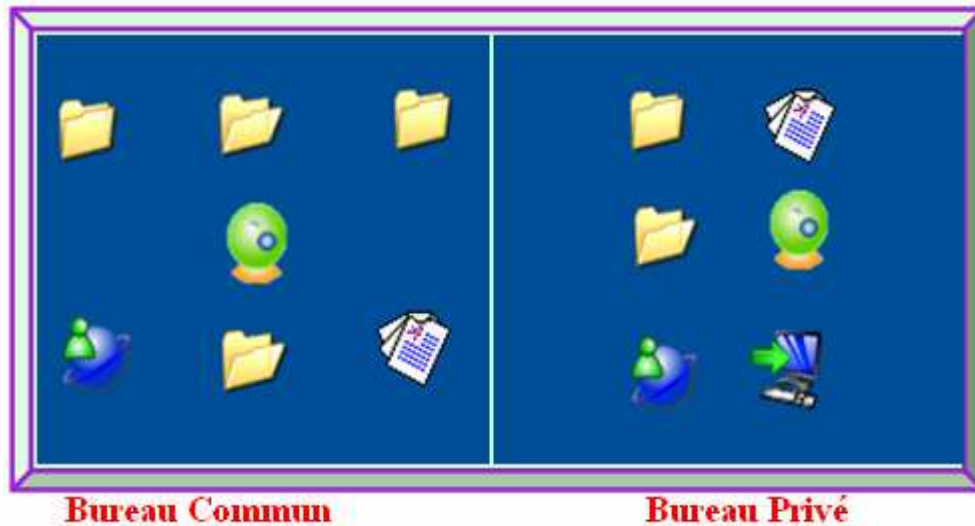
développé en 2004 à l'université de la Californie. Entropia utilise la MV (Machine Virtuelle) pour mettre en évidence le bureau de GRID. Cependant, l'utilisation d'une MV pour chaque session d'utilisateurs consomme une quantité considérable de ressource du système.

Plus récemment, il a été proposé le collabureau comme un espace de collaboration à la puissance GRID dans [94], représenté comme une autre vision de GSD [59]. Cependant, le collabureau se limite à une description en mode synchrone.

En effet, notre contribution à ce niveau s'appuie sur les deux modes de collaboration avec la possibilité d'accéder aux différents services de l'environnement collaboratif par un simple clique(One-Click) à travers un terminal léger et sans aucune installation, configuration ou mise à jour logicielle sur la machine.

Notre bureau partagé comporte deux bureaux qui peuvent être perçus simultanément :

1. **bureau privé** : propre à chaque utilisateur.
2. **bureau partagé** : commun à tous les membres et est mis à jour par la publication des contributions.



Bureau = Conteneur de Services

Icône sur bureau = service à lancer en « One Click »

Figure 4.6 - Bureau partagé

Chaque bureau est considéré comme un conteneur de services dont lequel s'exécute tout service sollicité et chaque service est symbolisé par des raccourcis (Icône par analogie au Microsoft Windows) sur le bureau qui peut être lancé en One Click. Par conséquent, on peut considérer les services à ce niveau comme des images des services collaboratifs de la couche inférieure directement accessibles par l'utilisateur de GRID.

4.3. Modèle intégré

Les couches que nous venons présenter ci-dessus sont simplifiées dans un modèle intégré illustré dans la figure 4.7. Ce modèle est vu de bas en haut. Initialement les ressources physiques de traitement et de stockage géographiquement distribuées, dites réelles, sont couplées à travers une infrastructure réseau en vue d'être virtualisées et utilisées ultérieurement par des services GRID collaboratifs. Pour ce faire, Elles subissent un processus de partage des ressources, au niveau de noyau de GRID. Une fois les ressources informatiques virtualisées, il est envisageable de mutualiser tout ou partie de ces ressources pour générer des services collaboratifs et satisfaire leurs besoins lors de l'exécution.

Dans son état initiale, l'environnement collaboratif ne dispose d'aucune instance de service GRID mais il est évolutif selon les besoins des communautés virtuelles. Deux services désactivés synchrone et asynchrone de type générateur, considérés comme des services bootables, sont créés par défaut dans un conteneur avec une portion des ressources

réifiées. Ils ont pour objectif d'instancier les services nécessaires aux sessions de collaboration. Après avoir authentifié auprès de CAS par l'intermédiaire d'un pointeur (Handle), un membre de l'organisation virtuelle, dit administrateur, accède aux services bootables pour les activer et lancer des instances de services nécessaires à la collaboration. Des autorisations d'accès aux services sont également communiquées aux autres membres sous forme d'un passeport électronique. Les instances sont générées dans des conteneurs de sessions collaboratives (environnement d'exécution des services) en fonction de leur mode et utilisant des ressources de façon dynamique. Une instance de service bureau partagé est aussi créée dans chaque conteneur visant à instancier des images de ces instances dans un conteneur bureau partagé accessible à travers un terminal léger par le biais d'un GSH.

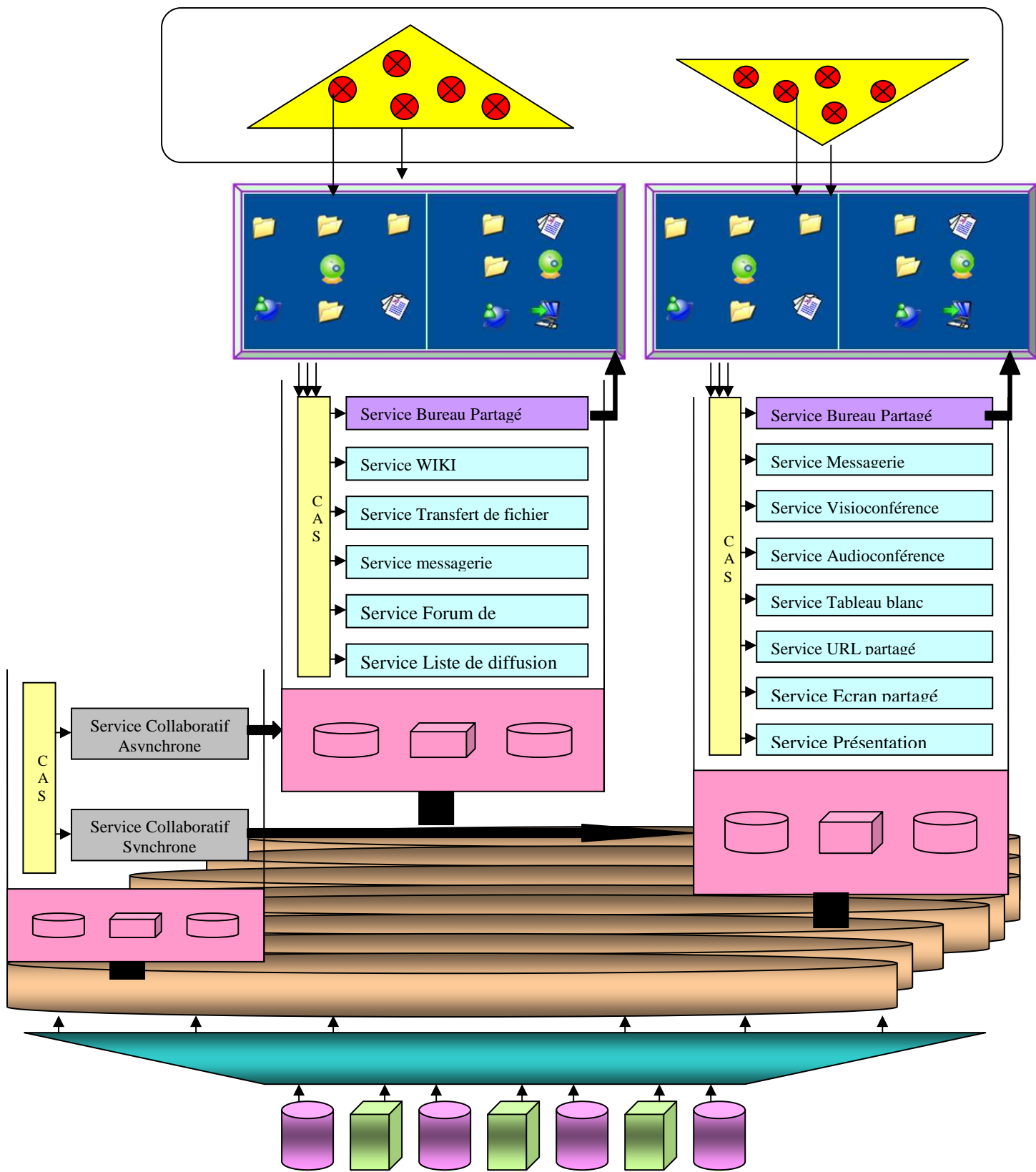


Figure 4.7- Modèle intégré de COLLAGRID

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

4.4. Conclusion

Nous avons présenté dans ce chapitre l'architecture générique de l'environnement COLLAGRID, permettant à des participants distants au sein d'organisation virtuelle de collaborer via un bureau partagé. Il est possible, via ce bureau de manipuler de manière coopérative des services collaboratifs de type GRID basés essentiellement sur les concepts de l'architecture OGSA, utilisant des ressources informatiques à la demande, pour atteindre des objectifs communs. COLLAGRID propose un support pour la collaboration, que celle-ci se fasse de manière asynchrone (les membre de la communauté virtuelle utilisent des services tels que : Service WIKI, Service transfert de fichiers, etc.), soit synchrone, en présentiel, en partageant des service genre Service visioconférence, Service tableau blanc, Service URL partagé, etc. Cet environnement nous a permis de mettre en œuvre les concepts qui ont été développé dans ce chapitre. Afin d'évaluer COLLAGRID, ainsi que nos propositions, nous avons réalisé un scénario des expérimentations que nous allons décrire au chapitre suivant.

Chapitre 5

Expérimentation

5.1. Introduction

Nous avons présenté au cours de ce mémoire un état de l'art relatif aux espaces collaboratifs sur les nouvelles orientations de la technologie GRID, ainsi qu'une proposition d'environnement collaboratif pour résoudre des problèmes au sein d'organisation virtuelle selon les concepts et les mécanismes de l'architecture de service GRID OGSA. Nous avons mis en place un environnement prototype mettant en oeuvre cette approche. L'expérimentation que nous présentons dans ce chapitre a un double objectif. L'objectif principal est d'évaluer l'impact de l'intégration du modèle de collaboration sur la construction collaborative d'ontologie partagée des turbines à gaz. Il s'agit également d'évaluer l'utilisabilité de notre environnement.

Nous décrivons dans un premier temps la turbine à gaz, nous y expliquerons en particulier les trois grandes sections et le principe de fonctionnement. Nous présenterons ensuite de manière détaillée la notion d'ontologie et la raison pour laquelle la développer, ainsi que les outils retenus dans notre expérimentation. Nous proposerons enfin un scénario de collaboration avec des résultats obtenus.

5.2. Turbine à Gaz

5.2.1. Description

La turbine à gaz [95] est un moteur thermique réalisant les différentes phases de son cycle thermodynamique dans une succession d'organes traversés par un fluide moteur gazeux en écoulement continu. Elle a pour but principal de transformer l'énergie thermique en énergie mécanique. Les turbines sont généralement utilisées pour entraîner : des compresseurs dans des stations de compression, des pompes dans des stations de pompage, des alternateurs, etc. Une turbine à gaz se compose de trois sections principales : décrites dans les paragraphes suivants.

Section Compresseur : Le compresseur est de type à flux axial. Le système à flux axial produit des débits d'air élevés, nécessaire pour obtenir des valeurs élevées de puissance utile avec des dimensions réduites. Un compresseur se compose d'une série d'étages d'aubes orientables, qui augmentent la vitesse de l'air en termes d'énergie cinétique, suivie alternativement par des étages d'aubes fixes, qui convertissent l'énergie cinétique en une pression plus élevée. Le nombre d'étage de compression est lié à la structure de la turbine à gaz et, surtout, au taux de compression à obtenir. Le compresseur sert également à fournir une

source d'air nécessaire pour refroidir les parois des directrices, des aubes et des disques de la turbine, qui sont atteints par l'intermédiaire de canaux à l'intérieur de la turbine à gaz, et par la tuyauterie de raccordement extérieur. En plus, le compresseur fournit l'air d'étanchéité aux joints à labyrinthe du palier.

