



HAL
open science

Adaptation dynamique d'applications multimédia à leur contexte d'exécution dans les réseaux du futur

Yves-Gaël Billet

► **To cite this version:**

Yves-Gaël Billet. Adaptation dynamique d'applications multimédia à leur contexte d'exécution dans les réseaux du futur. Autre [cs.OH]. Université Jean Monnet - Saint-Etienne, 2012. Français. NNT : 2012STET4008 . tel-00955366

HAL Id: tel-00955366

<https://theses.hal.science/tel-00955366>

Submitted on 4 Mar 2014

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

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



THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JEAN MONNET DE
SAINT-ÉTIENNE

Discipline : Génie Informatique, Informatique

Présentée et soutenue publiquement le 12 octobre 2012

par

Yves-Gaël BILLET

Ingénieur Télécom Saint-Etienne

Adaptation dynamique d'applications
multimédia à leur contexte d'exécution
dans les réseaux du futur

Directeur de thèse :
Jacques FAYOLLE

Co-Directeur de thèse :
Christophe GRAVIER

Composition du Jury :

PATRICK BRÉZILLON	Professeur à l'Université Pierre et Marie Curie	<i>Président</i>
TOUFIK AHMED	Professeur à l'Université Bordeaux I	<i>Rapporteur</i>
STEPHANE FRÉNOT	Professeur à l'INSA de Lyon	<i>Rapporteur</i>
OLIVIER BOISSIER	Professure à l'école des Mines de Saint-Etienne	
JACQUES FAYOLLE	Professeur à l'Université Jean Monnet	<i>Directeur</i>
CHRISTOPHE GRAVIER	Maître de Conférences à l'Université Jean Monnet	<i>Co-directeur</i>

Numéro d'ordre: xxxxxx

Notes

Remerciements

Ce travail a été réalisé au sein du Laboratoire Télécom Claude Chappe (LT2C) de Télécom Saint-Etienne et financé par le Conseil Général de la Loire (CG42).

Je souhaite en tout premier lieu remercier Jacques Fayolle et Christophe Gravier qui ont encadré cette thèse et qui m'ont accompagné bien au-delà de celle-ci. Je souhaite aussi remercier Frédérique Laforest pour le temps qu'elle a su me consacrer pour me prodiguer remarques et conseils avisés sur la présentation de mes travaux. Ainsi que Jean-Jacques Rousseau pour m'avoir accueilli au sein de son laboratoire.

Mes remerciements vont aussi à Toufik Ahmed et Stéphane Frénot pour l'intérêt qu'ils portent à mon travail en acceptant d'être rapporteur.

Je ne peux oublier mes collègues de travail au sein de l'équipe SATIN (Antoine, Pierre-Olivier, Benjamin, Nicolas, Merièmè, Jérémy, Mickael) avec qui j'ai partagé discussions techniques, relecture de papiers scientifiques, etc. Ainsi que la société Deliciapps, en particulier Mehdi et Kevin pour nos discussions quotidiennes aussi bien philosophiques que techniques.

Je tiens aussi à remercier l'ensemble des personnes avec qui j'ai pu travailler à Télécom Saint-Etienne, en particulier le personnel administratif (Gaétane, Julie, Chantal, Elodie, Stéphanie, Claudine, Florence, Viviane Frédérique, Emmanuelle, Jany, Jacqueline, Sophie, Valérie, Marion). Ainsi que les enseignants qui ont su me conseiller dans leurs domaines respectifs (Thierry, Gilbert, Richard, Bruno et Isabelle). Sans oublier le service informatique (Christian, Mathieu, Nelly et Charlotte) pour leur support et nos échanges techniques.

Je terminerais ces remerciements avec un mot pour ma famille : MERCI de votre soutien. En particulier, mon parrain qui a su me persuader de réaliser un doctorat en informatique et mes parents qui ont toujours été là pour me motiver dans les moments difficiles. Sans oublier ma femme qui a dû concilier la préparation de notre mariage avec la fin de ma thèse.

Table des matières

Table des matières	vii	
Table des figures	xi	
Liste des tableaux	xiii	
I	Contexte des travaux	1
1	Les fournisseurs de services et la problématique de l'adaptation continue	3
2	Scénario de référence	5
2.1	Déroulement du scénario	6
2.2	Problématiques soulevées	7
2.3	Adaptation continue des applications dans les réseaux du futur	8
3	La convergence : prochaine évolution des réseaux de télécommunications	11
3.1	Convergence de terminaux	11
3.2	Convergence de réseaux	13
3.3	Convergence de services	14
3.4	De la convergence aux réseaux du futur	15
3.5	Organisation du mémoire	15
II	Etat de l'art	17
4	Les réseaux du futur (NGN)	19
4.1	Définition	19
4.2	Convergence fixe-mobile	20
4.3	Séparation fonctionnelle	21
4.4	Gestion de la qualité de service	23
4.5	Fournisseurs de services	25

4.6	IMS, la réalisation concrète d'un NGN	27
4.7	Conclusion	36
5	Adaptation des applications en informatique	37
5.1	L'adaptation des applications dans les NGN	37
5.2	Les applications sensibles au contexte	53
5.3	Applications autonomes	61
5.4	Conclusion	66
6	Les ontologies en informatique	69
6.1	Le Web Sémantique	70
6.2	Web Ontology Language (OWL)	71
7	Bilan	77
III	Système de sensibilité au contexte pour les réseaux du futur	79
8	Introduction	81
9	Architecture de l'intergiciel	83
9.1	Périmètre fonctionnel	83
9.2	Principes de conception	86
9.3	Conclusion	93
10	Les blocs fonctionnels	97
10.1	Knowledge Base (Base de connaissances)	98
10.2	Harvester (Collecteur)	101
10.3	Context Broker (Courtier de contexte)	102
10.4	Semantic Formalizer (Formateur sémantique)	103
10.5	Context Querier (Interrogateur de contexte)	104
10.6	Conclusion	104
11	Les fonctions de l'intergiciel de sensibilité au contexte	107
11.1	Introduction	107
11.2	Déclaration d'une source de contexte	110
11.3	Déploiement d'un service numérique	112
11.4	Analyse de l'environnement lors de la consommation d'un service numérique	113
11.5	Suppression d'un service numérique	117
11.6	Conclusion	117
12	Particularités de notre modèle	119
12.1	Séparation des rôles	119
12.2	Signature de contexte	120

12.3	Gestion du déploiement	120
12.4	Traitement du contexte	121
12.5	Sensibilité au contexte et réseaux du futur	121
IV	Réalisation dans OpenIMS	123
13	Choix technologiques	125
13.1	Coeur de réseau IMS	125
13.2	serveur d'applications	126
13.3	Terminaux IMS	128
13.4	Contraintes liées à l'environnement	129
14	Mise en oeuvre	131
14.1	Les technologies utilisées pour la réalisation de l'intergiciel	131
14.2	Le moteur d'inférence	132
14.3	OpenIMS	132
14.4	Environnement construit	132
15	Validation du scénario de référence	137
15.1	Déploiement du service numérique sensible au contexte	137
15.2	Consommation du service numérique sensible au contexte	141
15.3	Conclusion	145
V	Conclusion	147
16	Conclusion générale	149
VI	Annexes	153
A	Le contrôle de session et le réseau coeur	155
A.1	La base utilisateur	155
A.2	P-CSCF	156
A.3	I-CSCF	157
A.4	S-CSCF	157
A.5	Conclusion	158
B	Technologies du web sémantique : les URI et RDF	159
B.1	URI	159
B.2	RDF	160
C	Composition d'un message SIP	165
C.1	La ligne de début	165

C.2	L'entête	167
C.3	Le corps du message	167
D	Installation d'OpenIMS	169
D.1	Préparation du système	169
D.2	Installation d'un environnement graphique persistant accessible via VNC	170
D.3	Installation d'OpenIMS	171
D.4	Configuration de l'environnement	172
D.5	Utilisation d'OpenIMS	176
	Bibliographie	179

Table des figures

3.1	Principes de convergence	12
4.1	Architecture d'un NGN [39]	22
4.2	Les trois couches d'IMS et l'indépendance de l'accès	29
4.3	Protocoles utilisés dans IMS	33
4.4	Architecture d'IMS	34
5.1	Modélisation du flux de tâche pour des environnements distribués [51]	38
5.2	Modèle pour des applications auto-adaptables [43]	39
5.3	Agent logiciel pour l'évaluation de la performance d'une session multimédia [41]	41
5.4	Architecture de signalisation QoS de bout en bout et adaptation des services [60]	42
5.5	Modèle de négociation dynamique de la qualité de service et d'adaptation de la QoS [69]	43
5.6	MCMS pour IMS [47]	44
5.7	Table MDP pour un service [25]	46
5.8	Établissement d'une session multimédia avec le B2BUA [4]	49
5.9	Architecture abstraite pour des applications sensibles au contexte [17]	56
5.10	Processus de transformation des données en connaissances [3]	57
5.11	Pyramide des systèmes autonomiques [67]	62
5.12	Diagramme de l'évolution de l'adaptation [58]	63
5.13	IBM MAPE-K	64
6.1	Feuille de route des rechnologies du Web sémantique	71
9.1	Un service numérique sensible au contexte	84
9.2	Boucle MCDA de l'intergiciel et services numériques sensibles au contexte au sein d'un serveur d'applications [10]	87
9.3	Schématisation de l'architecture sous la forme d'un modèle en couches	88
9.4	Répartition des fonctions d'adaptation	94
10.1	Chaîne de traitement du contexte	97
10.2	Schéma de l'ontologie	100

11.1	Modélisation UML : Diagramme de déploiement du système de sensibilité au contexte	108
11.2	Diagramme de cas d'utilisation général	109
11.3	Déclaration d'une source de contexte au sein de l'intergiciel - Diagramme d'activité	110
11.4	Déclaration d'une source de contexte au sein de l'intergiciel - Diagramme de séquence	111
11.5	Déploiement de l'application et de la signature de contexte	113
11.6	Déploiement d'une application - Diagramme de séquence	114
11.7	Consommation d'un service numérique - Diagramme d'activité	115
11.8	Ouverture d'une session - Diagramme de séquence	116
12.1	Intégration de notre proposition dans une architecture IMS	122
13.1	Architecture d'une plateforme JAVA EE 5 extraite de la documentation Oracle	128
14.1	Architecture IMS virtualisée	133
14.2	Architecture type de la plateforme EUROP	134
15.1	Aperçu de l'ontologie avant le déploiement du service numérique sensible au contexte	138
15.2	Ontologie après le déploiement du service numérique	141
15.3	Ontologie lors des ouvertures de sessions	142
15.4	Ontologie lors de l'ajout des valeurs des capteurs	143
15.5	Ontologie après l'inférence	144
15.6	Ontologie après un changement de situation (ici Bob avec une bande passante de 50 kbit/s)	145
15.7	Ontologie après les terminaisons des sessions	146
A.1	IMS : fonctions du HSS et son environnement [81]	156
B.1	Exemple d'un graphe RDF orienté	162
C.1	Composition d'un message SIP	166

Liste des tableaux

2.1	Modalités et règles de l'application	6
2.2	Évolution des modalités applicatives de l'application de téléconférence pour la session de Bob	7
5.1	Comparaison des approches pour l'adaptation des services numériques dans les réseaux du futur	52
11.1	Interactions majeures entre l'intergiciel et son environnement	109
13.1	Comparaison des serveurs d'applications	126
15.1	Situations d'usage	139
B.1	URI	159
B.2	Primitives RDF Schema	163
C.1	Code d'état SIP	166
C.2	Les 6 champs d'entête obligatoire	167

Première partie

Contexte des travaux

Chapitre 1

Les fournisseurs de services et la problématique de l'adaptation continue

Dans les modèles de réseaux unifiés (les NGN), l'accent est mis sur la séparation entre l'acheminement des données, la gestion du réseau et les applications. Cette approche est indispensable pour la réalisation d'un réseau qui propose une plateforme de diffusion de services numériques, et cela, quels que soient la technologie d'accès et le terminal utilisé.

Un nouvel acteur émerge dans la chaîne de valeur télécom : le fournisseur de services numériques. Il incarne la convergence des services en proposant ses services numériques à tous les utilisateurs du réseau, c'est-à-dire *quels que soient les terminaux et technologies d'accès utilisés*. Mais il ne s'agit pas de créer autant de versions du service numérique qu'il existe de couples possibles entre la technologie d'accès et les terminaux. Ni à l'inverse, de limiter le service numérique à certains terminaux ou à certaines technologies d'accès. Son objectif est de rendre le même service accessible à tous, *quels que soient le terminal et la technologie d'accès*, c'est-à-dire offrir la meilleure expérience d'utilisation possible en regard des capacités du réseau et du terminal. Dans les réseaux du futur, cette promesse d'adéquation du service numérique à son environnement d'exécution ne concerne pas uniquement le début de la session, mais doit perdurer tout au long de son exécution. En effet, les caractéristiques du réseau et du terminal peuvent évoluer au cours du temps. Il faut donc être en mesure de fournir la performance attendue en continu. Pour permettre cet alignement du service numérique avec le terminal et le réseau, les services numériques doivent prendre conscience de leur environnement.

Cela est d'autant plus important qu'IMS propose le changement de technologie d'accès (handover vertical) ou du terminal sans couture. Ce changement de contexte d'utilisation au cours d'une session applicative nécessite une réévaluation constante de l'environnement et un ajustement de l'application si nécessaire. Ceci est particulièrement

vrai pour le changement de technologie d'accès où l'utilisateur conserve son terminal. En effet, il ne souhaite pas subir l'arrêt inopiné de sa session applicative pour des raisons techniques comme le changement d'une technologie d'accès à une autre. D'autant plus que ces changements sont dus à des fonctions offertes par les réseaux du futur.

Nous pouvons voir ce problème sous un autre angle en reprenant le modèle OSI. Ce modèle décrit un réseau sous la forme de sept couches, chacune s'appuyant sur la précédente pour réaliser sa fonction. La première couche est la liaison physique, la dernière est l'application. IMS propose une continuité de la connexion au réseau lors du changement de technologie d'accès à ce même réseau, par exemple, le passage de WiFi à 3G. Autrement dit, la couche réseau conserve la connexion IP pour l'échange de données. Il semble donc évident que les couches supérieures ne devront pas être impactées par ce changement de technologie d'accès. Cependant, dans certaines situations, les utilisateurs expérimentent aujourd'hui une coupure du service en cours de consommation. En effet, les caractéristiques de chacune des technologies d'accès ne sont pas équivalentes, certaines offrant de meilleures caractéristiques que d'autres (p. ex. bande passante plus limitée pour une connexion 3G que pour une connexion 4G). Pour pallier ces problèmes liés aux situations d'usage, les services doivent s'adapter à leurs conditions d'utilisation tout au long de la session applicative.

Cette problématique d'adaptation dynamique d'une application à son environnement de consommation dans un réseau du futur motive l'étude présentée dans la suite de ce manuscrit.

Chapitre 2

Scénario de référence

Notre scénario de référence s'inscrit dans les réseaux du futur. Ce sont des réseaux *convergnents* qui proposent de nouveaux usages aux utilisateurs, tel que le changement de technologie d'accès au réseau sans déconnexion.

Dans ce scénario, Alice et Bob sont deux salariés d'une même société qui souhaitent communiquer en utilisant un système de téléconférence. Pour cela, ils s'appuient sur le service numérique fourni par leur opérateur de réseau : *téléconférence adaptative*. Ce service numérique permet de mettre en relation plusieurs participants au travers du réseau et de leurs terminaux. Par rapport à un service de téléconférence standard, sa particularité est sa capacité à adapter les flux audio et vidéo de chaque utilisateur du service par rapport à son environnement informatique. Il adapte les flux pour proposer une continuité du service de téléconférence en regard de la situation d'usage de l'utilisateur.

Dans notre scénario de référence, le service numérique propose quatre modalités d'exécution :

- Flux audio sans vidéo.
- Flux audio avec une vidéo en faible définition.
- Flux audio avec une vidéo en moyenne définition.
- Flux audio avec une vidéo en haute définition.

L'adaptation du service numérique est réalisée par le passage d'une modalité à l'autre lors de l'exécution du service numérique.

Pour réaliser ces adaptations, le service numérique évalue la performance de l'environnement informatique de l'utilisateur. Dans ce scénario de référence, l'évaluation de la performance est réalisée avec deux paramètres :

- La bande passante disponible,
- La définition de l'écran du terminal utilisé.

Le service de *téléconférence adaptative* propose la modalité de transmission la plus adaptée en fonction du contexte d'utilisation.

Ce service définit quatre situations d'usage (ou contextes d'utilisation) en fonction des valeurs du couple bande-passante/résolution de l'écran. Si la bande passante et la résolution de l'écran dépassent un certain seuil alors l'application est éligible à une ou plusieurs modalités. Nous représentons ces situations d'usage et les valeurs seuil correspondantes dans Tableau 2.1.

TABLEAU 2.1: Modalités et règles de l'application

Bande passante	Résolution de l'écran	Modalité
> 8 Mbit/s	> 1024*768	Audio et Vidéo HD
> 2 Mbit/s	> 800*480	Audio et Vidéo MD
> 256 Kbit/s	> 320*240	Audio et Vidéo BD
< 64 Kbit/s)	-	Audio seul

Ainsi au cours de l'exécution des sessions, le service numérique évalue de façon continue la situation d'usage de chaque utilisateur et modifie, si besoin est, le comportement du service numérique.

2.1 Déroulement du scénario

Notre scénario met en scène deux utilisateurs du service numérique de téléconférence adaptative : Alice et Bob.

Alice est dans son bureau. Son terminal est connecté au réseau au travers d'une connexion qui propose un débit supérieur à 8 Mbit/s et a un écran avec une définition jusqu'à 800x480. Elle n'est pas en situation de mobilité et conserve le même terminal tout au long de la session : sa situation d'usage ne va pas évoluer.

Le smartphone de Bob est enregistré sur le service numérique comme terminal accessible. Il dispose d'un écran offrant une définition jusqu'à 800x480 et permet l'accès au réseau au travers de connexions sans fil WiMax, WiFi, UMTS et EDGE. Bob est en situation de mobilité. Son terminal change de technologie d'accès au réseau en fonction de la couverture radio : sa situation d'usage va évoluer. Nous nous intéressons seulement au cas de Bob puisqu'il est en situation de mobilité.

Le trajet emprunté par Bob dispose d'une couverture WiMax en alternance avec une couverture EDGE. La technologie WiMax fournit une très large bande passante, offrant des débits bien supérieurs à 8Mbit/s. La technologie EDGE propose des ressources réseau beaucoup plus limitées, inférieur à 56 Kbit/s. Dans notre scénario, les conditions du réseau lors d'une connexion EDGE ne permettent pas de faire passer un flux vidéo.

Lorsque Bob utilise une connexion WiMax, il dispose d'une conférence audio et vidéo

en moyenne définition. La bande passante disponible est suffisante pour proposer la vidéo en haute définition, mais le terminal offre seulement une définition de 800x480. Lorsque Bob utilise une connexion EDGE, il ne peut pas afficher de vidéo, malgré la capacité du terminal : l'échange se fera seulement en audio.

TABLEAU 2.2: Évolution des modalités applicatives de l'application de téléconférence pour la session de Bob

Temps	Événement	Modalité
00	Ouverture de session (WiMax et définition de 800x480)	Audio & Vidéo MD
10	Changement de technologie d'accès (WiMax > EDGE)	Audio seul
15	Changement de technologie d'accès (EDGE > WiMax)	Audio & Vidéo MD

Tout au long de la session de téléconférence, le système offre à Bob la modalité la plus adaptée à sa situation d'usage (Tableau 2.2). Lorsque Bob reçoit la demande d'Alice, il utilise une connexion WiMax pour accéder au réseau avec un terminal offrant une résolution de moyenne définition. La situation d'usage correspond à *Audio & Vidéo MD*. Le système propose donc une modalité d'exécution comprenant l'audio et la vidéo en moyenne définition. Dix minutes après le début de la communication, la couverture WiMax n'est plus suffisante : son terminal change de technologie d'accès au profit de l'UMTS tout en gardant une connexion au réseau. Le service numérique de téléconférence adapte son exécution pour la session de Bob et coupe le flux vidéo pour offrir une continuité du service : la modalité utilisée est *Audio seul*. Quinze minutes après le début de la communication, le terminal utilise de nouveau la technologie WiMax pour la connexion au réseau. Le service numérique va de nouveau s'adapter à la situation d'usage de l'utilisateur en réactivant le flux vidéo en moyenne définition : la modalité utilisée est *Audio et Vidéo MD*.

2.2 Problématiques soulevées

Ce scénario met en avant deux problématiques que l'on retrouve lors d'un changement de terminal ou de technologie réseau : la continuité du réseau et la continuité du service numérique. La première problématique est en phase d'être résolue par l'environnement technique des réseaux du futur présentés dans la suite de ce manuscrit (chapitre 4) et ne concerne pas le coeur de cette thèse. La problématique de continuité du service numérique est par contre encore ouverte et nous la traitons dans cette thèse.

Les réseaux du futur sont caractérisés par la convergence des technologies d'accès et de terminaux. Cette convergence apporte de nouveaux modes de consommation, par exemple la possibilité de changer de technologies d'accès ou bien de terminaux au cours d'une session multimédia. Cependant, le support du handover vertical ou bien de la mo-

bilité de terminal ne permettent pas de répondre à la problématique de continuité du service. C'est en particulier le cas lors d'un changement vers une technologie ou un terminal avec des caractéristiques radicalement différentes. En effet, les technologies d'accès ou les équipements utilisateurs ont parfois des caractéristiques très différentes et n'offrent pas les mêmes capacités. Dans cette situation, il devient impossible de proposer un service unique qui soit compatible avec tous les terminaux et toutes les technologies d'accès, tout en proposant une expérience utilisateur qui tire pleinement parti des capacités du couple terminal-technologie d'accès.

Une des approches, que nous retrouvons déjà dans nos réseaux, est de proposer plusieurs versions du même service ; proposer soit une version haute-qualité, soit une version basse qualité en fonction des technologies réseau ou du terminal. C'est le cas des services de télévision numérique sur les réseaux IP. Ils proposent une version HD (High Definition) et une version SD (Standard Definition) pour l'ADSL ainsi qu'une version en basse définition pour la 3G. Le choix d'une des versions nécessite une intervention de l'utilisateur. Cette approche est possible puisque la diversité des terminaux et technologies d'accès est limitée. Avec la convergence des terminaux et réseaux d'accès, il deviendra impossible de concevoir, déployer et maintenir autant de services différents qu'il existe de couples technologies d'accès et terminaux.

Par ailleurs, cette solution ne prend pas en compte une des fonctions qui découle de cette convergence : le changement de technologie d'accès ou terminal sans couture. En effet, il ne devient plus seulement nécessaire de choisir la modalité de l'application correspondant à la situation au début de la session ; il faut aussi pouvoir changer la modalité de l'application en fonction de l'évolution de la situation d'usage.

Permettre l'émergence d'applications sensibles au contexte dans les réseaux du futur, c'est apporter un regard différent à la problématique exprimée ci-dessus. Vis-à-vis de l'utilisateur, il s'agit de lui proposer la meilleure expérience possible relative à sa situation et ceci de façon transparente. Du point de vue du réseau, il s'agit de favoriser la rationalisation des ressources, en proposant un service, et donc un contenu, adapté à la situation *numérique* de l'utilisateur.

2.3 Adaptation continue des applications dans les réseaux du futur

Dans ce scénario de référence, nous intervenons sur l'adaptation continue des services numériques d'un réseau du futur, plus précisément sur la gestion du contexte : l'analyse de la situation d'usage et le choix de la modalité d'adaptation.

Bien que le scénario de référence présente un service numérique de téléconférence, notre travail ne se limite pas à ce dernier. Nous proposons un système de gestion du contexte générique, sous la forme d'un intergiciel, qui enrichit les services numériques

avec une gestion du contexte d'exécution sous un partitionnement des rôles. Les développeurs d'applications définissent les situations d'usage et les modalités applicatives correspondantes, mais ils n'ont pas la charge de la création des fonctions d'évaluation de la situation et de déclenchement de l'adaptation.

L'intergiciel de gestion du contexte permet l'adaptation des services numériques dans un réseau du futur. Cette adaptation n'est pas pilotée par l'utilisateur, mais elle est automatique, décidée par ledit système. Ce dernier contrôle en permanence l'environnement d'exécution. Si un changement significatif de l'environnement est détecté pour une session, il demande au service numérique concerné de modifier son comportement.

L'apport d'un système de gestion de contexte externe et générique pour les services numériques dans les NGN permet :

- Aux concepteurs de logiciels, de s'appuyer sur une architecture déjà existante pour la création, le déploiement et la mise à disposition de services numériques sensibles à leur environnement dans les réseaux du futur
- Aux utilisateurs, d'obtenir une continuité dans la consommation de leurs services dans un réseau du futur.

Peu de travaux proposent une solution globale, qui convient à l'ensemble des services numériques qui seront présents dans les réseaux du futur. En effet, la majorité des travaux concerne soit des applications spécifiques soit uniquement certains types de flux (audio et vidéo). Les seuls travaux qui semblent se rapprocher de notre objectif, adapter les applications dynamiquement à leurs conditions d'exécution, s'intègrent difficilement dans le cadre de travail les NGN. Nous en parlons plus amplement dans la partie état de l'art et plus particulièrement dans la section 5.1.

Chapitre 3

La convergence : prochaine évolution des réseaux de télécommunications

Nous sommes à l'aube d'une nouvelle évolution majeure dans les réseaux de télécommunications : l'unification des réseaux. Elle est rendue possible par la numérisation (les réseaux parlent le même langage) et la commutation de paquets (les réseaux sont capables de s'échanger de manière quasi simultanée une grande quantité de données). Il s'agit de proposer un seul et unique réseau de télécommunications. Cette évolution, qui est connue sous le terme *convergence*, concerne les différents aspects (économiques, technologiques, régulation des marchés. . .) des réseaux de télécommunications. Un signe de la maturité imminente des réseaux convergents est l'étude par la commission européenne à la fin des années 90 des problèmes de régulation apportés par ces nouveaux réseaux de télécommunications [16].

Bien que certains déploiements de réseaux de télécommunications convergents aient déjà lieu à travers le monde, la partie technique est toujours à l'étude, aussi bien par les centres de recherche académiques qu'industriels. Cette convergence concerne les différents aspects d'un réseau :

1. Les terminaux,
2. Les supports de transmission et la gestion du réseau,
3. Les services.

3.1 Convergence de terminaux

Il y a encore quelques années, chaque terminal avait une fonction unique et son propre réseau. À une tâche correspondait un équipement et un réseau :

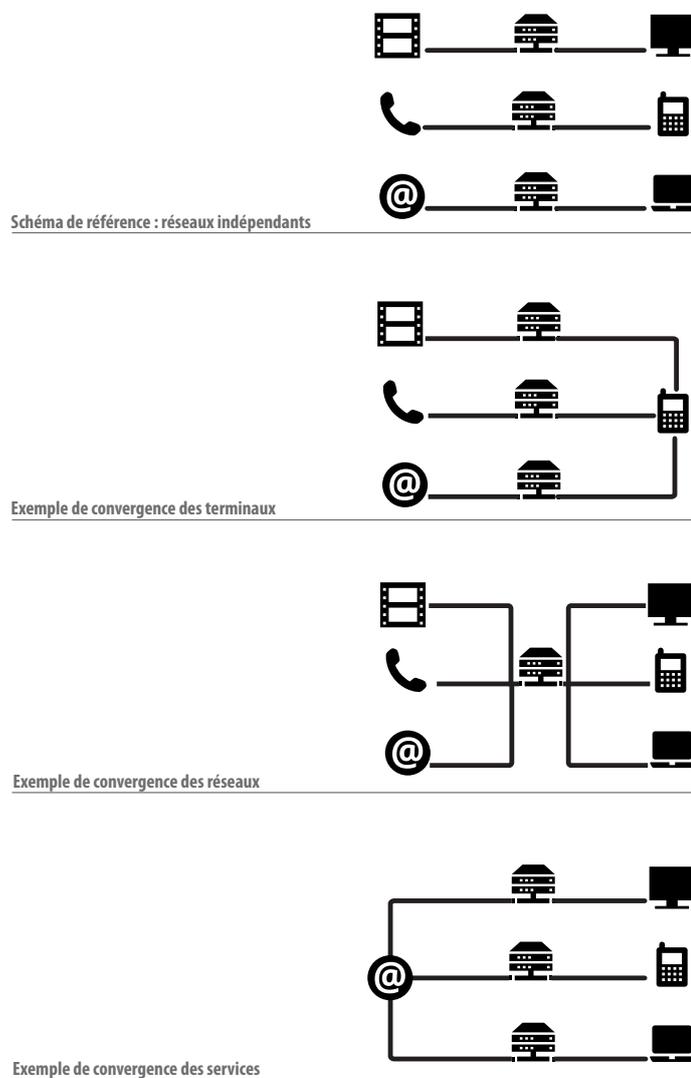


FIGURE 3.1: Principes de convergence

- Communiquer avec un téléphone sur un réseau téléphonique.
- Regarder le journal sur une télévision sur un réseau de télédiffusion.
- Écouter les émissions avec une radio sur un réseau de radiodiffusion.

De nos jours, les équipements utilisateurs regroupent de plus en plus de fonctions. L'exemple le plus répandu de cette convergence des terminaux est la démocratisation des ordiphones¹, ces téléphones semblables à des ordinateurs par leur puissance de calcul et les fonctions qu'ils proposent. Contrairement à leurs prédécesseurs qui ne permettaient que de téléphoner, il est désormais possible d'effectuer un appel vocal ou vidéo avec son terminal de poche, mais aussi d'échanger des courriers électroniques, naviguer sur le Web, regarder la télévision, écouter la radio, etc.

Ce phénomène ne touche pas uniquement les téléphones portables, mais aussi d'autres équipements de la vie de tous les jours comme les télévisions. Ces dernières se transforment en télévisions connectées. En plus de leur fonction principale, qui est la consommation des flux vidéos (chaines hertziennes, décodeur annexe, etc.), elles permettent d'accéder à Internet pour lire ses courriers électroniques, discuter en visioconférence (p. ex. : Skype) ou plus simplement regarder des vidéos à la carte².

La convergence des terminaux fournit la possibilité de regrouper plusieurs fonctions, jusqu'alors spécifiques à un type d'équipement, dans un unique terminal.

3.2 Convergence de réseaux

Parallèlement à la convergence des terminaux, nous observons une convergence des supports de transmission, que nous appelons convergence des réseaux. Actuellement, nous opposons deux catégories de réseaux de télécommunications : les réseaux filaires (xDSL, fibre optique, câble) et les accès hertziens (3G, 3G+, 4G). Chacun de ces réseaux dispose de ses propres ressources pour le transport des données, nécessite un abonnement pour son accès et n'offre qu'une seule fonction.

La convergence des réseaux apporte un réseau de transmission unique qui concentre les fonctions de gestion et les services. Les supports de transmission sont aussi bien les technologies filaires que non filaires (xDSL, fibre optique, 3G, 4G) qui permettent l'accès au réseau. Ces évolutions des réseaux de télécommunications nous mènent aujourd'hui à des réseaux *convergers* qui proposent :

- Une couche protocolaire unique, quelle que soit la technologie de transport,
- La mutualisation de certains points de l'architecture tel que le réseau cœur³.

1. De l'anglais *smartphone*. L'autre terme français consacré est *terminal de poche*

2. En anglais *Video-on-Demand* (VoD)

3. Nous appelons *réseau cœur*, la partie du réseau qui a pour fonction la gestion du réseau, la gestion des utilisateurs et la fourniture de services.

La convergence des réseaux se concrétise par la réutilisation des infrastructures d'accès actuelles et leur agrégation dans un réseau coeur unique. Les principaux bénéfices pour les opérateurs sont la réduction des coûts et l'accès à un plus grand nombre d'utilisateurs. En effet, pour permettre l'accès au réseau avec une nouvelle technologie de transport, il n'est plus nécessaire de concevoir et déployer un réseau entier, mais uniquement l'infrastructure d'accès. Cela réduit les coûts liés à l'investissement (CAPEX⁴), mais aussi ceux liés à l'exploitation (OPEX⁵). Le second bénéfice pour les opérateurs est l'augmentation du nombre d'abonnés sur leur réseau et donc du nombre potentiel de clients susceptibles de souscrire à leurs services. Ce dernier point est d'autant plus important que les attentes des utilisateurs ne sont plus limitées à l'accès au réseau de communications et au simple transfert d'informations entre des terminaux, mais à l'accès à des services à forte valeur ajoutée (vidéo à la carte, téléconférence...).

3.3 Convergence de services

La convergence des terminaux et des réseaux entraîne la convergence des services. Cette dernière est mise en oeuvre au niveau de la plateforme de diffusion de services, et peut être perçue sous deux angles :

- Fournir le même service, quel que soit le terminal ou la technologie d'accès, c'est-à-dire fusionner les services similaires déjà déployés sur les différents réseaux pour n'en proposer qu'un seul,
- Combiner plusieurs services existants pour en proposer un nouveau enrichi. Cette convergence est fortement liée à la convergence des terminaux qui apporte de plus en plus de capacités de traitement sur l'équipement final.

La télévision numérique est, dans un sens, un exemple de cette convergence de services. Le même contenu (au format numérique) est diffusé sur plusieurs réseaux de transmission :

- Réseau hertzien (utilisé pour la télévision analogique terrestre),
- Réseau satellitaire,
- Réseau ADSL (Asymmetric digital subscriber line),
- Réseau de téléphonie mobile.

Bien que ces réseaux de diffusion (hertzien, satellite, ADSL...) partagent un même contenu, ils restent indépendants. En effet, au-delà des différences d'ordre technique pour la diffusion, ils ne sont généralement pas interconnectés et chacun a une gouvernance différente.

4. Capital expenditure

5. Operational expenditure

3.4 De la convergence aux réseaux du futur

Cette nouvelle ère marquée par la convergence est la prochaine grande évolution des réseaux de télécommunications. On retrouve ces trois formes de convergence dans un seul et même réseau dans les futures architectures de réseaux de télécommunications. Les spécifications techniques qui sont les fondements de ces nouvelles architectures sont référencées sous la terminologie *réseaux du futur*.

Les attentes des utilisateurs évoluent avec ces changements. Elles ne sont plus les mêmes. Par exemple, l'utilisateur souhaite pouvoir disposer de l'ensemble de ces services, quels que soient le terminal (convergence des terminaux) et le réseau d'accès utilisé (convergence des réseaux). Un début de réponse à ce problème est apporté par le principe de convergence des services, en créant des services accessibles par plusieurs plateformes. Cependant, certaines problématiques liées demeurent ouvertes :

- Comment proposer le même service alors que les caractéristiques techniques des terminaux utilisateurs sont hautement variables (taille d'écran, puissance disponible, entrées/sorties, interfaces disponibles...) [50] ?
- Comment offrir une continuité de service, quelle que soit la variation des caractéristiques, du réseau d'accès, au cours du temps (bande passante, latence...)?

Ces questions, qui concernent la convergence des services, sont liées à la convergence des terminaux et la convergence des réseaux. Sans ces dernières, qui apportent de nouvelles fonctions et modifient les habitudes des usagers, la convergence de services n'aurait pas lieu d'être. Ainsi, pour exploiter pleinement le phénomène de convergence, il est indispensable que les services numériques puissent s'adapter à leur environnement pour :

- offrir un service numérique sans discrimination de terminal ou de réseau.
- proposer une continuité du service malgré les variations de l'environnement (changement de terminal ou de technologie d'accès).

3.5 Organisation du mémoire

Ce mémoire est organisé en quatre parties. La première partie est une introduction aux travaux de thèse de ce manuscrit avec une présentation du scénario de référence (chapitre 2) suivie d'une description de la problématique apportée par le phénomène de convergence (chapitre 3).

La seconde partie, l'état de l'art, propose deux chapitres : un premier sur les réseaux du futur (chapitre 4) et un second sur l'adaptation des applications en informatique (chapitre 5). Ce dernier chapitre présente plus particulièrement les travaux d'adaptation de services numériques dans les NGN (section 5.1), puis les applications sensibles au contexte (section 5.2) et se termine avec une présentation succincte des applications au-

tonomes (section 5.3). Nous terminons cette partie avec une présentation des ontologies dans le Web sémantique (chapitre 6).

Dans une troisième partie, je présente notre modèle pour un système de sensibilité au contexte dans les réseaux du futur. Cette présentation commence avec l'architecture de l'intergiciel (chapitre 9). Nous entrons dans le détail de l'architecture dans le chapitre 10 qui présente les blocs fonctionnels et chapitre 11 et les fonctions de l'intergiciel. Cette partie se termine avec la présentation des particularités du modèle dans le chapitre 12.

La quatrième partie présente la réalisation de notre modèle dans un NGN. Le chapitre 13 argumente nos choix technologiques, le chapitre 14 présente la mise en oeuvre de notre intergiciel dans OpenIMS. Nous terminons sur le chapitre 15 qui valide notre modèle avec le scénario de référence présenté précédemment.

Deuxième partie

Etat de l'art

Chapitre 4

Les réseaux du futur (NGN)

Les infrastructures où convergent réseaux et services sont regroupées sous l'expression *réseaux du futur* (ou *Next Generation Networks - NGN*), qui désigne, d'une façon générique, les architectures de réseaux de télécommunications à venir. Ainsi, il existera toujours des réseaux du futur, mais au cours du temps ils ne désigneront pas la même architecture. Dans la suite de ce manuscrit, les termes ci-dessous sont utilisés de façon interchangeable :

- NGN.
- Next Generation Networks.
- Réseaux du futur.
- Réseaux de demain.

4.1 Définition

Le cadre technique général qui définit les réseaux du futur est réalisé par les organismes de standardisation en télécommunications : l'ETSI¹, l'ITU-T², l'ATIS³ ainsi qu'une force commune de travail entre TTA⁴ (Corée), CCSA⁵ (Chine) et TTC⁶ (Japon). Chacune des entités citées s'est appropriée le problème dès 2004 en créant, au sein de son institution, un groupe de travail spécifique sur les architectures des réseaux convergents. Ainsi l'ETSI propose le groupe TISPAN⁷, l'ITU-T le groupe NGN-FG⁸ et l'ATIS le groupe NGN-FG. Pour une coordination entre ces différentes initiatives, le mouvement est piloté par l'ITU-T, qui publie l'ensemble des résultats, sous la forme de

-
1. European Telecommunications Standards Institute
 2. International Telecommunication Union Telecommunication Standardization Sector
 3. Alliance for Telecommunications Industry Solutions
 4. Telecommunications Technology Association en Corée
 5. China communications Standards Association
 6. Telecommunication Technology Committee
 7. Telecom and Internet converged Services and Protocols for Advanced Networks
 8. NGN Focus Group

recommandations, dont la série commence par Y.2xxx. L'ensemble de ces documents propose un modèle de conception pour les NGN. La toute première recommandation, Y.2001 [38] présente le contexte et définit un réseau du futur (NGN) de la façon suivante :

“A Next Generation Network (NGN) is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.”

Autrement dit, un réseau de télécommunications peut être qualifié de réseau du futur s'il répond aux conditions suivantes :

- Il utilise une commutation de paquets pour transporter les données,
- Il s'appuie sur différentes technologies d'accès aux réseaux (convergence des réseaux),
- Il propose une séparation entre l'infrastructure et les services,
- Il fournit un accès non restreint aux fournisseurs de services tiers,
- Il fournit un support de la mobilité dite *généralisée*,
- Il fournit une gestion de la qualité de service de bout en bout,
- Il fournit un accès non restreint au réseau pour les fournisseurs de services tiers.

Cependant un réseau du futur n'est pas uniquement un ensemble de spécifications techniques qui décrivent une architecture technique. C'est avant tout un changement vers un nouveau paradigme de communication orienté vers les services. Les services sont au coeur du réseau.

4.2 Convergence fixe-mobile

Dans un réseau du futur, le transport de données se fait aussi bien sur des technologies filaires (xDSL, fibre optique, câble) que hertziennes (HSPA, WiMax, LTE). De nos jours, ces deux catégories technologiques créent une segmentation, à la fois de l'offre commerciale et des réseaux. Les opérateurs disposent d'un réseau mobile et d'un réseau fixe, proposant respectivement le haut débit mobile et le haut débit fixe. La convergence fixe-mobile est l'unification de ces deux types de réseaux (filaires et hertziens) pour n'en former qu'un seul et unique. Quelle que soit la technologie d'accès utilisée (xDSL, fibre optique, WiMax, UMTS...), l'utilisateur accède à un seul et même réseau. Il ne souscrit qu'un seul abonnement pour l'accès au réseau et ses services sont accessibles, quelle que soit la connexion utilisée (fixe ou mobile). Bien qu'actuellement, il existe une interconnexion entre ces réseaux, nous ne pouvons pas encore parler d'unification. En effet, il n'est actuellement pas possible :

- De garantir la performance d’un échange pour des données traversant plusieurs réseaux,
- De changer de technologie d’accès ou de terminal sans perdre la connexion réseau. Ceci sous-entend qu’il faut couper le service en cours d’exécution. Par exemple, pour la technologie d’accès, il s’agit de conserver la connexion lorsque le téléphone doit commuter d’une connexion WiFi (via l’abonnement ADSL) à une connexion “3G” (lié au forfait mobile). Ce changement est appelé *handover vertical*,
- D’accéder aux services temps réels dont la contrainte de performance est forte en dehors du réseau de l’opérateur. Par exemple, il n’est pas possible d’accéder, depuis son mobile, à son service VoIP souscrit avec son abonnement ADSL.

L’unification passe par l’intégration complète de toutes les technologies d’accès dans un seul réseau coeur. Cela crée un réseau unique proposant des services identiques, quels que soient l’équipement et la connexion au réseau utilisé. Pour réaliser cette infrastructure, il est nécessaire d’avoir une couche d’abstraction entre les technologies d’accès et les services.

4.3 Séparation fonctionnelle

La couche d’abstraction crée une démarcation entre les fonctions relatives au transport (spécifiques pour chaque technologie d’accès) et les services. Le système se retrouve ainsi scindé en deux couches, chacune cachant sa complexité par rapport à l’autre. Nous trouvons d’un côté une couche de transport, et de l’autre une couche service. La couche transport a pour rôle l’acheminement des données entre les différents équipements, quelle que soit la technologie d’accès utilisée. Ses fonctions sont :

- Adapter le protocole de commutation de paquets à la couche de liaison physique,
- Authentifier le terminal sur le réseau,
- Gérer le transport des données (qualité de service, accès aux ressources...),
- S’interconnecter physiquement avec les réseaux traditionnels (passerelles).

La couche service est relative à l’utilisation du réseau et offre :

- L’authentification de l’utilisateur sur le réseau et les services associés,
- La fourniture de services et le contrôle d’accès à ces derniers.

Cette séparation des rôles entraîne un effet de bord : la redondance de certaines fonctions, comme l’authentification ou le contrôle d’accès. Par ailleurs, il est nécessaire de faire travailler de concert ces deux couches. C’est pourquoi il existe une couche transversale : la couche de gestion. Elle effectue une jonction entre les couches service et transport. Cette séparation fonctionnelle est présentée dans la recommandation Y.2012 de l’ITU-T [39] dont est extrait la Figure 4.1.

L’ITU-T définit trois interfaces pour dans un réseau du futur :

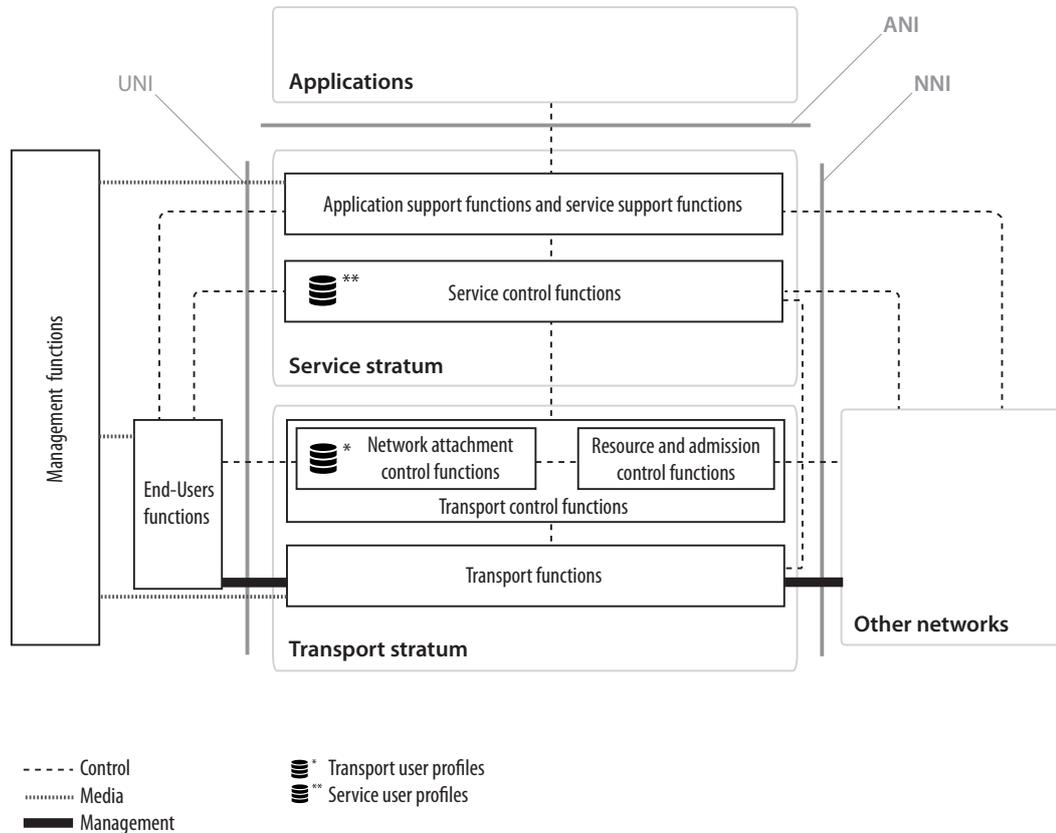


FIGURE 4.1: Architecture d'un NGN [39]

- UNI (User-Network Interface) - Démarcation entre l'utilisateur et le réseau.
- ANI (Application-Network Interface) - Séparation logique entre le réseau et les applications.
- NNI (Network-to-Network Interface) - Interconnection entre un NGN et d'autres réseaux (NGN ou traditionnels)

Les entités de chaque zone, délimitées par les interfaces, interagissent les unes avec les autres à l'aide de flux. Ces flux sont les liens entre les différents éléments qui composent le réseau et son écosystème. Nous distinguons trois catégories de flux dans un NGN :

- Les flux de contrôle (Control) qui transportent les données liées à l'établissement et au contrôle des sessions,
- Les flux de données (Media), c'est-à-dire les données échangées entre les applications et les terminaux utilisateurs,

- Les flux de gestion du réseau (Management).

4.4 Gestion de la qualité de service

Pour offrir des performances de transport qui permettent un échange du trafic en temps réel, les réseaux du futur s'appuient sur des mécanismes de qualité de service. Pujolle définit dans ses ouvrages [64] la qualité de service (QoS⁹) de la façon suivante :

“La qualité de service est le moyen de garantir que le service de transport sera capable de respecter une performance donnée.”

La problématique d'échange de données en temps réel dans les réseaux du futur est double :

- Les réseaux traditionnels sont liés à un seul cas d'utilisation (téléphonie, diffusion vidéo ...). Les technologies de transport sont optimisées pour cette tâche. Comment apporter la même performance en créant un réseau qui généralise la commutation de paquets et s'appuie sur les infrastructures de transport existantes, pour échanger des données qui concernent divers usages ?
- Les réseaux du futur sont destinés à s'interconnecter avec d'autres réseaux du même type et des réseaux traditionnels. Comment garantir la performance d'acheminement malgré le changement de catégorie ?

Le premier problème concerne principalement le changement de paradigme de communication. Nous passons d'un support réseau dédié et optimisé à un support réseau multiusage. Cette problématique est abordée depuis plusieurs années. Prenons l'exemple de la téléphonie traditionnelle. À l'origine, ce service s'appuie sur un réseau basé sur la commutation de circuits. Cette technique de commutation offre un temps d'acheminement fixe, ceci au détriment du taux d'utilisation des équipements. Depuis quelques années, le service de téléphonie change de support. Nous passons d'un réseau basé sur une commutation de circuits à une commutation de paquets (protocole IP). Le protocole IP se préoccupe uniquement de l'acheminement sans prendre en compte les contraintes de performance. Il est qualifié de réseau *best-effort*. Dans le cadre d'applications multimédia, il n'est pas possible de se contenter d'un acheminement *best-effort*. Ce changement de technologie a donc entraîné plusieurs challenges, dont l'application d'une performance d'acheminement. Dans le cas de la téléphonie sur IP, cela revient à limiter la variation et le temps d'acheminement. Si ce dernier dépasse une valeur seuil, la conversation devient désagréable par une perception d'écho ou de retard. Bien que l'appréciation de la qualité de la communication dépende principalement de l'utilisateur, c'est-à-dire une mesure subjective, nous pouvons dégager trois paramètres du réseau qui influent sur cette perception [31] :

9. Quality of Service

- La perte de paquets,
- La latence,
- La gigue (variation de la latence).

La latence exprime le temps que met un paquet à atteindre sa destination. C'est le paramètre le plus important. L'ITU-T dans sa recommandation G.114 recommande une latence inférieure à 150 ms, c'est-à-dire que les données doivent être acheminées en moins de 150 ms entre les terminaux engagés dans la communication. Pour assurer cette performance, les réseaux à commutation de paquets se sont dotés d'outils de gestion de la qualité de service.

Le second problème concerne l'interconnexion des réseaux. En effet, chaque réseau dispose de ses propres mécanismes de gestion de qualité de service. Cependant, cette gestion reste locale. Elle est appliquée au cas par cas au sein de chaque réseau. Chaque réseau est sous une administration et une autorité différente : nous parlons de domaine d'autorité. Ainsi, chaque opérateur de réseau dispose, sur son réseau, de ses propres fonctions de qualité de service. Puisqu'il n'a pas été prévu de qualité de service au sein des réseaux IP et que l'ajout ad hoc de ces protocoles ne prend pas en compte l'échange des politiques de gestion de QoS entre les réseaux ; il n'existe pas de continuité pour le support de la gestion de qualité de service.

La gestion de la performance tout au long du chemin de communication est désignée sous le terme qualité de service de bout en bout¹⁰. Cette mise en place d'une qualité de service sur des réseaux hétérogènes peut être réalisée au travers de plusieurs architectures [53]. Ce sujet est d'ailleurs l'objet du projet européen EuQoS [56].

L'approche retenue pour les réseaux de demain est une généralisation de la qualité de service au sein des réseaux. C'est donc au travers d'une gestion unique, quelle que soit la technologie d'accès, ainsi que d'une communication sur ses politiques de gestion avec les autres réseaux que le contrôle de la performance est réalisé. Ce sont les passerelles des réseaux qui permettent l'échange et la synchronisation des politiques de qualité de services. Une nécessité pour ces réseaux qui, à terme, vont remplacer nos réseaux traditionnels. Cependant, garantir une performance d'acheminement des paquets (rôle de la couche de transport) n'est qu'un aspect du problème de QoS dans les réseaux du futur. En effet, avant de pouvoir garantir l'acheminement, il faut que le réseau ait connaissance des contraintes qu'il doit respecter.

Dans les réseaux actuels, comme les réseaux haut débit filaires, les applications sont limitées (vidéo, internet ou téléphonie). Elles sont fournies par la même entité qui opère le réseau, en l'occurrence, le fournisseur d'accès à internet (FAI). La gestion de la qualité de service s'en trouve simplifiée puisque les types de flux sont connus et le réseau maîtrisé. Dans les réseaux de demain, les types de services sont hétérogènes et doivent être proposés par des fournisseurs de contenu tiers qui n'auront pas la main sur le réseau de

10. End-to-End QoS ou E2EQoS.

transport. Pour garantir le transport des données selon une performance donnée, l'ITU-T propose un mécanisme de négociation des paramètres de qualité de service lors de l'établissement d'une session multimédia entre les entités impliquées par la session [73]. S'ensuit une demande de réservation par les terminaux. Ce processus garantit l'indépendance des couches sur le schéma client/serveur, en sollicitant une réservation de ressources auprès de la couche transport.

D'un point de vue de l'architecture technique, la réservation des ressources est transmise par les applications et terminaux au module *Ressources Attachment and Admission Control Functions (RACF)*. Cette entité qui appartient à la couche de gestion s'appuie sur deux sous-entités, le PD-FE et le TR-FE, qui assurent respectivement :

- Le contrôle de la demande : Le PD-FE vérifie la disponibilité des ressources et l'autorisation de l'utilisateur.
- L'application de la demande : Le TR-FR applique les règles de contrôle et de gestion du trafic au sein des noeuds du réseau (routeurs, commutateurs...).

À noter que l'application de la QoS au niveau transport s'appuie sur deux entités, le TR-FE (couche services) et le PE-FE (couche de transport).

4.5 Fournisseurs de services

La séparation fonctionnelle du réseau crée une partition des métiers. Dans un cas idéal, nous avons d'un côté l'exploitation du réseau, proposée par un opérateur de réseau. De l'autre, la fourniture de services aux utilisateurs, à travers une plateforme de diffusion, proposée par un fournisseur de contenus et services. Dans la pratique, certains acteurs joueront les deux rôles. Cependant, l'ITU-T insiste sur la mise à disposition d'interfaces ouvertes, à l'aide d'API¹¹ ou de serveurs proxy, pour l'accès à cette plateforme à des fournisseurs de services tiers. Cette directive n'exclut pas que les opérateurs de réseaux deviennent des opérateurs de services. Elle doit apporter une concurrence égale entre les services de l'opérateur et les fournisseurs de services tiers. Ce qui semble évident d'un point de vue de l'utilisateur d'Internet ne l'est pas du point de vue de l'opérateur de télécommunications. Ceci fait écho aux récents débats sur la neutralité des réseaux¹².

Les réseaux de télécommunications comme nous les connaissons permettent d'accéder à des services fournis par les opérateurs. Nous avons accès à différents services. Ils sont liés à la souscription de l'accès au réseau. Par exemple, l'offre haut débit fixe propose la téléphonie sur IP, la télévision et l'accès à Internet. Il en va de même avec les offres mobiles, qui proposent en plus de l'accès au réseau GSM, la télévision et l'accès internet. Les services sont l'occasion, pour les fournisseurs d'accès à Internet de se différencier et

11. Application Programming Interface (Interface de programmation). Permet l'interopérabilité entre des applications grâce à des fonctions et spécifications accessibles depuis l'extérieur d'un programme.

12. <http://www.arcep.fr/index.php?id=8652>

de justifier leurs prix. Ces services à forte valeur ajoutée sont appelés *services gérés*¹³. Ces derniers sont la prérogative des fournisseurs d'accès à Internet entre autres pour des raisons techniques puisqu'ils nécessitent un contrôle de la qualité de service. La neutralité des réseaux propose donc de séparer les deux rôles, acheminement des données et fourniture de services, pour permettre une libre concurrence sur la couche service. Il n'existe plus de *services gérés* (services favorisés par l'opérateur).

L'argument souvent mis en avant est la nécessité par l'opérateur de contrôler son réseau. En effet, il est possible de gérer efficacement les flux temps réel sur un réseau IP, à condition de mettre en place une gestion de la qualité de service. Or ceci n'est réalisable que sur son propre réseau. Cela devient plus compliqué sur les réseaux tiers pour des raisons d'interaction de contrôles. Par voie de conséquence, il devient donc très difficile d'assurer une gestion de QoS sur un ensemble de réseaux sous des autorités différentes. Chaque réseau dispose de sa propre gestion, pas forcément alignée avec les réseaux auxquels elle est interfacée. Par ailleurs, aucun protocole n'a été proposé pour favoriser ces échanges. Cet état de fait explique le manque de gestion de qualité de service sur Internet. En effet, Internet est une interconnexion de réseau, donc un ensemble de réseaux sous des autorités différentes, simplement interconnectés entre eux. Les réseaux sans gestion particulière de la QoS sont appelés "best-effort", ce qui donna naissance à la classe de service QoS du même nom.

Même s'il est possible de souscrire à des services tiers sur Internet, certaines applications ne pourront pas offrir la performance attendue, comme pour la téléphonie sur IP. Nous avons d'un côté un réseau fermé et maîtrisé qui permet la fourniture de services, mais qui souffre essentiellement d'un défaut de nouveaux services numériques. De l'autre côté, un réseau ouvert, non maîtrisé et avec une faible barrière à l'entrée, où chaque jour de nouveaux services font leur apparition. Cependant, Internet ne permet pas, en raison de sa conception, de proposer des services avec une contrainte de transfert en temps réel, contrairement aux réseaux de télécommunications.

Pour l'ITU-T, un NGN est un écosystème proche de celui d'Internet dans un réseau extrêmement performant : le meilleur des deux mondes. Le réseau est géré par un opérateur, donc sous une juridiction unique, sur lequel des fournisseurs de services tiers proposent leurs applications. Nous avons ainsi l'assurance d'un réseau performant et d'une plateforme ouverte favorisant l'innovation de services.

Nous avons, au-delà d'une séparation des fonctions du réseau, une séparation des rôles avec l'apparition d'un métier : fournisseur de service multimédia. Il se différencie de ce que nous appelons communément *fournisseur de contenu*, par l'accès à un domaine qui jusqu'alors était réservé à l'opérateur : le réseau opérateur. Grâce aux performances offertes par les réseaux du futur, les barrières techniques pour la création d'applications

13. Les services gérés sont des services offerts par l'opérateur à travers son propre réseau. Ils ne sont pas accessibles depuis Internet et sont l'exclusivité des abonnés de l'opérateur. En somme, il ne passent pas par Internet.

ont été repoussées. Il ne s'agit plus de proposer simplement une interconnexion, mais des services. Les services, qui sont aujourd'hui l'apanage de certains terminaux ou réseaux seront accessibles à tous les utilisateurs, quel que soit le terminal ou la technologie d'accès.

4.6 IMS, la réalisation concrète d'un NGN

Les réseaux de demain, dont le cadre général est édicté par nos organismes de standardisation, ne restent pas au stade de rêve, mais deviennent une réalité. Ainsi, British Telecom a lancé en 2006 le projet 21CN (21st Century Networks) dont l'objectif est de proposer un réseau unifié basé sur IP. Bien que ce soit l'initiative la plus médiatique, ce n'est pas le seul opérateur à implémenter des NGN. On retrouve des programmes similaires chez KPN, Telecom Italia, Telefonica, Telefonica Espana, Swisscom, Verizon, Softbank Mobile Corp, France Telecom, Bell Canada... L'ensemble de ces projets concerne principalement la première étape d'un NGN : la mise en place d'un réseau unifié sur IP. Nous retrouvons d'autres initiatives mixtes (industrielles et scientifiques), dont le projet de recherche européen EuQoS [56]. L'objectif est de proposer une architecture qui propose une qualité de service de bout en bout [65] sur un réseau hétérogène : multi domaines, multi technologies et multi services. La proposition la plus aboutie est IP Multimedia Subsystem (IMS). C'est une réalisation de l'architecture d'un NGN tel que l'a défini l'ITU-T. Elle est pilotée par les organismes de standardisation en télécommunications :

- L'architecture est créée par l'ETSI et le 3GPP¹⁴.
- Les protocoles sont adaptés par l'IETF¹⁵.
- L'interopérabilité des services et du réseau sont pris en charge par l'OMA¹⁶.

Les contributions ne sont pas limitées à ces organismes. Des industriels participent activement à ce projet, comme l'opérateur Canadien CableLabs. Ce dernier apporte, au travers du projet PacketCable, son savoir-faire pour adapter la technologie d'accès au réseau câblé à un réseau IMS. C'est une initiative importante pour la viabilité du projet puisqu'elle enrichit techniquement le projet. De plus, elle montre l'intérêt des opérateurs pour ce projet. Pour conclure, l'objectif affiché d'IMS est d'offrir un réseau qui est capable de s'interconnecter avec d'autres réseaux IMS (ou réseaux traditionnels), à l'image d'Internet qui est une interconnexion de réseaux, tout en offrant une multitude de services. L'objectif est de garder la performance et la fiabilité (QoS) des réseaux de télécommunications actuels tout en offrant un environnement permettant la création et la diffusion de services avec une faible barrière à l'entrée. Les conditions de réalisation de cette architecture sont énoncées dans [2]. Les plus importantes sont :

14. Third Generation Partnership Project

15. Internet Engineering Task Force

16. Open Mobile Alliance

- Interconnexion avec les réseaux : Un réseau IMS s’interconnecte avec d’autres réseaux IMS mais aussi avec les réseaux traditionnels. Une communication doit pouvoir avoir lieu entre un utilisateur sur un réseau IMS et un utilisateur d’un autre type de réseau (p. ex. : cellulaire, à commutation, internet).
- Mobilité généralisée : Un utilisateur doit pouvoir accéder à ses services quelle que soit sa position géographique, c’est-à-dire, à travers un réseau IMS tiers.
- Connectivité IP : IMS doit être perçue comme un réseau IP. Tous les échanges s’effectuent en utilisant le protocole IP.
- Support des sessions multimédia : Chaque communication, que ce soit entre des terminaux ou une consommation de services, est perçue par le système comme une session multimédia. C’est un héritage des réseaux mobiles (UMTS par 3GPP) qui proposait ce terme pour les appels en visiophonie sur un réseau à commutation de circuits.
- Qualité de service : Une qualité de service de bout en bout. Elle est négociée par l’ensemble des parties engagées dès le début de la communication. Une fois négociées, les ressources réseau (bande passante, latence) sont réservées et appliquées sur le réseau.

IMS correspond en tout point à un réseau du futur, tel que défini par l’ITU-T. La suite de cette partie décrit les principes clés d’IMS, les protocoles implantés ainsi que son architecture.

4.6.1 Principes clés d’IMS

Séparation fonctionnelle

L’architecture d’IMS s’appuie sur deux principes, en apparence distincts, mais en réalité très proches :

- Les réseaux de télécommunications mobiles (standardisés par 3GPP) qui mettent l’accent sur la séparation entre le réseau coeur et le réseau d’accès
- Les recommandations de l’ITU sur l’architecture des NGN qui prônent la séparation fonctionnelle.

Les idées dominantes de ces concepts sont dans chaque cas :

- Un découplage entre les fonctions (transport et gestion) du réseau.
- Une gestion du réseau au travers d’un protocole de signalisation : Il s’agit de travailler de pair avec les applications/terminaux.

La séparation fonctionnelle d’IMS est logique, il n’existe pas d’infrastructure parallèle ou de circuit logique isolé pour échanger les données entre fonctions. Autrement dit, tous les flux (contrôle, gestion, applications) sont mélangés au sein du réseau. La séparation fonctionnelle est organisée autour de ces trois types de flux, débouchant sur un modèle composé de trois plans (Figure 4.2) :

- La *couche de transport* qui regroupe l'ensemble des fonctions de transport sur les supports physiques et qui achemine l'ensemble des flux (contrôle, gestion réseau ou applicatif).
- Le *coeur du réseau* qui regroupe l'ensemble des entités de gestion et de contrôle du réseau.
- La *couche application* qui regroupe l'ensemble des services proposés aux utilisateurs du réseau.

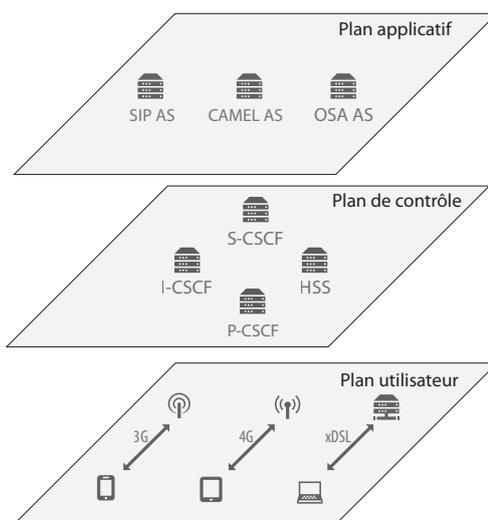


FIGURE 4.2: Les trois couches d'IMS et l'indépendance de l'accès

Ce découpage du réseau en trois plans n'est pas sans rappeler le partitionnement des réseaux mobiles avec le réseau coeur (CN - Core Network) et le réseau d'accès (AN - Access Network).

Les services au coeur du système

Depuis quelques années, les opérateurs de télécommunications réalisent leurs marges, non plus sur la transmission (le réseau), mais les services. France Télécom le rappelle dans son rapport annuel financier de 2010 [74].

“Pour rester compétitif, France Télécom doit maîtriser des technologies complexes au travers des systèmes et des processus en évolution permanente et capables de délivrer les services attendus des clients.”

L'enjeu est donc de pouvoir proposer aux clients finaux des services à valeur ajoutée, qu'ils soient éphémères (pour un événement) ou durables. Au-delà d'un réseau convergent, les NGN apportent aussi de nouvelles perspectives en terme de services [81]. IMS a été créé pour répondre à cet enjeu en proposant une plateforme de communication multimédia standardisée, se basant sur les recommandations formulées par l'ITU-T. Il s'agit de regrouper deux mondes :

- D'un côté Internet qui a eu un fort impact sur le développement de nouveaux usages et services. L'interconnexion de réseaux avec une faible barrière à l'entrée a permis l'émergence d'une multitude d'applications accessibles à un très grand nombre d'utilisateurs. De l'échange et l'affichage d'informations au format texte, nous sommes arrivés à des applications enrichies tels que les logiciels de bureautique ou de vidéoconférence accessibles depuis un navigateur web. Dans le même temps l'ouverture d'Internet, grâce à ce phénomène d'interconnexion, reste son plus grand défaut. Il est très difficile et complexe de réaliser des applications temps réels, puisque la qualité de service n'est pas assurée de bout en bout.
- De l'autre, les réseaux de télécommunications, qui sont de vastes réseaux fermés, mais maîtrisés. Les contraintes d'acheminement des données sont totalement maîtrisées, mais les nouvelles applications se font rares.

L'architecture d'IMS doit permettre de fusionner ces deux modèles en ne prenant que le meilleur : un réseau performant, maîtrisé et ouvert aux nouvelles applications.

Le contrôle de session

La session est l'élément de base d'un réseau IMS. Elle symbolise le lien entre l'origine de la transmission et sa destination. Elle permet d'échanger tous types de flux (audio, vidéo, données). Pour comprendre l'importance du concept de session et son implication dans un réseau du futur, il faut faire le parallèle avec le concept d'un appel téléphonique dans un réseau de télécommunications. La communication transporte la voix entre deux entités. La session transporte des données entre deux entités ; elle symbolise une communication dans un NGN.

Ce parallèle ne s'arrête pas au concept d'échange entre deux entités, mais reprend l'ensemble du mécanisme d'un appel dans un réseau traditionnel. Il reprend les principes d'établissement de la communication, de contrôle, de facturation, etc. Pour réaliser ces opérations de gestion, les réseaux de télécommunications s'appuient sur un plan de signalisation.

IMS étant un réseau de télécommunications, il possède lui aussi son plan de signalisation. Ce lien de contrôle permet :

- la négociation des paramètres de qualité de services,
- l'authentification des utilisateurs,
- l'ajout/suppression des flux de données au sein de la session,

- la facturation,
- etc.

Une session est donc une communication entre deux entités durant laquelle sont échangés des flux (audio, vidéo ou données). Cette session est supervisée via un protocole de signalisation qui permet l'échange des informations entre les équipements du coeur du réseau pour authentifier les utilisateurs, négocier la qualité de service, facturer les clients, etc.

La qualité de service

Bien qu'un modèle général de gestion de la QoS pour les NGN [18] ait été défini, il existe plusieurs versions en fonction des groupes de travail (l'ITU-T, l'ETSI, 3GPP, DSL Forum et PacketCable). Ces approches ont fait l'objet de comparaisons dans le monde scientifique [73] [79] [55]. La version 7 d'IMS par TISPAN (2008) a été l'occasion d'uniformiser le modèle de qualité de service [62] [45]. Ainsi, la qualité de service de bout en bout s'appuie sur deux principes : la QoS relative et la QoS garantie. La première se base sur la différenciation du trafic, et la seconde sur une réservation de ressources.

La procédure de réservation des ressources s'opère en trois étapes :

1. Négociation des paramètres lors de l'ouverture de session entre les entités concernées,
2. Demande de réservation auprès des éléments de gestion du réseau,
3. Application de la réservation des ressources par les équipements de la couche transport.

IMS offre aux services numériques la possibilité de renégocier les paramètres de la session au cours de l'exécution.

4.6.2 Les protocoles

IMS est un réseau de télécommunications qui utilise la commutation de paquets. Le protocole retenu est l'Internet Protocol (IP) qui est déjà largement présent et utilisé dans nos réseaux informatiques. Il est à la base de tous les autres protocoles présents dans IMS. Bien que les protocoles qui s'appuient sur l'IP soient nombreux et variés, seuls certains sont présents dans IMS. Cette sélection a été effectuée par les organismes de standardisation devant la nécessité de faire interagir des éléments hétérogènes de façon homogène. Nous pouvons classer ces protocoles en fonction de leur type dans un NGN :

- Les flux de contrôle,
- Les flux de gestion,
- Les flux de transport.

Les flux de contrôle permettent la gestion des utilisateurs et de session, ils s'appuient sur le protocole de signalisation Session Initiation Protocol (SIP). Il a été retenu pour établir et piloter les sessions multimédia. Ce protocole, déjà très populaire dans les logiciels de VoIP¹⁷, a été modifié pour répondre pleinement aux besoins d'IMS. 3GPP s'appuie sur l'IETF (qui spécifie les protocoles basés IP) pour apporter, s'il le faut, des modifications aux spécifications existantes.

Le protocole DIAMETER a été choisi pour les fonctions d'AAA (Authentification¹⁸, Authorization¹⁹, Accounting²⁰). Ce protocole est utilisé pour le transport des informations de gestion, c'est-à-dire pour les échanges entre les noeuds du réseau. Cela concerne les authentifications, le contrôle d'accès et la facturation. DIAMETER est une évolution de RADIUS. Ce dernier est un protocole largement présent dans les réseaux ; aussi bien dans les réseaux locaux que dans les réseaux d'opérateurs.

Le protocole IP propose plusieurs modes de transport de l'information : Transmission Control Protocol (TCP) et User Datagramm Protocol (UDP). Le premier garantit l'acheminement du message, il est dit fiable. Le second n'apporte aucune garantie d'acheminement en échange d'une plus grande simplicité de fonctionnement ; il est dit *non fiable*. Le plan de transport d'un réseau IMS peut utiliser tous les protocoles se basant soit sur TCP, soit sur UDP. Pour les applications qui ont une sensibilité temporelle, comme les flux multimédia en temps réel, c'est le protocole Real-time Transport Protocol (RTP) qui est utilisé. On le retrouve en conjonction avec Real-Time Control Protocol (RTCP). Cette (s)élection de protocoles de transport n'est pas nécessaire, mais elle permet de créer une base commune entre les différents intervenants du réseau IMS (équipementiers, fournisseurs de services et opérateurs).

Pour résumer, le protocole de base pour le transport de données est IP. Dessus vient se greffer le protocole de contrôle SIP, utilisé pour échanger des informations de gestion de sessions entre les terminaux, le réseau et les services. La gestion du réseau est accompagnée par le protocole DIAMETER pour l'échange sécurisé d'informations entre les noeuds coeurs d'IMS. Le transport des flux multimédia est assuré par les protocoles TCP et UDP ainsi que RTP pour les contraintes temps réel.

Les spécifications d'IMS ne font pas mention d'autres protocoles particuliers. L'ensemble des protocoles dits de haut niveau (HTTP, SMTP, etc.), c'est-à-dire relatifs aux applications, est utilisable dans IMS. En effet ils dépendent du service numérique et non pas du réseau.

17. Voice over IP

18. Authentifier - Vérifier l'identité d'une entité.

19. Autoriser - Permettre, ou non, l'accès d'une ressource spécifique à une entité.

20. Facturer - Récouter des informations sur la consommation de la ressource par l'entité pour auditer, gérer ou facturer l'usage.

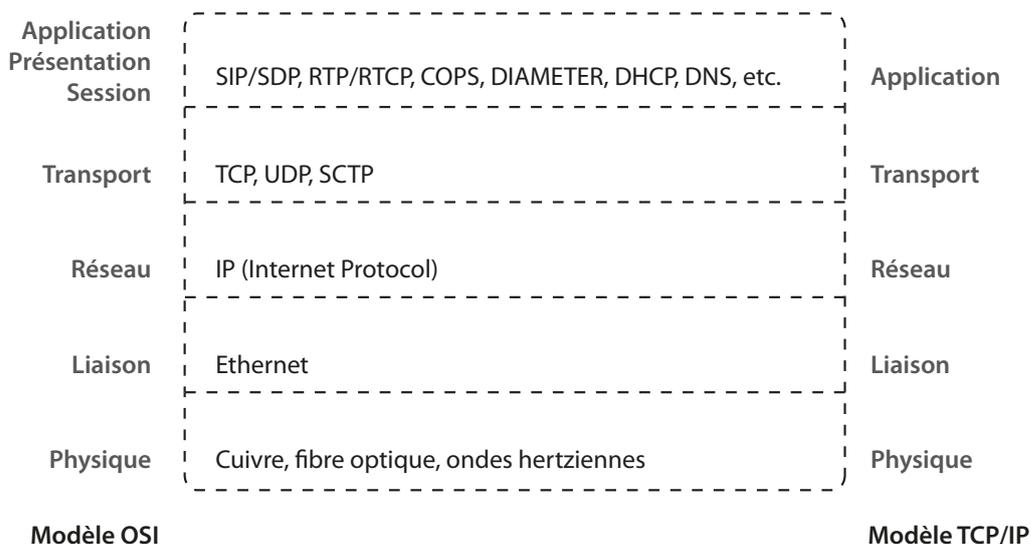


FIGURE 4.3: Protocoles utilisés dans IMS

4.6.3 Architecture d'IMS

Bien qu'IMS soit divisé en trois plans (Figure 4.2) nous pouvons aussi regrouper les entités d'IMS autour de 5 blocs :

- Les terminaux utilisateurs,
- Le coeur de réseau qui gère l'ensemble du réseau,
- Le réseau d'accès qui fait le lien entre les terminaux utilisateurs et le coeur de réseau,
- Les passerelles qui font le lien entre un réseau IMS et les autres réseaux (IMS et traditionnels),
- Les serveurs d'applications qui hébergent les services numériques pour les utilisateurs.

Dans cette section nous présenterons successivement chacun de ces blocs.

Le réseau coeur

Le coeur du réseau IMS est le centre névralgique sur lequel s'agrègent d'un côté les technologies d'accès, et de l'autre les services. C'est le point de passage obligé du trafic entre les terminaux utilisateurs et les services numériques. Les rôles du coeur sont :

- L'authentification,

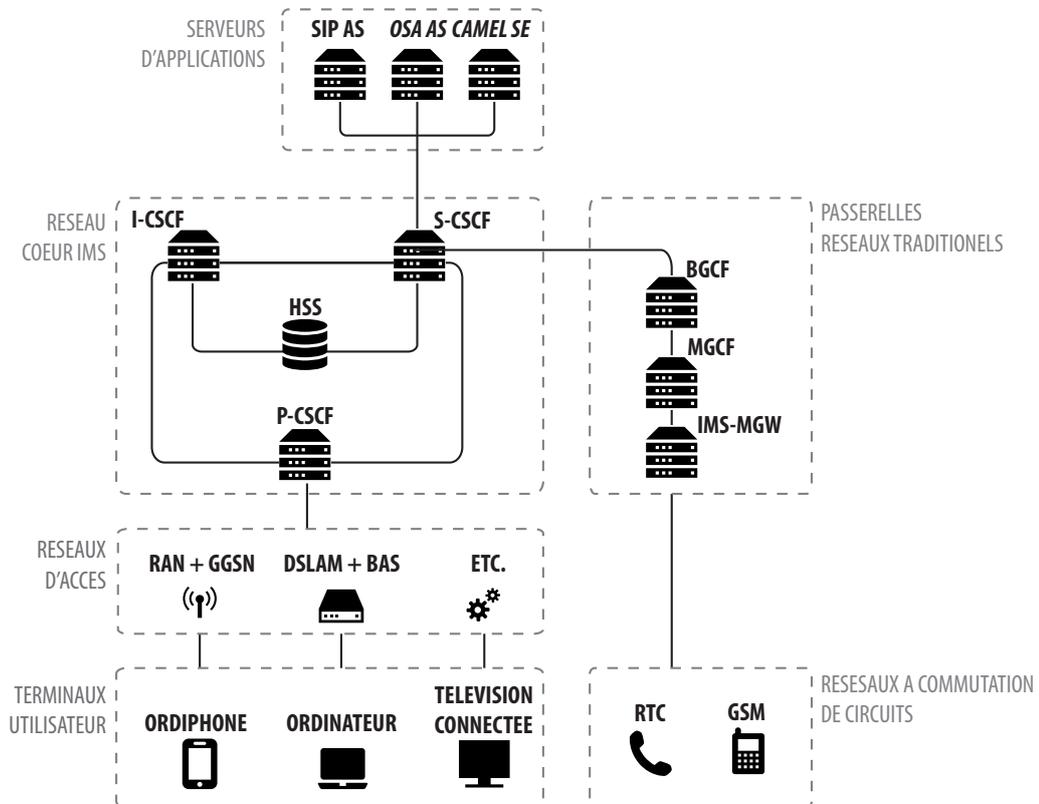


FIGURE 4.4: Architecture d'IMS

- La facturation,
- Le contrôle d'accès aux ressources,
- La gestion des sessions.

Pour réaliser ces missions, il s'appuie sur un ensemble d'entités atomiques [1] qui sont interconnectées entre elles. Chaque interconnexion est identifiée sous la forme d'un point de référence lié à un protocole de gestion.

Pour pouvoir proposer des services et assurer des échanges entre les utilisateurs, un coeur de réseau IMS doit être composé a minima :

- D'une base de données utilisateur,
- Des entités de gestion et de contrôle.

Ces fonctions sont assurées, en terme d'entités IMS, par le *Home Subscriber Server (HSS)* et les *Call Session Control Function (CSCF)* qui sont présentées en annexes (Appendice A). Une description détaillée des interconnexions et entités déjà ratifiées est disponible dans le livre *IMS - Concepts and Services* [63].

Les serveurs d'applications

IMS a été créé pour promouvoir les services au sein des réseaux de télécommunications. Les services sont hébergés et rendus disponibles par l'intermédiaire de serveurs d'applications. Ils appartiennent soit à l'opérateur soit à un fournisseur tiers. Ces serveurs sont situés soit au sein du réseau IMS soit dans un autre réseau. Nous distinguons trois catégories de serveurs d'applications (AS) en fonction de leurs interfaces [40] :

- Serveur d'applications SIP,
- Serveur d'applications OSA ²¹,
- Serveur d'applications CAMEL ²².

SIP étant le protocole de signalisation utilisé dans IMS, les serveurs d'application (AS) SIP sont dits natifs. Ils sont capables de recevoir et d'émettre directement les messages de signalisation sans passer par des passerelles.

Les serveurs OSA et CAMEL s'appuient sur un serveur intermédiaire pour traduire les messages SIP dans leur langage. Cette traduction de la signalisation (SIP vers OSA ou CAMEL) est réalisée par l'adjonction de médiateurs de services OSA-SIP et CAMEL-SIP. Ils sont respectivement dénommés dans IMS OSA-SCS ²³ ou IM-SSF ²⁴. 3GPP a souhaité intégrer des serveurs non-SIP pour promouvoir la fourniture de services au lancement du réseau en offrant une phase de transition qui permette la réutilisation des services déjà présents dans les réseaux traditionnels. Cependant les organismes de standardisation recommandent de créer des services en utilisant les serveurs d'applications SIP, les serveurs OSA et CAMEL n'étant présents que pour une phase de transition.

Un serveur SIP n'est autre qu'un serveur d'applications avec une interface SIP, ce qui permet de créer des services numériques avec différents rôles :

- Serveur proxy,
- Serveur de redirection,
- Entités utilisatrices,
- Serveur B2BUA ²⁵, jouant à la fois le rôle de client et de serveur.

21. Open Services Architecture est une architecture système ouvert pour les réseaux de télécommunication mobile à commutation de paquets (P. ex. UMTS). Cette API doit permettre de déployer des services indépendamment des technologies utilisés par le réseau.