Section de combustion : La section de combustion se compose d'un système comprenant une ou plusieurs chambres de combustions annulaires disposées symétriquement le long d'une circonférence, ces chambres de combustions reçoivent et brûlent du combustible au moyen d'un nombre égal des brûleurs (un pour chaque chambre de combustion). Au début, le processus de combustions est déclenché par une ou plusieurs bougies d'allumage. Une fois déclenchée, la combustion continue sans interventions, aussi longtemps que les conditions d'alimentation de combustible et d'air de combustion sont maintenues.

Section de la turbine : La section de la turbine comprend un certain nombre d'étages; chacun d'entre eux se compose d'un étage stator et d'étage rotor (aubes) ; Dans l'étage stator, les gaz à température et pressions élevées, fournis par la pièce de transition, sont accélérés et acheminés vers l'étage d'aubes du rotor montées sur un disque connecté à l'arbre de puissance. La conversion de l'énergie thermique et de la pression, en énergie cinétique a lieu dans l'étage du stator.

5.2.2. Principe de fonctionnement

La figure 5.1 montre le diagramme d'une turbine à gaz à deux arbres [96]. Ce diagramme est utile pour comprendre plus facilement le rôle de chacun de ses composants et la relation entre eux, ainsi que pour construire l'ontologie.

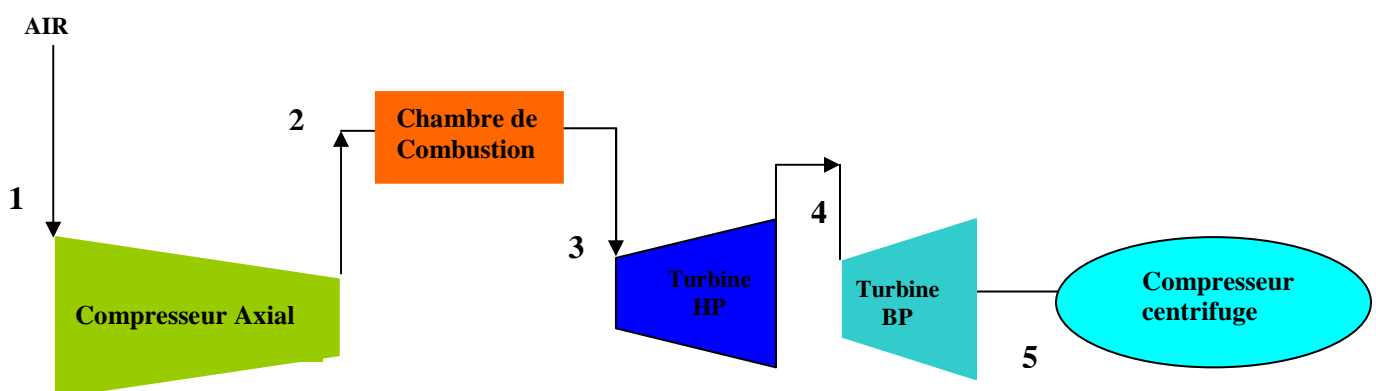


Figure 5.1 -Diagramme de fonctionnement de la turbine à gaz à deux arbres

L'air entre dans le compresseur au point (1), qui représente les conditions de l'air atmosphérique. Il est comprimé à l'intérieur du compresseur et sort dans la condition indiquée au point (2). Pendant la transformation de (1) à (2), aucune chaleur n'est transférée à l'air mais la température de l'air augmente, en raison de la compression poly-tropique, jusqu'à une valeur qui change en fonction du modèle de la turbine à gaz et de la température ambiante.

Après avoir traversé le compresseur, l'air entre dans la section de combustion, pratiquement dans les mêmes conditions de pression et de température qu'au point (2). Le combustible est injecté dans la chambre de combustion par l'intermédiaire d'un brûleur, et la combustion a lieu à une pression pratiquement constante. La transformation entre les points (2) et (3) ne représentent pas seulement la combustion. En effet, la température du processus réel de combustion, qui a lieu dans des conditions pratiquement stœchiométrique, atteint localement, dans la zone de combustions à côté du brûleur, des valeurs (d'environ 2000°C), qui sont trop hautes pour la résistance des matériaux en aval. Par conséquent, la température finale de la transformation du point (3), est inférieure, car c'est le résultat du mélange des gaz de combustion primaire avec l'air de refroidissement et de dilution comme décrit précédemment. La transformation suivante, entre les points (3) et (4), représente l'expansion des gaz à travers l'Étage haute pression qui entraîne le compresseur axial pour comprimer l'air de combustion. La transformation entre les points (4) et (5), représente la sortie d'étage HP et l'entrée d'étage BP (détente poly tropique). Qui sert à entraîner la charge (dans notre cas le compresseur centrifuge). En aval de la section (5), des gaz sont évacués dans l'atmosphère.

5.3. Ontologie

5.3.1. Notion

Le mot Ontologie vient du grec *Ontos* pour être et de *Logos* pour univers. C'est un terme philosophique introduit au XIX^{ème} siècle caractériser l'étude des être humains. Le mot Ontologie possède également différentes significations et demeure assez ambiguë. Il y a une vingtaine d'années la communauté de la représentation de la connaissance transforme ce concept philosophique en objet: une ontologie. Dans un sens large, on peut adopter pour la notion d'ontologie la caractérisation suivante [97] : «Une ontologie peut prendre différentes formes, mais elle inclura nécessairement un vocabulaire de termes et une spécification de leur signification. Cette dernière inclut des définitions et une indication de la façon dont les concepts sont reliés entre eux, les liens imposant collectivement une structure sur le domaine

et contraignant les interprétations possibles des termes.» Cette caractérisation convient pour des objets divers tels des glossaires, des terminologies, des thesaurus et des ontologies (au sens strict), mis en œuvre par différents professionnels (ingénieurs de la connaissance, terminologues, bibliothécaires, traducteurs) et se distinguant suivant que l'accent est mis sur les termes ou leur signification.

Gruber a proposé en 1993 une définition différente qui sert actuellement de référence : «Une ontologie est une spécification explicite d'une conceptualisation.» [98]. Cette définition fut reprise et modifiée par Borst en 1997, qui déclara : « An ontology is a shared specification of a shared conceptualization » [99]. Cette dernière définition met en avant le caractère collaboratif d'une ontologie, qui prend son sens dans un contexte partagé.

La plupart des ontologies sont structurées au moyen de la relation «est un» de subsomption, ou de généralisation, entre concepts (qui organise les concepts en une hiérarchie). La relation Tout-Parties «est composé de» est également utilisée. La taille des ontologies varie de quelques dizaines de concepts à plusieurs dizaines de milliers de concepts. Elles peuvent être générales (ontologies génériques) ou ne couvrir qu'un domaine particulier (ontologies de domaine). Les ontologies peuvent être informelles, formelles ou opérationnelles. Dans ce dernier cas, elles sont spécifiées dans un langage de programmation.

En conclusion, une ontologie peut être donc définie comme un ensemble d'informations dans lequel sont définis les concepts utilisés dans un langage donné et qui décrit les relations logiques qu'ils entretiennent entre eux.

5.3.2. Contenu d'une ontologie

Une ontologie contient donc tous les éléments nécessaires pour décrire de manière formelle un ensemble de connaissances. Les éléments nécessaires pour effectuer cette description sont spécifiés dans [100], basé sur [98] :

- **Des concepts** (encore appelés classes), qui représentent la brique de base de la construction ontologique. Pour parler simplement, un concept désigne « n'importe quelle chose au sujet de laquelle on peut dire quelque chose, et peut ainsi aussi bien être la description d'une tâche, d'une fonction, d'une action, d'une stratégie, d'un raisonnement, etc. » [100]. A noter que les concepts sont généralement organisés sous la forme d'une

taxonomie, et que, parfois certains considèrent que des taxonomies peuvent être des ontologies [101].

- **Des relations** : les relations représentent un type d'interaction entre deux ou plusieurs concepts. Des exemples de relation sont subclass-of ou encore connected-to.
- **Des fonctions** : les fonctions sont un cas particulier de relation à n éléments où le dernier élément de la relation est unique pour les n-1 premiers éléments.
- **Des axiomes**, qui représentent des propositions définies comme toujours vraies.
- **Des instances** pour représenter les éléments.

5.3.3. Pourquoi développer une ontologie ?

Toute activité humaine spécialisée développe son propre jargon (langue de spécialité) sous la forme d'une terminologie et d'une conceptualisation associée spécifiques. L'existence de tels jargons entraîne des problèmes de compréhension et des difficultés à partager des connaissances entre les acteurs de l'entreprise, les services d'une entreprise, les entreprises d'une industrie, qui font des métiers différents. Fondamentalement, le rôle des ontologies est d'améliorer la communication entre humains, mais aussi entre humains et ordinateurs et finalement entre ordinateurs. L'existence de vocabulaires différents au sein d'une entreprise (ex : bureau d'études) ou d'une industrie (ex : constructeur turbine, constructeur automobile) constitue un frein à la collaboration et aux partenariats. Les enjeux touchent donc directement la compétitivité de l'entreprise. Partager la compréhension commune de la structure de l'information entre les agents, est une des raisons les plus courantes qui conduit à développer des ontologies [102]. Pour l'entreprise, l'ontologie sert à :

- améliorer la compréhension entre les employés,
- favoriser la diffusion des information et leur exploitation,
- promouvoir une nouvelle approche de conception des systèmes d'information (réutilisation et interopérabilité de logiciels).

Pour ces besoins de standardisation du vocabulaire, une terminologie ou une ontologie informelle peuvent suffire [103].

5.3.4. Classification des ontologies

Les ontologies peuvent être de nature très diverse. Afin de mieux s'y retrouver, un certain nombre de classifications ont été proposées. La plus courante des classifications d'ontologies est la classification selon l'objet de conceptualisation [104]. On peut ainsi distinguer sept catégories [100] :

Les ontologies de représentation des connaissances : les ontologies de représentations des connaissances sont utilisées pour formaliser un modèle de représentation des connaissances.

Les ontologies supérieures (aussi appelées ontologies de haut niveau) : En particulier, les ontologies de haut niveau modélisent les concepts les plus généraux que l'on puisse définir.

Les ontologies génériques : elles contiennent des concepts généralistes, mais moins abstraits que ceux contenus dans les ontologies de haut niveau. On pourra réutiliser dans plusieurs domaines les connaissances que l'on y trouve [104].

Les ontologies des tâches [105] : ce type d'ontologie sert à modéliser les tâches d'un problème ou d'une activité donnée.

Les ontologies de domaine sont réutilisables à l'intérieur d'un domaine donné et modélisent le vocabulaire à l'intérieur de ce domaine. La plupart des ontologies existantes sont des ontologies de domaine [104].

Les ontologies de tâches-domaine : ce sont des ontologies de tâches spécifiques à un certain domaine.

Les ontologies d'application. Il s'agit du type d'ontologie le plus spécifique [104]. Les concepts que l'on trouve dans ce genre d'ontologies modélisent les concepts d'un domaine particulier dans le cadre d'une application donnée.

Parmi les outils de construction des ontologies on peut citer : ONTOLINGUA, ODE (Ontology Design Environment) et le plus célèbre PROTEGE 2000.

5.4. Outils utilisés

5.4.1. Protégé2000

Protégé2000 [106, 107] successeur de ProtégéWin, est un outil, une plate-forme et une librairie d'ontologies, qui permettent : 1) de construire une ontologie du domaine, 2) de personnaliser des formulaires d'acquisition de connaissances et 3) de transférer la connaissance du domaine. Il a été créé à l'Université de Stanford et est très populaire dans le domaine du Web Sémantique et au niveau de la recherche en informatique. Protégé est développé en Java. Il est gratuit et à code source libre.

Des « plugin » pour les langages RDF, DAML+OIL et OWL ont été développés pour Protégé. Ces « plugin » permettent d'utiliser Protégé comme éditeur d'ontologies pour ces différents langages, de créer des instances et les sauvegarder dans la plupart des formats d'ontologies RDF, RDFS, OWL, etc. Il est également possible de raisonner sur les ontologies en utilisant un moteur d'inférence général tel que JESS ou des outils d'inférence spécifiques au web sémantique basés sur des logiques de description tels que RACER. Ces deux outils peuvent être facilement intégrés à Protégé [108].

Le modèle de connaissances sous-jacent à Protégé2000 est issu du modèle des frames et contient des classes (concepts), des slots (propriétés) et des facets (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. Protégé2000 se veut compatible avec le modèle OKBC "The Open Knowledge-Base Connectivity" mais en diffère sur certains points. Par exemple, une frame, qui est soit une classe, soit un slot, soit une facet, est instance d'une classe et une seule dans Protégé2000 ce qui n'est pas toujours le cas dans OKBC, où une même frame peut d'ailleurs représenter plusieurs objets conceptuels. Protégé2000 autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie [109].