22. Customised Applications for Mobile Enhanced Logic est un standard pour la fourniture de services dans un réseau GSM.

23. OSA Service Capability Server

24. IP Multimedia Service Switching Function

25. Back-to-Back User Agent

Pour qu'un serveur d'application IMS soit natif, il faut qu'il dispose uniquement d'une interface SIP : c'est la seule contrainte technologique. Aucune préconisation n'est faite sur les autres technologies dans les serveurs d'applications (en particulier sur la logique métier des services numériques). Ce choix est laissé à la discrétion du fournisseur de services ou de l'opérateur.

Le réseau d'accès et la gestion de la QoS

La gestion de la QoS s'appuie sur une entité qui joue le rôle de pivot entre les couches de service et de transport. L'appellation de cette entité diffère selon les modèles, ce qui n'encourage pas la compréhension de ces principes. Le modèle de référence, proposé par l'ITU-T, nomme cette entité *Resource and Admission Control Function (RACF)*. Sous IMS il se transforme en *Policy Control and Charging Rules Function (PCRF)*. Dans les deux cas, il a pour fonction la prise de décision sur la politique de QoS et la facturation.

Le module de gestion de la qualité de service s'interface avec la couche de gestion qui extrait les requêtes de qualité de service des messages de contrôle échangés avec les terminaux utilisateurs et les serveurs d'applications. Il les transmet ensuite au PCRF, en vue d'une prise de décision sur la demande de réservation. En fonction des règles prédéfinies et des ressources disponibles, la demande est acceptée ou refusée. Si elle est acceptée, les paramètres de QoS sont alors communiqués à la couche transport pour être appliqués sur l'ensemble du réseau. Cette opération s'effectue auprès du *Policy and Charging Enforcement Function (PCEF)*. Ce dernier est une entité logique abstraite qui représente la passerelle d'un réseau d'accès (ADSL, UMTS...). Dans le cas d'un réseau d'accès UMTS, cette entité est nommée *Gateway GPRS Support Node (GGSN)*. Son rôle de passerelle l'oblige à contrôler le trafic aux extrémités du réseau en acceptant ou rejetant les données entrantes. Si un trafic n'est pas autorisé ou reconnu, il est annulé.

4.7 Conclusion

Les futurs réseaux de télécommunications sont axés sur les services numériques. Les communications, fonction basique, sont reléguées au second plan, c'est un acquis. Pour autant, ces réseaux *multimédia* n'ont pas vocation à remplacer immédiatement certains de nos réseaux traditionnels, preuve en est qu'ils s'interfacent avec ces derniers. Il ne s'agit pas non plus de se substituer à Internet. Ces réseaux NGN doivent être perçus comme une évolution et non une rupture. Ils se greffent sur les infrastructures existantes et intègrent les services existants. IMS est actuellement la réalisation la plus aboutie d'un modèle NGN. Il reste cependant de nombreux challenges techniques et scientifiques à surmonter pour proposer un réseau totalement opérationnel aujourd'hui [7], comme par exemple l'intégration complète de l'ensemble des technologies d'accès.

Chapitre 5

Adaptation des applications en informatique

5.1 L'adaptation des applications dans les NGN

La problématique d'adaptation des applications dans les réseaux trouve son origine à la fin des années 90 avec l'apparition des réseaux de données cellulaires. Dans un réseau sans fil, la performance de transport peut subir des variations dues à l'environnement extérieur. Ainsi, au cours d'une transmission, la performance du réseau qui s'exprime sur plusieurs paramètres (bande passante, latence, perte de paquets IP...) est susceptible de changer. Les applications réseau qui s'appuient sur un lien sans fil pour transporter les données subissent ces variations. Les conséquences vont de la dégradation du transfert à la coupure du service. Pour éviter ce désagrément, il faut adapter les applications aux conditions du réseau.

Les travaux liés à la qualité de service dans les réseaux et l'adaptation d'applications sont regroupés sous le terme de *adaptation de la qualité de service* (adaptive QoS). L'article de Li et coll. [51] présente en 1998 une architecture logicielle pour aligner le contenu des applications à la situation du réseau. Ils présentent le principe d'application *flexible*. Ce sont des applications dites *tolérantes* sur la qualité de service : elles ont la capacité de dégrader ou d'améliorer leurs demandes en ressources réseau en fonction de la performance offerte par ce dernier. Ainsi en cas d'impossibilité d'honorer la demande de qualité de service, l'application change ses besoins. Pour tirer profit de telles applications, l'article de Li et coll. propose un modèle théorique basé autour d'une boucle de rétroaction fermée (Figure 5.1). Les modules clés de cette boucle sont :

- *Observation Task* : il observe l'environnement et remonte les informations sur celui-ci en vue d'une comparaison avec une référence.
- *Adaptation Task* : il contient les règles d'adaptation et les applique en fonction de la dérive observée entre un environnement de référence et l'environnement observé.

- *Target Task* : il adapte la tâche en cours d'exécution d'après les actions de contrôle proposées par le module *Adaptation Task*.

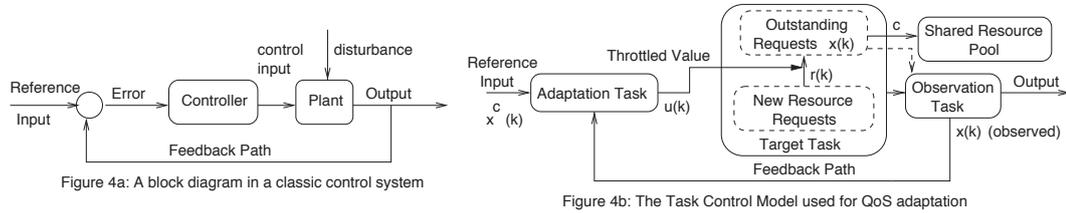


FIGURE 5.1: Modélisation du flux de tâche pour des environnements distribués [51]

Bien que les réseaux aient évolué et soient à même de répondre à la demande croissante des ressources par les applications réseau, nous retrouvons dix ans plus tard une problématique similaire. Elle reste fondamentalement la même : adapter l'application à son environnement variable. Cependant, il ne s'agit plus d'opérer en fonction de la performance du réseau et sur des équipements homogènes, mais dans sur des équipements hétérogènes dans un environnement avec une forte variabilité.

Plusieurs travaux scientifiques traitent de cette problématique dans le cadre des réseaux du futur. Pour synthétiser ces travaux nous nous appuyons sur la théorie du contrôle appliquée aux logiciels informatiques sur une partie du modèle proposé par Kokar et coll. [43]. Son modèle de contrôle pour les logiciels combine (Figure 5.2) :

- Une boucle de rétroaction,
- Une boucle d'adaptation,
- Une boucle de reconfiguration.

La *boucle de rétroaction* offre un contrôle direct du logiciel par une comparaison entre un objectif et la performance de réalisation de celui-ci. La mesure de performance est réalisée par un sous-module. Le résultat est injecté dans le module de contrôle pour une comparaison avec l'objectif initial. Le module de contrôle corrige, si nécessaire, le comportement du service numérique pour compenser la dérive entre la mesure et l'objectif de performance. Ce modèle considère que les paramètres extérieurs à la mesure de l'objectif n'influent pas sur sa réalisation.

Pour pallier ces suppositions, deux autres modèles de rétroactions sont proposés, la *boucle d'adaptation* et la *boucle de reconfiguration*. Le premier modèle enrichit celui de la boucle de rétroaction en proposant deux sous-modules qui permettent d'évaluer les paramètres du contrôleur et de choisir un modèle de comparaison adapté à la situation du logiciel. Le second propose une fonction de reconfiguration de l'ensemble des modules pour adapter le logiciel à sa nouvelle situation.

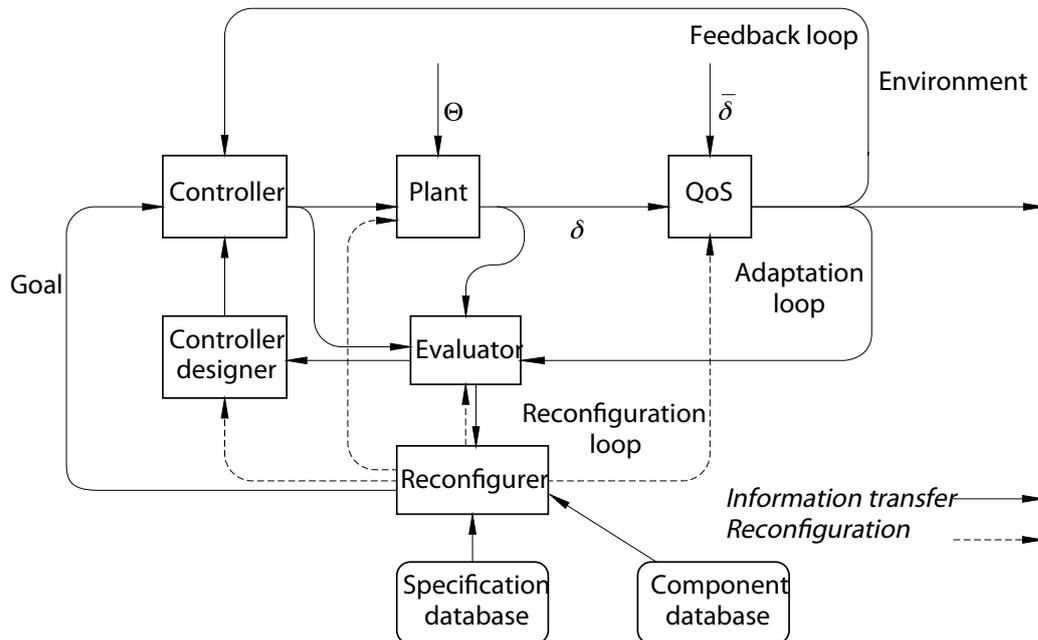


FIGURE 5.2: Modèle pour des applications auto-adaptables [43]

Nous utiliserons le principe de *boucle de rétroaction* présenté par Kokar ou Lee pour comparer les différentes approches d'adaptation des services numériques dans les réseaux du futur.

Dans la suite de ce chapitre nous comparons les travaux scientifiques cités ci-dessous, qui sont relatifs à la problématique d'adaptation des applications dans les réseaux du futur :

- L'adaptation des flux vidéos dans IMS (Park et coll.) [61]
- L'adaptation des flux non temps réel dans un NGN (Balakrishna et coll.) [4]
- L'évaluation de la QoS depuis le terminal utilisateur (Kim et coll.) [41]
- L'évaluation globale de la performance d'un échange depuis le terminal utilisateur en temps réel (Oczelebi et coll.) [60]
- L'adaptation d'un service de réalité virtuelle dans IMS (Skorin-Kpaov et coll.) [71] [69] [70] [25]
- La perception de la qualité de service et de l'expérience utilisateur dans IMS (Koumaras et coll.) [46] [47] [48] [11]
- La gestion des réseaux locaux pour une qualité de service de bout en bout dans IMS (Vidal et coll.) [78] [75] [76] [77]

Nous comparons les principes d'adaptations présentés dans ces travaux en s'appuyant sur une boucle de rétroaction telle que présenté par Kokar (Figure 5.2) ou Lee (Fi-

gure 5.1) sous trois aspects :

- L'évaluation de la performance : fonctions de collecte et d'agrégation des données sur l'environnement.
- Le contrôle : fonctions de comparaison entre l'objectif de performance attendu par le logiciel et sa performance actuelle.
- L'adaptation.

5.1.1 Mesure de la performance

La littérature comporte des travaux spécifiques à l'évaluation de la performance pour les services numériques ou la qualité de service de bout en bout dans les réseaux du futur. Ils font référence au module QoS dans les travaux de Kokar et coll. [43] ou *Observation Task* dans les travaux de Li et coll. [51]

L'article [41] de Kim et coll. présente une méthode d'évaluation de la performance de la consommation des services numériques au sein des réseaux. Ils proposent de réaliser les mesures sur le terminal de l'utilisateur en vue d'une comparaison avec la performance énoncée dans le contrat de service (SLA - Service Level Agreement¹) qui lie un utilisateur avec son opérateur. Habituellement ces mesures sont réalisées depuis le réseau de l'opérateur par lui-même.

Kim et coll. réalise ces mesures sur le terminal à l'aide d'un agent logiciel (Figure 5.3) composé de trois blocs :

- *Quality measurement processor* : capture et évalue la performance.
- *Measurement Result Processor* : transmet les données d'évaluation sur un serveur tiers (*Measurement quality server*).
- *Configuration Processor* : configure le système de capture.

L'agent logiciel capture un ensemble de paramètres tels que le *jitter*, *R-Value*², la *latence*, la *perte de paquets*, le *nombre de sessions lancés*, etc. Ces paramètres sont soit extraits des paquets IP (pour les protocoles ICMP, SIP, RTP, RTCP), soit calculés.

Cependant l'évaluation de la performance n'est pas transmise au fil de l'eau, mais uniquement à la fin de la session. Le niveau de détail est limité aux informations sur la moyenne, le minimum et le maximum de chacun des paramètres cités ci-dessus. Cette limitation s'explique par l'objectif même des travaux de Kim et coll. qui ne concernent que le principe d'évaluation d'une performance depuis le terminal utilisateur. Ils démontrent qu'une mesure depuis le terminal de l'utilisateur est aussi fiable qu'une mesure avec des

1. Le Service Level Agreement (SLA) est un document qui définit la qualité de service requise entre un prestataire et un client.

2. Détermine la qualité d'un réseau VoIP à partir d'une équation se basant sur un ensemble de paramètres mesurable (*jitter*, *perte de paquets*, ...)

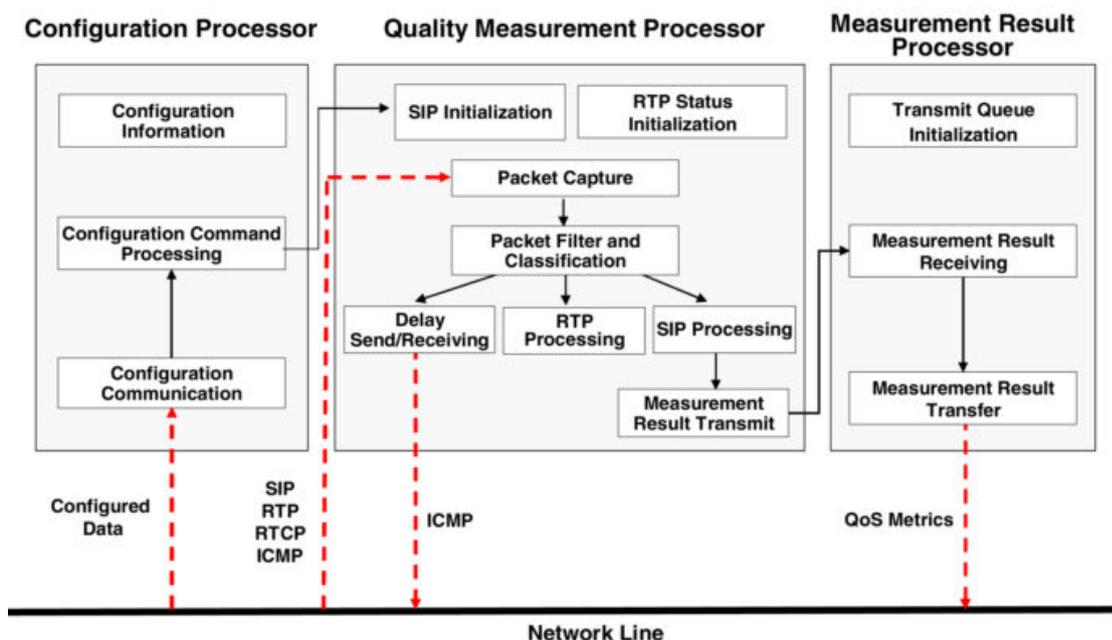


FIGURE 5.3: Agent logiciel pour l'évaluation de la performance d'une session multimédia [41]

outils dédiés au sein du réseau, pour l'évaluation de la performance d'un échange. Cette conclusion ne s'applique pas pour l'évaluation de la performance globale du réseau.

Les travaux de Kim et coll. s'intéressent uniquement à la performance du réseau dans les NGN. Ce n'est pas suffisant dans un objectif d'adaptation dans les réseaux du futur puisque les utilisateurs ont aussi la possibilité de changer de terminal. L'article d'Öçzelebi et coll. propose une approche similaire avec un agent logiciel qui effectue ses mesures de performance depuis le terminal client. Cependant il ne s'agit plus d'évaluer uniquement la performance du réseau, mais aussi celle des terminaux locaux et distants. Cette approche est particulièrement intéressante dans le cadre d'un échange entre terminaux. Pour évaluer cette performance globale de l'échange, chaque terminal concerné par un même échange :

1. Collecte des données sur le terminal et le réseau (appelés ressources locales).
2. Échange ses informations avec les autres terminaux engagés dans la communication.
3. Adapte les flux en fonction de la performance globale.

Les deux premières étapes concernent la mesure de la performance et sont réalisées par le module appelé *Resource Manager (RM)* dans l'architecture proposée par Öçzelebi

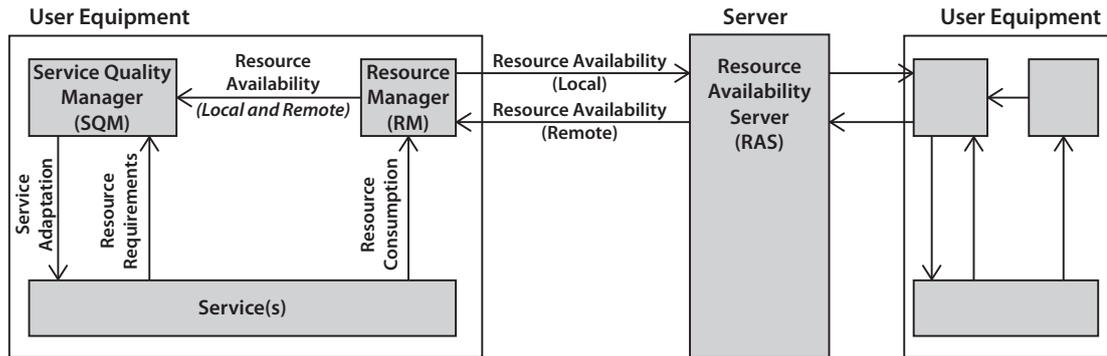


FIGURE 5.4: Architecture de signalisation QoS de bout en bout et adaptation des services [60]

(Figure 5.4). Ce module mesure localement la disponibilité de ressources et récupère les informations concernant la disponibilité des ressources distantes. Les informations distantes sont échangées en permanence. C'est un serveur tiers, le *Resource Availability Server (RAS)*, qui joue le rôle de relais entre les terminaux. Il utilise les messages SIP de notification d'évènements (RFC 4662) pour échanger les évaluations de performance entre les terminaux.

Un article [69] de Skorin-Kapov et coll. propose un principe similaire pour la mesure de la performance et la signalisation de bout en bout. Ils arguent que les standards définis pour la négociation ne sont pas adaptés aux futurs cas d'utilisations. En effet, pour que la négociation soit complète, il faut prendre en compte :

1. Les contraintes liées à l'environnement d'exécution (terminal et réseau) au début de la communication
2. Les préférences de l'utilisateur en terme priorité dans les composants de l'application.
3. La variabilité des contraintes de l'environnement.
4. Le compromis entre les préférences de l'utilisateur et les contraintes de l'application.

Ces observations mènent à l'élaboration d'un modèle de négociation de QoS réparti sur trois entités : le client, le système de contrôle d'accès et le serveur d'application. Il est présenté dans [69] sous la forme d'une architecture répartie (Figure 5.5). L'évaluation de la performance est réalisée sur le terminal de l'utilisateur par un agent logiciel propre à cette architecture [71].

Les travaux de Grgic et coll. viennent compléter la proposition de Skorin-Kapov, en s'intéressant à la couche transport [25]. Skorin-Kapov propose un modèle pour adapter

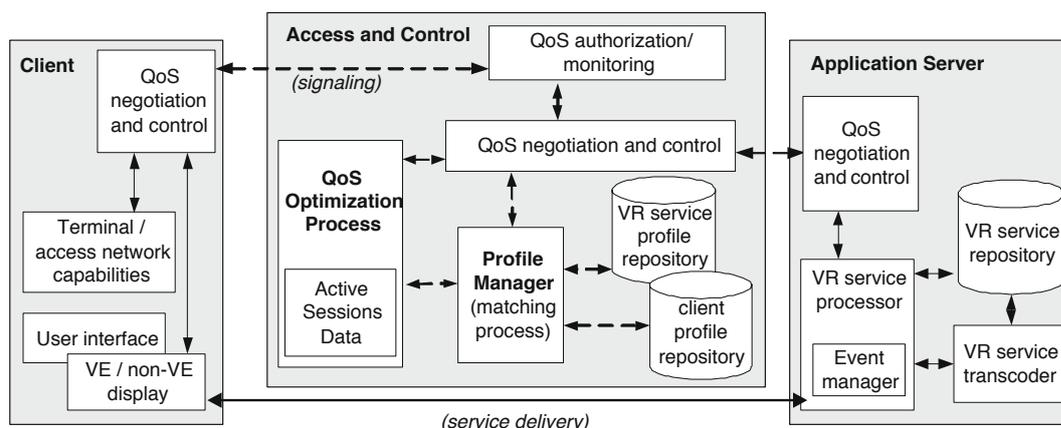


FIGURE 5.5: Modèle de négociation dynamique de la qualité de service et d'adaptation de la QoS [69]

les services numériques à leur environnement en fonction d'un ensemble de paramètres relatifs à la couche service. Grgic propose d'adapter ces services en effectuant les mesures au niveau de la couche transport. L'adaptation par la couche transport, approche *bottom-up*, permet de piloter le service numérique à partir des événements capturés depuis la couche réseau.

Les travaux de Park et coll. [61] s'intéressent aux événements des couches *basses*. La performance du réseau est déduite à partir d'informations extraites de la couche radio. C'est-à-dire la couche physique du modèle OSI. Pour rappel, le modèle OSI est un modèle abstrait utilisé pour représenter un système de communication. Il est composé de 7 couches indépendantes. Chaque couche fournit un service spécifique (transport point à point, transport au sein d'un réseau, contrôle de session, etc.). Le système de diffusion vidéo, qui est une application (niveau 7 du modèle OSI) utilise des informations de la couche physique (niveau 1 du modèle OSI). Cette approche casse l'indépendance des couches prônée par ce même modèle. En utilisant du *cross-layering*, Park rend sa proposition dépendante d'une technologie d'accès et donc un cas d'usage limité à certaines technologies sans fil.

Les travaux de Koumaras et coll. [47], dans le cadre du projet ADAMANTIUM, vont aussi dans le sens d'une capture des *couches basses* du réseau. La proposition s'articule autour d'un système de gestion de contenu (*MCMS - Multimedia Content Management System*) qui est composé de deux blocs, un premier pour la surveillance de l'environnement (monitoring) et un second pour l'adaptation (Figure 5.6).

Le *monitoring* est réalisé à l'aide de trois modules :

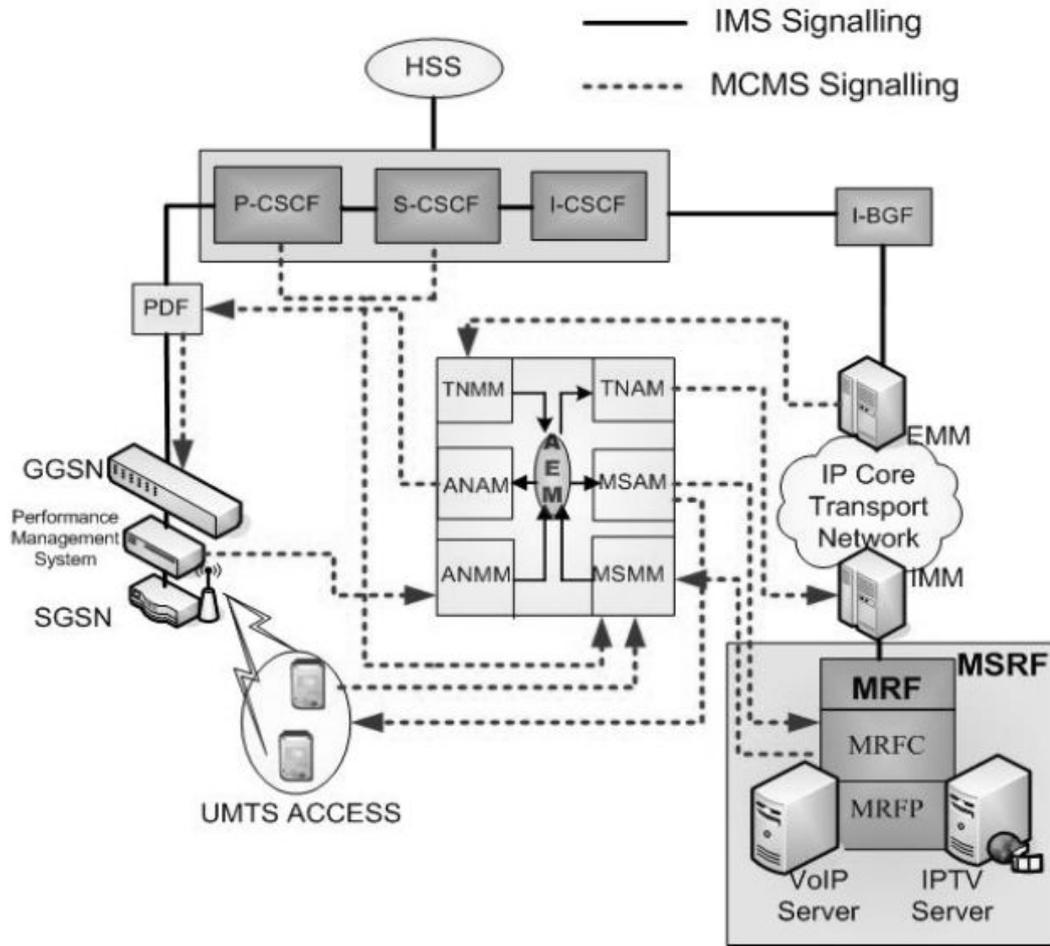


FIGURE 5.6: MCMS pour IMS [47]

- Multimedia Service Monitoring Module (MSMM) : extrait les informations de la couche de signalisation IMS.
- Transport Network Monitoring Module (TNMM) : surveille les caractéristiques du réseau de transport (perte de paquets, latence, jigue...).
- Access Network Monitoring Module (ANMM) : s'appuie sur les noeuds du réseau pour extraire des informations sur le lien réseau.

Cette approche combine une capture des informations présentes dans les couches basses et les informations de signalisation. Ceci permet de prendre en compte la performance propre du réseau ainsi que les données sur le service numérique, par exemple la qualité de service négociée entre les entités concernées par l'échange.

Nous retrouvons cette idée d'extraction de données de la couche de signalisation

dans la proposition de Balakrishna et coll. sur l'adaptation des flux audio et vidéo non temps réel. Ils s'appuient sur une capture de la signalisation SIP dans IMS [4] pour détecter les changements d'environnement. Cette capture est effectuée par un serveur mandataire installé entre le terminal de l'utilisateur et le serveur de média. Il négocie en lieu et place du serveur pour le choix du format (codec, bitrate, etc.). Cette dernière approche n'est pas une capture, mais plutôt une interception et retranscription du trafic de signalisation.

Pour conclure, nous retrouvons deux approches pour l'évaluation de la performance dans les NGN en vue d'une adaptation. La première s'appuie sur la capture du trafic au sein du réseau. C'est l'approche retenue dans [61], [25], [47] et [4]. Elle est non intrusive puisqu'elle ne nécessite pas l'installation d'un agent logiciel tiers sur les terminaux des utilisateurs. Elle permet de capturer les événements de changement de technologie d'accès et de terminal utilisateur ainsi que l'évaluation de la performance du réseau. Cependant, la performance du terminal utilisateur n'est pas prise en compte.

La seconde approche qui est retenue dans [41], [60], [69] met en oeuvre un agent logiciel sur les terminaux utilisateurs. Il propose une évaluation de la performance au plus proche de l'utilisateur et qui concerne le couple réseau-terminal utilisateur. Cependant, l'approche est intrusive et nécessite d'avoir un agent logiciel pour chaque type de terminal.

5.1.2 Décision d'adaptation

Cette section traite de la décision d'adaptation et du choix des adaptations.

Dans l'article d'Özçelebi et coll. les fonctions de décision d'adaptation sont confiées à un agent local dénommé *Service Quality Manager* (Figure 5.4). Il reçoit des applications exécutées sur le terminal les demandes en terme de ressources. Il propose des adaptations en fonction de la performance globale et des besoins de chaque application. Nous manquons de données sur le mécanisme de décision d'adaptation.

Dans l'architecture proposée par Skorin-Kapov et coll. (Figure 5.5), l'entité de décision est "Access and Control". Elle s'appuie sur des bases de profils contenant des données relatives aux utilisateurs et aux services pour proposer l'adaptation de l'application :

- *client profile repository* : la base contient les profils des utilisateurs. Chaque profil relate les caractéristiques des terminaux possédés par l'utilisateur, les réseaux à sa disposition, la liste des logiciels présents ainsi que les préférences de l'utilisateur en terme de flux (acceptés ou non, qualité de chacun).
- *VR service profile repository* : la base contient les profils des services numériques de réalité virtuelle. On retrouve, pour chaque profil, les processus d'adaptation et les contraintes de l'application.

Les données contenues dans ces bases de profils sont à la base de la décision d'opti-

misation de la qualité de service. Ce processus d'optimisation est opéré par le module QoS Optimization Process : pour chaque session, il propose un compromis entre les contraintes de l'utilisateur, de l'application et les préférences de l'utilisateur. L'algorithme utilisé pour le processus d'optimisation de la qualité de service (QOP) a fait l'objet d'une publication à part entière [70].

Grgic et coll. [25] proposent d'adapter les applications en fonction d'un processus de dégradation. Le service numérique propose une configuration optimale et plusieurs configurations alternatives. Elles sont enregistrées dans une *table* nommée **Media Degradation Path (MDP)** (Figure 5.7). Cette table présente les différents modes opératoires du service : du plus optimal au plus dégradé. Chaque mode présente les différents flux qui composent la session multimédia avec les paramètres d'applications (codec, résolution...) et les ressources nécessaires (bande passante, jitter...).

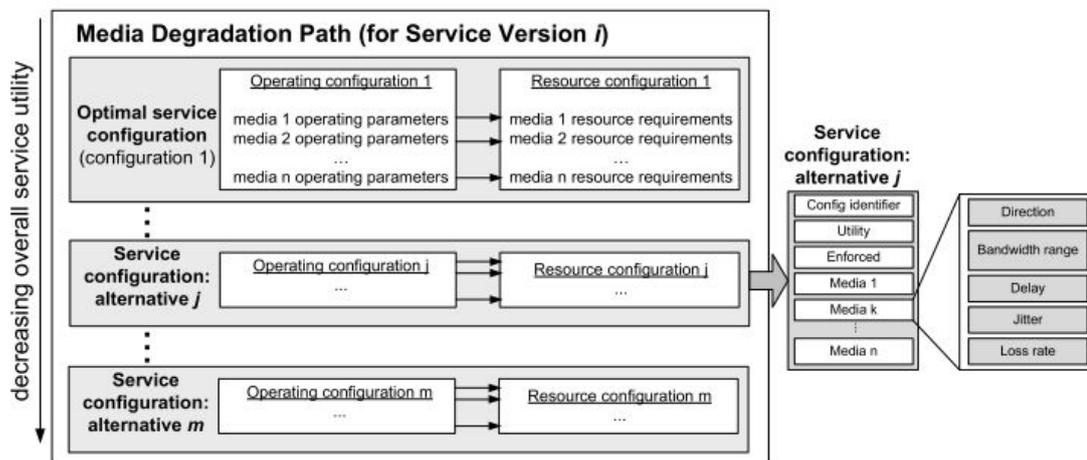


FIGURE 5.7: Table MDP pour un service [25]

À l'établissement d'une session, un croisement est effectué entre les paramètres de QoS (préférences utilisateurs, contraintes du service, contraintes des terminaux...) et le chemin de dégradation de l'application. Le résultat de cette opération est un chemin de dégradation personnalisé. Il est propre à chaque utilisateur puisque relatif aux situations qu'il est susceptible d'expérimenter. Ce chemin de dégradation personnalisé est communiqué aux fonctions de contrôles d'accès et de facturations (entité PCRF d'IMS). Ainsi en fonction de la situation de l'utilisateur, déduite grâce aux événements de la couche de transport, le PCRF élit la configuration la plus adaptée.

Il la transmet ensuite aux fonctions de réservation et d'application de la QoS, sous la forme de règles, pour une application dans le réseau de transport. Cette approche nécessite des modifications du modèle IMS proposé par les organismes de standardisation.

Dans la même lignée, nous retrouvons les travaux de Koumaras et coll. au sein du projet européen ADAMANTIUM. L'adaptation est réalisée par les trois blocs au sein du

MCMS (Figure 5.6) :

- Multimedia Service Adaptation Module (MSAM) : apporte des adaptations du point de vue du service, sur le terminal (codec, buffers, résolution . . .)
- Transport Network Adaptation Module (TNAM) : l'adaptation porte sur la qualité de service au sein du réseau de transport.
- Access Network Adaptation Module (ANAM) : modifie les politiques d'accès des passerelles d'accès.

Cette approche peut être qualifiée d'inter-couches puisque les services numériques sont adaptés sur plusieurs niveaux :

- Service,
- Réseau de transport,
- Réseau d'accès.

Dans les propositions de Park et coll. [61], Balakrishna et coll. [4] les mécanismes de décision sur l'adaptation ne sont pas présentés. De manière générale, les mécanismes de décisions d'adaptation sont très peu détaillés dans la grande majorité des propositions scientifiques.

5.1.3 Adaptation du service numérique

Cette section traite de l'application de l'adaptation sur les services numériques. Elle concerne les travaux de Park et coll., Balakrishna et coll. et Skorin-Kapov et coll. puisque ce sont les seuls travaux à présenter cet aspect. Leurs travaux sont spécifiques pour certains services numériques, ce qui leur permettent de présenter la réalisation de l'adaptation.

Les travaux de **Park et coll.** proposent de changer l'encodage des vidéos envoyées aux terminaux en fonction de la couverture radio. En effet, chaque cellule dispose d'une zone de rayonnement dont le centre est l'antenne (appelée *station de base*) utilisée pour transmettre les informations. Elle transmet les données aux terminaux présents dans sa zone d'influence (cellule). Cependant, la performance de transport offerte au sein de la cellule n'est pas uniforme : elle décroît de façon non proportionnelle à l'approche de ses bordures. Le type d'encodage de la vidéo est adapté en fonction des observations suivantes :

- Un codage de type non-SVC³ est très adapté au sein de la cellule mais pas du tout à ses bordures.
- Aux extrémités de cette dernière, c'est un codage de type SVC qui est le plus adapté.

3. Scalable Video Coding

La méthode qu'ils proposent met à profit les mécanismes déjà présents dans IMS pour proposer un codage dynamique de la vidéo :

1. Le support des flux multiples au sein d'une session : une session est composée d'au moins un flux (de type vidéo, audio ou données).
2. La modification de session au cours de son existence : sa modification correspond à une intervention sur les flux qu'elle encadre (ajout/modification/suppression).

Chaque session du service de diffusion vidéo est composée de deux flux vidéo : l'un utilise un codage de type SVC et l'autre utilise un codage de type non-SVC. Pour limiter l'encombrement de la bande passante, un seul des deux flux est envoyé sur le terminal de l'utilisateur. Nous le qualifierons de flux principal. Le flux secondaire est diffusé par le service, mais il n'est pas transporté par le réseau. Pour le terminal, ce flux alternatif est mis en attente. Le service numérique alterne le transport des flux en fonction de la situation de l'utilisateur. Ainsi, si l'utilisateur est proche de l'antenne, le service numérique diffuse le flux vidéo avec un encodage non-SVC. Si l'utilisateur est proche des bordures de la cellule, le service numérique diffuse le flux vidéo avec un encodage SVC.

D'un point de vue technique, la mise en attente d'un flux passe par la définition d'une demande en bande passante (QoS) de 0 kbit/s. Autrement dit, un flux diffusé a une demande de QoS supérieure à 0 kbit/s. Un flux qui n'est pas diffusé a une demande QoS égale à 0 kbit/s. Le service numérique change la demande en QoS des flux pour les diffuser ou non vers le terminal de l'utilisateur. Cette décision de diffusion est prise en fonction de la performance du réseau.

L'article de Balakrishna et coll. propose d'adapter les flux par un mécanisme de renégociation. Aucun mécanisme de renégociation n'est proposé (actuellement) dans l'architecture d'IMS. Plutôt que de mettre en place ce mécanisme au niveau de chaque service numérique, l'approche de Balakrishna est de généraliser ce mécanisme au sein du réseau. Pour cela, un serveur mandataire (proxy) prend en charge la négociation et l'adaptation du flux vidéo. Ainsi, pour chaque session multimédia, le mandataire joue son rôle d'intermédiaire entre le terminal et le serveur multimédia. Il intercepte et retransmet le trafic de signalisation (requêtes SIP) ainsi que le trafic multimédia (flux RTP).

Le framework est composé du couple serveur SIP (entité B2BUA⁴) et d'un serveur multimédia (Figure 5.8). Ces deux éléments composent le serveur mandataire et agissent de pair pour être perçus par le terminal utilisateur comme la source de diffusion.

Lors de l'initialisation d'une session, le terminal requête l'entité SIP pour une ouverture de session multimédia. Le serveur mandataire, en se faisant passer pour l'utilisateur, demande à son tour l'ouverture d'une session multimédia auprès du serveur d'applica-

4. Back-to-Back User Agent

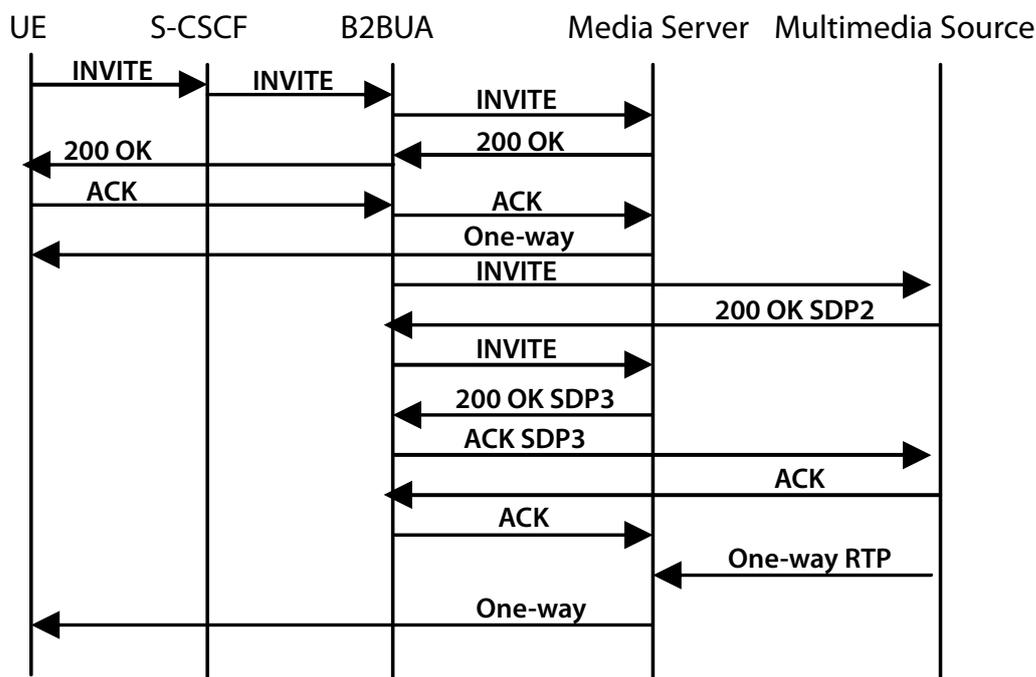


FIGURE 5.8: Établissement d'une session multimédia avec le B2BUA [4]

tions. Les flux multimédia, en provenance du serveur d'application, sont reçus par le serveur mandataire, transcodés et renvoyés à l'utilisateur.

Lorsque l'utilisateur change de réseau d'accès, le terminal notifie l'entité SIP qui compose le serveur mandataire en indiquant la nouvelle adresse IP et les paramètres de QoS. De son côté, le serveur multimédia du serveur mandataire transcode les flux multimédia et change leur destination.

Le changement de terminal fonctionne de façon analogue en ajoutant une mise en cache du trafic pour une rediffusion. Balakrishna et coll. proposent l'utilisation d'un serveur mandataire pour adapter les flux multimédia. Ces travaux sont parmi les rares qui traitent non seulement la mobilité du terminal, mais aussi la mobilité de session en utilisant un mécanisme de tampon.

L'article [72] de Skorin-Kapov et coll. présente la mise à l'épreuve d'un logiciel de réalité virtuelle autour de plusieurs cas d'utilisation [72] :

- Les ressources disponibles permettent de faire fonctionner l'application sans contrainte,
- Les ressources du réseau sont insuffisantes : l'adaptation est déclenchée par un

événement externe et s'exécute automatiquement ; l'application sélectionne les flux en fonction des contraintes et préférences de l'utilisateur.

- L'utilisateur souhaite dégrader le mode de fonctionnement de son application : l'adaptation est déclenchée par l'utilisateur.

L'adaptation des services numériques est présentée uniquement sur certains travaux : ceux qui proposent une application spécifique.

5.1.4 Conclusion

La problématique de l'adaptation des services numériques au sein des réseaux du futur ne cesse d'évoluer. C'est d'abord sous l'aspect de la qualité de service de bout en bout qu'elle a été traitée [78]. C'est ensuite l'évaluation de performance qui a fait l'objet de travaux scientifiques, dans un premier temps, seul le réseau a été pris en compte. Ensuite l'ensemble des entités qui participent à l'échange des données avec les services numériques ont été intégrés, à savoir : les terminaux utilisateurs et la capacité du réseau. Deux approches co-existent pour l'évaluation de la performance.

La première approche s'appuie sur les conclusions de Kim et coll. qui prouvent qu'une mesure depuis les terminaux utilisateurs est très fiable pour évaluer la performance d'un échange client-serveur [41]. Ces travaux ouvrent la voie à ceux de Özzelebi et coll. [60] et Skorin-Kappov et coll. [69]. La contrainte en terme de réalisation dans un réseau du futur reste très forte puisqu'il s'agit de pouvoir proposer un agent logiciel de mesure sur l'ensemble des terminaux susceptible d'être utilisée dans les NGN.

La seconde approche prône l'évaluation à partir de captures du trafic. Elles concernent la signalisation (changement de terminal) et les couches basses qui reflètent l'activité du média de transport (changement de technologie d'accès). Cela concerne les travaux de Koumaras et coll. [46] réalisés dans le cadre du projet européen ADAMANTIUM ou bien la variante des travaux de Skorin-Kapov proposée par Grgic et coll. [25]. Cette approche nécessite l'intégration d'entités de capture et donc la modification des architectures des NGN proposées par les organismes de standardisation.

La collecte de données dans les NGN fait encore l'objet de travaux scientifiques. C'est une problématique à part entière dans les travaux d'adaptation dans les NGN.

Les mécanismes de décision d'adaptation sont très peu discutés dans le thème des adaptations des applications dans les NGN. Les seuls travaux que nous avons recensés qui présentent en détail les mécanismes de décisions sont [70], [25]. Pour un aperçu plus complet des mécanismes de décisions, nous nous orientons vers les systèmes sensibles au contexte qui sont présentés dans la section 5.2.

Les mécanismes d'adaptation sont eux aussi très peu discutés pour des raisons de spécificité. En effet ils sont propres à chaque service numérique et dépendent en général du choix du concepteur de ce même service. Cette observation met en avant le fait qu'un

service numérique sensible au contexte est un logiciel avec deux logiques, une logique de sensibilité au contexte et une logique métier.

TABLEAU 5.1: Comparaison des approches pour l'adaptation des services numériques dans les réseaux du futur

Auteur	Résumé	Architecture	Apport	Remarque(s)
Park	Adaptation à la volée des vidéos dans IMS sur la partie réseau mobile	Diffusion vidéo sans architecture sous-jacente.	Intégration transparente dans IMS avec un système de commutations de flux au sein de la même session.	Ne concerne que la diffusion de flux vidéo et dépend de la technologie de transport.
Balakrishna	Adaptation à la volée des flux audio et vidéo dans IMS.	Utilisation d'un serveur mandataire pour transcoder les flux multimédia.	Prend en compte le concept de mobilité généralisée.	Ne concerne que les contenus multimédia non-temps réels.
Kim	Évaluation de la QoS depuis le terminal utilisateur.	Agents logiciels sur le terminal utilisateur interconnectés par un serveur tiers.	Mise en avant de la similitude entre une mesure depuis le terminal utilisateur et au sein du réseau.	Système de collecte des données sur l'environnement informatique.
Özgelebi	Adaptation des contenus lors d'une session entre deux terminaux.	Intergiciel	Adaptation d'applications sur un mode pair à pair	Centré sur la collecte et l'échange des données collectées et très peu sur l'adaptation.
Skorin-Kapov	Adaptation des contenus des applications de réalité virtuelle.	Architecture distribuée entre les entités qui composent un réseau.	Prise en compte des caractéristiques techniques du terminal, du réseau et des préférences de l'utilisateur.	L'intégration dans un réseau IMS nécessite la modification d'une partie des procédures et entités.
Koumaras	Optimisation des contenus multimédia en fonction de la perception de l'utilisateur.	Serveur d'applications mandataire	Approche orientée vers l'utilisateur. Adaptation sur plusieurs niveaux.	Projet parallèle aux standardisations d'IMS.
Vidal	Extension la gestion de la QoS au réseau local d'un abonné.	Serveur mandataire au sein du boîtier opérateur.	Qualité de service de bout en bout sur les réseaux haut débit filaires.	Ces travaux ne font pas référence à l'adaptation d'applications.

5.2 Les applications sensibles au contexte

Dans ce chapitre, nous élargissons notre état de l'art autour de la problématique d'adaptation des applications dans les réseaux du futur en nous intéressant de façon globale aux applications qui s'adaptent à leur environnement. De telles applications sont dites sensibles au contexte. Les travaux se référant à ce domaine sont définis par les mots-clés anglais suivants : context-awareness, context-aware applications, context-aware systems.

La première section discute brièvement de la nécessité de définir le contexte.

La seconde section décortique les architectures des applications sensibles. Nous terminons par une troisième section qui présente succinctement les applications autonomiques et leur lien avec les applications sensibles au contexte.

5.2.1 Définition du contexte

Tous les travaux scientifiques qui présentent des applications sensibles au contexte caractérisent leur environnement, c'est-à-dire les éléments qui composent son contexte. Cette notion de contexte a fait l'objet de travaux scientifiques. Pour mieux comprendre ce qu'est le contexte nous nous intéressons d'abord à sa notion littéraire puis à sa notion scientifique dans des applications informatiques.

Une définition générale du contexte, issue du dictionnaire *Antidote*, présente d'abord le contexte comme un élément linguistique :

“ Texte dans lequel s'insère une unité linguistique (mot, phrase, etc.) et dont elle tire sa signification ou sa valeur. ”

Ainsi, le contexte apporte un sens plus juste à une unité linguistique. Il permet d'aller au-delà de l'analyse sémantique et débusquer les faux-amis. Par exemple, la phrase “Dix vols par jour” [42] peut avoir deux sens. Seul le texte qui entoure cette phrase peut expliquer son sens. Le contexte permet de préciser ou déduire le sens d'une phrase.

Cette approche du contexte, spécialisée dans le domaine linguistique, nous donne une première idée de ce qu'est le contexte. Dans le domaine littéraire, le contexte permet de donner du sens pour comprendre le texte. Une définition plus globale (non spécialisée) vient compléter la définition présentée ci-dessus :

“ Ensemble des circonstances dans lesquelles s'insère un fait, un évènement. ”

Le contexte (au sens général) désigne un élément *vague* qui entoure un texte, un évènement, etc. Tout est contexte, à partir du moment où cela est relatif à un fait ou un évènements précis. Le sujet et son contexte sont évidemment liés. Puisqu'un fait ou un évènement peut être expliqué par son contexte, il est possible de reconnaître le

contexte pour supposer l'événement. C'est un raisonnement que nous tenons dans notre vie quotidienne. Par exemple, quand il pleut le ciel est gris. Ainsi voir le ciel gris nous indique qu'il est possible qu'il pleuve.

Les publications scientifiques qui traitent du contexte commencent donc par présenter leur contexte, qui est assimilé à une définition. Cette pratique débouche sur une large diversité de définitions, comme le souligne [6]. Chaque domaine (ou application) définit le contexte en fonction de son but. Pour identifier le contexte, il faut se rapporter au sujet de l'étude. Ensuite, déceler les éléments caractéristiques qui composent le contexte. Ce travail est réalisé de façon empirique.

Toutes les définitions du contexte s'articulent autour d'un patron commun. Ces similitudes motivent les travaux scientifiques sur la définition du contexte. Ils souhaitent répondre de manière globale à la question "Qu'est-ce que le contexte" ([68], [21], [14], [6], [82]). Ces travaux n'ont pas pour objectif d'imposer une définition universelle du contexte, mais de faire ressortir un substrat commun qui offre des pistes de réflexion pour définir *son contexte*. L'enrichissement apporté par les travaux cités ci-dessus permet de travailler sur un contexte de plus en plus précis.

La définition proposée par Dey, bien que critiquée par certains auteurs sur l'oubli de la composante temporelle, reste une référence.

“ Le contexte est n'importe quelle information qui caractérise la situation d'une entité. Une entité est une personne, lieu géographique ou un objet considéré comme pertinent par rapport à une interaction entre un utilisateur et une application. Ceci inclut les utilisateurs et les applications eux-même. [21] ”

Son aspect généraliste permet de l'enrichir avec d'autres définitions ou bien avec ses propres observations. L'article [82] de Zimmermann et coll. rappelle que la définition du contexte passe par l'observation des situations d'usage.

Parmi les spécialisations du contexte, une nous intéresse plus particulièrement.

Les travaux de Schilit font état d'une classification des informations de contexte [68] en informatique, sous la notion de *context-aware computing*. :

- computing context (contexte informatique) : bande passante, ressources du terminal. . .
- user context (contexte utilisateur) : profil utilisateur, position géographique. . .
- physical context (contexte réel) : bruit, éclairage. . .

Notre objectif d'adapter le service numérique à son environnement d'exécution centre la démarche autour du service numérique et non autour l'utilisateur. Bien qu'il n'existe pas de certitude sur la pertinence des éléments de contexte dans le cadre de services nu-

mériques du futur, c'est le contexte informatique qui est retenu comme la seule catégorie d'éléments pertinente à prendre en compte.

5.2.2 Context-aware systems

Plusieurs auteurs [14], [5], [34] ont comparé les systèmes sensibles au contexte. Ces états de l'art permettent de comprendre les problématiques du domaine et d'en dégager les tendances générales du domaine, en particulier sur les aspects conception. Dans cette section, nous commençons par une introduction à l'informatique pervasive et ubiquitaire, clé de voute de beaucoup de travaux sur la sensibilité au contexte. Ensuite nous présentons différentes architectures abstraites afin d'en dégager les bonnes pratiques, les fonctions essentielles et les technologies utilisées.

Les applications sensibles au contexte, une nouvelle vision de l'informatique

La majorité des applications sensibles au contexte font écho aux travaux de Mark Weiser, dont l'idée est d'utiliser l'informatique pour accompagner les humains dans leur vie quotidienne. Il définit les fondements de ce qui sera par la suite appelé les technologies calmes [80] :

- L'ordinateur doit aider les personnes dans leurs activités quotidiennes,
- Il doit être invisible,
- La technologie ne devrait pas nous opprimer.

Des applications aux systèmes sensibles au contexte

Les applications sensibles au contexte sont composées de deux logiques, une logique métier (l'application) et une logique de gestion du contexte (sensibilité au contexte). Les fonctions de gestion du contexte concernent :

- La capture de l'environnement,
- Rapatriement des données issues des capteurs,
- Pré-traitement des données capturées,
- Stockage/gestion des données sur l'environnement,
- L'échange des informations sur le contexte avec l'application.

Nous observons, dans la littérature, une externalisation des fonctions de gestion de contexte sous la forme de bibliothèques, APIs ou intergiciels. Plusieurs travaux scientifiques se proposent de comparer les différentes approches pour l'externalisation des fonctions de gestion du contexte [14], [5], [5]. Ils s'appuient sur une architecture abstraite réalisée autour d'un modèle en couche où chaque couche représente une des fonctions de gestion du contexte citées ci-dessus. L'idée du modèle abstrait en couches est retrouvée dans la proposition de Coutaz et coll. [17] pour l'émergence de système sensible au contexte.

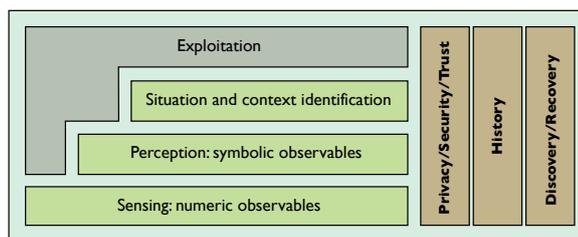


FIGURE 5.9: Architecture abstraite pour des applications sensibles au contexte [17]

Les fonctions transverses de Coutaz et coll. font référence à l'externalisation de la sensibilité au contexte et mettent en avant les problèmes d'accès et de confiance sur les données collectées. Ces préoccupations sont reprises dans l'analyse proposée par Ferry et coll. [22] sur les architectures de systèmes sensibles au contexte. Il présente des fonctions telles que l'apprentissage et la prise en compte des préférences utilisateurs. Les fonctions indispensables des systèmes sensibles au contexte sont :

1. La capture des données de l'environnement,
2. La modélisation du contexte et l'agrégation des données capturées dans ce modèle,
3. La compréhension du modèle et la capacité à informer les applications.

La section suivante passe en revue les outils et techniques utilisés pour chaque fonction d'un système sensible au contexte (capture, modélisation et exploitation du contexte) en s'appuyant sur les travaux de Baldauf [5] et Hong [34] puisque ce sont les plus récents dans le domaine.

5.2.3 Les fonctions clés d'un système sensible au contexte

Le traitement du contexte est l'opération qui transforme des données brutes de capteurs en modalités d'exécution de service numérique. Il met en jeu l'ensemble des composants de l'intergiciel.

Le processus de transformation des données en informations puis en connaissances est traité dans la littérature par Aamodt et coll. [3]. Il a été repris par Zins et coll. [83]. Ce premier définit les données brutes comme des éléments non interprétés issus directement des capteurs. La transformation de ces données brutes en informations est réalisée par un processus d'interprétation. Il enrichit sémantiquement les données, processus réalisé à l'aide des connaissances du système. Un processus analogue transforme les informations en connaissances qui sont intégrées au système. Ce processus est un apprentissage, car les informations sont *acquises* par le système. Ce cycle est illustré dans la Figure 5.10.

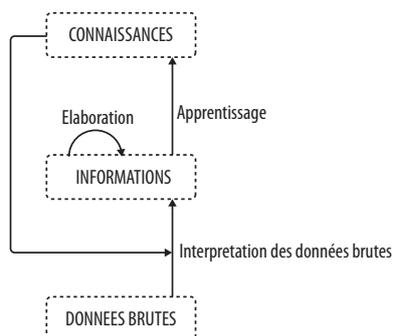


FIGURE 5.10: Processus de transformation des données en connaissances [3]

Capture des données de contexte

La première brique de toute application sensible au contexte comprend la lecture de l'environnement au travers de capteurs. Ils sont classés en deux catégories selon le niveau d'information qu'ils fournissent :

- Les informations de bas niveau, récupérées par des capteurs matériels (orientation, position GPS...).
- Les informations de haut niveau, récupérées par des capteurs logiciels (activité de l'utilisateur et position en fonction de son agenda).

Baldauf étend cette classification à une troisième catégorie : les capteurs logiques (combinaison d'un ou plusieurs capteurs).

La capture des données du contexte ne concerne pas uniquement le choix du capteur. L'objectif est d'acquérir une information sur l'environnement pour pouvoir la confronter à une référence et en déduire le contexte de l'application. Cette couche effectue donc tous les traitements relatifs à l'élément de contexte isolé, par exemple, l'extraction du sens de l'information capturée, l'agrégation des données de différents capteurs, etc. Certains modèles abstraits de systèmes sensibles au contexte proposent explicitement certaines de ces fonctions supplémentaires. Elles existent sous la forme de couches intermédiaires entre la capture des données et le traitement du contexte. L'article de Coutaz et coll. présente le terme anglais *interpretation layer*. Baldauf regroupe ces fonctions sous la dénomination *preprocessing*. Il peut s'agir de corriger une dérive lors de la capture, ou bien transformer une donnée de bas niveau (coordonnées GPS) en une donnée de plus haut niveau (nom de ville).

Pour résumer, la donnée est l'unité de traitement pour la première couche. Elle est capturée puis prétraitée pour être intégrée à un ensemble qui compose le contexte. Les traitements effectués dépendent du format brut de la donnée et des besoins des

applications. Les challenges scientifiques relatifs à cette couche de capture sont encore nombreux, comme l'illustrent les problématiques sur les réseaux de capteurs (autonomie des capteurs, communication des données capturées, etc.)

Modélisation et interprétation du contexte

La couche de capture fournit aux systèmes sensibles au contexte des données qui permettent par la suite de déduire et d'interpréter le contexte. L'analyse de la variation d'une ou plusieurs de ces données permet de détecter un changement de situation (événement ou fait). Pour cela, nous considérons deux étapes : la modélisation du contexte et son interprétation par le système. Les fonctions de ces étapes varient en fonction des problématiques.

La modélisation permet de transformer des informations isolées en contexte lisible et compréhensible par les applications. Pour cela, le système utilise ses connaissances pour transformer les données en informations puis en connaissances, comme le présente [3]. En effet, il s'agit de (re)connaître le contexte à partir d'informations qui composent l'environnement d'utilisation. C'est donc à partir d'une composition des différentes informations (à l'origine des données de capteur) que le système déduit le contexte. Cette étape, présente dans la littérature sous le terme *Modeling Context Information*, s'appuie sur différents outils et techniques, présentés plus loin dans ce chapitre.

Exploitation du contexte par la logique métier

Les deux étapes présentées précédemment permettent d'acquérir les données sur l'environnement puis de les exploiter sous la forme de contexte. La dernière étape consiste à communiquer le contexte à la logique métier (les applications), tout en gardant une séparation entre le coeur de l'application et la gestion du contexte. Pour garder cette séparation, l'architecture est basée sur une externalisation des fonctions de sensibilité du contexte. On peut retrouver ces dernières sous la forme d'un intergiciel. Ainsi il devient possible de mutualiser les fonctions citées précédemment et de les apporter aux applications. C'est en général l'ensemble des fonctions de traitement du contexte qui est externalisé (de la capture l'exploitation du contexte). Parmi les autres approches utilisées, nous trouvons :

- Aucune infrastructure. Un accès direct à l'ensemble de la brique apportant la logique de contexte.
- Serveur de contexte. Les informations de contexte sont accessibles à distance, par une interface réseau. En offrant une architecture client-serveur, le serveur effectue alors toutes les tâches intensives de déduction du contexte, sans déborder, en terme de ressources, sur les applications clientes. C'est une particularité de l'architecture à base d'intergiciel qu'une infrastructure à part entière.

Fonctions transverses

Nous terminons cette présentation des architectures d'applications sensibles au contexte par un aperçu des fonctions transverses. C'est-à-dire les fonctions qui n'ont pas de place définie dans le cycle de traitement du contexte, telles que l'apprentissage et la journalisation qui peuvent apparaître aussi bien au moment de la collecte de données qu'une fois que les données sont interprétées en contexte. Cette décision semble être du ressort du concepteur du système sensible au contexte, en fonction des besoins des applications. L'ajout d'un historique permet d'apporter l'axe temporel au système. Nous pouvons lier, dans certains sens, la notion d'apprentissage à la notion d'historique puisque l'apprentissage est la capacité du système à enrichir, par lui-même ou avec une aide extérieure, sa modélisation du contexte en utilisant l'évaluation des exécutions précédentes.

Ces fonctions transverses ne s'arrêtent pas à celles citées ici (p. ex. le contrôle d'accès). Elles ne sont pas citées dans ce travail qui concerne des systèmes sensibles au contexte traitant avec des machines plutôt que des personnes identifiables. En effet, des systèmes qui identifient les utilisateurs soulèvent le voile de la vie privée et mettent en cause des notions tel que le contrôle d'accès aux informations par les applications ou l'anonymisation.

5.2.4 Les outils de modélisation du contexte

Les outils présentés ci-dessous permettent de modéliser le contexte, de le stocker ou de l'interpréter.

Base clé-valeur

Les informations de contexte sont encodées sous la forme d'une clé (référence) et d'une valeur (information de contexte). Ce type de base est parfois appelé tableau associatif ou dictionnaire. Ce type de modélisation, très efficace pour la recherche, se retrouve de moins en moins utilisé pour le stockage des éléments de contexte dans les systèmes d'externalisation de sensibilité au contexte, car sémantiquement pauvre et source d'erreurs.

Langage de balises

Les langages de balises, qui appartiennent aux langages de description, proposent un enrichissement de l'information textuelle et une structuration. La modélisation du contexte sous cette forme offre un système de description hiérarchique et structurant des données. Il est dès lors possible de créer un système de priorité, de décrire exhaustivement une donnée de contexte, etc. La verbosité offerte par un système textuel permet de produire un modèle compréhensible par un système de traitement informatique et un être humain, ce dernier étant à l'origine du modèle.

Langage orienté objet

La modélisation en utilisant un langage orienté objet permet d'apporter une modélisation moins abstraite. Par ailleurs, le langage objet offre des composants encapsulables et réutilisables dans d'autres développements logiciels. Dès lors, chaque capteur ou modèle peut être perçu comme une interface. Bien qu'il soit aisé de modifier le modèle, il est plus compliqué de le réintégrer suite à une modification. En effet, chaque évolution du modèle nécessite de recompiler complètement l'application. Le moindre changement structurel devient alors très complexe à mettre en oeuvre. Ce modèle est qualifié de statique dans la littérature.

UML et les langages graphiques

La modélisation ou représentation du contexte à travers les outils graphiques (UML, graphes orientés) trouve son origine dans la nécessité de créer un modèle par des non spécialistes en informatique. En effet, ce n'est pas toujours l'équipe de développement de l'application qui est la plus à même de proposer la modélisation du contexte. Les outils de modélisation permettent donc de créer une base commune et formelle pour échanger de l'information.

Modèles logiques

Les modèles logiques s'appuient sur deux éléments : des données structurées et un langage de règles. Ce dernier décrit, en utilisant généralement la logique de premier ordre, un traitement à effectuer sur les données structurées. Ce traitement mène à un résultat appelé *conclusion*. Nous retrouvons ce principe de fonction dans les ontologies, présentées ci-dessous.

Ontologies

Les ontologies sont présentées, dans la littérature scientifique et littéraire, comme un système de représentation de la connaissance. En informatique, les connaissances sont représentées par des axiomes, qui sont reliés entre eux par des contraintes, et régis lors de l'inférence par des règles heuristiques. Une attention particulière doit être apportée à la création de cette ontologie [44] (cité par [5]).

Pour conclure sur les outils de modélisation, cette partition des rôles, parfois très atomique, oblige les concepteurs d'applications, selon leurs choix technologiques, à utiliser une combinaison de ces outils pour travailler sur le contexte. Par exemple l'article de Hofer et coll. [32] modélise le contexte avec le langage orienté objet et l'enregistrent avec un langage de balises. D'autres outils, comme les ontologies, permettent la modélisation, le stockage et l'interprétation. C'est une des raisons de la croissance des ontologies au sein des systèmes sensibles au contexte [34]. La modélisation du contexte permet d'organiser les données isolées, rapportées par les capteurs, pour former un ensemble : le contexte.

La représentation du contexte a fait l'objet de plusieurs travaux, dont ceux sur les graphes contextuels [12]. Ils offrent une solution de modélisation accessible à tous, quelles que soient les compétences informatiques des personnes qui modélisent le contexte. En effet, dans le cadre de la conception d'applications sensibles au contexte, les spécialistes du domaine ne sont pas forcément les concepteurs de l'application. Ils n'ont pas forcément les connaissances informatiques pour modéliser le contexte dans le langage de développement de l'application. Les outils de modélisation sous forme de graphes permettent de pallier cette situation. Une fois modélisé sous la forme de graphes, le contexte peut être modélisé automatiquement sous la forme de règles de production. Ces dernières sont ensuite intégrées à une application sensible au contexte pour identifier un évènement et appliquer la procédure correspondante.

5.3 Applications autonomes

Les systèmes autonomes ont la capacité d'adapter leur logique métier en fonction de leur environnement en automatisant leurs fonctions. Un parallèle est souvent fait avec le système nerveux humain : il est responsable des fonctions automatiques (digestion, muscles cardiaques...) qui ne sont pas soumises au contrôle volontaire. À l'instar du système nerveux, les applications autonomes disposent de fonctions qui leur confèrent une large autonomie.

Nous retrouvons dans la littérature plusieurs définitions des systèmes autonomes. Les deux définitions présentées ci-dessous sont représentatives des objectifs de recherche de ce domaine.

“Self-adaptive software evaluates its own behavior and changes behavior when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible [49].”

“Self-adaptive software modifies its own behavior in response to changes in its operating environment. By operating environment, we mean anything observable by the software system, such as end-user input, external hardware devices and sensors, or program instrumentation [58].”

Autrement dit, il s'agit d'applications qui ont la capacité de s'adapter aux changements de leur environnement avec un objectif d'optimisation de la performance globale du système. Ce but fait écho à l'objectif de cette thèse : adapter les services numériques en fonction des changements de leur environnement pour offrir la meilleure performance face à une situation d'usage.

Cependant, c'est un domaine de recherche très large. Il n'est pas évident de déterminer avec précision quelle partie du domaine de l'informatique autonome adresse notre problématique. En effet, la littérature fait mention de trois termes pour exprimer cette idée d'autonomie sur l'optimisation de la performance :

- Systèmes autoadaptatifs (self-adaptive systems),
- Systèmes autonomiques (autonomic systems),
- Systèmes auto-gérés (self-managing systems).

Ces termes sont souvent utilisés de façon interchangeable puisque dans la majorité des cas ceux-ci recouvrent la même idée. Cependant il subsiste quelques différences : Huebscher et coll. et Salehie et coll. préconisent, respectivement dans [36] et [67], une approche au cas par cas.

Les applications autonomes sont des systèmes complets dont l'objectif est de remplir une ou des tâches précises qui sont, le plus souvent, en rapport avec un système informatique.

Salehie propose une relation de filiation [67], entre les systèmes auto-* et les applications sensibles au contexte (Figure 5.11). Les deux types de systèmes remplissent un rôle similaire, mais à des degrés différents.

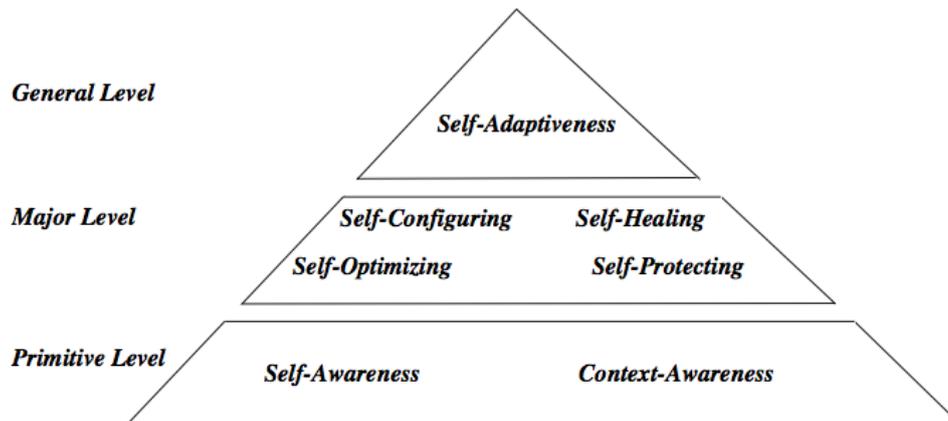


FIGURE 5.11: Pyramide des systèmes autonomiques [67]

La base de cette pyramide est composée des systèmes *conscients* (*-aware systems). Ils ont conscience de leur environnement (context-awareness) et de leur état (self-awareness). Les systèmes de cette couche primitive sont le fondement des systèmes adaptatifs. Le second niveau de la pyramide, la couche majeure, concerne les systèmes qui :

- s'auto-configurent : ils ont la capacité de se modifier en installant, supprimant ou modifiant les composants du système,
- s'auto-corrigent : ils peuvent diagnostiquer les problèmes puis corriger les erreurs, voire les prévenir,
- s'auto-optimisent : ils peuvent optimiser leur comportement en fonction des usages,
- s'auto-protègent : leurs fonctions défensives peuvent découvrir les failles et les corriger de manière autonome.

Le haut de la pyramide concerne les systèmes auto-adaptatifs de manière générale. La plupart de ces systèmes fonctionnent sur le principe d'une boucle de rétroaction fermée. Elle est mise en opposition avec la boucle ouverte, qui n'offre aucun contrôle ou retour d'information. Les systèmes en boucle ouverte n'ont pas de contrôle sur leur dérive. La boucle (fermée) de rétroaction est le modèle le plus commun et le plus utilisé. Cette boucle agit sur le système (lui-même) en s'appuyant sur des informations en provenance de son environnement d'opération (contexte). Nous retrouvons dans les systèmes adaptatifs les fondements de la théorie du contrôle. La réponse du système est contrôlée par les événements du contexte d'opération.

5.3.1 Boucle de rétroaction fermée

La boucle de rétroaction fermée est la base des systèmes auto-*, comme l'illustrent [15] et [58] (Figure 5.12).

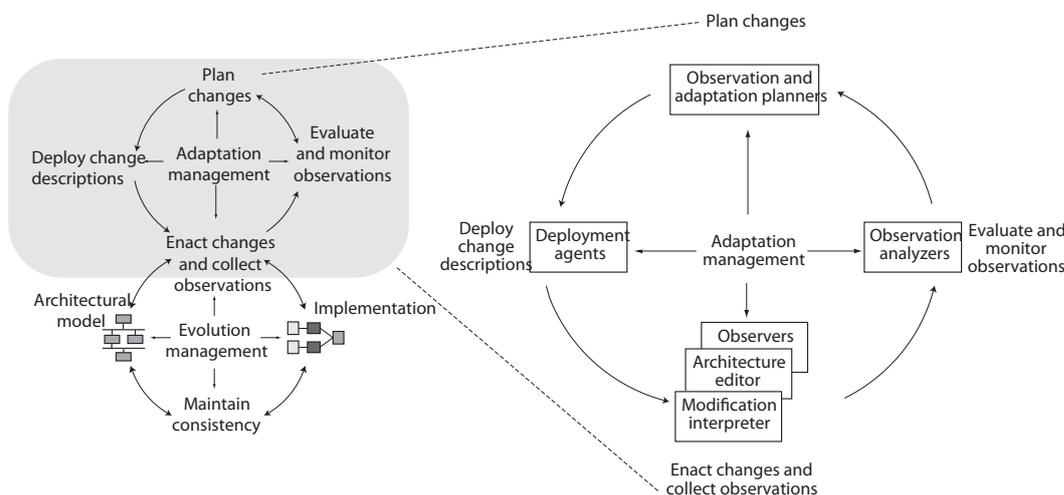


FIGURE 5.12: Diagramme de l'évolution de l'adaptation [58]

On retrouve dans cette boucle les quatre actions clés qui sont similaires à celles présentées dans la section 5.1 avec un découpage différent :

- Collecter les observations sur l'environnement opérationnel,
- Évaluer et surveiller,
- Planifier les actions d'adaptation,
- Exécuter les changements.

La dernière phase du processus adapte les modules du système pour le rendre compatible avec l'évènement. Cette phase peut être assez complexe dans certains cas, comme l'adaptation simultanée de modules.

Ces actions définissent les étapes pour une architecture totalement automatisée, c'est-à-dire sans l'intervention de l'homme.

L'article de Salehie et coll. représente le processus d'adaptation comme un flux plutôt qu'une boucle pour mettre en avant l'idée d'une transformation des données de contexte en action d'adaptation. Cette vision reprend les quatre étapes présentées ci-dessus et vient les compléter avec une cinquième étape : l'apprentissage. Le flux débute avec l'entité *capteurs*, qui recueille les données sur l'environnement. Il se termine avec l'entité *effecteur* qui applique l'adaptation. La représentation proposée dans [67] est inspirée du modèle MAPE-K [37] (Monitor, Analyse, Plan, Execute, Knowledge). Ce dernier est proposé par IBM pour modéliser de façon abstraite les éléments autonomiques.

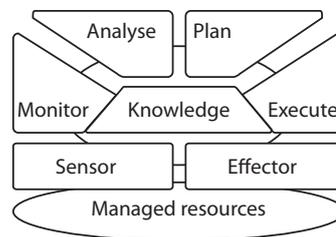


FIGURE 5.13: IBM MAPE-K

Le modèle d'IBM place la notion de connaissances au centre des étapes d'adaptation. Elle fait référence à l'apprentissage du système. C'est-à-dire la possibilité pour l'élément autonome d'apprendre de ces décisions d'adaptations puis de les exécuter. Ainsi, chaque réalisation d'une adaptation est évaluée, que ce soit de façon autonome ou bien avec une aide extérieure.

5.3.2 Les éléments clés d'un système auto-adaptatif

L'article [54] de McKinley et coll. présente un système auto-adaptatif comme une entité composée de deux couches : une couche *intergiciel* et une couche *applications*. Cette séparation des fonctions se retrouve souvent dans l'architecture de systèmes adaptatifs.

L'intergiciel fournit les fonctions d'adaptation aux applications, de l'observation à la proposition d'adaptation.

Dans cette boucle de rétroaction nous retrouvons comme éléments clés :

- un moteur de décision,
- des actions d'adaptation,
- un apprentissage.

Les adaptations (ou actions d'adaptations) sont proposées par un moteur de décision. Il décide des adaptations à apporter en fonction des changements de l'environnement et de la performance attendue par le système. C'est le coeur de la logique d'adaptation. Salehie [67] et Huebscher [36] citent les outils suivants pour réaliser le moteur de décisions :

- Arbres de décisions,
- Contraintes ADL (Langages de descriptions d'architectures),
- Règles ECA (Événement-Conditions-Actions),
- Machines d'états,
- Raisonners,
- Logique floue.

Ces moteurs permettent de choisir les actions d'adaptation à appliquer. Dans le cadre d'applications autonomes, ces actions peuvent concerner un ou plusieurs éléments de celle-ci. Les actions sont distinguées sous deux catégories, en fonction de leur appartenance au système : externes ou internes au système. Une action interne mélange le moteur d'adaptation et l'application au sein du même système. Alors qu'une action externe repose sur un moteur d'adaptation externe au système. Dans le second cas, le moteur d'adaptation est commun à plusieurs applications, comme proposé par [24][Goergiadis et coll.].

Il existe d'autres paramètres pour décrire et classer les adaptations [67] dans le cadre d'applications autonomes :

- **Discrètes.** Les actions d'adaptations qui ont lieu dans les couches inférieures (au niveau du logiciel médiateur) sont silencieuses. Aucune notification n'est envoyée pour avertir les applications qu'un changement a eu lieu [66]. L'adaptation est transparente pour le système.
- **Statiques/Dynamiques.** Les actions d'adaptations sont *statiques* lorsqu'il n'est pas possible d'en ajouter au cours de l'exécution de l'application. A contrario, un fonctionnement dynamique propose d'étendre le comportement défini lors du lancement par l'ajout d'actions au cours de l'exécution. Ces nouvelles actions d'adaptations peuvent concerner des entités non encore mises en jeu.
- **Réactives/Proactives.** Ce paramètre fait référence à la propriété (ou capacité) d'anticipation. Les actions sont *réactives* si elles ont lieu après le changement d'environnement. Dans cette situation le système subit son environnement. Les adaptations *proactives* préviennent les changements de l'environnement. Le système est capable de prédire ces derniers et anticipe la réponse.
- **Pilotées/Autonomes.** Les actions pilotées s'appuient sur un modèle qui définit le système et son environnement. Il est défini lors de la conception et est chargé au lancement du logiciel autonome. En revanche, les actions autonomes (*model-free adaptations*) ne suivent pas un modèle prédéfini pour adapter le comportement du système par rapport à son environnement. Dans ce cas, le système est totalement

autonome.

Une des caractéristiques des systèmes adaptatifs autonomes est la capacité à apprendre pour enrichir le modèle d'adaptation. Cet apprentissage se réalise sur la base d'une évaluation de l'efficacité d'une adaptation. L'évaluation peut être effectuée aussi bien par un système informatique que par un humain. Dans ce dernier cas, un opérateur évalue le résultat des adaptations. S'il est automatique, c'est un moteur qui évalue l'adaptation en fonction de la performance réalisée.

Pour conclure, les principes qui régissent les applications autonomes sont similaires à ceux des applications sensibles au contexte. Les processus d'adaptation sont réalisés avec les mêmes technologies et leur mode opérationnel est commun :

- Surveillance de l'environnement,
- Analyse les données,
- Planification des changements,
- Adaptation des composants du système,
- Apprentissage.

Les différences entre ces deux modes de fonctionnement concernent leur utilisation et le degré d'autonomie. L'objectif à atteindre pour les applications autonomes est d'avoir un système qui se suffit à lui-même, tout au long de son cycle de vie, quel que soit l'évènement. Le système est piloté par sa performance au sein de son environnement. Une application sensible au contexte est un système qui réagit aux évènements de son environnement, sans forcément inclure un objectif de performance ou d'auto-suffisance.

5.4 Conclusion

Une application sensible au contexte est divisée en deux parties : le coeur de métier de l'application et les fonctions d'adaptation. La réalisation de la logique de gestion du contexte passe par plusieurs questions :

- À quel problème répond l'application ? Quels sont les évènements qui pilotent son adaptation ?
- Quelles caractéristiques permettent d'identifier l'évènement ?
- Comment récupérer ces caractéristiques ?
- Comment agréger ces caractéristiques pour donner du sens à l'environnement et déduire le contexte ?
- Comment propager les évènements qui permettront aux applications d'adapter leur comportement ?

La première étape se compose de la définition du contexte en le spécialisant et en identifiant les situations d'exécution de l'application. L'identification du contexte permet de choisir les capteurs d'acquisition des données qui composent le contexte. Il s'agit

d'avoir une liste de données qui permet d'identifier une situation d'exécution. Cette identification permet ensuite d'effectuer un changement sur l'application pour l'adapter à son contexte.

Les changements à opérer pour correspondre à la situation d'usage sont décidés par l'application, sur la base d'un modèle de transformation. Ce dernier apporte les connaissances nécessaires à l'interprétation des données brutes, qui transforme les données brutes issues des capteurs en informations puis en connaissances du système. La modélisation du contexte est l'étape la plus cruciale pour la conception d'un système sensible au contexte.

L'article [34] de Hong et coll. revient sur les problématiques afférentes aux applications sensibles au contexte. On pourra citer entre autres les challenges scientifiques autour de l'interprétation et de la classification des informations de contexte. Le contexte peut être composé d'une large quantité de données. L'optimisation du traitement de ces données pour améliorer la rapidité des déductions logiques passe par des travaux dans d'autres domaines. Cet aspect est tout à fait contemporain avec l'émergence forte, il y a un an, des problématiques autour des *Big Data*.

La dernière étape consiste à propager le contexte jusqu'à l'application pour qu'elle adapte son comportement.

Le section suivant présente la notion d'ontologie en informatique au travers du Web Sémantique. Nous retrouvons cette technologie dans plusieurs modèles de système sensible au contexte. C'est par ailleurs la solution retenue dans notre modèle pour l'analyse et la déduction du contexte pour notre intergiciel.

Chapitre 6

Les ontologies en informatique

Le terme ontologie a fait son apparition en informatique au cours des années 1990, au sein de la communauté scientifique travaillant sur l'intelligence artificielle. Ce terme apporté par Tom Gruber en 1992 a fait l'objet de plusieurs travaux autour de sa définition et de son implémentation. C'est aujourd'hui une technologie utilisée pour concevoir des applications aussi bien dans le monde académique que dans le monde industriel [52].

Cette notion est importée de la philosophie, domaine dans lequel on définit l'Ontologie comme une branche fondamentale de la Métaphysique. Cette dernière s'intéresse à la notion d'existence, aux catégories fondamentales de l'existant et étudient les propriétés les plus générales de l'être [23].