5.4.2. Compendium

Compendium [110] est un médium visuel dialogique de modélisations du discours concernant des problèmes à résoudre, directement issu du prototype gIBIS de Concklin ainsi que du produit QuestMap. Un outil qui, entre les mains d'utilisateurs compétents, facilite la capture et la structuration des idées, et permet non seulement de modéliser le discours, mais aussi de modéliser les domaines de problèmes d'une manière qui invite et structure les contributions, et ce de manière synchrone ou asynchrone. Il est optimisé afin d'être utilisé

dans ce qui constitue sans doute l'environnement d'utilisation le plus extrême pour un outil de représentation de la connaissance : la modélisation collaborative en temps réel. Le logiciel est une application Java gratuite fonctionnant sur toutes les plateformes et incluant le code source. Téléchargements et autres ressources communautaires sont coordonnés par le Compendium Institute, une association à but non lucratif [111]

L'approche Compendium porte l'accent sur la modélisation collaborative de l'information, des idées et arguments, afin d'obtenir un ajout de valeur immédiat pour les utilisateurs (mémoire à court terme utile), ainsi que de poser les bases d'une mémoire à long terme nécessaire à la gestion des connaissances. L'affichage de Compendium présente un certain nombre d'affordances visuelles permettant de saisir d'un trait de l'information concernant l'état d'une analyse, ce qui n'est immédiatement évident ni dans un document textuel traditionnel, ni dans d'autres approches de cartographie conceptuelle. Cela inclut les problèmes irrésolus, les idées concurrentes, dans quelle mesure des faits extérieurs sont utilisés pour renforcer les idées, ainsi que la 'profondeur' de réutilisation de noeuds et d'étiquetage (un indicateur du degré de modélisation utilisé). Compendium permet d'exporter le Map sous forme HTML, XML, JPG etc. Compendium est actuellement utilisé dans des grands projets tel que la représentation des métadonnées scientifiques sur l'exploration de la planète Mars de la NASA, [112], etc.

5.5. Le Scénario

Pour valider notre approche nous avons créé une petite communauté virtuelle appelée CVOnto constituée d'étudiants ayant une petite expérience dans le domaine des turbines considérés comme Experts de domaine et un spécialiste des sciences cognitives situés à des endroits différents dont l'objectif est la construction collaborative d'ontologie partagée des turbines à gaz en vue d'être utilisée plus tard pour remplacer les experts en leur absence. Pour ce faire, Nous nous sommes fondés sur notre expérience en utilisant Compendium comme service dialogique pour la modélisation du discours des membres de la communauté durant la session collaborative et Protégé-2000 [106] comme service d'éditeurs d'ontologies. De ce fait le scénario de collaboration se déroule en deux grandes étapes illustrées dans la figure suivante :

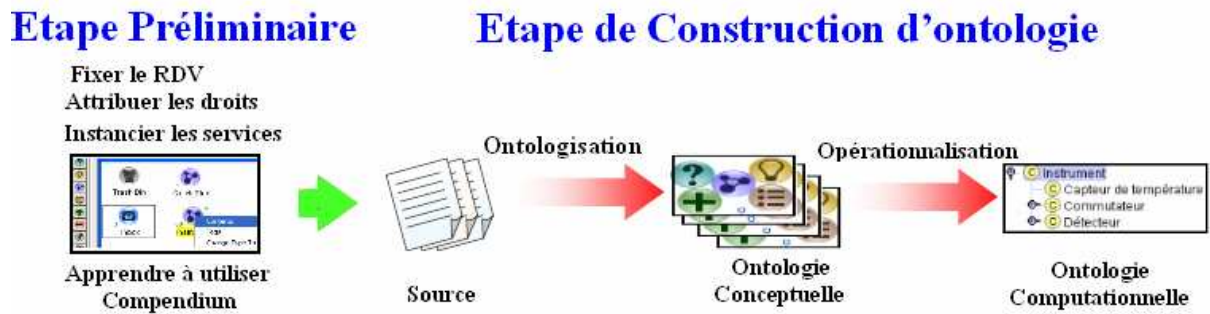


Figure 5.2- Déroulement du scénario de collaboration pour construire une ontologie partagée

5.5.1. Etape Préliminaire

Cette situation initiale consiste à planifier la réunion des membres de la communauté virtuelle qui vont participer à la session collaborative. Un Service Messagerie Electronique est utilisé pour que les participants se mettent d'accord sur un rendez-vous. Puis, à chaque participant est attribué un passeport électronique qui correspond à un service d'authentification reposé sur CAS (Community Autorisation Service) et qui consiste à définir les accès aux sessions ainsi qu'aux services disponibles après une opération d'identification. Par la suite, l'administrateur s'occupe de l'instanciation des services nécessaires à cette collaboration. Initialement ces instances sont : Service Visioconférence pour se réunir et communiquer en même temps, Service Messagerie instantanée pour échanger des messages, Service Compendium pour modéliser le discours concernant l'ontologie et Service Protégé 2000 pour décrire formellement cette dernière. Chaque membre de la communauté virtuelle peut accéder à la session collaborative et y participer grâce à un terminal léger connecté directement via Internet au serveur Web. Dans ce cas précis, le terminal léger tel que nous le définissons ici peut être vu comme un navigateur Web compatible JVM (Java Virtual Machine). Enfin une fois les membres de la communauté virtuelle se sont réunis, l'administrateur commence à leur apprendre comment utiliser le service Compendium tout en respectant la stratégie définie dans le tableau 5.1, (voir figure 5.7) .

5.5.2. Etape de construction de l'ontologie

Pour les travaux que nous présentons dans cette étape, n'étaient pas de proposer de nouveaux concepts en termes d'ingénierie ontologique, mais d'utiliser les principes existants pour les adapter à notre situation. En effet, la méthode que nous avons utilisée pour la création de notre ontologie s'inspire de la méthode Ontospec [113], du fait de sa simplicité de mise en oeuvre. Dans cette étape du scénario, on se limite aux instruments de la turbine en se

focalisant sur la description présentée dans [114]. Les instruments sont prévus pour contrôler les paramètres de la turbine et du système de l'huile de graissage et assurent des fonctions de commande d'avertissement et de déclenchement.

L'entité « instruments » est une classe qui regroupe tous les instruments qui surveillent la turbine tels que : capteur de température, détecteur de gaz, indicateur de pression, thermostat, etc.

Une première phase d'*ontologisation*, qui correspond à « l'acquisition et la modélisation des connaissances ontologiques des turbines à gaz ». Dans cette phase, les participants (experts) échangent leurs points de vue sur les termes utilisés ainsi que sur l'ensemble de sources (documents, Base de données, interviews d'experts enregistrés, Site Internet ...) dont ils auront besoin pour construire leur ontologie de turbine à gaz. Un service de visioconférence en technologie Flash est utilisé pour la communication.

Pour bien capturer leur idées et modéliser leur discours, les experts définissent chaque concept mis en place dans l'ontologie au travers de Compendium partagé en utilisant des nœuds et des flèches suivant les formes illustrées dans le tableau suivant :







Représentation	Type du nœud	Signification
	Map	Une classe principale contenant un ensemble de sous classes, propriétés et instances. Ex : Instrument
	Décision	Une classe à la quelle les participants se sont mis d'accord. Ex : la classe Commutateur
	Question	<p>Pour poser une question ou présenter un problème. Généralement ce genre de nœud est utilisé pour identifier les sous concepts (sous classe), les slots (propriétés) et les instances du concept en question, en effet ce nœud peut être représenté de la manière suivante :</p> <p><i>Concepts = Nœud question (sous-classe ?, propriété ?, Instance ?).</i></p> <p>Ex : poser des questions pour décrire les sous classes, les propriétés et les instances de la classe Commutateur.</p>
	Réponse	Pour répondre à une question posée par l'un des participants sur les sous classes d'une classe c-à-d fournir des éventuels sous concepts à une classe. Un nœud de type réponse lui même peut être considéré comme un concept. Ex : les sous classe de Commutateur sont : Commutateur de position, Commutateur de pression (pressostat), Commutateur – niveau.
	Pro	Pour décrire des propriétés d'une classe, Ex : les propriétés de la classe Commutateur niveau sont :Description, Référence , Composant surveillé,etc.
	Liste	Pour représenter les instances des concepts, ex : Commutateur de niveau a une seule instance LS2

Tableau 5.1- Illustration des nœuds utilisés pour modéliser le discours des membres de la CV dans la description de la turbine à gaz

Les flèches sont employées par les participants de la CVOnto pour lier les réponses aux questions c-à-d pour répondre à une question il faut créer un nœud de type réponse et le relier par une flèche au nœud qui modélise une question.

La figure 5.3 montre un exemple de modélisation de la discussion autour de la classe Commutateur en décrivant explicitement ses sous classes, et les instances et les propriétés de

chacune d'entre elles en répondant respectivement aux questions représentées par des nœuds de type question dont les étiquettes sont : sous-classe ?, propriété ? et Instance ?

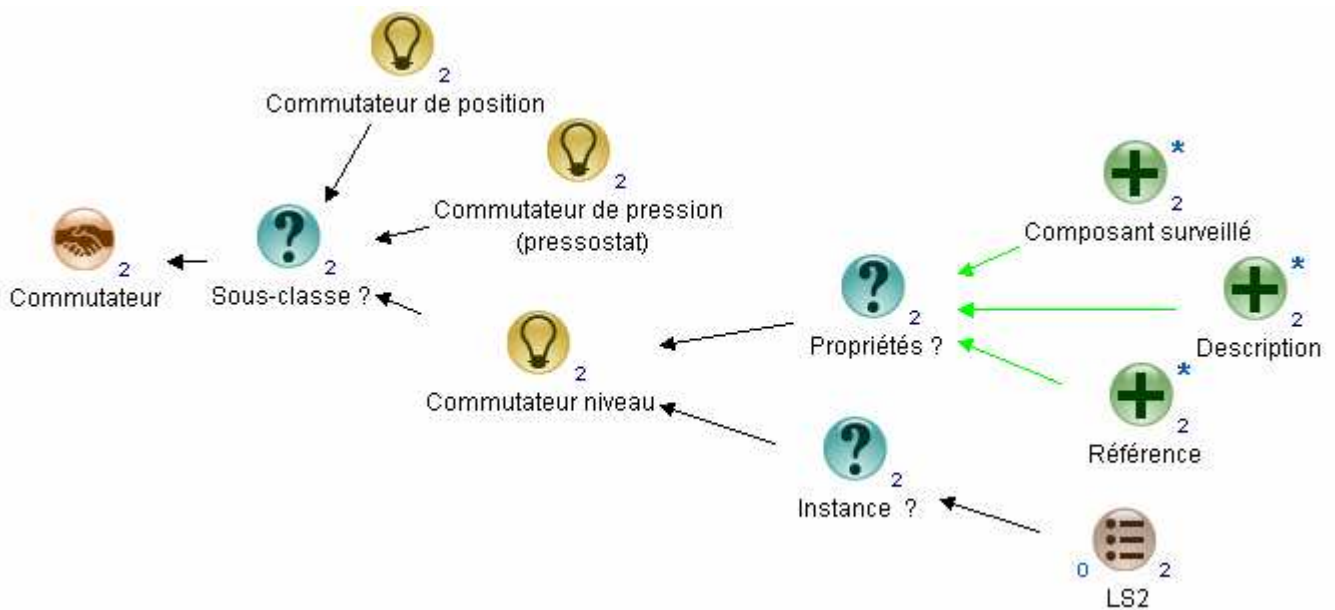


Figure 5.3 - Exemple de modélisation de discussion autour de la classe Commutateur par Compendium

Cette structuration des idées est une méthodologie formelle explicite et facilement compréhensible du côté ingénieur cognitif pour l'aider à transformer aisément par la suite les captures des idées d'experts en ontologie interprétable par Protégé 2000.

Cette étape aboutit à une ontologie conceptuelle générée par Compendium au format HTML, indépendante de l'implémentation sur Protégé 2000, pour laquelle la définition de chaque classe contient les éléments suivants (voir la figure 5.4 ci-dessous) :



Figure 5.4- Exemple de définition de la classe Commutateur générée par Compendium de genre HTML

En cours de session, l'un des participants propose d'importer un Service URL dans le bureau partagé en cliquant sur son icône sur le bureau afin de pouvoir co-naviguer sur le site Internet du constructeur des turbines à gaz SIEMENS. Immédiatement des instances de ce service se lancent dans les bureaux des membres de la communauté virtuelle, figure 5.10. Ils peuvent à leur tour diffuser leur bureau pour échanger leurs impressions sur ce service.

Une deuxième phase d'*opérationnalisation*, permettant de passer de l'ontologie conceptuelle à l'ontologie computationnelle (dans le sens ontologie décrite en Protégé 2000). Cette étape correspond aux choix de représentation, ainsi qu'à la transposition des affirmations réalisées dans l'ontologie conceptuelle vers le formalisme OWL. Nous avons utilisé le service Protégé pour nous assister dans cette étape. A partir du fichier HTML de différentes définitions le cognicien crée l'ontologie en protégé 2000 en vu d'être visualisée, discutée et validée par les experts de domaine, Figure 5.5. Cette étape aboutit à une ontologie des turbines à gaz validée.

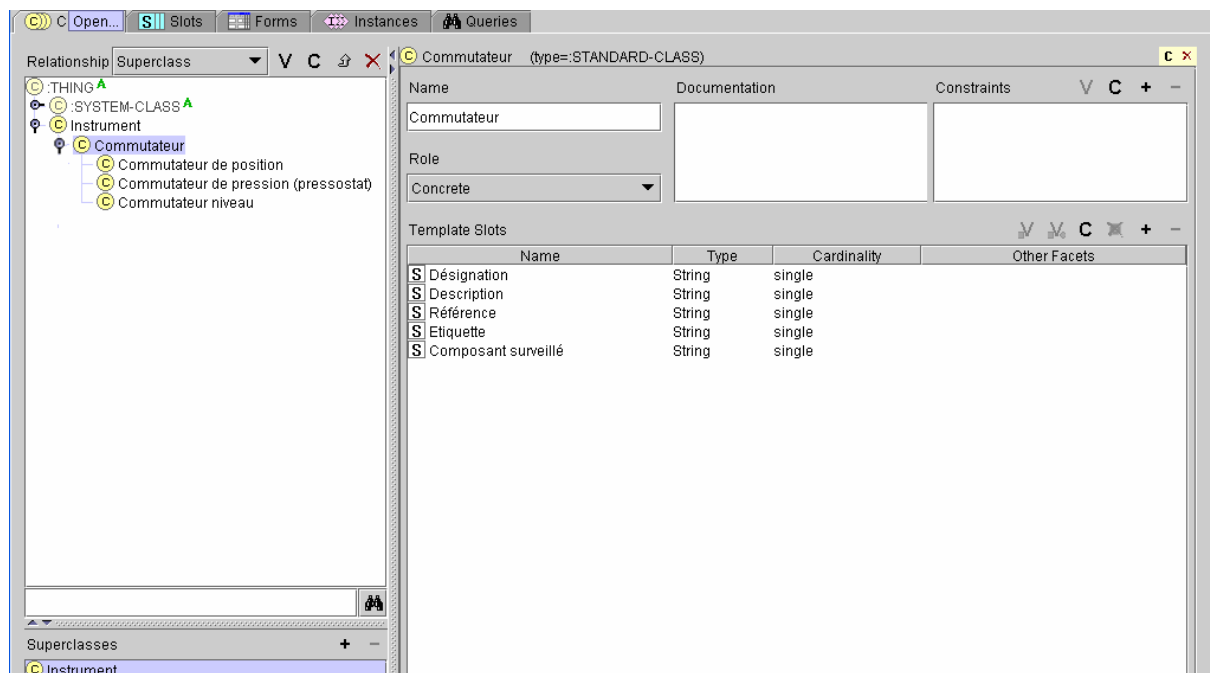


Figure 5.5- Exemple de représentation de la classe Commutateur en Protégé2000

Une première solution technique que l'on pourrait envisager pour expérimenter notre environnement consisterait à utiliser un outil de partage d'application, comme l'outil VNC (Virtual Network Computing – réseau virtuel de calcul) initialement développé à l'Université de Lancaster (R.U.) [115, 116, 117, 118] qui a ensuite été intégré à la plate-forme collaborative Platine [119, 22] développée au LAAS-CNRS. Le logiciel VNC permet de visualiser le fonctionnement d'un ordinateur A et d'interagir avec lui à partir de n'importe quel autre ordinateur raccordé. Notre architecture (Figure 5.6) montre les différents composants du système :

Serveur VNC : VNC est l'acronyme de Virtual Network Computing conçu en tant que logiciel libre fonctionnant sur des plateformes très diverses. Il se base sur le protocole Remote

Frame Buffer (RFB) et permet d'utiliser virtuellement un bureau à travers un réseau ou Internet depuis n'importe quel ordinateur connecté à Internet et possédant un navigateur web compatible Java ou une application cliente VNC. Le module « serveur » installé sur la machine dont on héberge tous les bureaux partagés et privés de la communauté virtuelle. Plusieurs clients peuvent se connecter en même temps sur un même serveur VNC.

Serveur Web : Il permet aux participants de la communauté virtuelle d'accéder à leurs bureaux partagés à partir d'un navigateur (browser) installé sur leur ordinateur distant. Le serveur Web est connecté d'une part au serveur VNC par l'intermédiaire du protocole RFB et d'autre part aux utilisateurs distants de la session collaborative par le moyen de l'applet VNCViewer pour pouvoir prendre contrôle à leurs bureaux.

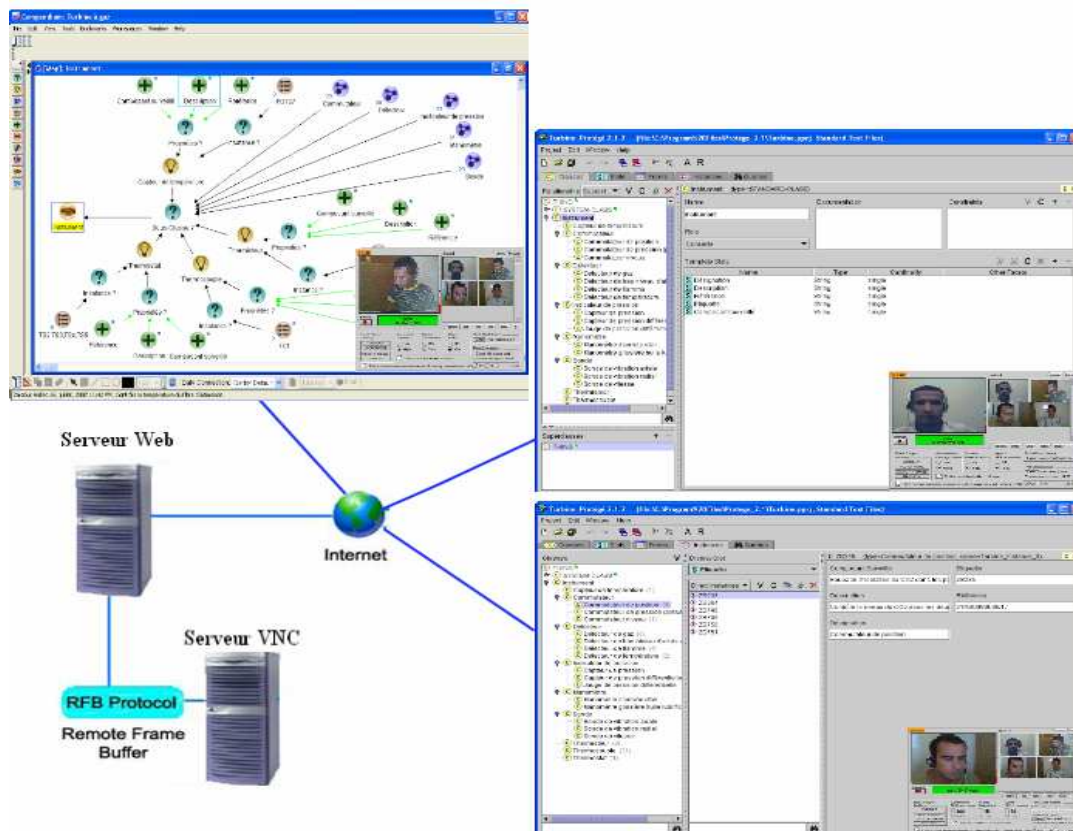


Figure 5.6- Architecture générale de notre système collaboratif

Le système permet de partager Compendium et Protégé 2000 en terme de service entre les membres de la CVOnto, Dans ce cas précis, Compendium et Protégé 2000 sont considérés comme deux services collaboratifs synchrones partagés.

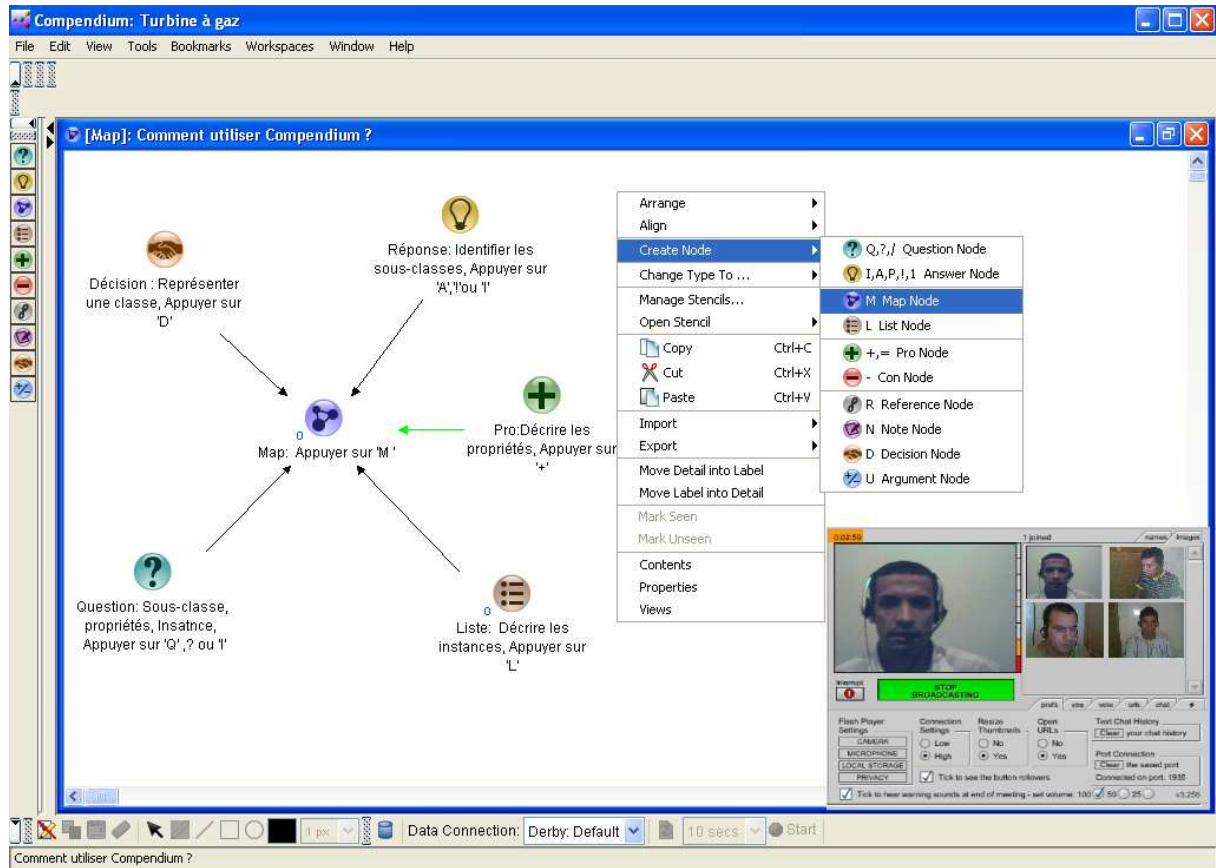


Figure 5.7- L'administrateur apprend aux membres de la CVOnto comment utiliser le service Compendium