En informatique, la définition d'une ontologie fait débat [30]. En effet, certaines définitions propres au domaine informatique sont en contradiction [29]. La définition la plus complète, de mon point de vue, est celle proposée par Tom Gruber. Il fut un des premiers à proposer une définition [27] et revient sur celle-ci, quinze ans après (traduction de [28]) :

“Dans le contexte de l'informatique et des sciences de l'information, une ontologie définit un ensemble de représentations primitives qui permettent la modélisation d'un domaine de connaissances. La représentation des primitives comprend des classes (ensembles), des attributs (propriétés) et des relations (relations entre les membres des classes). Les primitives sont utilisées pour présenter des informations sur le sens et les contraintes logiques de l'application. À l'instar d'une base de données, une ontologie propose un niveau d'abstraction du modèle de données avec des caractéristiques similaires puisque nous retrouvons les relations entre les modèles. Cependant l'organisation de l'information est différente. Les ontologies modélisent des connaissances autour d'une unité : l'individu en précisant ses attributs et les relations avec les autres individus. Les ontologies sont spécifiées dans un langage qui offre une abstraction des structures de données et de l'implémentation. En

pratique, ce langage est plus proche de la logique descriptive que d'autres langages utilisés pour modéliser les bases de données. Pour cette raison, les ontologies sont dites *d'un niveau sémantique*, alors que les bases de données sont dites *d'un niveau logique ou physique*. Grâce à leur indépendance par rapport aux modèles de données de bas niveau, les ontologies peuvent être utilisées pour aligner des bases de données hétérogènes, ajouter une interopérabilité entre différents systèmes ou simplement être des interfaces pour les services de bases de connaissances."

Dameron [19] définit les ontologies autour de deux notions :

"Une ontologie est un accord sur une conceptualisation partagée et éventuellement partielle." "Une ontologie formelle est un développement systématique, formel et axiomatique de la logique de toutes les formes et tous les modes d'existence."

Une idée de collaboration est mise en avant. Elle sous-entend l'existence d'un consensus sur la conceptualisation d'un domaine. Cette idée, mixte de consensus et de conceptualisation, est partagée par [13] qui classe les ontologies selon 4 catégories :

- Les ontologies **d'application**. Elles sont spécifiques à une application. En dehors de cette tâche, leur validité n'est pas vérifiée. Elles sont très précises.
- Les ontologies **de domaine**. Elles sont propres à un domaine et leur validité couvre un ensemble d'applications.
- Les ontologies **générales**. La précision de ces ontologies est limitée, mais en contrepartie elles englobent les domaines sous-jacents au sens général.
- Les ontologies **supérieures**. Elles représentent les concepts généraux admis par tous. C'est le cas de l'ontologie *DOLCE*¹ du projet européen WonderWeb². Elle est dite fondationnelle et présente des concepts abstraits pour des domaines de connaissances.

Pour conclure, en philosophie l'Ontologie est une science qui étudie les propriétés de ce qui *est*. En informatique c'est un outil qui permet *la formalisation d'une convention de langage entre plusieurs entités* associée à un raisonnement logique. Dans la suite de ce chapitre, nous nous intéressons aux aspects techniques des ontologies avec leur implémentation au sein des outils du Web sémantique.

6.1 Le Web Sémantique

Ce mouvement est né sous l'impulsion de Tim Berners-Lee qui propose de créer un web de données libres et liées (*linked data*), c'est-à-dire un espace non plus de documents que seul l'humain peut comprendre, mais un espace où les ordinateurs peuvent interagir

1. <http://www.loa.istc.cnr.it/DOLCE.html>

2. http://cordis.europa.eu/projects/rcn/60325_en.html

avec les données³. Un espace numérique où les données deviennent intelligibles pour les machines et les humains. Pour que les données soient intelligibles par les machines, il faut faire émerger une *sémantique* compréhensible par les machines.

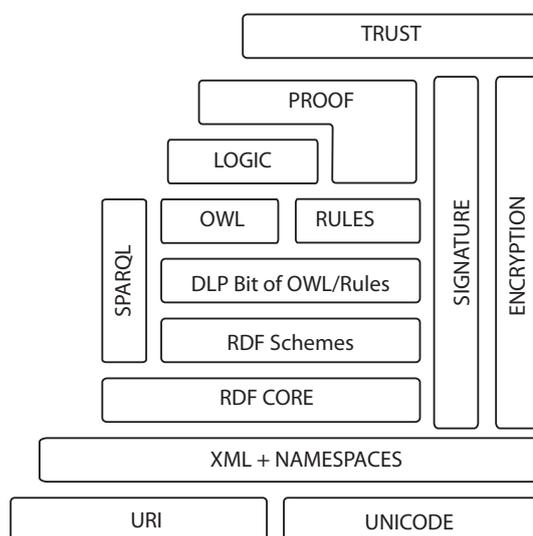


FIGURE 6.1: Feuille de route des technologies du Web sémantique

Pour remplir cet objectif, Tim Berners-Lee propose une feuille de route technologique : la *pile du web sémantique* (Figure 6.1). Ce modèle en couche s'appuie sur deux briques essentielles, les URI et XML. Nous retrouvons la description sémantique avec les technologies RDF et les la déduction logique avec les ontologies pour le web (OWL).

Nous ne présenterons dans cet état de l'art que les ontologies pour le web sémantique (OWL). Une présentation succincte des URI, de XML et de RDF est fournie en annexe (Appendice B).

6.2 Web Ontology Language (OWL)

La première couche d'outils (XML et URI) constitue les fondations des outils sémantiques, en apportant syntaxe structurée et identification. La seconde couche (RDF et RDF Schema) offre un modèle de données pour décrire sémantiquement les ressources. La troisième (OWL) complète le vocabulaire de RDF en apportant les notions indispensables à un raisonnement logique. Grâce à cette capacité de déduction logique, les machines ont la capacité, à l'instar des êtres humains, de traiter les données au niveau sémantique. En effet, *RDF* apporte la description et l'identification des ressources. Ce mécanisme est indispensable, mais insuffisant pour un raisonnement sémantique. Pour

3. http://www.ted.com/talks/lang/en/tim_berniers_lee_on_the_next_web.html

offrir un raisonnement sémantique, il faut compléter la description des ressources avec des concepts logiques (cardinalité, équivalence, union, disjonction...).

Le W3C a défini trois sous versions d'OWL, qui offrent plus ou moins de fonctions. Elles sont hiérarchisées. Ainsi, OWL Lite est compris dans OWL-DL et OWL-DL est compris dans OWL Full :

- OWL Lite : Base du langage OWL avec des concepts simplifiés. Il ne permet pas beaucoup d'expressivité dans la description. Par exemple, la cardinalité est limitée à 0 ou 1.
- OWL DL. DL signifie Description Logic (Logique de description). C'est un formalisme permettant la description et le raisonnement. OWL DL est une extension d'OWL Lite pour apporter une expressivité élevée (cardinalité, union...), mais implique un raisonnement plus lent qu'OWL Lite.
- OWL Full est la version d'OWL la plus complète. Elle offre une plus grande capacité de description au détriment de la complexité du raisonnement.

Le choix de la sous-version dépend de la complexité de la description souhaitée. Mais au final, ce choix semble être limité puisque guidé par les implémentations existantes de moteurs de raisonnement. En effet les moteurs d'inférence supportent principalement OWL DL et OWL Lite. Le choix est donc limité à ces deux derniers.

6.2.1 Caractéristiques d'une ontologie

Une ontologie s'appuie sur trois notions pour modéliser un domaine : le concept, le rôle et l'individu. Nous pouvons faire un parallèle avec la programmation orientée objet (la classe, la propriété et l'instance). Un individu représente un élément. Cet élément (*instance*) fait partie d'un ensemble, appelé concept (*classe*). Les liens entre les individus sont appelés les rôles (*propriétés*). L'ensemble de ces éléments compose la base de connaissances qui contient la formalisation du domaine et son instanciation. Ceci débouche sur deux niveaux :

- La terminologie (TBox) qui est la modélisation de la connaissance. Elle est formée de concepts et de rôles. C'est le cadre axiomatique du domaine.
- Les assertions (ABox) qui regroupent les individus. C'est la réalisation de la modélisation (TBox).

Nous retrouvons toutes les caractéristiques d'une base de données, c'est-à-dire un modèle (TBox) qui permet de structurer les données (ABox). La TBox et la ABox composent la connaissance du domaine. La différence majeure avec une base de données standard réside dans la capacité d'une ontologie à s'enrichir avec les données dont elle dispose à l'aide de son moteur d'inférence. Le raisonnement permet de déduire de nouvelles informations à partir des informations contenues dans la base. Ce raisonnement logique par déduction s'appuie sur deux principes, présentés ci-dessous. Autrement dit, une base de connaissances est une base de données spécialisée dans le stockage et la

déduction de connaissances.

Monde ouvert et monde fermé

Dans un raisonnement basé sur OWA (Open World Assumption - Monde ouvert), certaines informations contenues dans la base de connaissances ne sont pas déductibles. Tant qu'un énoncé non déductible n'est pas prouvé, tout est encore possible. Il est considéré comme inconnu. Pour mieux comprendre ce principe, il faut se référer à l'opposé d'OWA, CWA (Closed World Assumption - Monde fermé). Dans ce type de raisonnement, tous les énoncés qui ne sont pas prouvés comme justes sont faux.

Énoncé : Yves-Gaël a un permis B

Question : Est-ce qu'Yves-Gaël a un permis A ?

OWA : Peut-être.

CWA : Non

Le monde ouvert conserve la possibilité d'une réponse vraie si les informations sont incomplètes, alors qu'en monde fermé, la réponse sera fausse.

Monotonie (monotonic)

Un système de déduction logique basé sur la monotonie ne remet pas en question les déductions établies, même si de nouvelles assertions entraînent des déductions opposées. Par exemple, en reprenant l'illustration précédente.

Énoncé 1 : Yves-Gaël a un permis B.

Question : Est-ce qu'Yves-Gaël a un permis B ?

Réponse : Oui

Nouvel Énoncé : Yves-Gaël a un permis A.

Question : Est-ce qu'Yves-Gaël a un permis A ?

Réponse : Oui

Question : Est-ce qu'Yves-Gaël a un permis B ?

Réponse : Oui

Ceci peut mener à des inconsistances de raisonnement pour les systèmes de raisonnement utilisant les ontologies. Si le rôle “a un permis” est fonctionnel en OWL, cela signifie qu’on ne peut avoir qu’un et un seul permis. Dans cet exemple, le nouvel énoncé rend la base de connaissances inconsistante. Il y a une violation du principe de satisfaisabilité. Pour pallier cette situation, il faut un système de “retour en arrière” qui supprime les conclusions invalides suite à un changement.

6.2.2 Mécanisme d’inférence

Le mécanisme d’inférence ajoute de nouvelles conclusions à partir de propositions existantes. Ces conclusions font alors partie intégrante de la base de connaissances. Le mécanisme de déduction permet la création de nouveaux concepts, rôles ou individus. Puisque les déductions s’appuient sur les connaissances existantes, il faut dans un premier temps vérifier leur validité, c’est-à-dire vérifier qu’elles n’entrent pas en conflit avec des individus ou des concepts déjà existants. En effet, le système n’est pas capable de décider de lui-même quelle information favoriser si deux éléments sont en conflit. Le même travail est effectué avec les déductions à intégrer. Ces mécanismes s’assurent que les connaissances de la base ne rentrent pas en conflit d’un point de vue logique.

Au niveau de la TBox, le raisonneur contrôle et complète le modèle à partir des connaissances explicites. Il :

- Vérifie la satisfaisabilité des concepts : tous les concepts doivent être réalisables.
- Déduit les subsomptions entre concepts : Relation hiérarchique entre les concepts, appartenance des concepts à d’autres concepts.
- Déduit les équivalences et les disjonctions entre concepts.

Au niveau de la ABox, le raisonneur vérifie les assertions. Il :

- Contrôle les instances : Toutes les instances doivent appartenir à un concept,
- Vérifie la cohérence des instances avec leurs concepts (TBox),
- Recherche le concept le plus spécifique pour chacune des instances.

Ainsi pour éviter toute conclusion erronée et donc toute instabilité, le système contrôle la cohérence de la base lors de l’inférence, sur les concepts, les rôles et les individus.

6.2.3 Langages de règles

Les langages de règles sont utilisés pour contrôler ou prendre des décisions, en décrivant les concepts d’une organisation. Ils formalisent le savoir-faire et permettent d’automatiser certaines actions métier. Cette approche permet de proposer une approche logique, proche d’un raisonnement humain, à partir d’observations.

L'article [35] de Horrocks et coll. propose de compléter OWL, plus particulièrement la version DL, en ajoutant un langage de règles de production nommé SWRL (Semantic Web Rule Language). Ce système de règles métier, mélange de OWL-DL et RuleML⁴ (Rule Markup Language) ajoute des fonctions d'expressivité logique à OWL. Il s'agit d'avoir un levier supplémentaire pour la manipulation des individus et de leurs propriétés au sein de la base de connaissances. Bien que ces règles puissent utiliser l'ensemble des éléments d'une ontologie, elles ne concernent que les individus (ABox) et non les concepts ou rôles (TBox). Une règle SWRL se construit sur le schéma suivant :

$$\text{antécédents}(\text{body}) \Rightarrow \text{conséquence}(\text{head})$$

L'antécédent et la conséquence sont composés de zéros ou plusieurs atomes, qui forment une conjonction d'atomes. Le premier (antécédent) exprime les conditions nécessaires à la réalisation de la seconde partie (conséquence).

Il existe deux cas particuliers, en dehors de la forme générale d'une règle (composée d'une conjonction d'atomes pour l'antécédent et la conséquence) :

- Un antécédent vide traduit le fait que toutes les conditions satisfont la réalisation.
- Une conséquence vide traduit le fait qu'aucune condition ne satisfait la réalisation.

Un atome peut être un concept, un rôle OWL ou le résultat d'une opération SWRL prenant une ou deux variables. Les variables permettent de sélectionner les individus en fonction des concepts auxquels ils appartiennent, des relations entre ces concepts ou des opérateurs propres à SWRL (comparaisons et builtin).

$$\text{parent}(?x, ?y) \wedge \text{brother}(?y, ?z) \Rightarrow \text{uncle}(?x, ?z)$$

L'ajout des "Built-Ins" dans la syntaxe des règles est motivé par une approche modulaire qui permet d'étendre le système. Les modules officiellement supportés existants sont :

- la comparaison,
- la manipulation mathématique,
- la gestion des chaînes de caractères,
- la gestion des dates,
- l'accès à des URIs,
- les listes.

4. RuleML est un langage basé sur XML qui permet d'exprimer des règles pour déduire ou transformer des éléments lors d'inférences. Il est proposé par une initiative composée d'industriels et d'académiques (<http://www.ruleml.org>).

Un aspect important de ce langage de règles est l'impossibilité d'ordonner ou rendre prioritaires les règles SWRL. Ceci peut dès lors rendre certaines règles très complexes voire impossibles puisque tout doit être réalisé en une seule requête. Pour pallier ce problème, l'article [26] Grosz et coll. introduit le système (*SILK*)⁵ pour prioriser les règles en cas de conflit. L'assertion utilisée pour ce système de priorité est *@override*.

6.2.4 Langages de requêtes

Les parties précédentes présentent les ontologies OWL et les règles SWRL. Une ontologie permet de stocker et gérer des connaissances au moyen d'une logique de description. Les règles SWRL apportent un niveau supplémentaire d'expressivité en logique descriptive. La dernière pièce du tableau est un langage de requêtes qui permet d'accéder aux informations contenues dans la base de connaissances.

RDF dispose de son langage d'interrogation, SPARQL, pour extraire des données d'une base RDF. Les requêtes sont basées sur les modèles de graphes et proposent une variété d'extension (filtres, modificateurs...). OWL utilisant RDF, il est possible d'interroger une ontologie OWL en utilisant SPARQL mais il ne permet pas d'offrir des requêtes utilisant les concepts OWL. Nous dénombrons trois solutions qui permettent de requêter une ontologie OWL au niveau sémantique d'OWL. OWL-DL, DIG's ASK et SQWRL. OWL-DL est resté à l'état de spécifications et ASK manque d'expressivité.

O'Connor propose SQWRL (Semantic Query-enhanced Web Rule Language) [57]. C'est un langage qui emprunte la syntaxe des requêtes de sélection de SQL tout en utilisant la forme des règles SWRL. L'antécédent joue le rôle de filtre de la requête et la conséquence définit le type de requête et les valeurs à interroger. Cette extension SWRL n'est malheureusement pas supportée par tous les moteurs d'inférences.

5. Scalable and Expressive Semantic Rules

Chapitre 7

Bilan

L'adaptation des contenus à leur environnement de consommation est un problème qui a été soulevé lors de l'arrivée des réseaux mobiles. Cette problématique resurgit avec l'arrivée imminente des réseaux convergents puisqu'au cours d'une même session, l'environnement informatique peut subir des variations qui nécessitent une adaptation des services numériques en cours de consommation. On retrouve dans la littérature plusieurs travaux (Tableau 5.1) qui traitent de l'adaptation des services numériques dans les réseaux du futur. La majorité des travaux concernent uniquement la phase d'observation du contexte d'exécution ou bien la mécanique d'adaptation du service numérique. Nous trouvons très peu de propositions autour de la phase de décision d'adaptation. Les travaux de Skorin-Kapov et coll. [71] [69] [70] font exception à cette observation. Cependant l'approche proposée est complexe à mettre en oeuvre. Elle nécessite une modification de certaines entités d'IMS et un déploiement réparti des différentes entités pour permettre de rendre un service numérique adaptable. Ce dernier point est particulièrement délicat à mettre en oeuvre dans le cadre d'un réseau NGN. En effet, une des philosophies est la séparation des rôles, avec d'un côté un fournisseur pour l'accès au réseau et de l'autre un fournisseur pour l'accès aux services.

Par ailleurs, l'adaptation d'applications ne se cantonne pas aux réseaux du futur. Aucun des travaux étudiés au cours de mon doctorat ne fait référence aux applications sensibles au contexte ou bien aux applications autonomes. Les applications sensibles au contexte accompagnent les humains dans leurs tâches quotidiennes. Elles adaptent la réponse globale de l'application en fonction de l'environnement d'exécution et ceci de façon transparente (sans intervention de l'utilisateur). L'évolution de cet axe de recherche mène aujourd'hui à des travaux sur la conception de solutions qui permettent d'ajouter la sensibilité au contexte à un ensemble d'applications, en limitant les développements spécifiques. Les applications autonomes, de leur côté, sont pilotées par la performance des différents éléments qui la composent. Il s'agit d'assurer la meilleure performance possible, quels que soient les changements de l'environnement dans lequel elles évoluent. Bien que les deux domaines soient distincts dans leurs applications, l'objectif général et

les modèles sont similaires.

Nous avons donc souhaité prendre une approche transverse en nous inspirant des pratiques existantes dans le domaine des applications sensibles au contexte et des applications autonomes. Ces systèmes reposent sur le principe d'une boucle de rétroaction fermée au cours de laquelle nous retrouvons quatre phases :

- Collecter les observations sur l'environnement opérationnel,
- Évaluer et surveiller,
- Planifier les actions d'adaptation,
- Exécuter les changements.

On retrouve aussi le principe de séparation des rôles et d'architecture en couche. Celui-ci crée une indépendance entre la logique métier d'une application sensible au contexte et les fonctions de gestion du contexte de celle-ci. D'ailleurs on observe une évolution du concept avec l'externalisation et la mutualisation des fonctions de gestion du contexte au sein d'un intergiciel pour apporter la sensibilité au contexte pour une ou plusieurs applications.

Les étapes de collecte des observations, d'évaluation, de surveillance et d'exécution ont déjà été largement traitées dans les travaux concernant l'adaptation des applications dans les réseaux du futur. Nous nous intéressons donc à l'étape de planification des actions d'adaptation qui concerne l'identification de la situation d'usage et la réponse à apporter.

Pour conclure, les travaux relatifs aux NGN, présentés dans cette partie, n'offrent pas une réponse complète pour l'adaptation dynamique des applications à leur environnement informatique. Par ailleurs les quelques travaux qui proposent un modèle complet s'éloignent de certains principes clés qui sont à la base des réseaux du futur, avec comme conséquence des propositions difficilement exploitables. Nous souhaitons combler le fossé qui sépare les travaux sur l'adaptation dans les réseaux du futur avec les applications sensibles au contexte et autonomes tout en restant dans les compétences de notre équipe de recherche, le web sémantique.

C'est donc une approche hybride, qui s'appuie sur plusieurs domaines, qui a été retenue pour répondre à la problématique soulevée dans cette thèse.

Troisième partie

Systeme de sensibilité au contexte pour les réseaux du futur

Chapitre 8

Introduction

Dans un réseau du futur, un service numérique désigne une application réseau consommable par un utilisateur. Le terminal de l'utilisateur exécute une instance client du service numérique. Elle communique, au travers du réseau, avec une autre instance présente sur un serveur. Cet usage se trouve renforcé par l'ensemble des fonctions apportées par les réseaux du futur : mobilité généralisée, communications multimédia enrichies, convergence des technologies d'accès, etc. Cependant, la plupart de ces fonctions sont relatives au réseau et ne s'opèrent qu'à son niveau du réseau, c'est-à-dire en dehors du niveau applicatif.

Le *vertical handover* illustre parfaitement ce problème. Les terminaux des utilisateurs ont la possibilité de changer de technologie d'accès au cours de leurs sessions. Or, tous les liens d'accès au réseau n'ont pas les mêmes caractéristiques. Prenons par exemple, un des paramètres qui a souvent une influence importante sur les services numériques : la bande passante. Ce paramètre représente la quantité de données que peut transporter le lien réseau. Puisque les liens réseau n'ont pas les mêmes caractéristiques, le changement d'une technologie d'accès à une autre entraîne, dans notre exemple, un changement de cette capacité de transport. Ainsi, si la bande passante devient insuffisante pour le transport des données, le service ne peut plus être acheminé jusqu'à l'utilisateur. Dans cette situation, malgré la capacité du réseau à maintenir la connexion, l'utilisateur expérimente une coupure du service numérique en cours de consommation. Le réseau a su s'adapter à ce changement de conditions, mais pas le service numérique. Pour pallier ce problème, il est nécessaire que les services numériques :

1. Soient informés des changements de situation,
2. Aient la capacité d'adapter la logique métier.

Le service numérique doit fournir plusieurs modalités de diffusion pour un même contenu, chacune est associée à une ou plusieurs situations d'usage. Cette vision de multimodalité applicative est un premier pas vers une adaptation des services numé-

riques à leur environnement d'exécution. Il faut ajouter à cela une capacité du service numérique à changer automatiquement et dynamiquement d'une modalité à une autre en fonction de son environnement. Il doit comprendre les changements de son environnement d'exécution et s'adapter en conséquence, autrement dit adapter le contenu ou la logique d'exécution en fonction de l'environnement, sans intervention de l'utilisateur. Ce partitionnement de la solution (multimodalité et autogérance) illustre les deux idées fondamentales qui sont à l'origine de notre vision des services numériques adaptatifs :

- une multimodalité de la logique métier,
- une gestion autonome.

Nous différencions un service numérique standard d'un service numérique adaptatif par sa capacité à modifier automatiquement et dynamiquement son interaction avec l'utilisateur, au cours de son exécution.

Dans le cadre des réseaux du futur, il s'agit d'**adapter les flux échangés entre l'instance sur le serveur et l'instance sur le terminal client**. Cette adaptation résulte des changements pertinents observés dans l'environnement informatique (technologie d'accès, terminal, etc.). Cet environnement est présent dans la littérature scientifique, sous le terme de *contexte informatique*. Nous bornons le contexte des services numériques en spécialisant le contexte d'exécution d'un service numérique au domaine de l'informatique. Ainsi, quel que soit le service numérique, les éléments qui caractérisent son environnement sont finis puisqu'ils sont limités aux caractéristiques des terminaux et technologies d'accès.

L'objectif de ces travaux est d'apporter une réponse à l'adaptation automatique et dynamique des services numériques dans les réseaux du futur pour les raisons évoquées ci-dessus. La suite de cette partie présente notre proposition pour un tel système.

Nous commençons par une présentation générale de l'architecture de notre proposition (chapitre 9) puis des blocs fonctionnels qui la composent (chapitre 10). Nous présentons ensuite les fonctions de l'intergiciel (chapitre 11) qui sont au coeur de notre proposition. La conclusion (chapitre 12) insistera sur les particularités de notre modèle.

Chapitre 9

Architecture de l'intergiciel

Dans ce chapitre, nous précisons, dans une première partie, nos définitions pour chaque élément contribuant à l'architecture de l'intergiciel¹. Dans une seconde partie, nous présentons les principes de conception utilisés pour réaliser notre intergiciel.

9.1 Périmètre fonctionnel

En informatique, les applications client/serveur s'exécutent sur deux instances nommées *client* et *serveur*. Ces deux entités travaillent de pair en s'échangeant des données au travers d'un réseau.

L'instance serveur est exécutée sur un *serveur d'applications*, l'instance client est exécutée sur un *terminal utilisateur*. L'environnement d'exécution de ces applications est composé :

- D'un ordinateur serveur qui exécute une instance serveur du service,
- D'un équipement utilisateur qui exécute une instance cliente du service,
- D'un réseau qui permet l'échange de données entre ces deux instances.

C'est ce paradigme (client/serveur) qui est utilisé pour les services numériques dans un NGN. Dans un réseau du futur, les performances du réseau et des terminaux sont susceptibles de varier au cours d'une session d'exécution. En effet, le principe de mobilité permet à l'utilisateur de changer, au cours d'une session applicative, de terminal ou de technologie d'accès. Autrement dit, le contexte d'exécution dépend d'un environnement non figé qui dépend lui-même de l'évolution de la situation de l'utilisateur.

Notre objectif est d'adapter **n'importe quel service numérique** hébergé par les

1. Etymologiquement composé du préfixe d'origine latine 'inter' qui veut dire entre et du suffixe 'iciel' qui signifie logiciel, ce mot dénomme les logiciel qui fonctionnent sur le modèle client-serveur et servent d'intermédiaire transparent entre un logiciel d'application et une entité (un réseau, un autre logiciel, ...) Le terme anglais consacré est middleware.

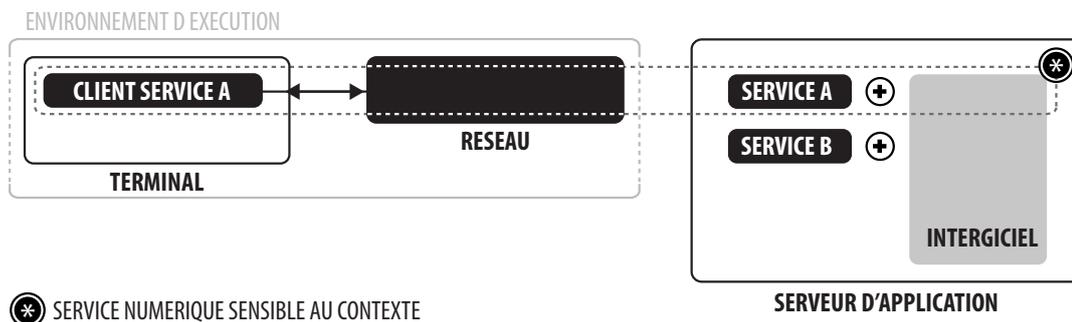


FIGURE 9.1: Un service numérique sensible au contexte

serveurs d'applications au sein des réseaux du futur. Nous ne voulons pas nous limiter à un type d'application particulier. Pour cela, nous souhaitons apporter des fonctions génériques de gestion du contexte informatique. Ces dernières permettent d'adapter l'instance serveur du service à des conditions d'exécution identifiées.

Dans la suite de cette sous-section nous présentons les différentes entités qui composent le périmètre fonctionnel de notre proposition (Figure 9.1) :

- Le service numérique sensible au contexte (hébergé par le serveur d'applications),
- L'intergiciel de sensibilité au contexte (présent sur le serveur d'applications),
- L'environnement d'exécution (composé du terminal client et du réseau).

9.1.1 Le service numérique sensible au contexte

Un service numérique sensible au contexte est un service numérique qui a la capacité d'adapter sa logique métier en fonction de son environnement d'exécution. Il dispose de deux logiques d'exécution : une logique métier et une logique de gestion du contexte. Cette dernière permet au service numérique de s'adapter à son environnement d'exécution. Dans notre proposition, cette logique supplémentaire (gestion du contexte) est fournie par une entité additionnelle sous la forme d'un intergiciel. L'intergiciel de gestion du contexte prend en charge l'ensemble des fonctions de sensibilité au contexte (context-awareness) pour des services numériques sensibles au contexte. Ainsi, l'instance du service numérique est dépourvue de tout ce qui est relatif à la gestion du contexte. Elle n'est composée que de la logique métier du service numérique. Cependant, l'adaptation, qui est une modulation de la réalisation de la logique métier du service numérique, est réalisée par le service numérique. Le résultat est une modification de l'instance serveur du service numérique et dans certaines conditions l'instance client. Dans ce dernier cas, elle sera communiquée au client par le serveur.

Dans notre scénario de référence, présenté en introduction (chapitre 2), cette logique métier est la mise en relation de plusieurs individus à l'aide de flux audio et vidéo. Pour la fourniture de ce service, les modalités d'exécution existent sous la forme :

- d'une communication audio seulement.
- d'une communication audio et vidéo (sous différentes qualités).

9.1.2 L'intergiciel

Comme présenté ci-dessus, l'intergiciel vient compléter le service numérique en lui proposant les fonctions de gestion du contexte. Ces dernières correspondent à l'observation de l'environnement d'exécution, l'identification des situations d'usage et le choix de la modalité d'exécution à employer. Ce travail est réalisé pour chaque session de chaque service numérique sensible au contexte présent sur le serveur d'applications. Autrement dit, l'intergiciel a pour fonction la gestion du contexte des sessions au profit des services numériques sensibles au contexte hébergés sur le serveur d'applications.

L'intergiciel, au travers de ces fonctions de gestion du contexte, transforme les données sur l'environnement d'exécution en choix d'adaptation. Ces choix sont utilisés par les services numériques sensibles au contexte pour adapter leurs sessions. Ces adaptations, qui représentent le changement de modalités d'exécution, portent sur les flux échangés entre le client et le serveur concernés par une session. Autrement dit, chaque service, après s'être informé sur les adaptations nécessaires pour ses sessions ouvertes, adapte le flux à destination du client.

9.1.3 L'environnement d'exécution

Un service numérique sensible au contexte s'adapte à son environnement d'exécution. Ce dernier permet à l'intergiciel d'identifier la situation d'usage et de proposer la modalité d'exécution à utiliser pour chaque session ouverte.

Dans un réseau du futur, seules les caractéristiques du réseau et du terminal sont susceptibles d'évoluer au cours de l'exécution. Nous avons donc défini l'environnement d'exécution pertinent pour les décisions d'adaptation autour de ces deux éléments. La caractérisation de cet environnement est réalisée par le prélèvement d'informations grâce à des capteurs physiques ou logiciels.

9.1.4 Conclusion

Le périmètre fonctionnel de notre proposition est composé de trois entités :

- Un serveur d'applications qui héberge le service numérique sensible au contexte (instances serveur) et l'intergiciel de sensibilité au contexte.
- Un terminal client qui exécute la logique client du service numérique.

- Un réseau qui transporte les données entre l'instance client et l'instance serveur du service numérique.

Parmi les caractéristiques qui évoluent au cours du temps, nous nous intéressons à celles qui sont relatives au terminal et au réseau. Ce sont les seuls éléments que nous retenons pour composer notre contexte d'exécution. Ainsi, l'évolution de la performance du serveur d'applications n'a pas été prise en compte puisqu'elle n'est pas relative aux réseaux du futur. Nous souhaitons adapter les sessions de chaque service numérique grâce à l'observation de ces caractéristiques. Dans notre architecture, la session d'un service numérique sensible au contexte est composé (Figure 9.1) :

- D'une instance du service numérique sur le serveur d'applications.
- D'une instance du service numérique sur le terminal client.
- D'une prise en charge des fonctions de gestion du contexte par l'intergiciel.

L'intergiciel de sensibilité au contexte, coeur de notre proposition, prend en charge l'évaluation de l'environnement d'exécution et la commande d'adaptation pour les sessions de services numériques. Les principes employés pour sa conception sont présentés dans la suite de ce chapitre.

9.2 Principes de conception

L'architecture de l'intergiciel de sensibilité au contexte emprunte les recommandations sur la conception d'applications sensibles au contexte. Elles sont issues des travaux existants dans les domaines des applications autoadaptatives et des applications sensibles au contexte. Nous avons retenu les éléments suivants :

- La boucle autonome.
- La séparation des logiques.
- L'externalisation sous la forme d'une infrastructure.

9.2.1 La boucle autonome

Un service numérique sensible au contexte réalise le travail d'adaptation autour d'un cycle composé de quatre étapes :

1. Mesurer : récolter et consolider les informations qui composent le contexte.
2. Contrôler : identifier la situation d'usage du service numérique avec les situations possibles (définies par les concepteurs).
3. Décider : proposer la modalité d'exécution correspondante à la situation.
4. Adapter : moduler la logique métier.

Ces étapes sont liées sous la forme d'une boucle que nous appelons la boucle MCDA (Mesure, Control, Decide, Action) (Figure 9.2).

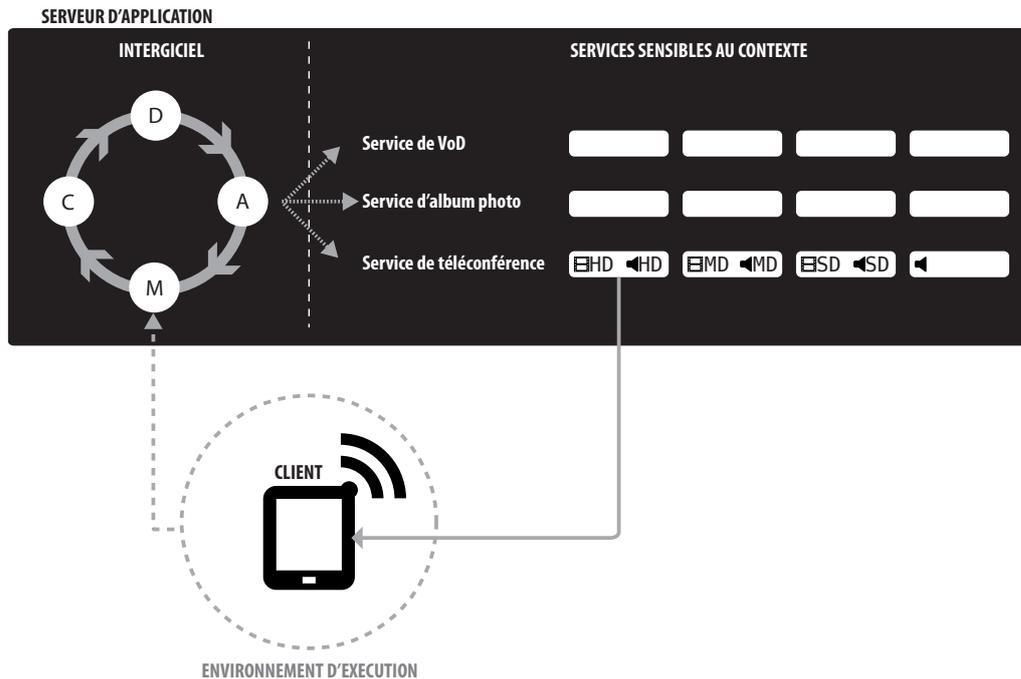


FIGURE 9.2: Boucle MCDA de l'intergiciel et services numériques sensibles au contexte au sein d'un serveur d'applications [10]

Le traitement effectué par chaque itération de ce cycle peut être modélisé sous la forme d'une architecture en couches. (Figure 9.3) Une *représentation en couches* des différentes étapes de ce cycle met en avant les différents niveaux d'abstraction qui permettent la transformation de données brutes issues d'un capteur en une décision d'adaptation pour un service numérique.

Capture et collecte

La première étape concerne la mesure de l'environnement.

C'est d'abord par une *capture* des données sur l'environnement d'exécution que commence cette étape. La couche de capture a la capacité de collecter l'ensemble des informations sur l'environnement d'exécution des services numériques sensibles au contexte. Ces informations, qui sont relevées par des capteurs, sont mises à disposition par ce que nous appelons des sources de contexte.

Associée à cette couche de capture, nous retrouvons, la *couche de collecte*. Cette couche ne conserve que les informations pertinentes pour l'évaluation de l'environnement d'exécution des services numériques sensibles au contexte. Elle réalise un filtrage sur les

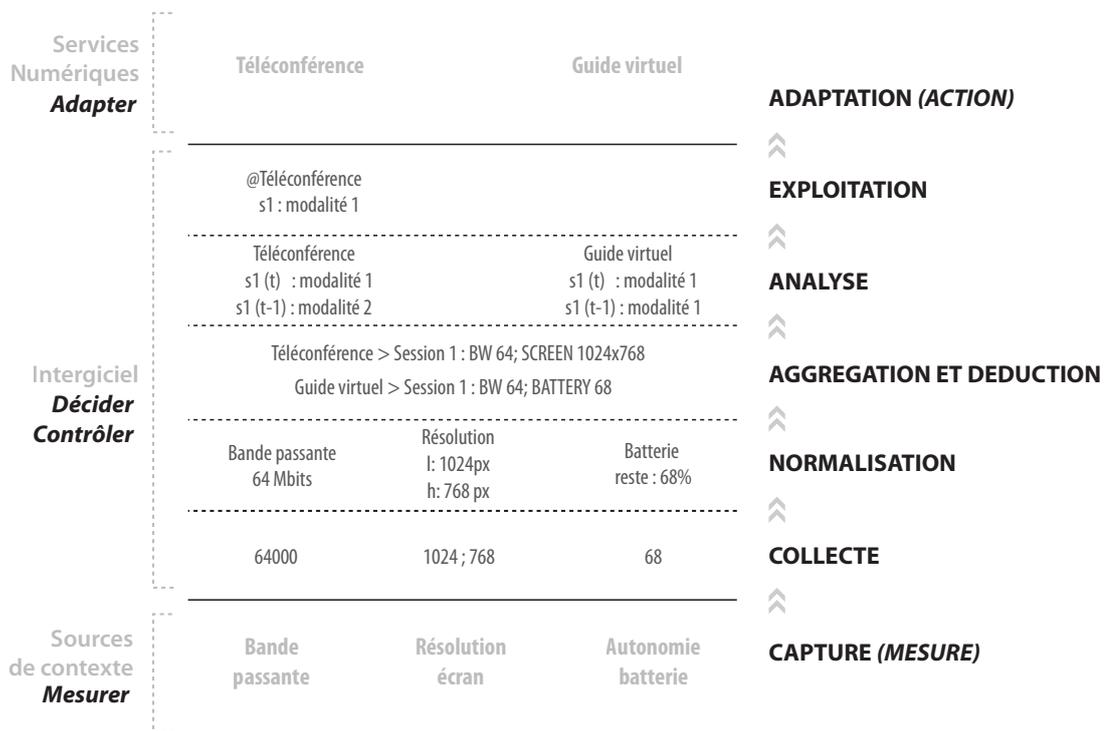


FIGURE 9.3: Schématisation de l'architecture sous la forme d'un modèle en couches

données en provenance de la couche de capture.

Normalisation

Au-dessus de la couche de collecte, nous trouvons la *couche de normalisation* des données. Son travail est la mise conformité des données pertinentes en provenance des différentes sources de contexte pour créer un ensemble homogène au sein de notre base de connaissances. Ce travail est réalisé en deux temps, d'abord une normalisation puis un enrichissement sémantique des données. C'est l'étape de transformation de la donnée brute en information de contexte.

Agrégation et déduction

La troisième couche de l'intergiciel concerne l'*agrégation* des informations de contexte et la *déduction* des contextes d'utilisation des services numériques. L'agrégation correspond à la combinaison des informations de contexte pour identifier la situation d'usage de chaque session ouverte. L'identification de la situation d'usage implique une déduction sur les modalités d'exécution possibles pour chacune des sessions. Cette étape permet la transformation des informations, qui composent l'environnement, en un contexte d'utilisation. De ce dernier seront ensuite déduites les modalités d'exécution.

Analyse

La *couche d'analyse* fait le lien entre l'état actuel des sessions des services numériques et les situations d'usage. En effet, la couche d'agrégation et de déduction n'a qu'une vocation de conseil par rapport aux observations immédiates. Elle n'a pas connaissance :

- de la modalité d'exécution la plus adaptée en terme de performance,
- de l'état des modalités applicatives utilisées par les services numériques,
- des précédentes préconisations.