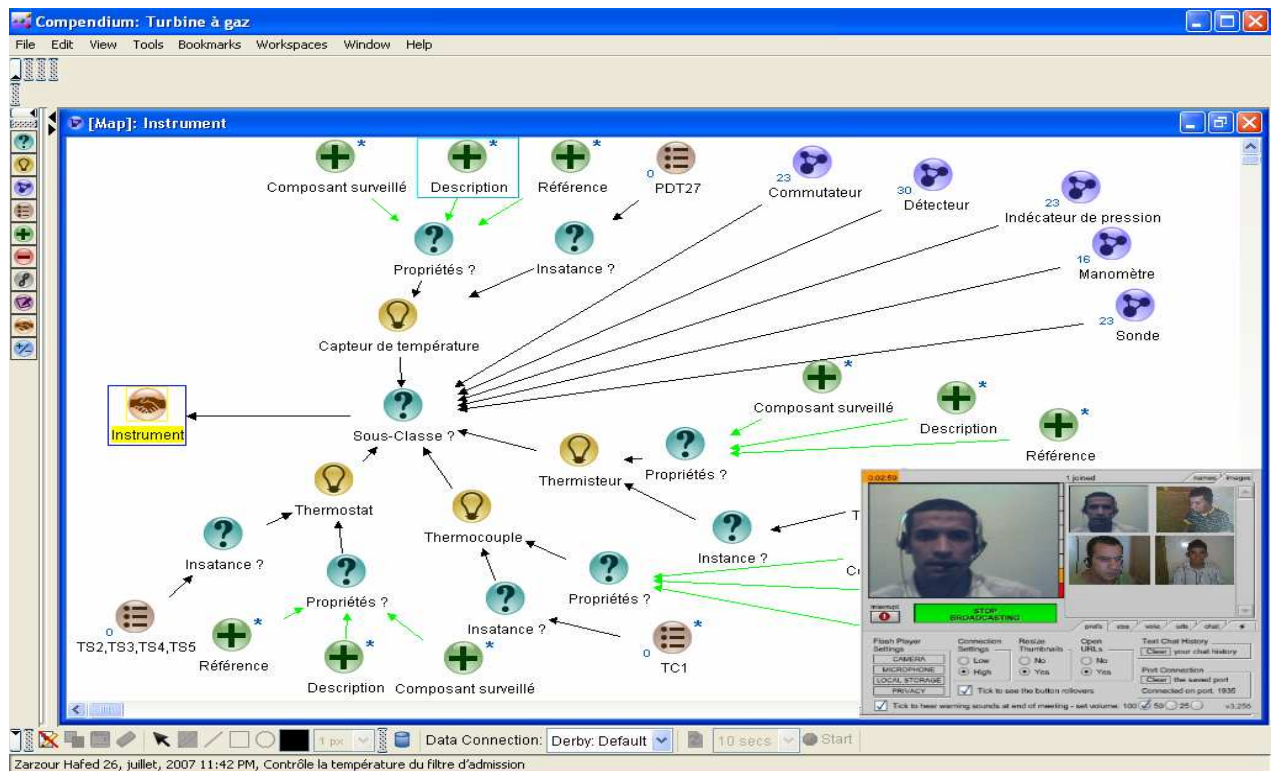


Figure 5.8- Discussion autour de la classe Instrument par le biais du service Compendium partagé

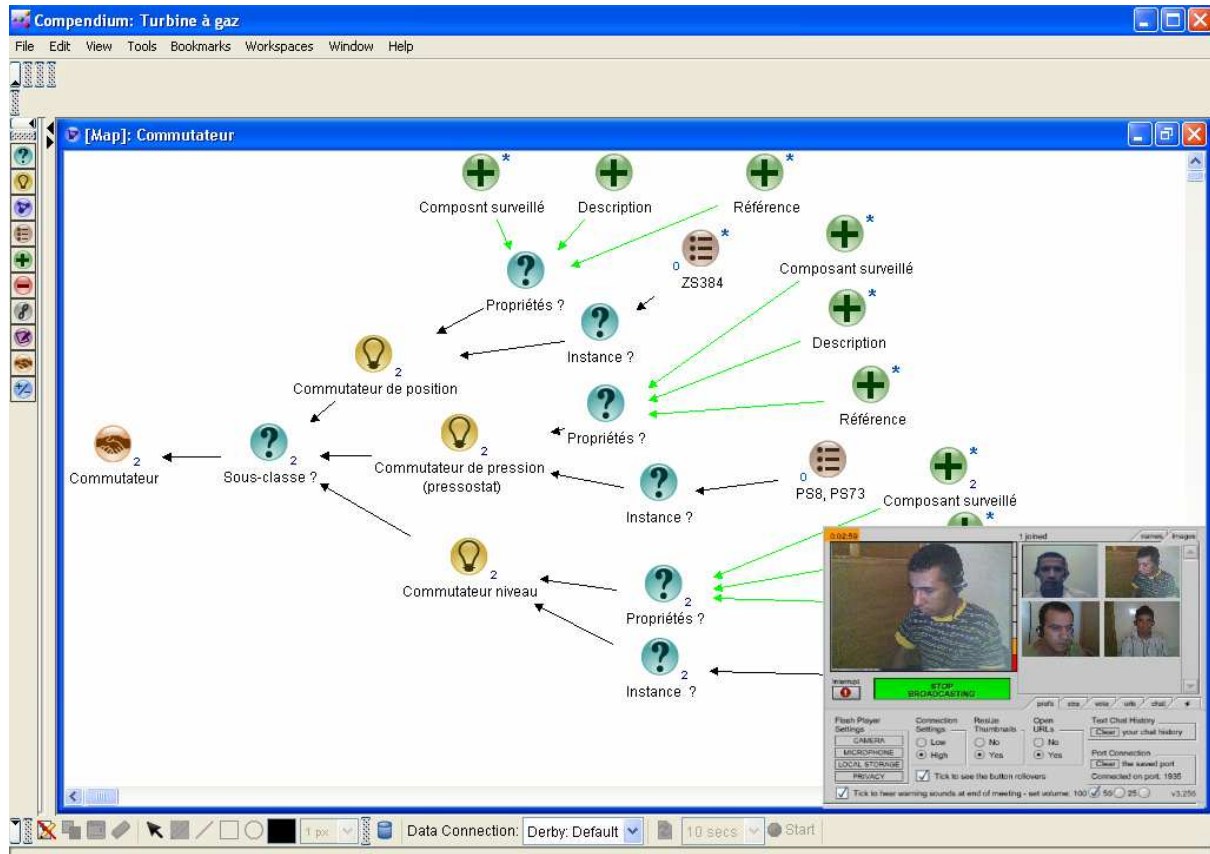


Figure 5.9- Exemple de description de la sous-classe Commutateur en Compendium partagé

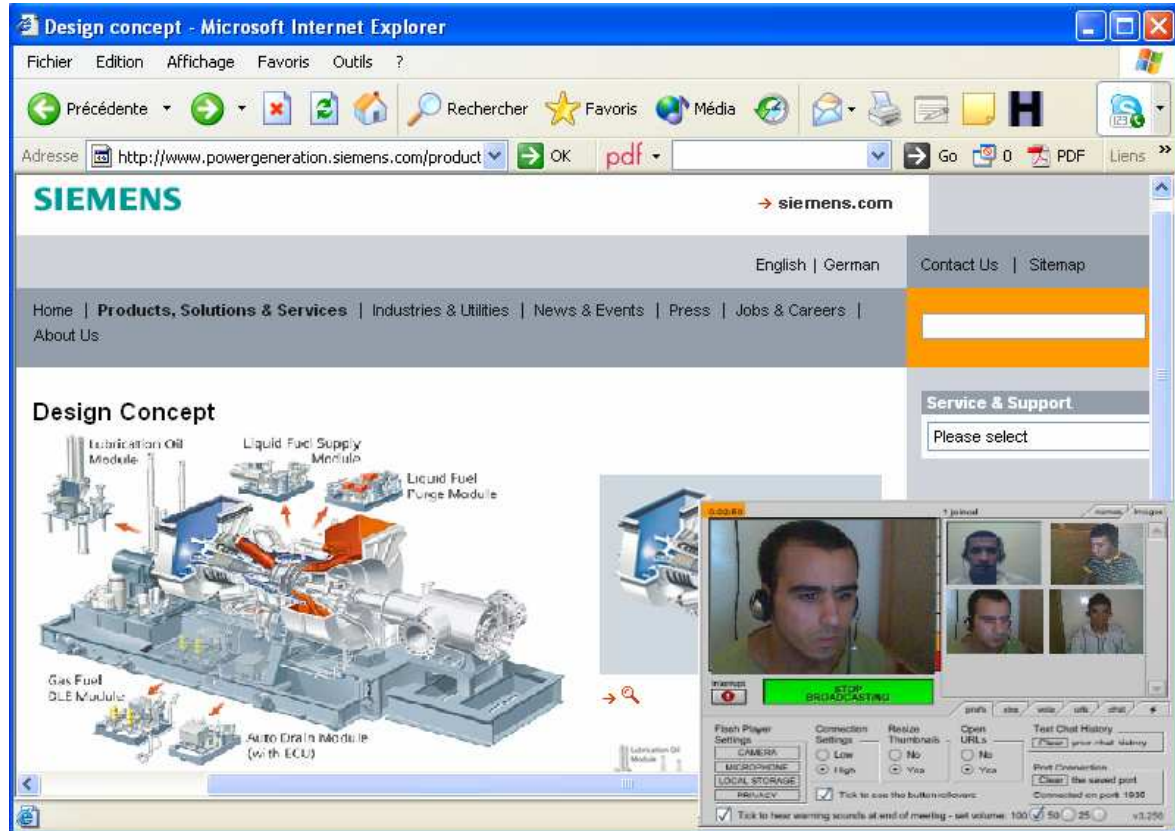


Figure 5.10- L'un des participants invite les membres de la CVOnto à consulter un site des turbines en important un Service URL dans le bureau partagé



Figure 5.11-Génération de la version HTML de la classe Instrument

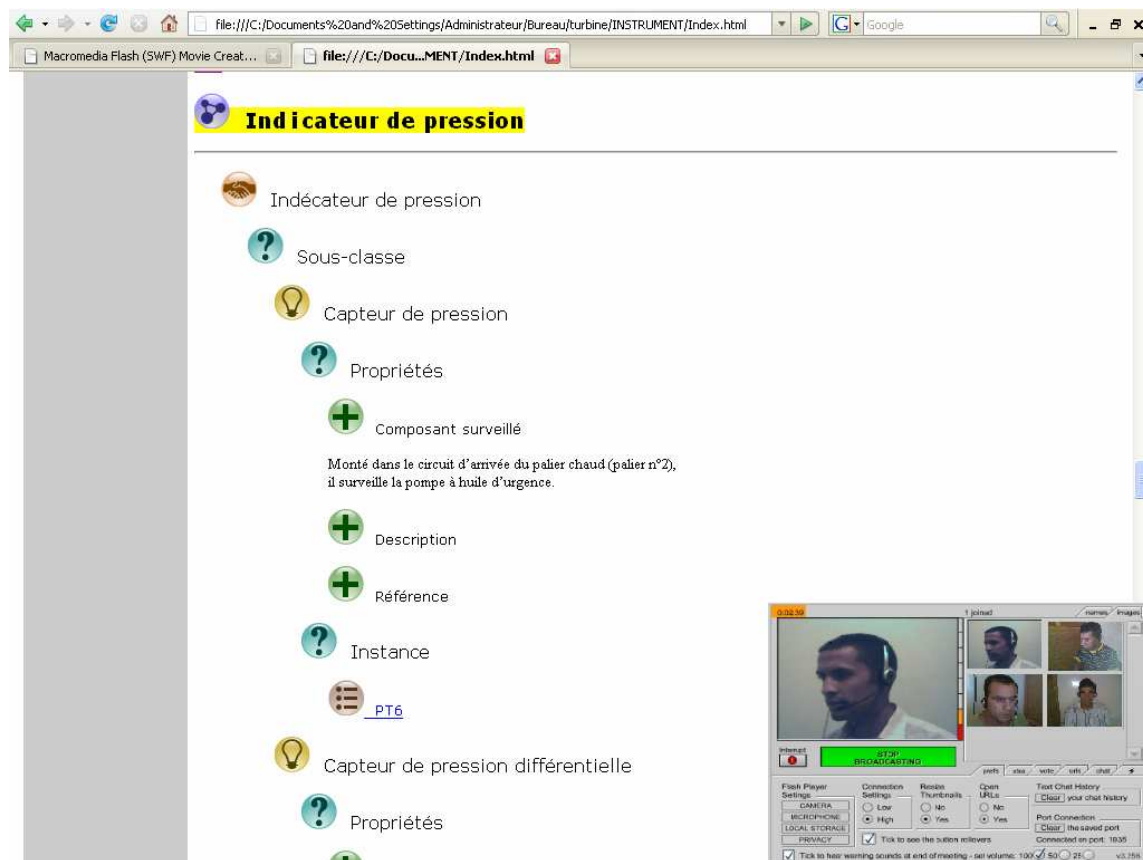


Figure 5.12- Description de la classe Indicateur en HTML généré par Compendium

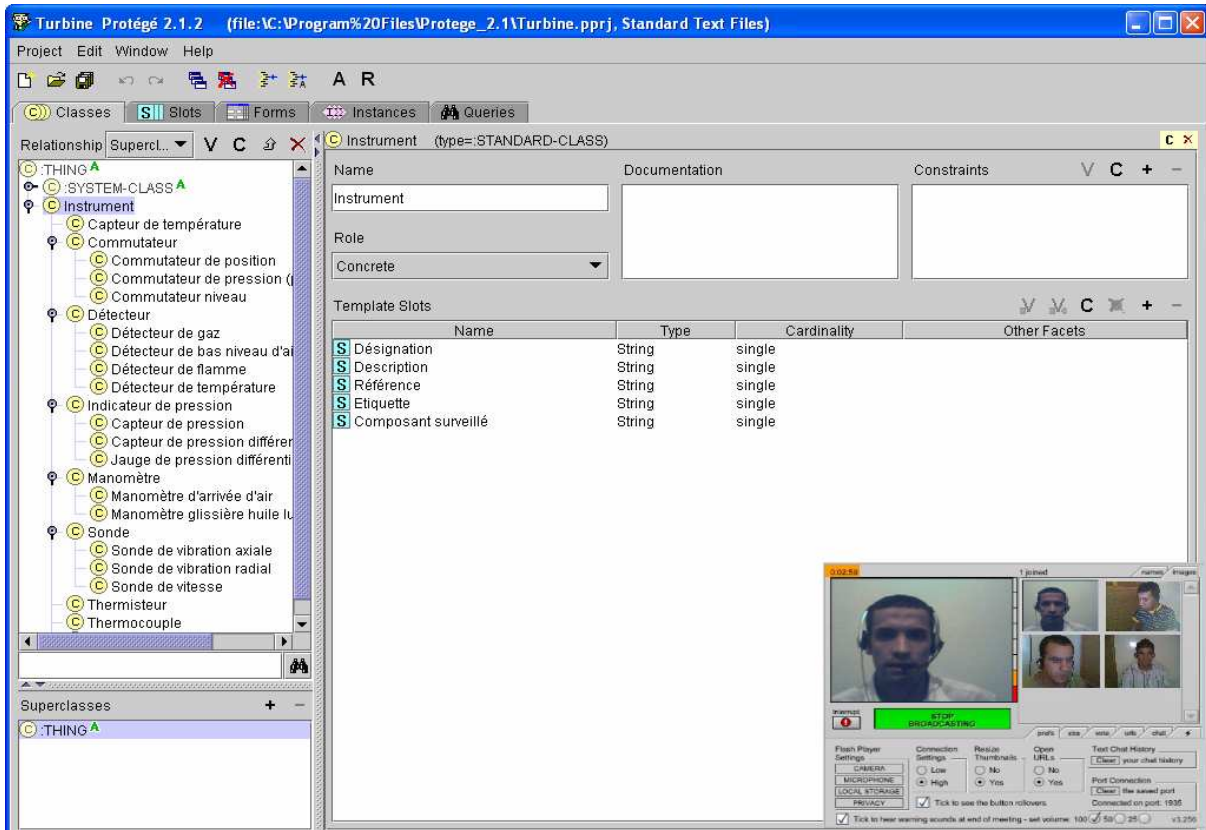


Figure 5.13- Transformation et Visualisation de l'Ontologie des turbines à gaz en Protégé2000

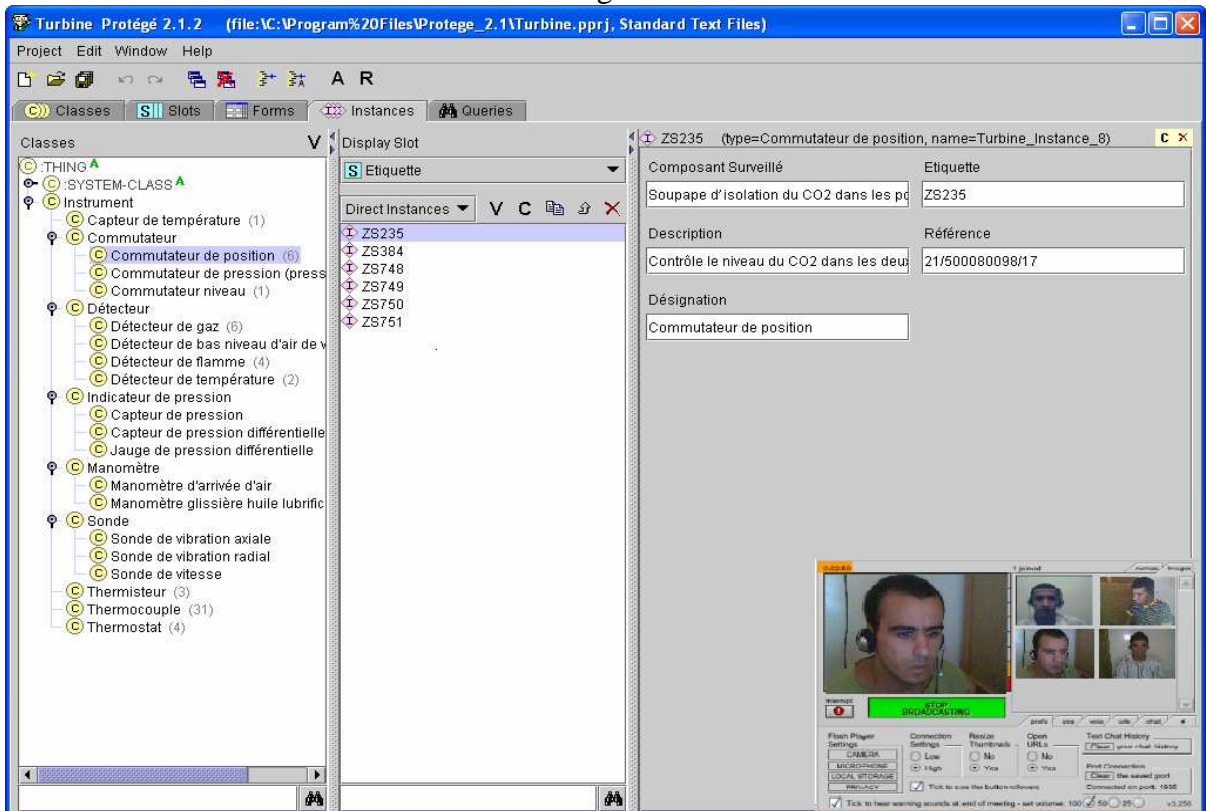


Figure 5.14- Exemple de visualisation et validation de la classe Commutateur par les membres de la CVonto

5.6. Conclusion

Nous avons vu dans ce chapitre une expérimentation sur l'utilisation et le déploiement de notre environnement autour d'un scénario de construction interactive d'ontologie partagée, et plus particulièrement sur l'aspect collaboratif fourni par les diverses instances des services synchrones et/ou asynchrones. Les résultats que nous avons obtenus sont encourageants: On note que notre environnement a permis de capturer et de structurer les idées des intervenants en leur permettant de prendre en main le service Compendium après une session de collaboration, avec une bonne compréhension de ce qu'est une ontologie. Par exemple la description de la classe commutateur a été discutée de façon compréhensible par les participants à partir des représentations bien définies en compendium. L'apport du nouvel service URL partagé n'était pas programmé au départ mais s'est avéré être un plus déterminant dans la construction collaborative d'ontologie. Les utilisateurs ont eu une impression de travailler en local et face à face. D'un point de vue des performances en termes de temps d'ingénierie, les participants ont pu résoudre le problème de construction d'ontologie en mettant en commun leurs compétences. Un problème concret qui demanderait 5 ou 6 jours à un ingénieur a ainsi pu être résolu à quatre ingénieurs en une heure seulement. L'interdisciplinarité entre turbine à gaz et sciences cognitives a été l'une des réussites de ce scénario. Cette expérimentation a cependant un certain nombre de limitations, à commencer par le petit nombre (4 personnes) y ayant participé. D'autre part, les personnes ayant participé à l'expérimentation sont des étudiants, et ceux-ci n'ont pas les mêmes connaissances et méthodes d'opération que des utilisateurs professionnels. Il serait donc intéressant de réaliser des expériences avec des experts et des cognitivistes professionnels, éventuellement sur des projets complets.

Conclusion Générale & Perspectives

6.1. Conclusion

Nous avons apporté avec ce mémoire les premiers éléments dans la construction d'une approche mettant en relation deux domaines auparavant distincts : les environnements collaboratifs d'une part et l'infrastructure GRID dans le contexte orienté service d'autre part. En conclusion, nous récapitulons les principales contributions et les principaux résultats auxquels nous sommes arrivés. Le manuscrit a été organisé en deux parties.

La première partie a été composée de deux chapitres qui décrivent l'état de l'art sur les thèmes de recherche abordés par ce mémoire. Le deuxième chapitre a introduit les environnements collaboratifs actuels sur le Web, nous avons tout d'abord défini les principaux termes et mots-clés qui font partie du travail collaboratif. Par la suite, nous avons procédé à une classification des principaux environnements collaboratifs sur le Web selon leurs solutions offertes aux utilisateurs, et nous avons analysé différentes approches de développement ainsi que leurs limitations pour pouvoir positionner notre approche en partant du besoin d'exploiter le potentiel important de l'infrastructure GRID dans le développement des plateformes de collaboration. Le troisième chapitre a dressé un état de l'art relatif aux principes fondamentaux des services Web/GRID et leurs apports au concept de collaboration. Nous avons commencé par aborder les principaux concepts de l'architecture des services Web et ses technologies standards. Partant du cycle de vie, nous avons tiré les avantages et les inconvénients présentés par les services Web. Ensuite, nous avons présenté les concepts clés et les différentes définitions fournies dans la littérature de GRID, ainsi que son anatomie. Nous avons par la suite mis en évidence l'architecture des services GRID OGSA et le point de convergence des deux technologies services GRID et Web. Enfin, Nous avons terminé ce chapitre par la présentation des grands projets et l'évolution de GRID.

La deuxième partie a été composée de deux chapitres qui présentent les contributions apportées par nos travaux de mémoire. Le chapitre 4 a présenté une approche originale visant à développer des environnements collaboratifs à base de GRID couplé à une solution de Bureau partagé. La proposition représente une architecture générique d'une plateforme collaborative sûre et flexible, exprimée en terme de couche, et offrant des solutions de travail intégré à une communauté virtuelle pour atteindre des buts communs en partageant de façon efficace et fiable des ressources informatiques sur le Web. Notre proposition s'appuie sur les concepts clés de l'architecture OGSA (Service GRID, Conteneur, Organisation virtuelle, etc.), ainsi que sur ses processus de partage des ressources (Virtualisation et Réification). Le

concept de bureau partagé vu comme un outil typique de collaboration a été introduit également comme une nouvelle dimension d'interaction à distance en deux modes entre les participants des communautés virtuelles et les instances des services GRID dans le contexte d'une session. Avec la présence de la gestion de l'état dans les services, les services GRID sont adaptés aux services collaboratifs synchrones et asynchrones. Sur le plan fonctionnel, notre environnement offre des services collaboratifs synchrones et asynchrones accessibles à partir du Web par le biais d'un terminal léger de façon sécurisée. Sur le plan architectural, il fournit un nouvel paradigme de partage et d'utilisation efficace et bien réparti des ressources physiques, géographiquement distribuées, en fonction de leur disponibilité et à la demande. Sur le plan technologique, il repose sur des technologies basées sur des standards ouverts du Web. Le chapitre 5 a présenté une démonstration de notre approche autour d'un scénario original et applicatif. Pour ce faire, nous avons développé un prototype fondé sur deux serveurs Web et VNC utilisant le protocole RFB pour communiquer et Applet JAVA pour mettre en œuvre le concept du bureau partagé. Ce scénario de co-construction d'une ontologie partagée des turbines à gaz au sein d'une communauté virtuelle, se déroule en deux grandes étapes en se focalisant sur deux outils: Protégé2000 et Compendium considérés comme deux services collaboratifs synchrones, nous a donné des résultats encourageants et a montré la flexibilité et l'efficacité de notre approche en palliant aux problèmes souvent rencontrés pour une telle collaboration: problème de la distance, ressources hétérogènes, nombre important de participants, coût élevé de la collaboration, absence d'historique, collaboration limitée dans le temps, rentabilité et productivité critiquables.