Pour chaque session, une déduction est effectuée à l'aide de la base de connaissances pour permettre de choisir la modalité d'exécution la plus adaptée. Autrement dit, la couche d'analyse réalise un travail de filtrage sur les conclusions de la base de connaissances pour ne conserver que les informations utiles aux applications.

Exploitation et adaptation

La dernière couche de l'intergiciel est la *couche d'exploitation*. Elle transfère les informations jugées utiles aux services numériques. C'est elle qui assure l'interaction entre l'intergiciel de sensibilité au contexte et les services numériques adaptatifs. L'intergiciel préconise les modalités d'exécution adaptées pour chaque session des services numériques. Les services numériques réalisent les adaptations de la logique métier qui correspond aux situations d'usage identifiées par l'intergiciel. Chaque application négocie ensuite, avec l'instance client, la mise en place du changement de modalité.

9.2.2 Séparation des logiques

La représentation du traitement du contexte sous la forme de couches (Figure 9.3) s'articule autour de quatre étapes qui correspondent à trois logiques d'exécution :

- L'observation de l'environnement (Mesure) : les sources de contexte capturent les données sur l'environnement.
- L'identification de la situation d'usage (Contrôle) et la déduction d'adaptation (Décision) : l'intergiciel traite le contexte.
- L'adaptation du service numérique (Adaptation) : les services numériques délivrent la modalité d'exécution adaptée à la situation d'usage du client.

La répartition des fonctions d'un service numérique sensible au contexte sous la forme de trois blocs met en avant l'idée de séparation des logiques. Les trois logiques présentées ci-dessus sont prises en charge, respectivement, par trois entités : les sources de contexte, l'intergiciel et les services numériques adaptatifs.

Un service numérique sensible au contexte comme présenté dans le scénario de référence repose sur deux principes.

- Une logique de gestion du contexte (adaptation de l'application) qui regroupe les fonctions qui permettent l'*observation de l'environnement*, l'*identification de la situation d'usage* et la *déduction de l'action* à effectuer pour l'adaptation.
- Une logique métier de l'application (fonction principale) qui réalise le service numérique et ses modalités (p. ex. téléconférence entre deux utilisateurs).

Seule la **logique de gestion du contexte** est identique quel que soit le service numérique. Nous l'avons donc **mutualisé au sein du serveur d'applications** sous la forme d'un intergiciel. Cette externalisation permet de dégager les concepteurs de services numériques de la contrainte de réalisation des fonctions indispensables à une gestion du contexte (Mesure, contrôle et décision). L'adaptation fait partie intégrante du service numérique puisqu'elle reste spécifique à chacun.

En effet, les adaptations sont fortement liées à la logique métier du service numérique. La logique de gestion du contexte et la logique métier fonctionnent de pair pour offrir un service numérique sensible au contexte.

La logique de gestion du contexte est elle-même composée de deux logiques distinctes : l'observation de l'environnement d'exécution d'un côté ; l'analyse et la déduction de l'autre côté.

La problématique d'évaluation des performances du réseau et du terminal dans les réseaux du futur fait déjà l'objet de plusieurs publications scientifiques (p. ex. [59]). Par ailleurs, il nous semble que c'est un sujet à part entière dans les NGN puisqu'il faut une solution qui prenne en compte tous les types de terminaux susceptibles d'exister dans de tels réseaux. C'est pourquoi nous décidons de nous appuyer sur des solutions tiers pour l'observation de l'environnement. La séparation des fonctions de capture et des fonctions de traitement du contexte offre un modèle où la capture est réalisée au travers de fournisseurs de contexte. Un fournisseur de contexte est une solution tiers qui propose de mesurer différents paramètres de l'environnement informatique. Nous appelons ces paramètres, sources de contexte. Nous retrouvons ainsi une source de contexte pour la mesure de la bande passante, une autre pour la mesure de la résolution de l'écran, etc. À l'image d'un système informatique s'appuyant sur de multiples sources données, il faut une interface qui s'interconnecte avec chaque source de contexte.

Pour résumer, notre modèle est séparé sous la forme de trois logiques :

- Lecture du contexte : fournis par des solutions tiers.
- Analyse et déduction des situations d'usage : réalisée par l'intergiciel.
- Adaptation et réalisation du service métier : le service numérique.

Dans ce modèle, nous concentrons nos travaux sur les fonctions d'identification des situations d'usage et de déduction des modalités d'exécution pour chaque service numérique.

Signature de contexte

La signature de contexte est une des contributions de ces travaux de recherche. C'est un jeu de règles métier qui permettent à l'intergiciel d'identifier les situations d'usage de chaque service numérique et d'associer les modalités d'exécution. Il décrit le contexte du service numérique, c'est-à-dire les paramètres de l'environnement qui ont une influence sur son exécution. C'est grâce à cette information que l'intergiciel a la capacité d'observer le contexte des services numériques et de proposer la modalité d'exécution appropriée. En effet, nous retrouvons dans la signature de contexte les modalités d'exécution liées à chaque situation d'usage.

Chaque service numérique sensible au contexte est déployé avec sa signature de contexte. Elle fait le lien entre l'intergiciel et les services numériques ainsi qu'entre l'intergiciel et les sources de contexte. Autrement dit, dans notre modèle, elle est le point de jonction entre les trois logiques d'un service numérique sensible au contexte : la lecture du contexte, l'analyse/déduction et l'adaptation. La signature de contexte répond à une problématique de communication du contexte du service numérique à l'intergiciel. En effet, elle permet à l'intergiciel d'avoir connaissance :

- Du contexte du service numérique,
- Des situations d'usage du service numérique,
- Des modalités d'exécution liées aux situations d'usage.

Cette signature de contexte est créée par les concepteurs du service numérique. Elle est propre à chaque service numérique. Le service numérique comme la signature de contexte ne sauraient exister seuls, c'est pourquoi elle doit suivre le service numérique. Pour cela, elle est embarquée dans le service numérique sensible au contexte et communiquée à l'intergiciel lors du déploiement.

Communication entre l'intergiciel et les entités liées

Une contrainte importante est la nécessité de respecter un faible couplage, entre l'intergiciel, les services numériques et les solutions de monitoring, pour contraindre au minimum les concepteurs.

Autrement dit, nous devons permettre à un service numérique sensible au contexte d'indiquer :

- Aux sources de contexte : quelles sont les données pertinentes pour l'identification des situations d'usage.
- À l'intergiciel : quelles sont les situations d'usage et les modalités d'exécution liées.

L'intergiciel doit être en mesure, au cours de l'exécution du service numérique, de notifier chaque service numérique des changements de modalité à réaliser pour chaque session ouverte.

Dans ce dernier cas, pour conserver un couplage faible, un lien virtuel est réalisé en utilisant le paradigme MOM (Message Oriented Middleware). Ce dernier est parfois utilisé pour intégrer des applications métier ou modules d'applications fonctionnant sur des plateformes hétérogènes.

Dans notre cas, l'intergiciel et les applications communiquent au travers d'un service de messagerie, présent dans le serveur d'applications, sous le mode *publish/subscribe*. Nous distinguons deux types de messages susceptibles d'être échangés entre l'intergiciel et les services numériques :

- Les ordres ou commandes d'adaptation en provenance de la logique de gestion du contexte et à destination des services numériques adaptatifs. Ces messages sont issus de la nécessité d'adapter une session.
- Les notifications en provenance des services numériques et à destination de l'intergiciel. Ces notifications ont lieu lors d'une nouvelle session ou bien lorsqu'une session prend fin.

Le principe *publish/subscribe* (Publication/Abonnement) propose d'avoir d'un côté un émetteur de message, de l'autre des abonnés à des *sujets* (topics). Les émetteurs placent un message dans une file d'attente correspondant à un sujet. Les abonnés ont alors accès aux messages qui les concernent en raison de leur abonnement préalable.

Nous proposons deux types de sujets, ceux à destination de l'intergiciel et ceux qui concernent les services numériques. Dans le premier cas, les messages sont des notifications d'ouverture ou de fermeture de session. Dans le second cas, ce sont des ordres d'adaptation pour les sessions.

9.2.3 Externalisation sous la forme d'une infrastructure

Nous présentons la logique d'adaptation, nécessaire aux services numériques sensibles au contexte, sous la forme d'un service logiciel présent dans le serveur d'application. Ce choix fait écho aux préconisations de Hong et coll. présente dans l'article [33] qui recommande d'externaliser la gestion du contexte des applications sensibles au contexte sous la forme d'une infrastructure. Les fonctions de gestion du contexte sont dès lors accessibles de manière centralisée pour plusieurs applications. Elles ne sont pas embarquées dans chaque application. L'utilisation d'un point de mutualisation pour la gestion du contexte permet de favoriser :

- La maintenance et l'évolution de la logique de gestion du contexte,
- La gestion des accès concurrents à une ou plusieurs ressources.

Cet article de Hong et coll. attire l'attention sur les problèmes de passage à l'échelle lors du choix d'un mode de réalisation sous la forme d'une infrastructure et plus particulièrement sur la capacité du système à absorber la demande d'accès à l'infrastructure de gestion du contexte par les applications. Le passage par un point central pose un

problème pour l'accès concurrent à plusieurs ressources. Dans cette situation, le goulot d'étranglement se situe sur le point de mutualisation. Dans notre cas, ce goulot se situe au niveau de l'intergiciel de gestion du contexte. Il est indispensable que le système supporte la charge d'accès créée par les services numériques. Cette problématique est relative à la répartition de notre système sensible au contexte au sein d'un réseau du futur. Pour notre proposition, nous estimons qu'il est plus simple d'un point de vue de la conception, de limiter l'intergiciel de sensibilité au contexte à un seul serveur d'applications puisque ce sont ces derniers qui hébergent les services numériques. Autrement dit, nous proposons un intergiciel de sensibilité au contexte par serveur d'applications.

Ainsi, l'intergiciel prend en compte chacun des services numériques sensibles au contexte qui sont hébergés par le serveur. Il peut dès lors être perçu par les applications comme un service mis à leur disposition, à l'instar de la journalisation, l'authentification, etc. Cette incarnation, sous la forme d'un médiateur d'accès au contexte rattaché au serveur d'applications, nous semble être la meilleure représentation de l'externalisation de la logique de contexte. En effet, ce choix permet de répondre à deux des problèmes induits par cette architecture (infrastructure) :

- La sécurité d'accès aux données, car l'accès est limité aux applications du serveur d'applications.
- Le passage à l'échelle, car le serveur d'applications limite le nombre d'applications prises en charge par l'intergiciel.

9.3 Conclusion

Un service numérique sensible au contexte s'appuie sur un cycle composé de quatre étapes que nous regroupons sous trois logiques d'exécution :

- L'observation de l'environnement (Mesure).
- L'analyse de l'environnement et la déduction des situations d'usage (Contrôle et Décision)
- L'adaptation du service numérique (Adaptation)

Nous répartissons ces trois logiques entre trois entités (Figure 9.4) :

- L'observation est prise en charge par des solutions de "monitoring".
- La décision d'adaptation est proposée par l'intergiciel.
- L'adaptation est réalisée par le service numérique.

Le coeur de notre proposition, autour de l'adaptation des services numériques à leur environnement, repose sur l'intergiciel de sensibilité au contexte qui prend en charge les fonctions de décision d'adaptation. En effet, la phase d'observation est une problématique largement adressée dans les réseaux et l'adaptation est propre à chaque service numérique.

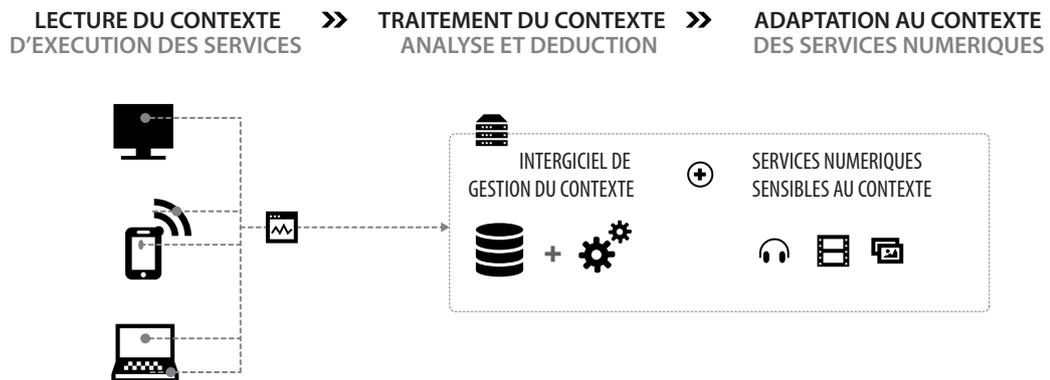


FIGURE 9.4: Répartition des fonctions d'adaptation

Il est indispensable que les concepteurs des services numériques puissent définir les situations d'usage des services numériques et les adaptations liées (modalités d'exécution). Nous regroupons une description des situations d'usages et les modalités liées dans ce que nous définissons comme la *signature de contexte*. Pour rappel, elle permet à l'intergiciel d'avoir connaissance des éléments caractéristiques pour l'observation de l'environnement pour permettre l'identification des situations d'usage et la déduction des modalités d'exécution appropriées.

Ainsi l'intergiciel est en mesure, pour chaque session de chaque service numérique, de proposer les adaptations nécessaires en regard de la situation de l'utilisateur.

Pour conclure, un service numérique sensible au contexte est composé :

- D'une signature de contexte,
- D'une logique métier,
- De plusieurs modalités de réalisation de ces logiques métiers.

La signature de contexte est utilisée par l'intergiciel pour lire l'environnement du client et déduire la modalité d'exécution.

L'approche retenue est la mise en place, par le concepteur du service numérique, de plusieurs logiques métier qui correspondent chacune à une ou plusieurs situations d'usage. Le coeur de métier de l'application reste le même, mais sa forme de consommation sur le terminal client est différente en fonction de la situation. Nous utilisons le terme *modalité d'exécution* pour caractériser ce polymorphisme de services. Ces situations d'usage sont identifiées grâce à la *signature de contexte*. Cette dernière est réalisée par le concepteur du service numérique. Puis elle est exploitée par l'intergiciel pour analyser l'environnement et ensuite déduire les situations d'usage.

Le chapitre suivant présente la composition fonctionnelle de notre intergiciel de sensibilité au contexte.

Chapitre 10

Les blocs fonctionnels

La fonction principale de notre intergiciel est le traitement du contexte. Cette opération permet, à partir d'observations sur l'environnement, d'identifier les situations d'usage et de proposer la modalité d'exécution adaptée pour chaque session. Cette transformation est réalisée par ce que nous appelons la **chaîne de traitement du contexte** (Figure 10.1).

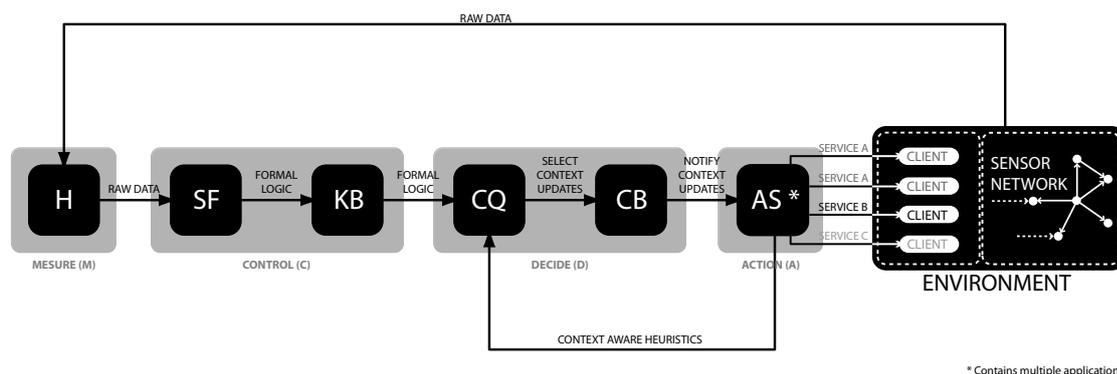


FIGURE 10.1: Chaîne de traitement du contexte

Elle met en jeu l'ensemble des modules présents dans l'integiciel. La première étape est l'acquisition des données issues des capteurs. Ce travail est réalisé par le module *Harvester*. Il confie ensuite les données au module *Semantic Formalizer* qui les enrichit sémantiquement. Cette transformation permet de formaliser les données pour leur insertion dans la base de connaissances. Le bloc fonctionnel *Knowledge Base* agrège l'ensemble des informations avec l'aide des connaissances du système, qui sont les signatures de contexte, et les connaissances sur l'état des services (nombre de sessions ouvertes). Avec l'ensemble des connaissances, il déduit les modalités d'exécution possibles pour chaque session de chaque service numérique. Le *Context Querier* interroge régulièrement la base de connaissances et il extrait les déductions sur les modalités d'exécution

de chaque session. C'est lui qui va choisir la modalité d'exécution la plus adaptée pour chaque session. Le choix de la modalité d'exécution de chaque session est communiqué aux services numériques par l'intermédiaire du *Context Broker* sous la forme d'une commande. L'application de cette commande entraîne une modification du code d'exécution de la session concernée du service numérique. Cette modification du code est laissée à la discrétion des concepteurs de services numériques.

Nous nous attachons, maintenant, à décrire un à un les cinq blocs dans l'ordre suivant :

- Knowledge Base (KB) - Base de connaissances
- Harvester (H) - Collecteur
- Context Broker (CB) - Courtier de contexte
- Semantic Formalize (SF) - Adaptateur sémantique
- Context Querier (CQ) - Interrogateur de contexte

L'ordre de présentation est volontairement différent de celui de la Figure 10.1 pour faciliter la compréhension du module. L'enchaînement des différents blocs sous la forme de la chaîne de traitement du contexte est présenté dans le section 11.4.

10.1 Knowledge Base (Base de connaissances)

La base de connaissances est l'élément central de l'architecture de l'intergiciel. Elle contient les connaissances sur les services numériques sensibles au contexte à l'instant présent.

Ces connaissances ont une durée de validité limitée dans le temps. Nous définissons trois catégories d'informations qui ont chacune une validité temporelle différente :

- Le contexte d'exécution des services numériques côté client (caractéristiques du réseau et du terminal utilisé). Leur durée de validité est très courte, de l'ordre de quelques secondes. Ce choix dépend de l'intervalle d'interrogation des sources de contexte par l'intergiciel.
- Les sessions ouvertes sur les services numériques. Leur durée de validité correspond au temps d'exécution d'une session.
- Les services numériques (modalités d'exécution, situations d'usage). Leur durée de validité correspond au temps de déploiement du service numérique concerné.

Le premier groupe d'informations caractérise l'environnement d'exécution du service numérique. Ces informations sont issues des capteurs (présents sur les terminaux, dans le réseau, etc.) qui récupèrent les caractéristiques de l'environnement. Elles sont rapatriées grâce aux solutions de monitoring. Leur durée de vie est limitée dans le temps en fonction du paramétrage de l'échantillonnage. C'est une image instantanée de l'environnement d'exécution des sessions du service numérique.

Le second groupe d'informations est relatif aux sessions actives des services numériques adaptatifs. Chaque session de chaque service numérique est représentée dans la base de connaissances. On retrouve les informations tels que son identifiant au sein du réseau du futur et la modalité d'exécution de la logique métier. Ces données sont fournies par le service numérique lors de l'ouverture d'une session (consommation du service). C'est un instantané de l'exécution du service numérique.

Le troisième groupe d'informations concerne les services numériques sensibles au contexte déployés sur le serveur d'applications. On retrouve les services numériques hébergés avec leurs situations d'usage potentielles et leurs modalités d'exécution correspondantes (éléments de la signature de contexte). Ces connaissances sont formalisées par les concepteurs des services numériques et elles sont communiquées par le service numérique lors de son déploiement.

La suite de cette section présente les deux éléments qui composent ce module :

- La base de connaissances sous la forme d'une ontologie,
- Le moteur d'inférence.

10.1.1 Une ontologie d'application

La base de connaissances existe sous la forme d'une ontologie informatique. Ce choix repose sur plusieurs éléments :

- La nécessité de modéliser et formaliser notre environnement d'une façon intelligible par l'humain et les machines : notre système fait intervenir des données définies par les concepteurs de service numérique, par le serveur d'applications et par des sources logicielles externes.
- L'utilisation d'un raisonnement par déduction logique pour identifier l'environnement d'exécution : déduire la situation d'usage de chaque session du service numérique et proposer la modalité d'exécution correspondante.

La structure de la base de connaissances dépend des différentes informations destinées à être hébergées et de leurs relations entre elles. Dans ce travail, les informations sont spécifiques. Elles mènent à la création d'une *ontologie d'application*. Pour nous aider dans ce travail de conception, nous avons résumé nos observations sur les éléments à inclure et leurs interactions sous la forme d'une liste :

- Un serveur d'applications héberge plusieurs services numériques adaptatifs. Cette observation nous a mené au terme "context-aware application server" (CAAS) pour définir les serveurs hébergeant qui hébergent des services numériques sensibles au contexte.
- Chaque service numérique propose plusieurs modalités de fonctionnement.
- Le choix de la modalité est déterminé par le contexte d'exécution.
- Une session ne peut appliquer qu'une seule et unique modalité en même temps.

- Une session délivre un ou plusieurs flux à son instance cliente, à travers un environnement que nous appelons le contexte informatique.
- Le contexte informatique est obtenu par l'intermédiaire de capteurs.
- Les capteurs sont pilotés par des solutions de monitoring, que nous appelons fournisseurs de contexte
- Nous appelons sources de contexte un ensemble de capteurs qui fournissent la même information.

Les sources et les fournisseurs de contexte sont inconnus lors du déploiement de l'intergiciel de sensibilité au contexte. Ils sont ajoutés/supprimés en fonction des besoins.

Le résultat de ce travail de conception est une ontologie composée de deux classes principales : Context Provider et Session Figure 10.2.

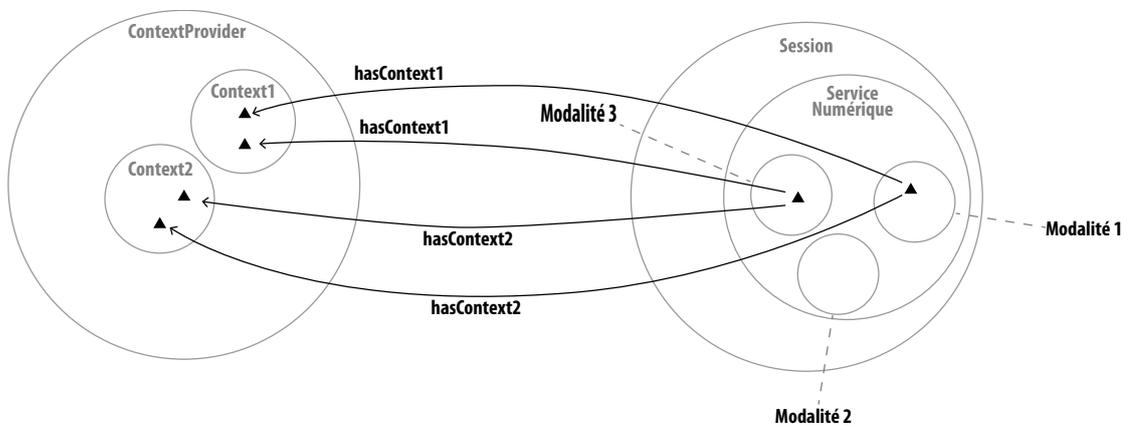


FIGURE 10.2: Schéma de l'ontologie

Dans notre ontologie (Figure 10.2), l'ensemble de gauche (Context Provider) regroupe les informations liées à l'environnement de l'utilisateur, cela concerne la partie client du service numérique. L'ensemble de droite regroupe les informations liées au serveur d'applications, cela concerne la partie serveur du service numérique.

L'ensemble *Context Provider* est composé d'autant de sous-ensembles qu'il existe de sources de contextes. Une source de contexte est une catégorie d'information, par exemple la bande passante ou la résolution de l'écran. Elle est ajoutée manuellement par l'administrateur du serveur d'applications. Lors de cet ajout, un sous-ensemble est créé dans *Context Provider*. Ces sous-ensembles contiennent les données des capteurs qui sont exploitées pour déduire le contexte. Autrement dit, chaque capteur est représenté par un individu qui appartient à une source de contexte. Il contient dans ses propriétés la valeur courante du capteur. Si nous reprenons notre scénario de référence, nous disposons de deux fournisseurs de contextes : bande passante et taille d'écran. Nous trouverons donc deux individus dans chaque sous-ensemble cité, un pour chaque session (la session de Bob et la session d'Alice). L'évolution de l'ontologie, basée sur le scénario de référence,

est présentée plus loin dans ce manuscrit (chapitre 15).

L'ensemble *Session* regroupe les services numériques sensibles au contexte déployés sur le serveur ainsi que leurs sessions. Chaque service numérique est représenté par un sous-ensemble de *Session*. Il est lui-même composé de sous-ensembles qui représentent les modalités d'exécution du service numérique, c'est-à-dire les adaptations du service numérique. Dans notre scénario de référence, nous retrouvons donc un sous-ensemble *téléconférence* qui contient quatre sous-ensembles (AudioVideoHD, AudioVideoMD, AudioVideoFD, AudioSeul). Chaque session ouverte est présente dans l'ontologie sous la forme d'un individu qui appartient au sous-ensemble de *Session* correspondant au service numérique concerné. Nous retrouvons cet individu dans différents sous-ensembles relatifs aux modalités d'exécution en fonction de son état de fonctionnement.

10.1.2 Le moteur d'inférence

L'ensemble des informations de la base de connaissances permet, par déduction, de définir les modalités d'exécution possible pour chacune des sessions, du service numérique, ouvertes sur le serveur d'applications. Ce mécanisme de déduction est réalisé par l'intermédiaire du moteur d'inférence et de règles métiers. Un classement est réalisé, dans la base de connaissances, par le changement d'appartenance des individus aux sous-ensembles qui représentent les modalités d'exécution.

Ce changement d'appartenance des individus aux sous-ensembles est dicté par des règles métier qui sont spécifiques à chaque service numérique. Elles identifient les situations d'usage qui sont spécifiées en terme de capteurs et valeurs seuils. Et pour chaque situation d'usage, nous retrouvons une modalité d'exécution liée. Une même modalité d'exécution peut-être associée à plusieurs situations d'usage. Ce jeu de règles métier que nous appelons *signature de contexte* est propre à chaque service numérique. Il est intégré à la base de connaissances lors du déploiement de celui-ci.

Chaque modalité d'exécution correspond à une modulation de la logique métier, du service numérique sensible au contexte, qui réalise l'adaptation.

Si nous reprenons notre scénario de référence, nous retrouvons quatre règles métier qui permettent d'identifier, chacune, une situation d'usage et une modalité correspondante (Tableau 2.1). Les règles qui composent la signature de contexte sont présentées plus loin dans ce manuscrit (chapitre 15).

10.2 Harvester (Collecteur)

Ce module joue le rôle de passerelle entre les sources de contexte et l'intergiciel. Il incarne la *couche de collecte* présentée précédemment puisqu'il récupère les données pertinentes pour identifier l'environnement d'exécution. Nous distinguons deux mécanismes pour l'acquisition des données sur l'environnement :

- Une collecte des données auprès des sources de contexte.
- Un filtrage des données envoyées par les sources de contexte.

Le premier mécanisme est une sollicitation des sources de contexte à intervalle régulier sous la forme d'une requête question-réponse tout au long de la session de l'utilisateur. Le second mécanisme est un abonnement à la source de contexte pour la durée de la session.

Dans notre scénario de référence, ce module récupère les informations sur la bande passante disponible et la résolution de l'écran du terminal de Bob et d'Alice. Les autres informations relatives à l'environnement d'exécution de Bob et d'Alice sont laissées de côté. Elles ne sont donc pas collectées par le *Harvester*.

Pour que l'opération de filtrage soit possible il faut que ce module ait connaissance :

1. Des sources de contexte accessibles pour récolter les données sur l'environnement.
2. Parmi ces dernières, de celles qui sont pertinentes pour identifier le contexte d'utilisation de chaque session ouverte.

Pour cela, l'administrateur du système déclare les sources de contexte dans l'intergiciel. Il ajoute le type d'information accessible (résolution des écrans, bande passante disponible, etc.) et la méthode d'accès (collecte ou abonnement) puis configure l'accès. À partir des sources de contexte déclarées, les concepteurs des services numériques sensibles au contexte créent les règles métier qui constituent la *signature de contexte*.

La déclaration de la source de contexte est une étape indispensable à tout déploiement de service numérique sensible au contexte. Elle permet d'identifier le type d'information de contexte (bande passante, résolution d'écran, etc.) que chaque source de contexte est en mesure de fournir et la façon de solliciter les capteurs. C'est une déclaration des sources de contexte, accessibles par l'intergiciel de sensibilité au contexte, pour adapter les services numériques. À partir de ces sources de contexte, les concepteurs de services numériques définissent les éléments à surveiller pour identifier l'environnement d'exécution et adapter le service numérique à la bonne modalité d'exécution.

Pour conclure, le rôle du *Harvester* est de solliciter les sources de données pour récupérer les informations nécessaires à l'identification de l'environnement d'exécution.

10.3 Context Broker (Courtier de contexte)

Le module *Context Broker*, traduit littéralement par courtier de contexte, a une mission de routage des informations relatives au contexte qui transitent entre l'intergiciel et les services numériques. Plus globalement, il joue le rôle de passerelle entre l'intergiciel et :

- les services numériques sensibles au contexte.
- les interfaces de configuration de l’intergiciel.

Il est piloté par des événements que nous classons sous trois catégories :

- Ouverture et fermeture de session. Cet événement est généré par les services numériques. Il informe l’intergiciel d’une ouverture ou d’une fermeture de session.
- Changement de modalité d’exécution. Cet événement est généré par l’intergiciel. Il informe les services numériques de la nécessité de changer de modalité d’exécution pour une ou plusieurs sessions.
- Ajout et suppression d’un service numérique au sein du serveur d’applications. Cet événement est généré par l’intergiciel pour lui-même. Il s’agit de prendre en main le déploiement d’un nouveau service numérique ou bien son retrait.

Pour résumer, le *Context Broker* joue le rôle de routeur de messages entre les services numériques et l’intergiciel.

10.4 Semantic Formalizer (Formateur sémantique)

Le *Semantic Formalizer* prend en charge toutes les opérations d’insertion dans la base de connaissances et assure ainsi l’intégrité de celle-ci grâce à deux mécanismes :

- Une mise en conformité du format des données avant leur insertion.
- Un contrôle de conflit avec les données déjà présentes et celles qui vont être insérées.

Nous distinguons deux provenances pour les données à destination de la base de connaissances :

- Les solutions de monitoring à travers le module *Harvester*,
- Les services numériques et les opérations de maintenance à travers le module *Context Broker*.

Les solutions de monitoring fournissent les données relatives au contexte d’exécution des services numériques. Leur mise en conformité est réalisée par l’enrichissement sémantique des données brutes collectées par les capteurs. Cette transformation est réalisée en indiquant le type de données récolté ainsi que la session à laquelle la donnée appartient. Ce travail est indispensable pour une insertion dans la base de connaissances puisqu’il indique le sous-ensemble d’appartenance et la relation avec la session ouverte. Pour éviter un conflit avec les données déjà présentes, ce module prend en charge l’opération de remplacement des informations sur le contexte dans la base de connaissances. Ceci mène à une suppression des propriétés existantes de l’individu représentant la source de contexte.

Les services numériques alimentent la base de connaissances avec des données relatives aux sessions, c’est-à-dire la création ou suppression des individus lors de l’ouverture

ou la fermeture d'une session sur un service numérique. Ce module prend en charge la création de l'individu dans le sous-ensemble *Session* correspondant au service numérique et des individus dans les sous-ensembles *Context*. Il met en place, dans l'ontologie, l'ensemble des individus et leurs relations, lors de l'ouverture d'une session. Ce schéma d'instanciation est déduit des règles qui composent la signature de contexte. De même, lors de la fermeture d'une session, il supprime ces individus.

Un travail de coordination est effectué entre le *Semantic Formalizer* et le *Harvester*. En effet, dès qu'une session est ouverte ou fermée le *Semantic Formalizer* a le devoir d'avertir le *Harvester* pour que ce dernier démarre ou arrête la collecte des données.

10.5 Context Querier (Interrogateur de contexte)

Le *Context Querier* prend en charge les opérations d'extraction de la base de connaissances. C'est l'interface entre les modules *Knowledge Base* et *Context Broker*.

Il extrait de la base de connaissances les modalités d'exécution possibles pour chaque session de chaque service numérique pris en charge. En effet, la base de connaissances, couplée au moteur d'inférence, déduit du contexte d'exécution les modalités d'exécution auxquelles les sessions sont éligibles. Pour chaque session nous retrouvons, ainsi, une ou plusieurs modalités d'exécution. C'est le *Context Querier* qui choisit celle qui est la plus adaptée aux performances proposées par l'environnement de la session. Il réalise ce choix en croisant les modalités d'exécution éligibles et la dégradation possible du service numérique. Il compare ensuite cette décision avec la modalité d'exécution en cours d'utilisation. Si un changement de modalité est identifié par le *Context Querier*, il émet alors un ordre d'adaptation à destination du service numérique sensible au contexte concerné. Le *Context Querier* réalise ce travail après chaque inférence de la base de connaissances.

10.6 Conclusion

Ce chapitre présente les 5 blocs qui composent notre intergiciel :

- Harvester (Collecteur) : il collecte les données sur le contexte d'exécution des services numériques.
- Semantic Formalizer (Adapteur sémantique) : il enrichit sémantiquement les données sur l'environnement.
- Knowledge Base (Base de connaissances et moteur d'inférence) : il agrège les données pour les transformer en connaissances et déduit les situations d'usage des services numériques.
- Context Querier (Interrogateur de contexte) : il détecte les changements dans les situations d'usage des services numériques.

- Context Broker (Convoyeur de contexte) : il informe les services numériques du changement de modalité d'exécution si nécessaire.

Le chapitre suivant présente les fonctions générales de l'intergiciel autour des 5 blocs fonctionnels.

Chapitre 11

Les fonctions de l'intergiciel de sensibilité au contexte

Le traitement du contexte est un des modes de fonctionnement de l'intergiciel [8]. Nous retrouvons d'autres modes d'opérations tels que le déploiement de services numériques ou l'ajout de sources de contexte.

11.1 Introduction

Notre système de sensibilité au contexte s'insère dans le monde des NGN. Il est composé d'un serveur d'applications, d'un réseau, d'un terminal et d'un système de monitoring (Figure 11.1).

Nous plaçons ces travaux dans les NGN et nous nous appuyons sur le modèle IMS. Une application IMS standard est composée d'une instance dite "locale" (terminal) et d'une instance distante (serveur d'applications). Les deux instances échangent des flux de gestion qui concernent la session en cours, ainsi que des flux multimédia (audio, vidéo et/ou données) relatifs à la logique métier. C'est ce que nous appelons dans ce manuscrit un service numérique. En conséquence, une application IMS sensible au contexte diffère du modèle standard par l'ajout de deux fonctions supplémentaires :

- La capture des données relatives à l'environnement : cette capture du contexte est réalisée par des solutions de *monitoring*.
- L'adaptation de la logique métier.

De tels services sont composés d'une logique d'adaptation et d'une logique métier. L'approche retenue est l'intégration de la sensibilité au contexte au travers d'un intergiciel. Il prend en charge l'ensemble des services numériques sensibles au contexte hébergés sur le même serveur d'applications (Figure 11.1). Ainsi, la sensibilité au contexte est rendue opérationnelle par l'intergiciel de sensibilité au contexte : c'est un activateur de

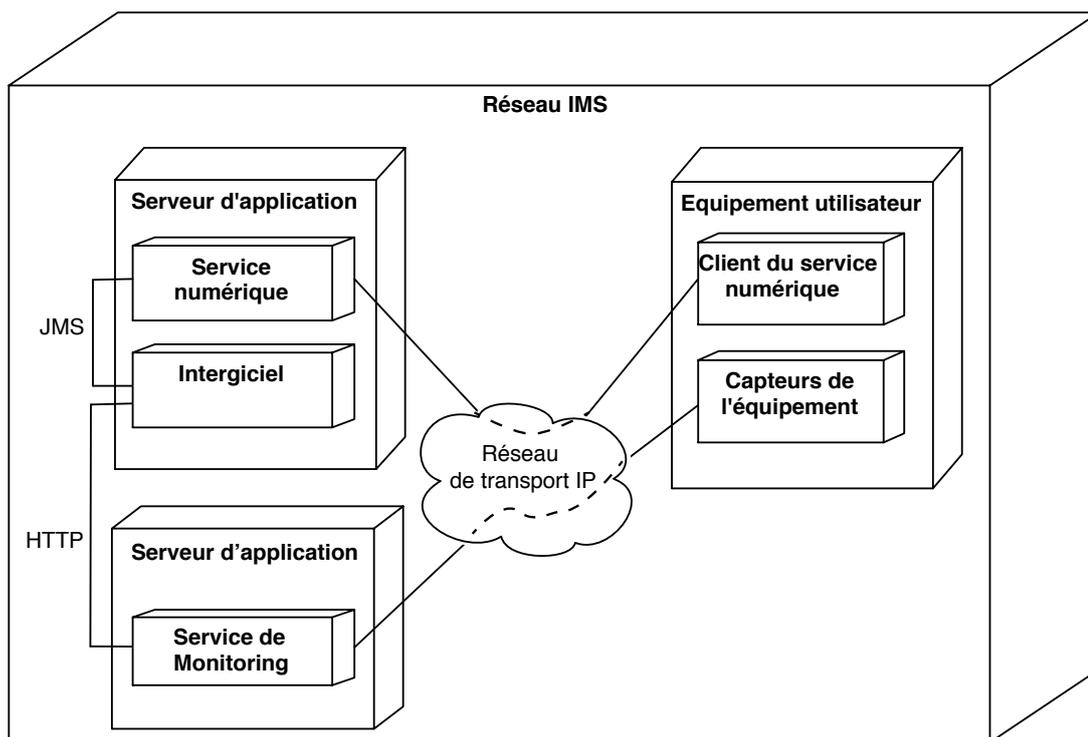


FIGURE 11.1: Modélisation UML : Diagramme de déploiement du système de sensibilité au contexte

contexte.

Le diagramme de cas d'utilisation général (Figure 11.2) met en avant les différentes interactions possibles avec l'intergiciel (Tableau 11.1). Les *solutions de monitoring* récupèrent les données sur l'environnement à l'aide de *capteurs*. L'*intergiciel* interroge à intervalles réguliers les *solutions de monitoring* pour collecter les données pertinentes. C'est une relation entre des agents logiciels qui est réalisée au travers du réseau. Grâce aux données collectées, l'*intergiciel* pilote les *services numériques* en proposant les adaptations pour chaque session ouverte par un utilisateur final. Ce travail est automatique et réalisé par l'échange de messages.

La configuration de l'intergiciel est réalisée par l'administrateur du système. Ce travail de configuration concerne particulièrement l'ajout de lien vers les sources de contexte.

Le service numérique est développé et déployé par le concepteur logiciel. Ce dernier a aussi à sa charge la création de la signature de contexte.