6.2. Perspectives

Plusieurs pistes de recherches peuvent être explorées afin d'approfondir le travail présenté dans ce mémoire. Nous avons plus particulièrement identifié trois points qui nous paraissent particulièrement intéressants :

1. Notre premier objectif consiste à poursuivre nos travaux pour une mise en œuvre complète de notre environnement .Pour ce faire, nous développerions un nœud de GRID local et propre à l'université qui sera fondé sur la boîte à outils GLOBUS en reliant l'ensemble des ressources informatiques y disponibles. Une connexion sera également établie au nœud de GRID EUMEDGRID, dont l'Algérie fait partie, pour rejoindre la communauté des chercheurs euro méditerranéens dans ce domaine.

2. Développement d'un modèle d'intégration SMA (Système multi Agents) et GRID où le concept *service* se situe à l'intersection de ces domaines. Les services sont échangés par des agents grâce à et au travers de l'infrastructure et des mécanismes de GRID. Ian Foster [120] 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. Clément Jonquet [121, 122, 123] expose notamment les analogies que présentent les interactions entre agents et les services d'une architecture GRID.
3. Développement de nouveaux scénarii d'expérimentations pour valider notre modèle dans des situations d'utilisation concrètes, notamment pour l'e-learning.

Les travaux présentés dans ce mémoire constituent un premier pas vers une approche transversale entre les environnements collaboratifs et l'infrastructure GRID. Nous espérons qu'ils ouvriront la voie à de nouveaux travaux à la croisée de ces deux domaines. Notre travail a fait l'objet de quelques publications [43, 44, 124, 125, 126, 127].

Références Bibliographiques

- [1] D DeRoure, R.N, Jennings, and N. Shadbolt, “Research agenda for the Semantic GRID: a future e-science infrastructure”, Technical report, University of Southampton, UK, Report commissioned for EPSRC/DTI Core e-Science Programme, June 2001.
- [2] P. Ritrovato, C. Allison, S. Cerri, T. Dimitrakos, M. Gaeta, and S.Salerno, “Towards the Learning GRID: advances in Human Learning Services”, volume 127 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, November 2005.
- [3] C. Allison, S. A. Cerri, P. Ritrovato, A. Gaeta, and M. Gaeta. “Services, Semantics and Standards: Elements of a Learning Grid Infrastructure”, *Applied Artificial Intelligence Journal*, Special issue on Learning Grid Services, Vol. 19, No. 9-10, November 2005, pp. 861–879.
- [4] S. Wesner and K. Wulf, “How GRID could improve E-Learning in the environmental science domain”, *1st LeGE-WG International Workshop on Educational Models for Grid Based Services*, Lausanne, Switzerland, Electronic Workshops in Computing (eWiC), September 2002.
- [5] T. O’reilly, What is Web 2.0, 2005, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=3>
- [6] J. DeRosnay, futurologue (Entretien de décembre 2005).
- [7] I. Salah, A. Kadmi, E. Reyes, *L’hypermédia au service du travail collaboratif*, Juin 2005, édition Hermès.
- [8] M. Robial, “Le travail collectif à distance en documentation”, Université du Mans1, juin 2004, <http://savoircsdi.cndp.fr/CulturePro/actualisation/ENTintranet/robial/travcoll.pdf>
- [9] J. Lonchamp, *Le travail coopératif et ses technologies*, Hermès Science, 320 pages, 2003.
- [10] C. Ellis, S. Gibbs et G. Rein, “Groupware, Some issues and experiences”, *Communications of the ACM*, Vol. 34, No. 1, Janvier 1991, pp. 38-58.
- [11] K.L. Kraemer and J.L. King, “Computer-Based Systems for Cooperative Work and Group Decision Making”, *ACM Computing Surveys*, Vol. 20, No. 2, Juin 1988, pp. 115-146.
- [12] H.P. Dommel and J.J.G. Luna, “Aceves, Group Coordination Support for Internet collaboration“, *IEEE Internet Computing Magazine*, Special Issue on Multimedia and Collaborative Computing over the Internet, Vol. 3, No. 2, Mars-Avril 1999, pp. 74-80.
- [13] K Drira, A. Martelli and T. Villemur, “Cooperative Environments for Distributed Systems Engineering”, *The Distributed System Engineering Report*, State of the Art Survey, Lecture Notes in Computer Sciences, Vol. 2236, N°ISBN 3-540-43083-0, Springer, 2001, pp.107-151.
- [14] S. Khoshafian, M. Buckiewicz, “Groupware et Workflow”, Interéditions, ISBN 2-225-82926-8, 1998.

- [15] G.J. Hoyos-Rivera, R. Lima-Gomes and J.P. Courtiat, “A Flexible Architecture for Collaborative Browsing”, *11th IEEE WETICE, Workshop on Web-Based Infrastructures and Coordination Architectures for Collaborative Enterprises*, 2002, Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA.
- [16] WebEx Communications, [http:// en.wikipedia.org/wiki/WebEx](http://en.wikipedia.org/wiki/WebEx)
- [17] WikiWikiWeb: <http://c2.com/cgi/wiki?WikiWikiWeb>, 2007.
- [18] WikiPedia, <http://www.wikipedia.org/>, 2007.
- [19] H. Guyennet, E. Garcia, and J.C. Lapayre, “Multimedia integration in cooperative work”, *Proceedings of the fifth international conference on information systems analysis and synthesis*, ISAS99, Orlando, USA, 1999.
- [20] OMG, Real-time CORBA, joint revised submission, OMG TC document orbos, 1999.
- [21] E. Garcia, H. Guyennet, J.C. Lapayre, N. Zerhouni, “A new industrial cooperative tele-maintenance platform”, *Computers & Industrial Engineering*, Vol. 46, No. 4, 2004, pp. 851–864.
- [22] Platine Web site, <http://www.laas.fr/PLATINE>, 2007.
- [23] Tixeo Soft <http://www.tixeo.com/>, 2007.
- [24] NetDive <http://www.netdive.com/>.
- [25] Microsoft Office Live Meeting <http://office.microsoft.com/en-gb/livemeeting/FX101729061033.aspx>.
- [26] WebEx Web site <http://www.webex.com/>.
- [27] Lotus Sametime <http://www.lotus.com/sametime>, 2007.
- [28] Produit WebCT. – <http://www.webct.com>, 2007.
- [29] S. Gance, Description de WebCT, Portland State University, <http://www.webct.pdx.edu/faculty/basics.html>.
- [30] Sakai, <http://sakaiproject.org/>, 2007.
- [31] Moodle, <http://moodle.org/>, 2007.
- [32] M.P. Singh and M.N. Huhns, *Service-oriented Computing, Semantics, processes, agents*, John Wiley & Sons, 2005.
- [33] W3C – World Wide Web Consortium, “Requirements for the Internationalization of Web Services”, W3C Working Group Note 16 November 2004, <http://www.w3.org/TR/ws-i18n-req/>.

- [34] “Extensible Markup Language”, <http://www.w3.org/XML/>, 2007.
- [35] M.C. Daconta, L.J. Obrst, K.T. Smith, *The semantic Web*, Wiley, Indianapolis, 2003.
- [36] F. Curbera, W.A. Nagy, and S Weerawarana, “Web services: Why and how?”, *OOPSLA 2001 Workshop on Object-Oriented Web, Services*, 2001.
- [36] R. Perrey, M. Lycett, “Service-oriented architecture”, *Symposium on Applications and the Internet Workshops (SAINT 2003)*, IEEE Computer Society Press ISBN: 0-7695-1873-7, Orlando (USA), Jan. 2003, pp.116-119.
- [37] D. Booth, H. Haas, F. McCabe, and E. Newcomer, “Web services architecture”, august 2003, <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>.
- [38] Wrox Author Team, *Service Web XML Professionnel*, Wrox, Paris, dec 2001.
- [39] J.M. Chauvet, *Services WEB avec SOAP, WSDL, UDDI, ebXML*, Eyrolles, Paris, 2002.
- [40] W3C – World Wide Web Consortium, “SOAP Version 1.2 Part 1: Messaging Framework”, W3C Recommendation 24 June 2003, <http://www.w3.org/TR/soap12-part1/>.
- [41] W3C – World Wide Web Consortium, “Web Services Description Language (WSDL) 1.1” W3C Note 15 March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.
- [42] OASIS – Organization for the Advancement of Structured Standards *UDDI Version 3.0.2*, UDDI Spec Technical Committee Draft, Dated 20041019 <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>.
- [43] M. Sellami, H Zarzour, M.T. Khadir, Y. Slimani, “A Grid Based Environment for SARIMA and other North-African projects”, *IST-Africa 2007 Conference Proceedings*, Paul Cunningham and Miriam Cunningham (Eds), IIMC International Information Management Corporation, ISBN: 1-905824-04-1, 2007.
- [44] H. Zarzour, M. Sellami, “Environnement collaboratif à base de GRID pour la construction interactive d'ontologies partagées”, *International Confernece on Computer Integrated Manufacturing, CIP'2007*, Ferhat ABBAS University, Setif, Algeria, November, 03-04 2007
- [45] Définition de GRID, <http://websemantique.free.fr/grille/node3.html>.
- [46] L. Miguel. B. Yannis A. Dimitriadis, and E. Gomez-Sanchez, “GRID Characteristics and Uses: a GRID Definition”, *Postproc. of the First European Across GRIDs Conference (ACG'03)*, Springer-Verlag LNCS 2970, Santiago de Compostela, Spain, Feb 2004, pp. 291-298.
- [47] R. Buyya and S. Venugopal, “A Gentle Introduction to GRID Computing and Technologies”, CSI Communications, July 2005.
- [48] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the GRID: Enabling Scalable Virtual Organizations”, *International J. Supercomputer Applications*, Vol. 15, No. 3, 2001.

- [49] C. Kesselman and I. Foster attempted a definition in the book “The GRID: Blueprint for a New Computing Infrastructure”, 1998, <http://www.mkp.com/GRIDs>.
- [50] Journal le Monde Informatique N1010, (23-01-2004), 2004.
- [51] K. Jeffery, Knowledge, Information and Data. Research report, Central Laboratory of the Research Councils, UK, September 1999.
- [52] F.O. Rana and L. Moreau, “Issues in building agent based computational GRIDs”, *3rd Workshop of the UK Special Interest Group on Multi-Agent Systems, UKMAS’00*, Oxford, UK, December 2000.
- [53] The Globus Alliance, <http://www.globus.org>.
- [54] I. Foster and C. Kesselman, “Globus: A Toolkit-Based GRID Architecture”, In Foster, I. and Kesselman, C. eds. *The GRID: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, pp. 259-278.
- [55] I. Foster, and C. Kesselman, “The Globus Project: A Status Report”, In *Proc, Heterogeneous Computing Workshop*, IEEE Press, 1998, pp. 4-18.
- [56] I. Foster, C. Kesselman, J. Nick, S. Tuecke, “The Physiology of the GRID : An Open GRID Services Architecture for Distributed Systems Integration”, *Open GRID Service Infrastructure WG, Global GRID Forum*, The Globus Alliance, June 2002.
- [57] I. Foster et al., “The Open GRID Services Architecture, Version 1.0”, <http://www.GRIDforum.org/document/s/GWD-I-E/GFD-I.030.pdf>.
- [58] J. Treadwell et al, ”Open GRID Services Architecture Glossary of Terms Version 1.5”, March 10, 2006, <http://forge.GRIDforum.org/projects/ogsa-wg> .
- [59] P. Dugénie, P. Lemoisson, C. Jonquet, M. Crubézy, ”The GRID Shared Desktop: a bootstrapping environment for collaboration”, *Advanced Technology for Learning, Special issue on Collaborative Learning*, 2006. Accepted for publication -Expected end of 2006.
- [60] C. Jonquet, P. Dugenie, SA. Cerri, “Service-based integration of GRID and multi-agent systems models”, Research report 06012, University Montpellier 2, France, February 2006. www.lirmm.fr/~jonquet/Publications.
- [61] P. Dugenie, “GSD: Conceptual framework for the future GRID Shared Desktop”, IEEE, 2005, www.lirmm.fr/~jonquet/Publications.
- [62] S. Tuecke, K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling et P. Vanderbilt, “Open GRID Services Infrastructure (OGSI), Version 1.0”, Global GRID Forum, 27 Juin 2003.
- [63] S.A. Cerri, “Open Learning Service Scenarios on GRID”, *Proceedings of the 3rd International Workshop on GRID Infrastructure to Support Future Technology*, 2003.

- [64] D. Snelling, I. Foster, T. Maguire, “OGSA-WG recommendation: OGSA WSRF basic profile”, Recommendation 1.0, Open Grid Forum, 2005.
- [65] Globus: OGSA - The Open Grid Services Architecture, <http://www.globus.org/ogsa>.
- [66] J.P. Prost, “Open GRID Services Architecture Evolution”, IBM Montpellier, France, Ecole Bruide 2004.
- [67] C. Comito, D. Talia, P. Trunfio, “GRID Services: Principles, Implementations and Use”, *International Journal of Web and GRID Services*, Vol. 1, No. 1, 2005, pp. 48-68.
- [68] I. Foster and C. Kesselman, *The Grid: blue print for a new computing infrastructure*, San Francisco: Morgan Kaufmann Inc, CA, USA, 1999, pp. 82-87.
- [69] Enabling Grids for E-science, <http://www.eu-egee.org/>.
- [70] Web site gLite (Lighthouse Middleware for GRID Computing), <http://glite.web.cern.ch/glite/>.
- [71] Empowering eScience across the Mediterranean, <http://www.eumdgrid.com>.
- [72] Seti@home, <http://setiathome.berkeley.edu>, 2007.
- [73] Genome@home, <http://genomeathome.stanford.edu>, 2007.
- [74] World community Grid, <http://www.worldcommunityGRID.org>, 2007.
- [75] Décrypton, <http://www.decryphon.org>, 2007.
- [76] GRID'5000 Home page , <https://www.grid5000.fr/>, 2007
- [77] The Data GRID Project , <http://eu-datagrid.web.cern.ch/eu-datagrid/>
- [78] EUROGRID Project, <http://www.eurogrid.org/>, 2007.
- [79] I. Foster and C. Kesselman, *The Grid2: blue print for a new computing infrastructure*, San Francisco: Morgan Kaufmann Publishers Inc, CA, USA, 2003.
- [80] F. Berman, G. Fox, and A.J.G. Hey, *Grid computing: making the global infrastructure a reality*. John Wiley & Sons, 2003.
- [81] C. Leeand, D. Talia, “Grid programming models: current tools, issues and directions”, G. Fox F. Berman and A.J.G. Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*, chapter 21 , John Wiley & Sons, 2003, pp. 555–578.
- [82] D. DeRoure, N.R. Jennings, and N.R. Shadbolt, “The Semantic Grid: past, present, and future”, *Proceedings of the IEEE*, Vol. 93, No. 3, March 2005, pp. 669–681.

- [83] M. Geldof, “The Semantic Grid: will Semantic Web and Grid go hand in hand ?”, Technical report, European Commission DG Information Society Unit Grid technologies, June 2004.
- [84] M. Cannataro and D. Talia, “The Knowledge Grid”, *Communications of the ACM*, Vol. 46, No. 1, January 2003, pp. 89–93.
- [85] H. Zhuge, “The Knowledge Grid”, *World Scientific*, Singapore, 2004.
- [86] P. Ritrovato, C. Allison, S.A. Cerri, T. Dimitrakos, M. Gaeta, and S. Salerno, “Towards the Learning Grid: advances in Human Learning Services”, volume 127 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, November 2005, pp. 41–62.
- [87] S. Wesner and K. Wulf, “How Grid could improve learning in the environmental science domain”, *1st LeGE-G International Workshop on Educational Models for Grid Based Services*, Lausanne, Switzerland, September 2002. *Electronic Workshops in Computing*.
- [88] R. Nkambou, G. Gouardères and B.P. Woolf, “Toward learning Grid infrastructures: an overview of research on Grid learning services”, *Applied Artificial Intelligence*, Special issue on Learning Grid Services, Vol. 19 No. 9-10, October-November 2005, pp. 811–824.
- [89] C. Allison, S.A. Cerri, P. Ritrovato, A. Gaeta, and M. Gaeta, “Services, semantics and standards: elements of a learning Grid infrastructure”, *Applied Artificial Intelligence*, Special issue on Learning Grid Services, Vol. 19, No. 9-10, October-November 2005, pp. 861–879.
- [90] R. Yatchou, Gouardères, G, N'kambou R., “Ubiquitous Knowledge Prosthesis for GRID Learning Services: The GRID-e-Card”, *GRID Learning Services Workshop*, GLS'04, in conjunction with ITS'2004, Maceió, Brazil, 2004.
- [91] “*Grid Application Toolkit and Testbed*”, <http://www.gridlab.org/Welcome/News.html>.
- [92] J. Novotny, M. Russell, “GridSphere: A Portal Framework For Building Collaborations”, <http://www.gridsphere.org/gridsphere/wp-content/uploads/2004/08/RioBabyRio/gridsphere.pdf>.
- [93] B. Calder, A.A. Chien, J. Wang, D. Yang, “The Entropia Virtual Machine for Desktop Grid”, 2004, <http://www-csag.ucsd.edu/papers/vm2004.pdf>.
- [94] P. Dugénie and S.A. Cerri, “Le Collabureau: un espace de collaboration à la puissance Grid”, *Technologies de l'Information et de la Communication pour l'Education*, TICE06, novembre 2006, hal-lirmm.ccsd.cnrs.fr/lirmm-00113818.
- [95] Manuel d'utilisation des turbines à gaz.
- [96] H. Zarzour, S. Khedairia, “Description multi niveaux, gestion des pannes d'une station de compression et simulation du comportement thermique des turbines à Gaz”, Mémoire d'ingénieur, Université de Badji Mokhtar, Annaba, Soutenu le 28 juin 2005.

- [97] M. Uschold, “Knowledge level modelling: Concepts and terminology”, *Knowledge Engineering Review*, Vol. 13, No. 1, 1998.
- [98] T. Gruber, “A translation Approach to Portable Ontology Specifications”, *Knowledge Acquisition*, Vol. 5, No. 2, pp.199 – 220.
- [99] W.N.Borst, “Construction of Engineering Ontologies”, Mémoire de Thèse, Center for Telematica and Information Technology, University of Twente, NL, Soutenue le 5 septembre 1997.
- [100] A.Gomez-Perez, “Ontological Engineering: a State of the Art”, Expert Update. *British Computer Society*. Vol. 2, No. 3, 1999, pp. 33 – 43.
- [101] R. Studer, V.R. Benjamins, D. Fensel, “Knowledge Engineering: Principles and Methods”, *Data and Knowledge Engineering*, Vol. 25, 1998, pp.161 – 197.
- [102] J.Conklin, and M.L.Begeman, “gIBIS: A Hypertext Tool for Exploratory Policy Discussion”, *ACM Transactions on Office Information Systems*, Vol. 4, No.6,1988, pp. 303-331.
- [103] D.E. O’Leary et al, “Using AI in Knowledge Management: Knowledge Bases and Ontologies”. *IEEE Intelligent Systems*, May/June 1998, pp. 34-39.
- [104] V. Psyché, O. Mendes, J. Bourdeau, “Apport de l’ingénierie ontologique aux environnements de formations à distance”, *revue sticef.org*, Vol. 10, 2003.
- [105] R.Mizuguchi, “Tutorial on ontological engineering Introduction to Ontological Engineering”, *New Generation OhmSha & Springer*, Vol. 21, No.4 , 2003, pp. 365 – 384.
- [106] PROTEGE2000, Protege2000 Ontology Editor Home Page, <http://protege.stanford.edu/>, 2007.
- [107] N.W. Fergerson and M.A. Mausen, “The knowledge model of Protégé2000: combining interoperability and flexibility”, *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW’00)*, 2000.
- [108] “La Création d’Ontologies Web Sémantique avec Protégé-2000”, 2006, http://www.cetic.be/imprimer.php3?id_article=138.
- [109] F. FÜRST, “L’ingénierie ontologique”, Rapport de recherche No 02-07, Institut de Recherche en Informatique de Nantes, Octobre 2002.
- [110] B. Shum, “Towards Next Generation Tools for Crafting Hypermedia Discourse”, *Semaine de la Connaissance 2006*, Nantes, 26-30 June 2006, http://sdc2006.org/cdrom/contributions/Shum_SDC2006.pdf.
- [111] Compendium, <http://www.CompendiumInstitute.org>, 2007.
- [112] Compendium Community - 2007 Workshop, <http://www.compendiuminstitute.org/community/workshops/2007/>.

- [113] G. Kassel, “Ontospec : une méthode de spécification semi-informelle d’ontologies”, *IC2002 proceedings*, Rouen, 28–30 mai 2002.
- [114] F.Z. Laallam, M. Sellami. “Gas Turbine Ontology for the Industrial Processes”, *Journal of Computer Science* Vol. 3, No. 2, ISSN 1549-3636, 2007, pp. 113-118.
- [115] Real Virtual network Computing Web site, <http://www.realvnc.com/>, 2007.
- [116] S.F. Li, ; M. Spiteri, J. Bates, et Al, “Capturing and Indexing Computer-based Activities With Virtual Network Computing”, *Proceedings of the 2000 ACM Symposium on Applied Computing Villa Olmo*, Via Cantoni 1, 22100 Como, Italy March 19-21, 2000, ACM ISBN 1-58113-239-5.
- [117] S.F. Li, Q. Stafford-Fraser and A. Hopper, “Integrating Synchronous and Asynchronous Collaboration with Virtual Network Computing”, *IEEE Internet Computing*, Vol. 4, No. 3, May/June 2000.
- [118] T. Richardson, Q. Stafford-Fraser, K.R. Wood, et al, “Virtual Network Computing”, *IEEE Internet Computing*, Vol. 2, No. 1, January/February 1998.
- [119] D. Raymond, V. Baudin, K. Kanenishi, et al , “Distant e-learning using synchronous collaborative environment Platine”, *IEEE 6th International Symposium on Multimedia Software Engineering*, MSE'2004 Miami (USA), December 13-15, 2004, pp. 88-95.
- [120] I. Foster, N.R. Jennings, and C. Kesselman, “Brain meets brawn: why Grid and agents need each other”, *3rd International Joint Conference on Autonomous Agents and Multi agent Systems*, AAMAS04, New York, NY, USA, Vol. 1, July 2004, pp. 8–15.
- [121] C. Jonquet, D. Service, “Generation: Agent interaction for service exchange on the GRID”, PhD thesis, Université de Montpellier II, Montpellier, 2006.
- [122] C. Jonquet, P. Dugénie, and S.A. Cerri, “Intégration orientée service des modèles Grid et Multi-Agents”, *14èmes Journées Francophones sur les Systèmes Multi-Agents*, JFSMA06, pp. 271–274, Annecy, France, octobre 2006, Hermès. hal-lirmm.ccsd.cnrs.fr/lirmm-00107500.
- [123] C. Jonquet, P. Dugénie, and S.A. Cerri, “Agent-Grid Integration Language, Multi agent and Grid Systems”, (à paraître), 2007.hal-lirmm.ccsd.cnrs.fr/lirmm-00139691.
- [124] H. Zarzour, M. Sellami, “Environnement collaboratif intégré sur le Web à base de GRID”, *Séminaire national sur le langage naturel et l’intelligence artificielle*, LANIA'2007, Université Hassiba Benbouali, Chlef, Algérie, 20-21 Novembre 2007.
- [125] H. Zarzour, M. Sellami, “Bureau partagé pour un environnement collaboratif orienté GRID”, *Conférence internationale des 3èmes journées internationales de l’information graphique*, JIG'2007, Sélectionné pour une publication dans la revue ANDRU(Agence National de Développement de la Recherche Universitaire) qui a consacré un numéro spécial pour JIG'2007, pp.166-172, Université Mentouri de Constantine, Algérie, 29 - 30 Octobre 2007.

[126] H. Zarzour, M. Sellami, “Environnement collaboratif orienté GRID”, *Journées Ecole Doctorale & Réseaux de Recherche en Sciences et Technologies de l'information*”, JED'07, Université de Annaba, Algérie, 27-28 Mai 2007.

[127] H. Zarzour, M. Sellami, “Environnement collaboratif Multimodal à la puissance GRID”, Accepté au *9e Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées*”, CARI'08, Rabat, Maroc, 27-30 Octobre 2008.