Les paragraphes précédents mettent en avant les différentes interactions possibles avec l'intergiciel. Nous classons en deux catégories : celles qui nécessitent une inter-

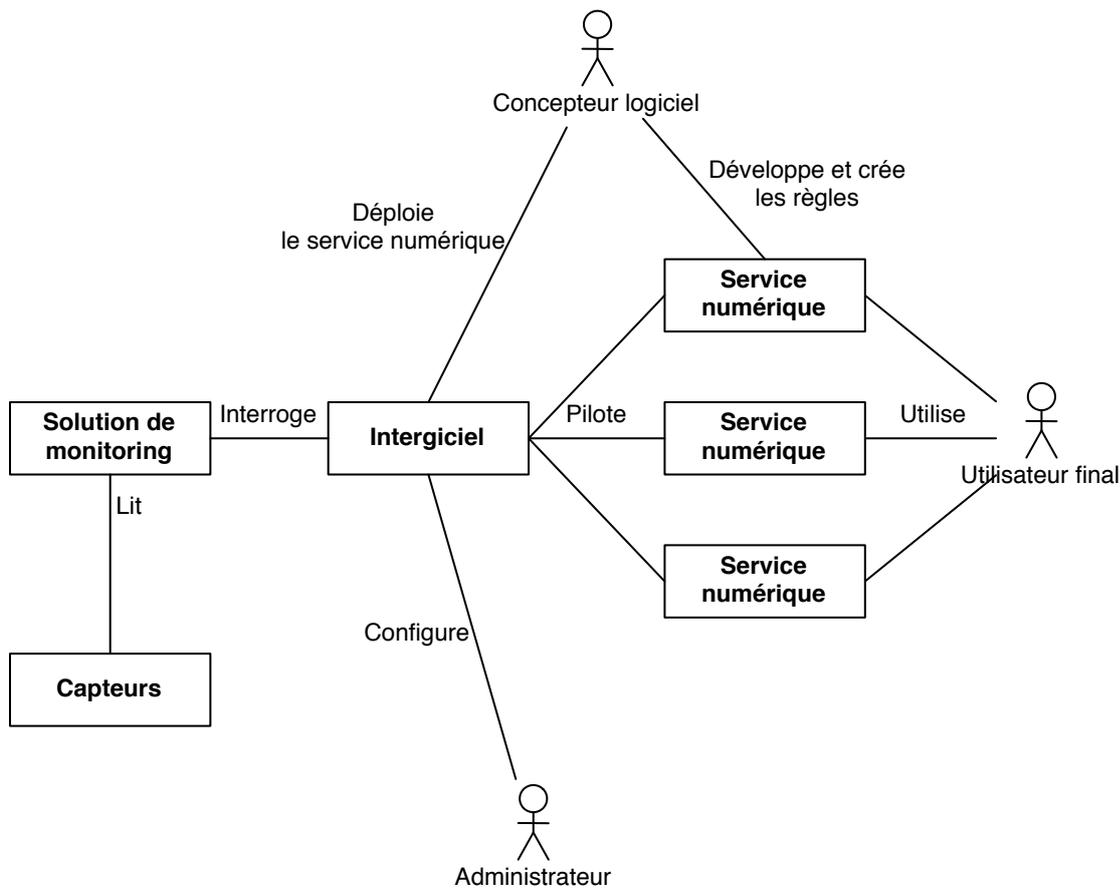


FIGURE 11.2: Diagramme de cas d'utilisation général

face graphique et correspondent donc à des interactions avec un utilisateur ; celles qui nécessitent une API pour interagir avec des applications ou composants logiciels.

TABLEAU 11.1: Interactions majeures entre l'intergiciel et son environnement

Acteur	Type	Action
Administrateur du système	Interface graphique	Configuration du serveur d'applicaiton
Développeur de service numérique	Interface graphique	
Application	API	Communication avec l'intergiciel de sensibilité au contexte
Outils de monitoring	API	Couche de capteurs

Toutes ces fonctions sont modélisées au travers des 5 blocs logiques qui sont présentés dans le chapitre 10 : *Harvester, Semantic Formalizer, Knowledge Base, Context Querier*

et *Context Broker*.

La suite de ce chapitre présente les interactions entre ces modules, au travers des différentes fonctions de l'intergiciel :

- Déclaration de sources de contexte pour la collecte des données sur l'environnement d'exécution en section 11.2
- Déploiement d'un service numérique en section 11.3
- Analyse de l'environnement pour un service numérique lors de la consommation d'un service numérique en section 11.4
- Suppression d'un service numérique en section 11.5

11.2 Déclaration d'une source de contexte

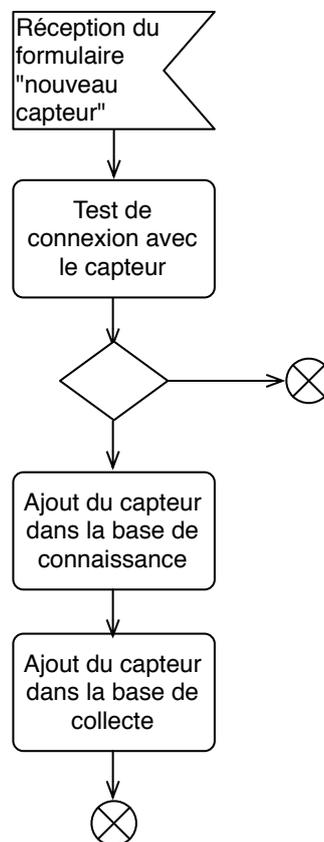


FIGURE 11.3: Déclaration d'une source de contexte au sein de l'intergiciel - Diagramme d'activité

L'intergiciel ne possède pas de fonction propre pour la capture de l'environnement.

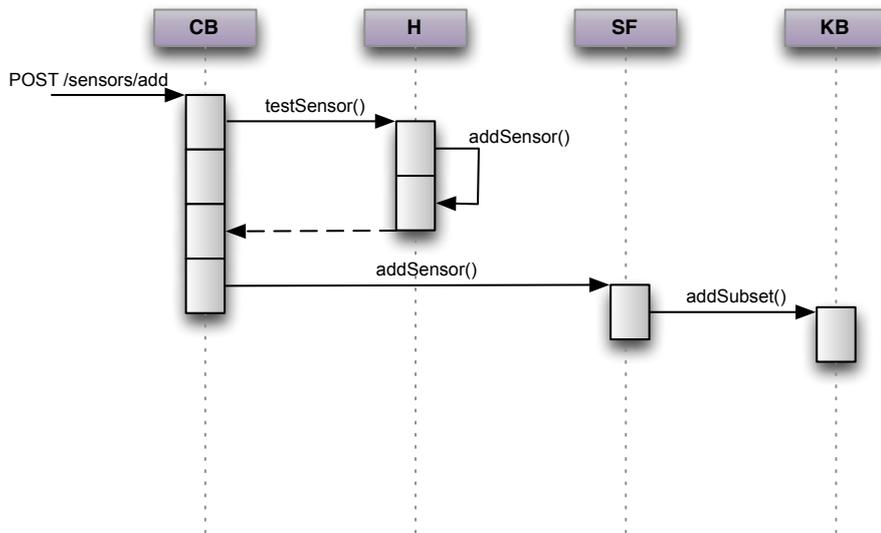


FIGURE 11.4: Déclaration d'une source de contexte au sein de l'intergiciel - Diagramme de séquence

Ce choix a été justifié dans le chapitre précédent. Pour avoir accès aux informations sur l'environnement il s'appuie sur des solutions tiers, dites solutions de monitoring. Chacune de ces solutions réalise la capture d'une partie de l'environnement dont l'accès aux données doit être connu par l'intergiciel.

Nous n'avons pas choisi une découverte automatique des capteurs, pour les mêmes raisons que celles évoquées pour l'intégration de notre propre couche de capture de l'environnement. Par conséquent, l'accès aux données se fait en deux temps :

- déclaration des sources de contexte accessibles par l'intergiciel de sensibilité au contexte.
- interrogation des sources de contexte.

Nous ne traiterons dans cette section que la première étape : la déclaration. Cette étape nécessite une intervention humaine, l'administrateur du serveur déclare une nouvelle source de contexte (Figure 11.3). Lors de cet ajout, le système vérifie la connexion avec la source de contexte qui fournit l'accès à cette nouvelle information. Dans le cas d'un accès réalisable à cette nouvelle information, le système déclare au module *Harvester* la présence d'une nouvelle source de contexte pour l'évaluation de l'environnement. Cette déclaration est aussi réalisée auprès du module *Knowledge Base* avec la création d'un nouveau sous-ensemble dans l'ensemble *ContextProvider*. C'est dans cet ensemble que nous retrouverons les individus qui identifieront les données de l'environnement à traiter.

Cette interaction administrateur-intergiciel est prise en charge par le module *Context Broker* (Figure 11.4).

La déclaration des sources de contexte est une étape indispensable à la mise place de l'intergiciel puisqu'elle définit l'accès aux capteurs qui permettent la capture de l'environnement d'exécution. Par ailleurs, puisque la signature de contexte identifie les situations d'usage elle est liée aux sources de contexte accessible. Pour qu'une signature de contexte soit validée par l'intergiciel de sensibilité au contexte, lors du déploiement d'un service numérique, il faut qu'elle se base sur les sources de contexte accessibles par l'intergiciel.

11.3 Déploiement d'un service numérique

Le déploiement d'un service numérique sensible au contexte est une étape clé de notre système. C'est lors de son déploiement que la séparation des logiques (adaptation et métier) est effectuée.

Traditionnellement avant son déploiement, un service numérique est empaqueté dans un conteneur. On retrouve le binaire du service numérique et des ressources associées, telles que des média, des librairies et/ou des fichiers de configuration. Ainsi le serveur d'applications reçoit un seul et unique document, le paquet de déploiement. Charge à lui de déplacer les fichiers (binaires, média, librairies, etc.) à l'endroit approprié. Ainsi, lorsqu'un service numérique est déployé sur un serveur d'applications, il n'est pas exécuté, mais simplement déposé sur ce dernier, avec ses ressources associées (Figure 11.5. Ce n'est que lors d'une connexion client qu'un processus représentant le service numérique est créé.

Notre choix de réaliser ce processus de déploiement par l'intergiciel de sensibilité au contexte permet de limiter les modifications à opérer sur le serveur d'applications. En effet, traditionnellement c'est le serveur d'applications qui prend en charge le déploiement des services numériques. Dans notre situation, la contrainte de cette opération de déploiement est d'intégrer la signature de contexte à l'intergiciel et la logique métier du service numérique au sein des conteneurs (EJB, SIP, etc.) du serveur d'applications. Pour bien comprendre la problématique, il faut imaginer l'intergiciel comme une application tierce déployée : le serveur d'applications n'a aucune autorité sur elle. Or le processus de déploiement d'une application est pris en charge par le serveur d'applications. Il devient donc très compliqué et pas forcément élégant :

- De modifier le processus de déploiement sur le serveur d'applications,
- De s'assurer du déploiement correct de la signature de contexte.

Pour pallier ces problèmes, nous proposons de prendre en charge le déploiement de l'application depuis l'intergiciel. Ceci permet d'extraire la signature de contexte du paquet contenant le service numérique sensible au contexte, pour ensuite l'injecter dans la base de connaissances.

Avant le déploiement

Après le déploiement

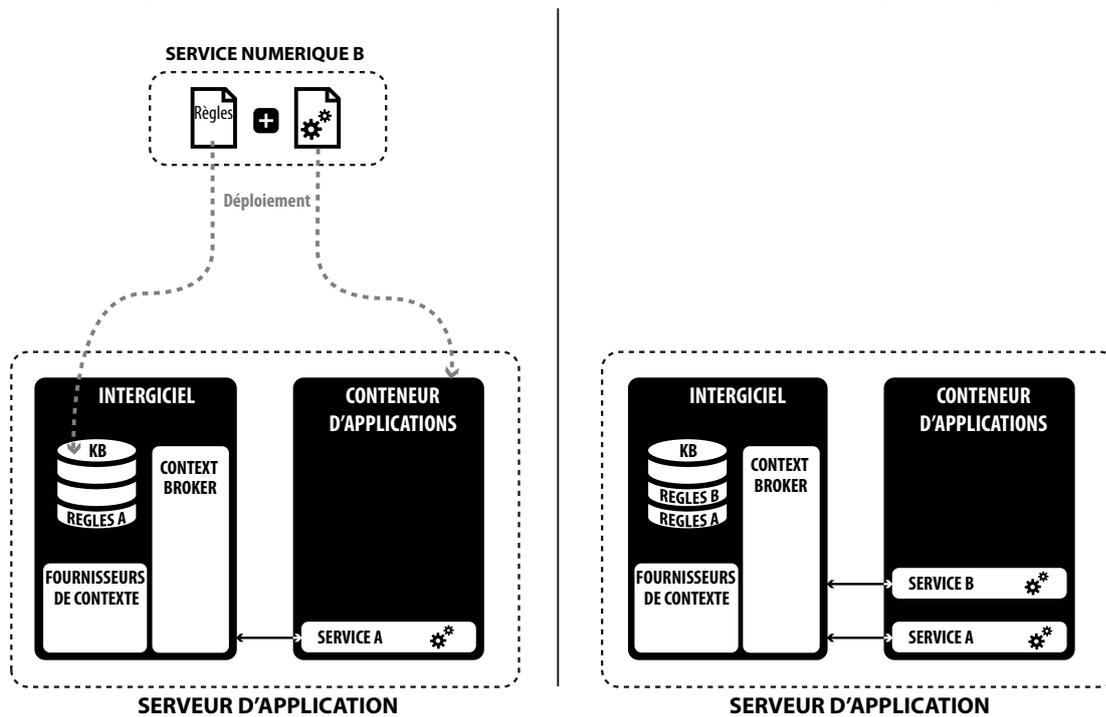


FIGURE 11.5: Déploiement de l'application et de la signature de contexte

Ainsi le concepteur fournit le service numérique sensible au contexte sous la forme d'un paquet. Ce paquet est ensuite déployé directement par l'intergiciel au travers d'une interface d'administration fournie par le *Context Broker*. Le déploiement du service numérique est réalisé en deux étapes : dans un premier temps la séparation des logiques, dans un second temps leur déploiement. Les éléments de la logique métier de l'application sont placés dans les conteneurs d'exécution. C'est-à-dire que les éléments binaires du service numérique sont eux confiés au serveur d'application en vue d'un déploiement standard. La logique d'adaptation (les règles de sensibilité au contexte) est prise en charge par l'intergiciel pour être injectée dans la base de connaissances du module *Knowledge Base* (Figure 11.6).

11.4 Analyse de l'environnement lors de la consommation d'un service numérique

Nous distinguons trois phases autour de la consommation d'un service numérique sensible au contexte : l'ouverture de session, son exécution et la terminaison.

La phase de consommation d'un service numérique a fait l'objet d'une modélisation

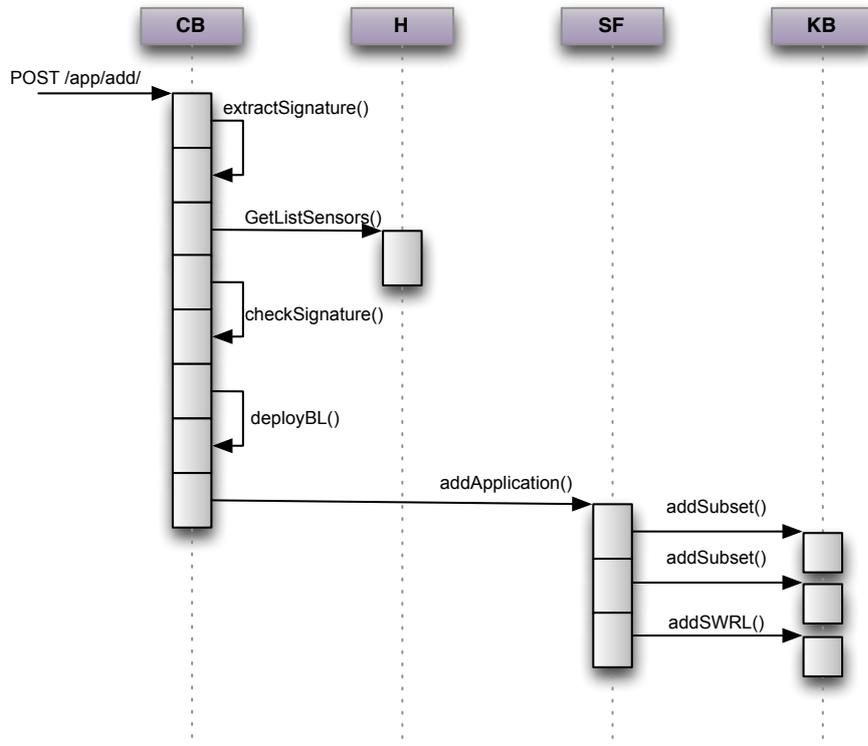


FIGURE 11.6: Déploiement d'une application - Diagramme de séquence

UML sous la forme d'un diagramme d'activité présentée en Figure 11.7.

11.4.1 Ouverture d'une session (service numérique)

L'ouverture de session se fait à partir du service numérique, selon la procédure standard d'IMS. Lors d'une ouverture de session par un client, le service numérique notifie l'intergiciel qu'une nouvelle connexion a lieu en envoyant un message au *Context Broker* qui joue le rôle de chef d'orchestre pour les autres modules de l'intergiciel (Figure 11.8). Il demande au *Semantic Formalizer* de créer, dans la base de connaissances, les individus qui représentent la session, ainsi que ceux qui représentent les sources de contexte à interroger. Il notifie ensuite le *Harvester* de l'existence d'une nouvelle session pour lancer le processus de collecte des données sur l'environnement d'exécution tel que défini plus haut.

De son côté, le service numérique se met en écoute des notifications envoyées par l'intergiciel. Ces notifications arrivent sous la forme de messages adressés au service numérique.

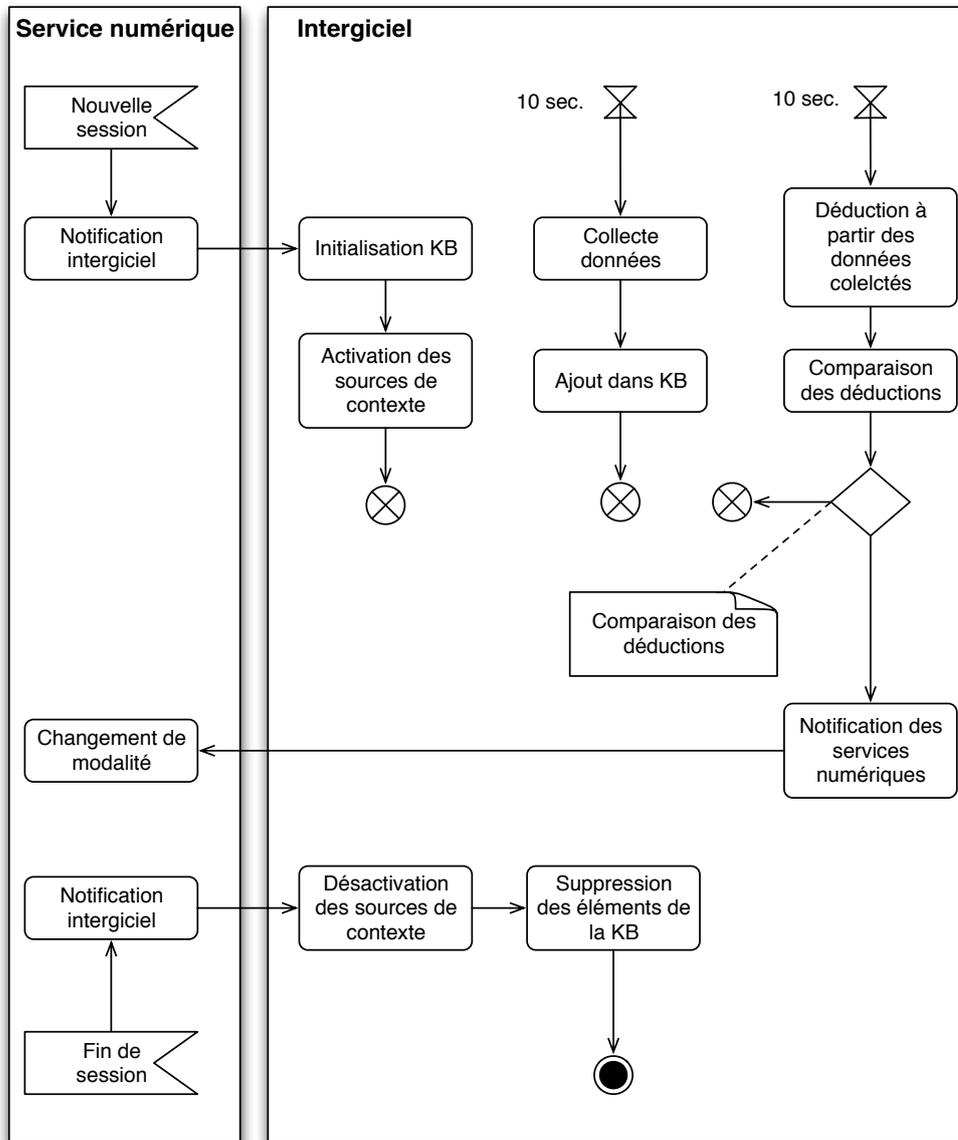


FIGURE 11.7: Consommation d'un service numérique - Diagramme d'activité

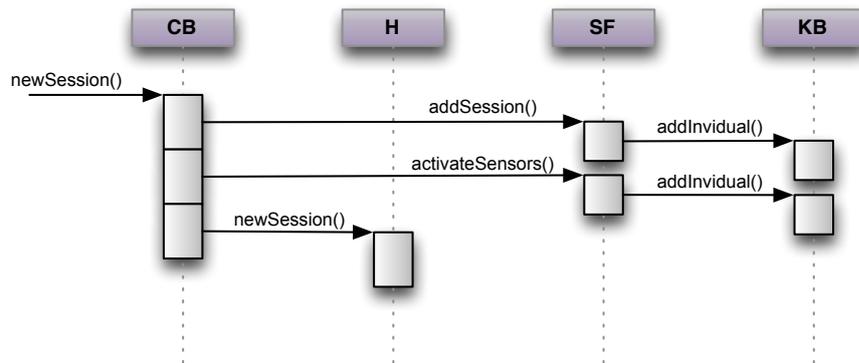


FIGURE 11.8: Ouverture d'une session - Diagramme de séquence

11.4.2 Exécution de la session

Pendant l'exécution de la session, qui commence dès son ouverture et se termine avec sa fermeture, l'intergiciel de sensibilité au contexte observe l'environnement pour déduire le contexte d'exécution de la session et choisir la modalité d'exécution adaptée.

La lecture de l'environnement est un processus commun à toutes les sessions applicatives. Il est exécuté à intervalle régulier, sans distinction de session. Le module *Harvester* récupère les données de l'environnement auprès de fournisseurs tiers pour l'ensemble des sessions ouvertes. À intervalles réguliers, le système récupère les données et les insère dans la base de connaissances.

Cette insertion fait intervenir un processus d'oubli. Il s'agit de supprimer les informations de contexte précédentes pour permettre de déduire la situation d'exécution actuelle.

La lecture de l'environnement est associée au processus de déduction puis de notification du changement de modalité. La première étape est la déduction des modalités d'exécution adaptées à la situation actuelle de l'utilisateur. Le choix des modalités d'exécution éligibles pour chaque session est réalisé par le module *Knowledge Base* à l'aide d'une inférence de la base de connaissances. Cette inférence est le résultat d'une confrontation des données de l'environnement avec les règles de sensibilité au contexte issues de la signature de contexte.

Le *Context Broker* élit ensuite, parmi les modalités d'exécution possibles, celle qui offre la meilleure performance. Ce processus d'élection est réalisé par un croisement entre les modalités d'exécution possibles et la dégradation du service numérique. Dans un second temps, il compare le résultat de l'élection avec la situation actuelle des sessions.

La seconde étape, la notification du changement de modalité, est effectuée par le mo-

dule *Context Querier*. Ce dernier compare les nouvelles connaissances avec les déductions précédentes. Chaque différence de modalité d'exécution correspond à un changement de la situation d'usage. Le *Context Broker* notifie les services numériques dont les sessions ont subi un changement de contexte, qu'un changement de modalité est nécessaire. Cette notification est réalisée sous la forme d'un message qui indique au service numérique les nouvelles modalités reprises pour les sessions concernées.

11.4.3 Terminaison d'une session (service numérique)

Le cycle de lecture de l'environnement se termine avec la fin de la session. Toutes les données relatives à la session sont supprimées. La suppression de ces éléments est réalisée dans la base de connaissances et elle concerne aussi bien les données des capteurs et que les données sur la session. En parallèle, le module *Harvester* supprime de sa liste de tâches l'accès aux données relatives à cette session. En effet, le rapatriement des données sur l'environnement est piloté par le *Harvester*. Ces deux tâches sont réalisées par l'intergiciel lors de la notification du service numérique de la fin d'une session.

11.5 Suppression d'un service numérique

La suppression du service numérique est une action déclenchée par une intervention humaine, elle est donc pilotée par le *Context Broker*. Tout comme le déploiement d'un service numérique, la suppression nécessite un travail sur le serveur d'applications et au sein de l'intergiciel de sensibilité au contexte. De fait, la suppression d'un service numérique sensible au contexte est pilotée par l'intergiciel, pour s'assurer de la suppression de la signature de contexte et du service numérique sensible au contexte.

Autrement dit, la suppression nécessite une intervention :

- Sur le serveur d'applications : retrait des binaires correspondants dans les conteneurs JEE,
- Sur l'intergiciel : suppression des références du service numérique (signature de contexte, ensemble dans la base de connaissances).

11.6 Conclusion

Ce chapitre présente les fonctions de notre intergiciel de gestion du contexte. Elles s'appuient sur les cinq modules qui composent notre intergiciel.

Nous distinguons trois phases d'opération de notre intergiciel :

- La maintenance de l'intergiciel avec l'ajout et la suppression de sources de contexte,
- La gestion des services numériques sensibles au contexte au sein du serveur d'applications (ajout et suppression),
- Le traitement du contexte lors de la consommation d'un service numérique.

Le prochain chapitre présente les particularités de notre modèle, en particulier le découplage des logiques proposées (section 12.1) et la signature de contexte (section 12.2) qui fait le lien entre les logiques. Ainsi que les différents modes opératoires de notre intergiciel inhérent à notre architecture : le déploiement (section 12.3) et le traitement du contexte (section 12.4)

Chapitre 12

Particularités de notre modèle

Un service numérique sensible au contexte est un service qui s'adapte à son environnement d'exécution. Ce dernier est restreint au contexte informatique tel que défini par les travaux de Schilit [68]. Il ne concerne que les ressources du terminal et du réseau d'accès. Cependant, il n'est pas figé, il peut évoluer en fonction des besoins et des disponibilités de sources de contexte.

Le modèle proposé sépare la gestion du contexte de la logique métier du service numérique selon les recommandations de Dey [20]. Nous nous inspirons aussi des remarques de Hong et coll. [34] sur l'externalisation sous la forme d'une infrastructure pour proposer un intergiciel sensible au contexte qui prend en charge l'ensemble des services numériques présents sur le serveur d'applications.

Dans cette partie nous avons présenté les apports de cette thèse au travers de quatre éléments caractéristiques :

- La séparation des rôles,
- La signature de contexte,
- La gestion du déploiement,
- Le traitement du contexte.

Dans ce chapitre, nous revenons individuellement sur chacun d'eux pour mettre en avant leurs interactions.

12.1 Séparation des rôles

Nous modélisons un service numérique sensible au contexte autour de trois logiques (sous-section 9.2.2) :

- La logique métier du service numérique,
- La logique d'identification des situations d'usage et la déduction des adaptations,

- La logique de collecte des données sur l’environnement d’exécution.

Notre travail se concentre sur l’identification des situations d’usage et la déduction des adaptations. La collecte est confiée à des solutions tiers. La logique métier du service numérique est réalisée par les concepteurs du dit service numérique sensible au contexte. Notre proposition s’articule autour d’un intergiciel de sensibilité au contexte qui permet de réaliser une séparation des logiques et de faire le lien entre elles.

L’intergiciel identifie les situations d’usage (contexte d’exécution du service numérique) et déduit l’adaptation à effectuer pour chaque session du service. Dans notre modèle l’adaptation est laissée à la discrétion du concepteur du service numérique. En effet, nous modélisons un service numérique sensible au contexte comme plusieurs logiques métier qui correspondent chacune à une modalité d’exécution. L’intergiciel identifie donc le contexte d’usage et préconise, pour chaque session de chaque service numérique sensible au contexte, la modalité d’exécution.

Le lien entre l’intergiciel et le service numérique sensible au contexte est réalisé par la signature de contexte.

12.2 Signature de contexte

Le moteur d’inférence est enrichi à l’aide de règles métier qui permettent d’identifier les situations d’usage et de déduire les modalités d’exécution. Ces règles d’adaptation sont le mécanisme clé du processus d’adaptation. Il ne faut cependant pas confondre adaptation et processus d’adaptation. Les règles qui composent la signature de contexte ne précisent pas l’adaptation de la logique métier du service numérique, mais identifient les situations d’usage [9]. De fait, chaque service numérique dispose d’au moins deux règles métier qui formalisent la modulation de la logique métier à utiliser en fonction du contexte de la session.

Elles font le lien entre une logique métier et une situation d’usage. La signature de contexte doit suivre le même cycle de vie que le service numérique, c’est-à-dire que la signature doit être déployée, supprimée ou modifiée en même temps que celui-ci.

C’est le concepteur du service numérique sensible au contexte qui fournit la *signature de contexte*. À l’instar d’une signature qui permet d’identifier une personne, elle identifie le contexte d’utilisation du service numérique.

12.3 Gestion du déploiement

Un service numérique sensible au contexte, tel que nous le modélisons, est réparti sur plusieurs entités (chapitre 9). Il faut donc gérer l’éclatement des éléments qui le composent sur ces différentes entités.

Pour cela nous confions la gestion du déploiement du service numérique sensible au contexte à l'intergiciel de sensibilité au contexte (section 11.3). Ce dernier décompose le service numérique sensible au contexte pour extraire les différentes logiques et les confier aux entités liées :

- La signature de contexte est confiée à l'intergiciel (logique d'identification et lien avec la logique de collecte).
- Le code du service numérique est confié au serveur d'applications (logique métier).

Ceci permet, lors du cycle de déploiement, d'injecter les règles dans la base de connaissances tout en déposant l'application dans son espace d'exécution [10].

12.4 Traitement du contexte

La transformation des données brutes (issues des capteurs) en information est réalisée avec l'aide des connaissances acquises lors de la déclaration des sources de contexte. L'enrichissement sémantique de la donnée fournie par le capteur concerne le type de capteur dont il est issu, la session relative à la donnée et l'unité.

Le second traitement est l'agrégation dans la base de connaissances de l'ensemble des informations pour former le contexte. Les connaissances qui permettent de transformer les informations isolées en contexte sont issues de la signature de contexte. Cette dernière fournit aussi le troisième traitement qui est la transformation du contexte en modalité d'exécution des sessions de chaque service numérique.

12.5 Sensibilité au contexte et réseaux du futur

La gestion du contexte pour les services numérique, telle que proposée dans ce manuscrit, a été motivée par les nouvelles situations d'usage apportées par les réseaux du futur. Ces réseaux n'étant pas encore totalement opérationnels ce modèle reste volontairement générique.

Notre contribution, un intergiciel de sensibilité au contexte, a été conceptualisée pour s'insérer dans un NGN, en l'occurrence IMS (Figure 12.1).

L'infrastructure du réseau IMS reste inchangée, nous retrouvons le réseau coeur, le réseau d'accès et les terminaux utilisateurs. Notre intergiciel est mis en place sur les serveurs d'applications. C'est donc tout naturellement que nous le retrouvons au même niveau que les autres serveurs d'applications. Comme nous avons les serveurs d'applications natifs, les serveurs d'applications OSA et les serveurs d'applications CAMEL, nous trouvons les serveurs d'applications natifs sensibles au contexte. C'est dans cette zone que nous trouverons aussi les applications tierces sur lesquelles notre intergiciel s'appuie pour la capture des données sur l'environnement d'exécution.

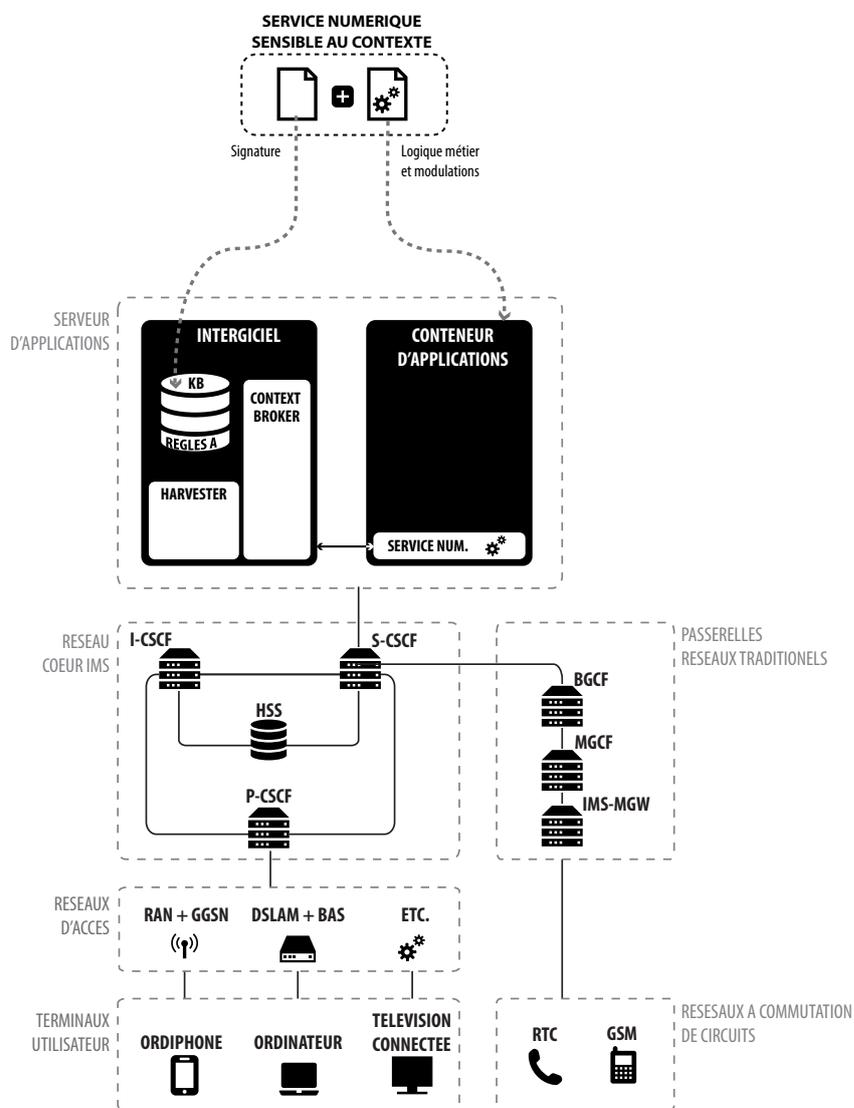


FIGURE 12.1: Intégration de notre proposition dans une architecture IMS

Quatrième partie

Réalisation dans OpenIMS

Chapitre 13

Choix technologiques

Les travaux de ce manuscrit s'inscrivent dans les réseaux du futur. Ce chapitre justifie nos choix technologiques pour la mise en place de l'environnement de travail.

Nous avons retenu IP Multimedia Subsystem (IMS) comme infrastructure NGN car c'est la plus aboutie actuellement. Un réseau IMS est composé de plusieurs éléments (sous-section 4.6.3) :

- Le coeur de réseau qui concentre les fonctions vitales du réseau (base d'utilisateurs, authentification, contrôle d'accès, etc.)
- Les services numériques.
- Les terminaux des utilisateurs.

13.1 Coeur de réseau IMS

Il n'existe, à notre connaissance, que deux solutions open source de coeur de réseau IMS :

- LittleIMS
- OpenIMS Core

LittleIMS est un produit de la société NEXCOM Systems. Ce produit est présenté comme une solution clé en main pour les développeurs d'applications IMS : un coeur de réseau, un softphone et un serveur d'applications. Dans sa version open source, lancée en 2009, littleIMS ne propose que les éléments du coeur de réseau.

OpenIMS Core est un projet open source qui se propose d'implémenter le coeur d'un réseau IMS section 4.6.3 tel que spécifié par 3GPP et l'ETSI. Ce projet est aussi connu sous le nom d'OSIMS¹ (Open Source IMS Core System). Il est développé par

1. OSIMS - The FOKUS Open Source IMS Core : http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osims/index.html

l'institut de recherche allemand Fraunhofer FOKUS. L'objectif de FOKUS est de fournir un environnement IMS pour favoriser le développement de prototypes pour la recherche. Nous avons retenu la solution OpenIMS pour la mise en place de notre environnement de tests, car c'est, à notre connaissance le projet le plus mature² et le plus complet. En effet, Fraunhofer FOKUS a développé autour de OSIMS un banc d'essai IMS. Il est composé des éléments suivants :

- Call Session Control Function.
- Home Subscriber Server.
- Passerelle OSA/Parlay.
- Clients IMS.
- Operation Support System (OSS³) pour OpenIMS.
- Architecture de facturation (PCC⁴).

D'autres éléments sont développés sur la base d'OSIMS pour compléter cet environnement. Le coeur de réseau seul ne permet pas de dépasser l'aspect pédagogique du fonctionnement d'IMS. En effet, il ne propose que la gestion de sessions réseau et applicative, mais il offre la possibilité d'utiliser des applications SIP. Ces dernières restent assez limitées puisque l'utilisation la plus courante reste la téléphonie sur IP. Pour pouvoir tester d'autres concepts, il faut adjoindre au coeur de réseau une plateforme de services.

13.2 serveur d'applications

Les services numériques au sein d'un réseau IMS utilisent les protocoles SIP et parfois Diameter. Nous avons enrichi les critères de sélection en ajoutant une contrainte sur la licence du serveur : nous avons exclu tous les serveurs sous licences commerciales, pour des raisons de budget. La sélection a été affinée avec les fonctionnalités et les langages disponibles (Tableau 13.1).

TABLEAU 13.1: Comparaison des serveurs d'applications

	Interface SIP	Interface Diameter	Modèle de programmation
Cipango (1.0)	X		JEE 5
Cipango (2.0)	X	X	JEE 5
GlassFish	X		JEE 6
Mobicents Jain-SLEE	X	X	JSLEE
Mobicents SIP-Servlets	X	X	JEE 5
OpenSIPs	X	X	JEE 5
Sailfin	X		JEE 5

Nous distinguons deux modèles de programmations, Java Enterprise Edition (JEE)

-
2. OpenIMS a été créé en 2006 LittleIMS a été lancé en 2009
 3. Ce sont des systèmes de surveillance et gestion des réseaux.
 4. Policy and Charging Control

et Java SLEE. Malgré l'intérêt des fonctionnalités de JSLEE pour l'industrie des télécommunications et le potentiel de ce type d'environnement nous nous tournons vers JEE. En effet, JSLEE ne coïncide pas avec l'objectif de réalisation d'un prototype et de surcroît nécessite l'apprentissage d'un nouveau modèle de programmation.

Lors des premières phases de réalisation, seuls les serveurs d'application OpenSIPS et Mobicent SIP-Servelets disposaient d'une interface SIP et Diameter⁵. Malgré la modularité proposée par OpenSIPS, notre choix s'est porté sur le produit proposé par Mobicents. Ce dernier est composé d'une plus grande communauté d'utilisateurs et la documentation proposée est complète. Par ailleurs, la base du serveur est JBoss, produit de la société RedHat. Cette situation offre l'avantage de pouvoir profiter de la communauté JBoss pour les questions ou problèmes relatifs à JEE et de nous appuyer sur la communauté Mobicents pour les spécificités des interfaces SIP et Diameter. Cependant nous regretterons l'absence d'OSGi dans ce serveur d'applications, qui aurait pu apporter un axe de réflexion supplémentaire pour la réalisation de notre intergiciel de sensibilité au contexte en renforçant la vue service du serveur d'applications, transformant la proposition en un serveur d'applications sensible au contexte.

13.2.1 Modèle JEE

JEE 5 est la cinquième version de Java Enterprise Edition, une des trois versions de Java⁶. C'est un composé de technologies issu de Java qui offre un environnement complet pour héberger et exécuter des applications Java sur un élément distant. Nous trouvons sur un serveur d'applications Java, l'environnement d'exécution Java, des services/ressources et des applications. Les services et ressources sont, par exemple, les bases de données, services de courrier électronique, etc. Nous nous intéresserons à des composants SIP/WEB et EJB déployés, gérés et exécutés sur le serveur d'applications⁷. Ils s'appuient sur des médiateurs d'accès aux ressources (p. ex. pour accéder à une base de données) et des services standards.

Les applications sont présentes dans des conteneurs. Un conteneur est un environnement d'exécution pour les applications, dont la responsabilité est la création des instances et la gestion de leur cycle de vie. Ces conteneurs gèrent le cycle de vie des applications et proposent les interactions avec les autres composants et applications de ce serveur. La logique métier des applications est enfermée dans ces conteneurs. Ils sont organisés de la façon suivante : un container WEB pour l'ensemble des technologies web et l'autre pour les composants EJB (Enterprise Java Beans). Les composants web sont les Java Server Pages (JSP) ainsi que les servlets HTTP ou SIP. Une application SIP est un programme

5. Lors du choix du serveur d'application, Cipango n'était pas encore au stade d'une version 1.0 stable.

6. <http://www.oracle.com/technetwork/java/javaee/tech/javaee5-jsp-135162.html>

7. Ceci exclut les composants exécutés entièrement ou partiellement en dehors du serveur d'application.

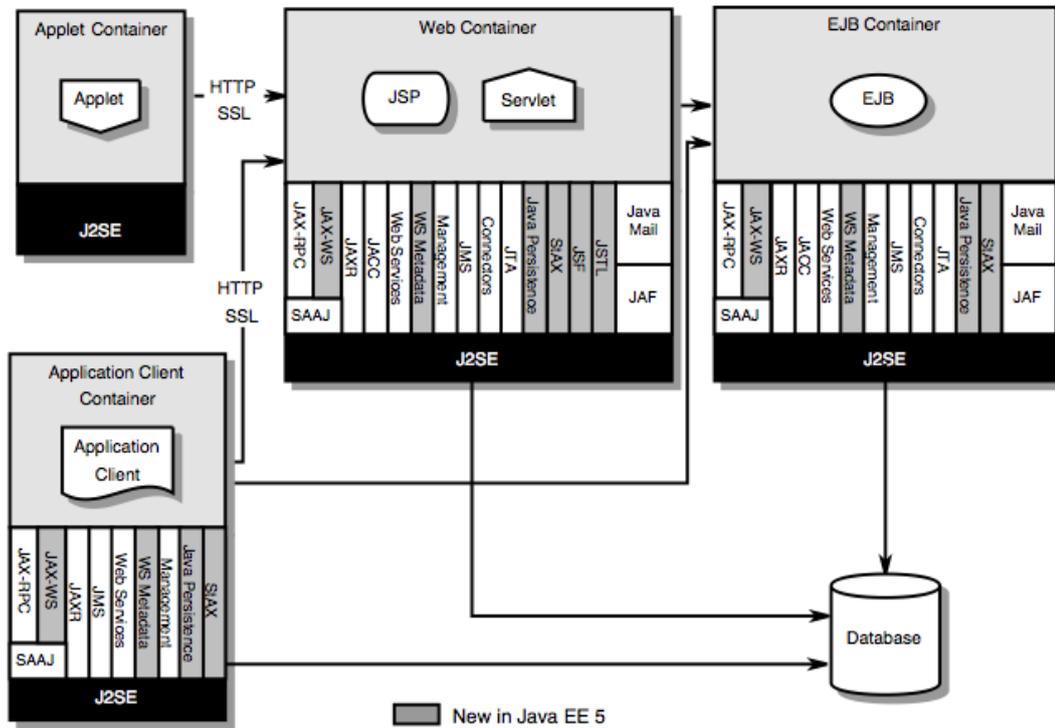


FIGURE 13.1: Architecture d'une plateforme JAVA EE 5 extraite de la documentation Oracle

qui s'appuie sur au moins une servlet SIP. Les servlets SIP pour JAVA sont définies par la JSR 2.11

13.3 Terminaux IMS

Pour la réalisation de notre prototype, nous préférons un ordinateur plutôt qu'un terminal mobile. L'ordinateur est un système virtualisé sur un des serveurs de la plateforme EUROP. Cet ordinateur est équipé de deux programmes, un qui prend en charge la session IMS et un qui prend en charge la session de l'application. Autrement dit, un premier logiciel s'authentifie sur le réseau IMS et crée la session IMS. Un second logiciel interagit avec le service numérique mis à disposition sur un serveur d'applications. Cette approximation d'un terminal IMS, à partir d'un équipement informatique quelconque, est rendue possible grâce à la convergence des télécommunications et de l'informatique. En effet, un réseau de télécommunications devient un réseau convergent basé sur IP.

13.4 Contraintes liées à l'environnement

Les technologies utilisées pour réaliser le prototype sont, en partie, définies par les éléments disponibles pour la réalisation de notre banc d'essai. Notre travail concerne les services numériques proposés dans un réseau IMS, c'est-à-dire le serveur d'applications. Nous avons retenu le serveur d'applications Mobicents, qui propose un environnement d'exécution JEE. Puisque nous avons implémenté la sensibilité au contexte des services numériques sous la forme d'un intergiciel ce dernier sera développé en utilisant le langage JAVA et les composants JEE.

Chapitre 14

Mise en oeuvre

14.1 Les technologies utilisées pour la réalisation de l'intergiciel

Chacun des 5 composants de l'intergiciel est implémenté en Java 5 et utilise des bibliothèques supplémentaires qui sont présentées dans la suite de cette section.

14.1.1 Le langage web d'ontologies

Nous nous appuyons sur les technologies du Web Sémantique pour la réalisation de notre intergiciel. Plus particulièrement sur les technologies liées au langage web d'ontologies :

- la logique de description au travers du langage OWL.
- les règles métier SWRL.
- le langage de requêtes SQWRL.

Nos choix technologiques sont imposés par le langage de requête SQWRL. C'est une extension personnalisée de SWRL réalisée par Martin O'Connor. Elle est présente dans SWRLTab¹ qui s'appuie sur l'API de Protégé-OWL².

Le langage web d'ontologie est principalement utilisé pour notre base de connaissances qui est hébergée par le module *Knowledge Base*. On le retrouve dans les modules *Semantic Formalizer* et *Context Querier* car ils ont en charge les accès en lecture et écriture pour la base de connaissances :

- Semantic Formalizer (SF) - Ce module prend en charge l'ensemble des opérations d'écriture dans la KB, il s'appuie sur l'API Protege-OWL.

1. <http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab>

2. <http://protege.stanford.edu/overview/protege-owl.html>

- Context Querier (CQ) - Il prend en charge l'interrogation de la KB dont les requêtes SQWRL. On retrouve les fonctions de la librairie SQWRL Query API.

14.2 Le moteur d'inférence

Le module *Knowledge Base* concentre les informations sur l'environnement au format d'une ontologie et utilise les règles métiers pour maintenir la connaissance. Nous retrouvons donc dans ce module la base de connaissances et le moteur d'inférence.

C'est le moteur d'inférence Jess³ qui a été retenu, car il est le seul moteur d'inférence qui supporte les extensions personnalisées de SWRL (contrainte induite par l'utilisation de SQWRL) pour l'inférence des données de la base de connaissances. Depuis le premier trimestre 2012, il est possible d'utiliser Drools pour exécuter des requêtes SQWRL.

14.2.1 Routage des messages

Pour mettre en place un couplage faible entre l'intergiciel et l'application, nous utilisons le paradigme MOM (Middleware Oriented Message). L'ensemble des échanges entre les services numériques sensibles au contexte et l'intergiciel s'effectue par l'intermédiaire de messages. Le routage des messages s'appuie sur la librairie Java Apache Camel⁴.

Apache Camel est un environnement d'intégration d'applications pour Java, basé sur un moteur de règles et de médiation. Il intègre plusieurs API et technologies de transport accessibles par un langage de domaine spécifique (DSL - Domain Specific Language). Les concepts proposés par cette librairie reposent sur les modèles d'intégration EIP - Enterprise Integration Patterns⁵. Ce sont des solutions éprouvées et approuvées pour résoudre des problèmes d'intégration en génie logiciel.

Nous retrouvons cette librairie au sein du *Context Broker*.

14.3 OpenIMS

14.4 Environnement construit

Pour comprendre et mettre en pratique les mécanismes des réseaux du futur, j'ai mis en place un environnement IMS basé sur OpenIMS (section D.3). Il est composé des éléments coeur d'un réseau IMS (CSCF), d'une base utilisateurs (HSS), d'une gestion QoS (PDF) ainsi que deux équipements utilisateurs (UE). Les équipements utilisateurs sont des softphones⁶. Ces éléments permettent de mettre en place les sessions IMS et de réaliser des communications entre terminaux.

3. <http://jessrules.com/>

4. <http://camel.apache.org/>

5. <http://www.eaipatterns.com/>

6. Logiciels de téléphonie par internet.

Pour compléter cette architecture, nous avons ajouté une couche de services. Cet ajout est réalisé par l'intégration d'un serveur d'applications Mobicents qui héberge l'intergiciel et les services numériques (sensibles au contexte ou non).

Les serveurs (CSCFs, HSS, AS, UE) sont virtualisés : nous exécutons, sur un même serveur physique, plusieurs systèmes d'exploitation en parallèle.

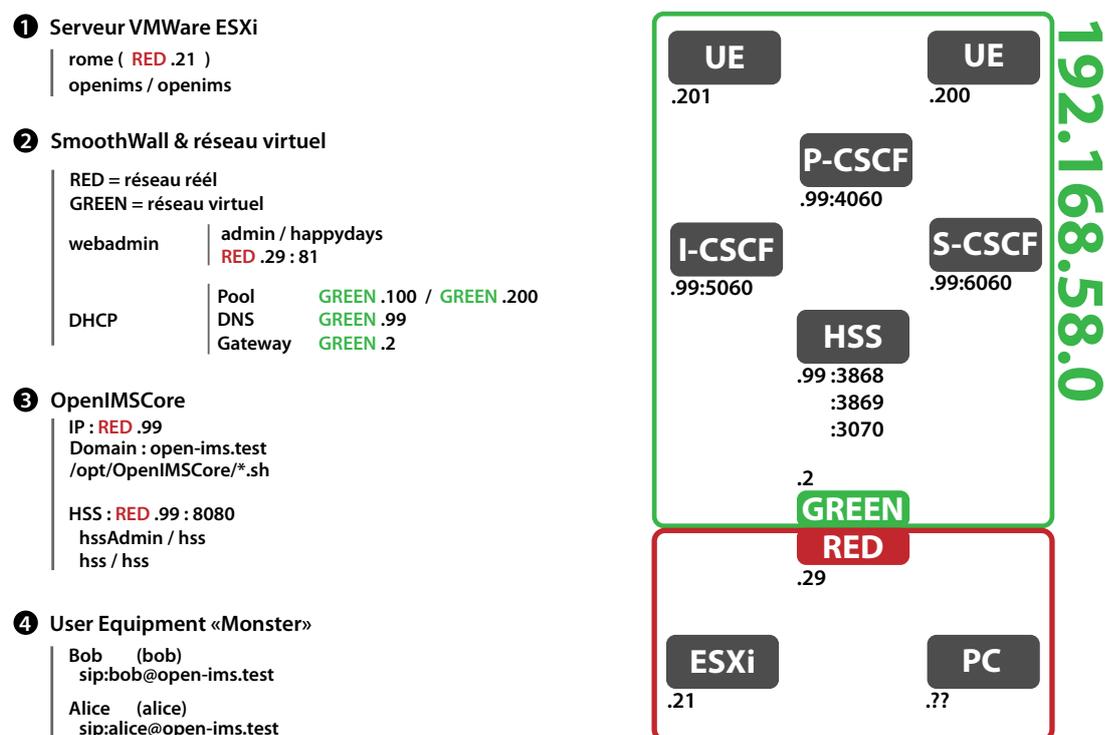


FIGURE 14.1: Architecture IMS virtualisée

Le serveur virtualisé qui héberge le coeur IMS fait partie de la plateforme EUROP, une des plateformes technologiques de Télécom Saint-Étienne.

14.4.1 Plateforme EUROP

J'ai eu l'opportunité de travailler sur la conception puis la réalisation de la plateforme EUROP pendant mon doctorat. Cette plateforme de tests et simulations de réseaux haut débit filaires permet la mise en place d'un environnement complet avec les technologies cuivres (ADSL, ADSL2+, SHDSL, VDSL) et optiques (point à point, GPON). De par sa conception modulaire, elle permet de recréer différentes architectures utilisées pour les réseaux haut débit.

Les équipements utilisés pour recréer de tels réseaux sont :

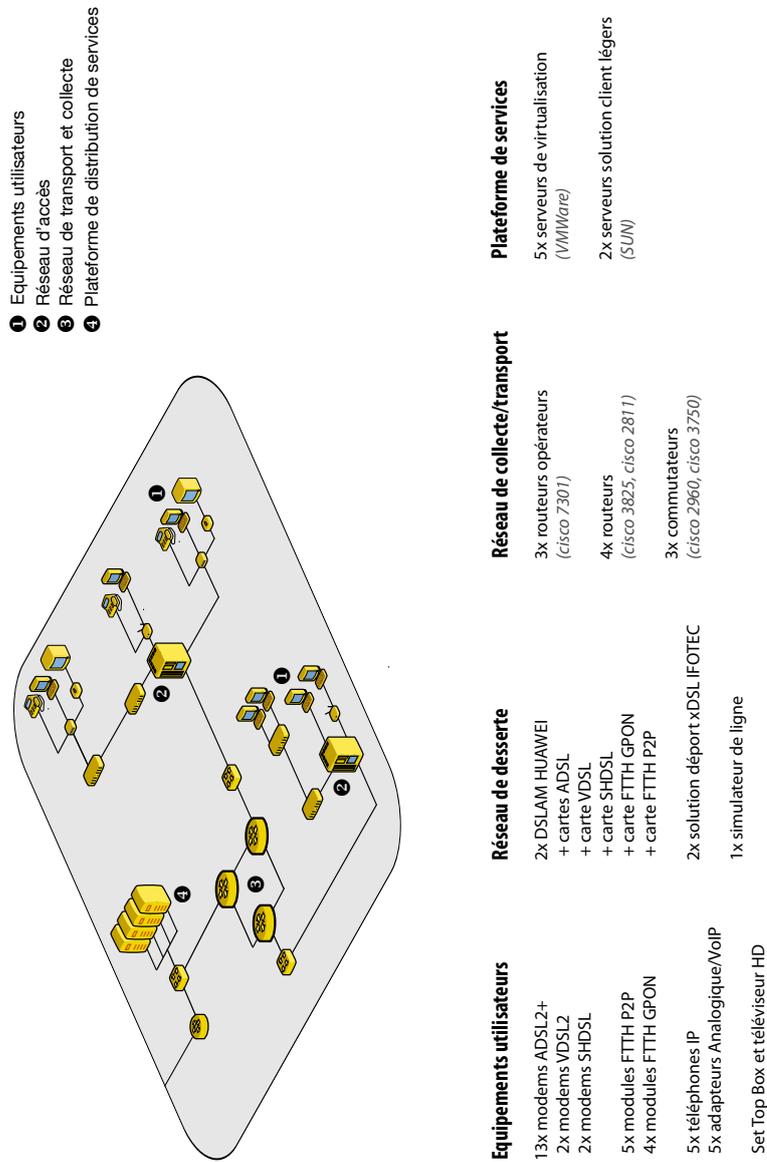


FIGURE 14.2: Architecture type de la plateforme EUROP

- Serveurs virtualisés sous VMWare Infrastructure.
- Coeurs de réseaux CISCO avec fonctions fournisseurs de services et opérateurs.
- Routeurs cisco (3825 et 2811) avec fonctions opérateurs.
- 2 DSLAMs avec des cartes ADSL, ADSL2, ADSL2+, VDSL2, G.SHDSL, G.SHDSL.BIS, GPON, FE (FTTH, FTTH Pro et FTTB).
- Solution de déport de signaux xDSL sur fibre optique (IFOTEC).
- Simulateurs de ligne xDSL.
- Outils de mesure de bande passante et de signaux xDSL.
- Terminaux (modems, médiaconverters, ONTs, CPEs FTTH, STB, ordinateurs, télévision).
- Solution de clients légers SUN.

L'architecture présentée en Figure 14.2 permet de mettre en oeuvre un réseau haut débit filaire multiservices qui permet la mise à disposition de services par l'opérateur. Nous retrouvons les services avec de la vidéo (vidéo à la demande, télédiffusion), de l'audio (Voix-sur-IP), des transferts de données (échange de courriels, navigation internet).

Chapitre 15

Validation du scénario de référence

La validation de notre scénario de référence (chapitre 2) se réalise en deux temps :

1. Le déploiement du service numérique sensible au contexte.
2. La consommation du service numérique par un utilisateur externe.

15.1 Déploiement du service numérique sensible au contexte

Un service numérique sensible au contexte se présente au serveur d'applications sous la forme d'un *paquet* composé d'un binaire et d'une signature de contexte. Lors de son déploiement l'intergiciel décompose le service numérique en séparant ces deux éléments. Il confie le déploiement des binaires du service numérique au serveur d'applications et il prend en charge l'interprétation puis l'intégration de la signature de contexte.

Dans notre scénario de référence, l'intergiciel présent sur le serveur d'applications est connecté à deux sources de contexte : une pour la lecture de la bande passante et une pour la lecture de la résolution de l'écran. Cette interconnexion a été réalisée par l'administrateur du système. Il n'y a encore aucun service numérique déployé sur le serveur d'applications.

L'ontologie (Figure 15.1) ne contient donc aucun sous-ensemble dans son ensemble *Session* et deux sous-ensembles dans l'ensemble *ContextProvider* qui correspondent à la source de contexte pour la bande passante disponible (*Bandwidth*) et un pour la résolution de l'écran (*Resolution*).

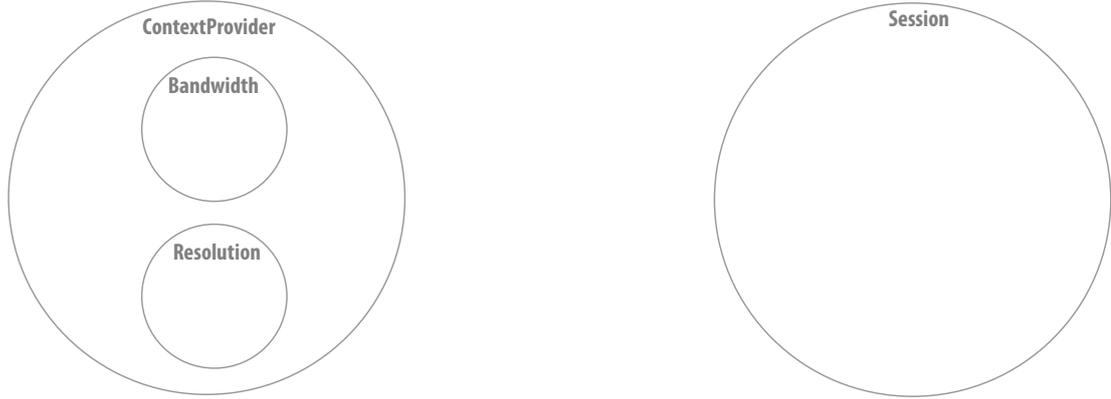


FIGURE 15.1: Aperçu de l'ontologie avant le déploiement du service numérique sensible au contexte

15.1.1 La signature de contexte

Dans notre scénario de référence, présenté en début de ce manuscrit, le service numérique *téléconférence* propose 4 modalités d'exécutions pour 4 situations d'usage définies par des seuils de bande passante et de résolution d'écran (Tableau 15.1). Ces éléments se retrouvent dans la signature de contexte. Ci-dessous voici la signature de contexte de notre service numérique. Elle est composée de règles métier au format SWRL (7.1 à 7.4) et SQWRL (7.5 à 7.8).

$$\begin{aligned}
 & \text{TeleConference}(?s) \wedge \\
 & \text{hasBandwidth}(?s, ?b) \wedge \\
 & \quad \text{value}(?b, ?bv) \wedge \\
 & \text{swrlb : greaterThanOrEqual}(?bv, 8192) \\
 & \text{hasResolution}(?s, ?r) \wedge \\
 & \quad \text{value}(?r, ?rv) \wedge \\
 & \text{swrlb : greaterThanOrEqual}(?rv, 786432) \Rightarrow \text{AudioVideoHD}(?s) \quad (15.1)
 \end{aligned}$$

$$\begin{aligned}
 & \text{TeleConference}(?s) \wedge \\
 & \text{hasbandwidth}(?s, ?b) \wedge \\
 & \quad \text{value}(?b, ?bv) \wedge \\
 & \text{swrlb : greaterThanOrEqual}(?bv, 2048) \wedge \\
 & \text{hasResolution}(?s, ?r) \wedge \\
 & \quad \text{value}(?r, ?rv) \wedge \\
 & \text{swrlb : greaterThanOrEqual}(?rv, 384000) \Rightarrow \text{AudioVideoMD}(?s) \quad (15.2)
 \end{aligned}$$

$$\begin{aligned}
& \text{TeleConference}(?s) \wedge \\
& \text{hasbandwidth}(?s, ?b) \wedge \\
& \text{value}(?b, ?bv) \wedge \\
& \text{swrlb : greaterThanOrEqual}(?bv, 256) \wedge \\
& \text{hasResolution}(?s, ?r) \wedge \\
& \text{value}(?r, ?rv) \wedge \\
& \text{swrlb : greaterThanOrEqual}(?rv, 76800) \Rightarrow \text{AudioVideoFD}(?s) \quad (15.3)
\end{aligned}$$

$$\begin{aligned}
& \text{TeleConference}(?s) \wedge \\
& \text{hasbandwidth}(?s, ?b) \wedge \\
& \text{value}(?b, ?bv) \wedge \\
& \text{swrlb : lessThan}(?bv, 64) \Rightarrow \text{AudioOnly}(?s) \quad (15.4)
\end{aligned}$$

$$\text{AudioVideoHD}(?s) \Rightarrow \text{sqwrl : select}(?s) \quad (15.5)$$

$$\text{AudioVideoMD}(?s) \Rightarrow \text{sqwrl : select}(?s) \quad (15.6)$$

$$\text{AudioVideoFD}(?s) \Rightarrow \text{sqwrl : select}(?s) \quad (15.7)$$

$$\text{AudioOnly}(?s) \Rightarrow \text{sqwrl : select}(?s) \quad (15.8)$$

TABLEAU 15.1: Situations d'usage

Bande passante	Résolution de l'écran	Modalité	Règle liée
> 8 Mbit/s	> 1024*768	Audio et Vidéo HD	7.1
> 2 Mbit/s	> 800*480	Audio et Vidéo MD	7.2
> 256 Kbit/s	> 320*240	Audio et Vidéo BD	7.3
< 64 Kbit/s)	-	Audio seul	7.4

L'analyse de cette signature est prise en charge par le *Context Broker*. Il extrait les différentes informations nécessaires pour les autres modules de l'intergiciel :

- modalité d'exécution de la plus optimale à la plus dégradée,
- sources de contexte pertinentes.

Notre signature propose quatre modalités d'exécution :

- Audio avec vidéo en haute qualité (7.5),
- Audio avec vidéo en qualité standard (7.6),
- Audio avec vidéo en basse qualité (7.7),
- Audio sans vidéo (7.8).

Et deux sources de contexte :

- hasBandwidth,
- hasResolution.

Il informe à l'aide des informations extraites :

- Le module *Context Querier* des modalités d'exécution du service numérique.
- Le module *Semantic Formalizer* des modalités d'exécution, du nom du service numérique et de la signature de contexte complète.
- Le module *Harvester* des sources de contexte nécessaires pour la collecte du contexte du service nouvellement déployé.

15.1.2 Les modalités d'exécution SQWRL (Context Querier)

Le module *Context Querier* reçoit les modalités d'exécution et leur ordre de réalisation. En effet l'ordre de présentation de ces règles dans la signature de contexte a son importance puisqu'il détermine, pour le module, quelle est la modalité d'exécution optimale pour le service numérique. En effet, lors d'une inférence, nous retrouvons l'ensemble des modalités d'exécution possibles pour chaque session. C'est au *Context Querier* de faire le tri et de proposer la modalité optimale.

Ces modalités sont donc classées dans l'ordre de la plus optimale à la moins optimale. Ce formalisme est propre à notre intergiciel. Le moteur d'inférence, dans sa version standard, n'accorde aucune importance à l'ordre des règles SQWRL.

15.1.3 Les ensembles et sous-ensembles de la base de connaissances (Semantic Formalizer)

Le *Semantic Formalizer* reçoit du *Context Broker* la signature de contexte brute ainsi que les modalités d'exécution et l'identifiant du service numérique. Nous faisons ici une opposition entre les éléments issus de l'analyse de la signature de contexte par le *Context Broker* et la signature de contexte sans analyse. L'ensemble des informations est à destination de la base de connaissance qui est piloté par le module *Knowledge Base*.

Ces informations permettent de construire les ensembles de la base de connaissances qui sont relatifs au service numérique sensible au contexte nouvellement déployé. Cela concerne l'application et ses modalités d'exécution. Dans notre scénario, nous retrouvons un ensemble *TeleConference* avec quatre sous-ensembles qui sont *AudioOnly*, *AudioVideoFD*, *AudioVideoMD*, *AudioVideoHD* (Figure 15.2) Ces données sont indispensables à la base de connaissances. La signature de contexte est insérée directement dans l'ontologie.

15.1.4 Les sources de contexte (Harvester)

L'analyse de la signature de contexte permet d'extraire la liste des capteurs utilisés par le service numérique sensible au contexte. Ces informations sont identifiées par

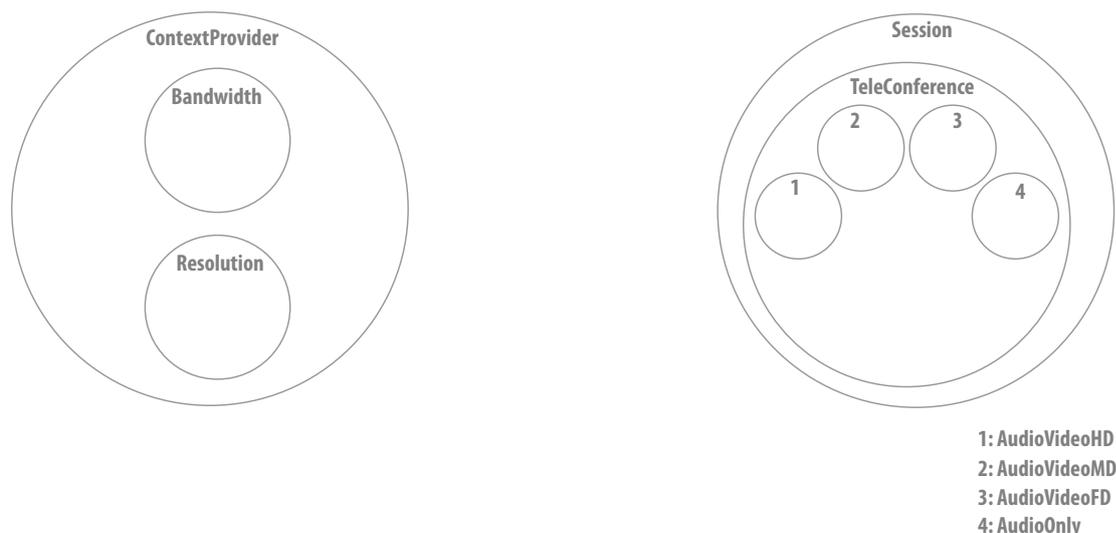


FIGURE 15.2: Ontologie après le déploiement du service numérique

analogie avec l'intergiciel qui dispose déjà des passerelles vers ces capteurs. Dans notre scénario nous avons deux sources contexte :

- hasBandwidth : la bande passante.
- hasResolution : la résolution de l'écran.

15.2 Consommation du service numérique sensible au contexte

Une fois le service numérique sensible au contexte mis en place sur le serveur d'applications et l'intergiciel prêt à prendre en charge la gestion du contexte des sessions ouvertes, nous abordons la phase de consommation du service numérique. Cette phase s'appuie sur la chaîne de traitement du contexte (Figure 10.1).

15.2.1 Ouverture d'une session

L'ouverture d'une session par un utilisateur se fait au travers du service numérique. Lorsqu'il reçoit une demande, le service numérique notifie l'intergiciel d'une nouvelle connexion pour initialiser les fonctions de gestion du contexte proposées par l'intergiciel.

Nous nous intéressons ici uniquement au déroulement de la session du côté de l'intergiciel. Ce cycle débute par la réception d'une notification d'*ouverture de session* de la part d'un service numérique. Cette notification est reçue et traitée par le *Context Broker* puisqu'il joue le rôle de passerelle entre l'intergiciel et les services numériques.

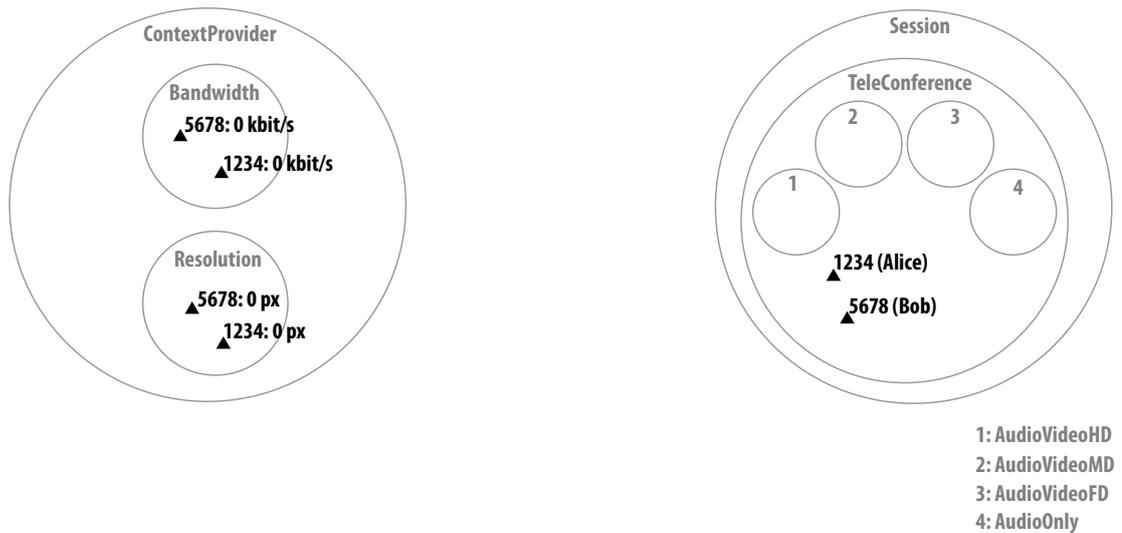


FIGURE 15.3: Ontologie lors des ouvertures de sessions

Dans notre scénario, le service numérique informe de la création d’une session lors de la connexion d’Alice. Il en va de même lors de la connexion de Bob au service numérique. Lors de la création, le *Context Broker* demande au *Semantic Formalizer* de créer de nouveaux individus dans la base de connaissances (Figure 15.3 :

- un pour la session du service numérique,
- plusieurs autres pour les capteurs d’information de contexte : un pour chaque capteur nécessaire au service numérique.

Pour l’instant les individus qui personnifient la session n’appartiennent à aucun sous-ensemble. Ils appartiennent seulement à l’ensemble global *TeleConference*.

Les individus qui personnifient les capteurs d’informations de contexte sont ajoutés dans leurs sous-ensembles respectifs. Dans notre scénario :

- Deux individus dans l’ensemble *Bandwidth* (un pour Alice, un pour Bob).
- Deux individus dans l’ensemble *Resolution* (un pour Alice, un pour Bob).

15.2.2 Traitement du contexte : collecte

La mesure des données sur l’environnement est réalisée par des solutions tiers. Ces données sont ensuite collectées par l’intergiciel grâce au module *Harvester*. Il ne collecte que les données nécessaires, c’est-à-dire qui concernent le contexte d’une session ouverte. À intervalles réguliers il interroge les sources de données pour récupérer les données sur l’environnement. Dans cet exemple, le *Harvester* collecte les données suivantes de nos deux sources de contexte :

```

bandwidth = { "1234": "10000", "6789": "3000" }
resolution = { "1234": "76800", "6789": "76800" }

```

Les résultats sont groupés par source de contexte puisque le *Harvester* demande l'ensemble des données concernant une source de contexte. Chaque session dans IMS est identifiée par un numéro que nous avons représenté ici par *1234* (Bob) et *6789* (Alice).

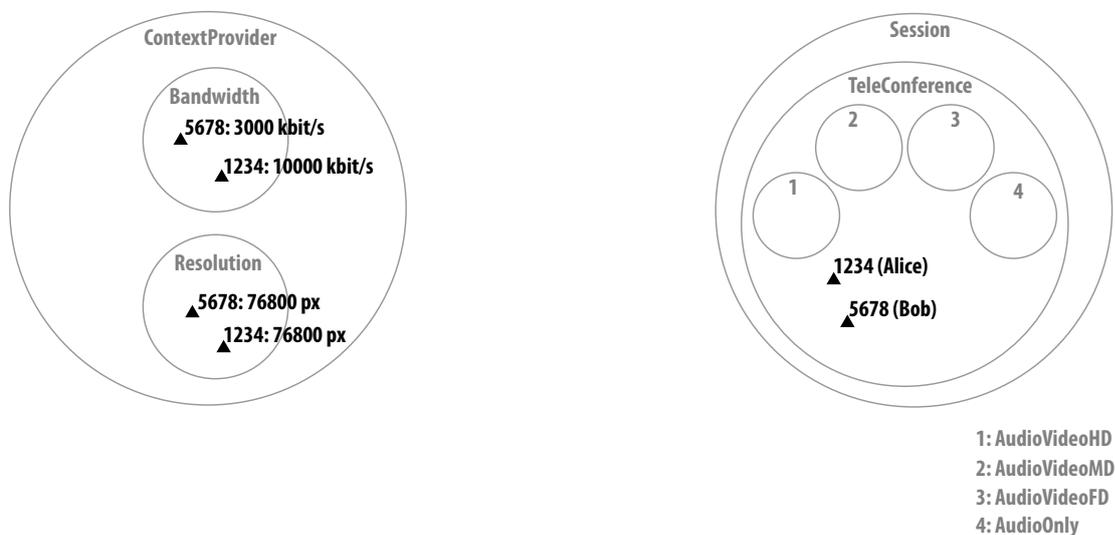


FIGURE 15.4: Ontologie lors de l'ajout des valeurs des capteurs

Ces données sont transmises au *Semantic Formalizer* qui les enrichit sémantiquement et les insère dans la base de connaissances. L'ajout dans la base de connaissance de nouvelles données issues des capteurs nécessite la suppression des anciennes données. En effet, le système de déduction des ontologies en web sémantique sont basé sur la monotonie (section 6.2.1). Si nous n'effectuons pas un remplacement des données, le système peut se retrouver dans une situation d'indécision. Cependant l'opération de remplacement n'est pas une opération standard dans une ontologie. Il faut donc que le *Semantic Formalize* supprime les données collectées précédemment et les conclusions liées. Il peut ensuite ajouter les nouvelles valeurs et effectuer de nouvelles déductions. Une fois les données insérées et avant l'inférence, l'intergiciel n'a pas connaissance des modalités d'exécution possibles pour les différentes sessions (Figure 15.4).

15.2.3 Traitement du contexte : analyse de l'environnement d'exécution

Lorsque les données sur l'environnement sont insérées dans la base de connaissances, c'est le moteur d'inférence qui détermine quelles sont les modalités d'exécution possibles pour chaque service numérique. Pour réaliser ce travail de déduction, il s'appuie sur les données de l'environnement et les règles SWRL de la signature de contexte.

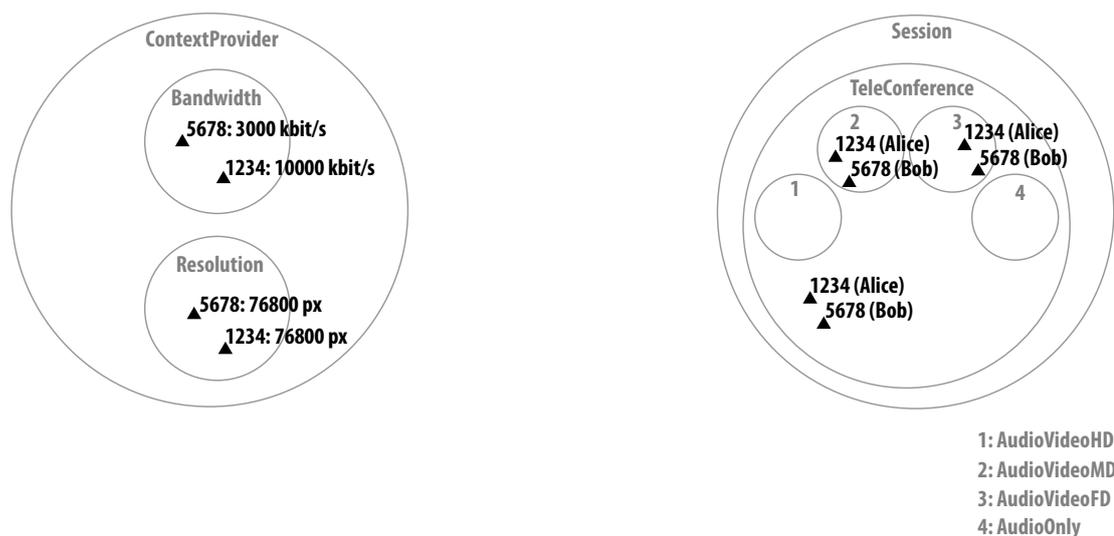


FIGURE 15.5: Ontologie après l'inférence

Dans cet exemple, Alice a une bande passante suffisante pour pouvoir exploiter pleinement l'affichage de son terminal qui propose une résolution en basse définition. Les modalités d'exécution possibles sont *AudioVideoMD* et *AudioVideoFD*. Bob est dans la même situation d'usage qu'Alice : il a la possibilité d'utiliser les modalités d'exécution *AudioVideoMD* et *AudioVideoFD* (Figure 15.5).

15.2.4 Traitement du contexte : déduction de la modalité adaptée

Le module *Context Querier* interroge régulièrement la base de connaissances pour connaître des modalités d'exécution éligibles pour chaque session.

Il récupère les nouvelles valeurs des sessions d'Alice et Bob. Dans la situation d'Alice et Bob, la base de connaissances déduit deux modalités possibles : *AudioVideoMD* et *AudioVideoFD*. D'après la signature de contexte, la modalité optimale pour cette session est *AudioVideoMD*.

Nous sommes à la toute première inférence du système. Il n'existe pas de situation précédente. Le *Context Querier* transmet donc au service numérique par l'intermédiaire du *Context Broker* les nouvelles modalités.

15.2.5 Traitement du contexte : détection d'un changement de situation

Après plusieurs minutes de fonctionnement, la situation de Bob évolue : sa bande passante est de 50 Kbit/s. La situation d'Alice reste inchangée.

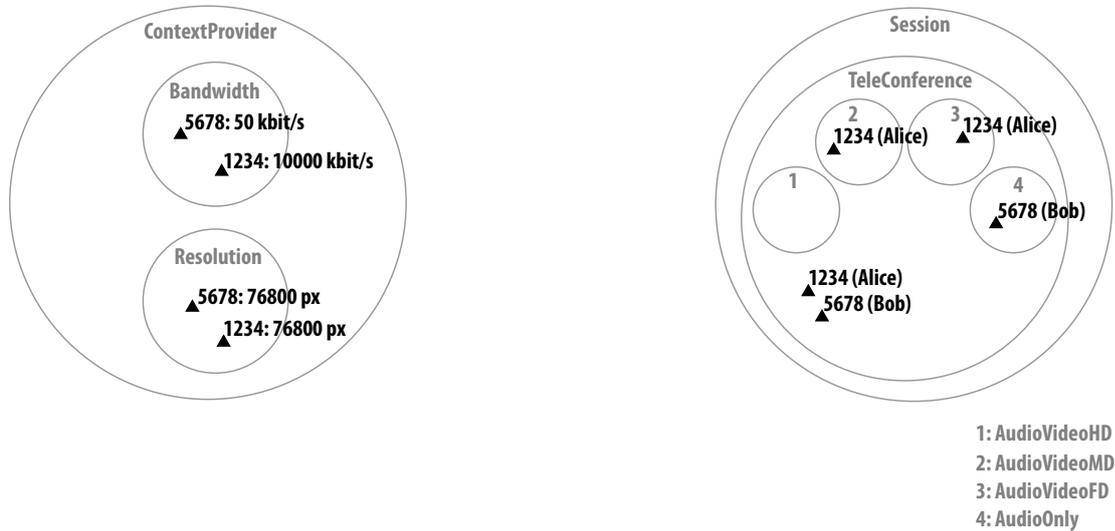


FIGURE 15.6: Ontologie après un changement de situation (ici Bob avec une bande passante de 50 kbit/s)

L'inférence propose pour la session de Bob la modalité d'exécution *AudioOnly* (Figure 15.6) et pour la session d'Alice les modalités *AudioVideoMD* et *AudioVideoFD*.

Le *Context Broker* détecte un changement de modalité d'exécution pour la session de Bob. Elle était précédemment avec la modalité *AudioVideoMD* et elle est maintenant à *AudioOnly*. Le *Context Broker* notifie le service numérique d'un changement de modalité uniquement pour la session de *Bob*. Le service numérique adapte ensuite la modalité d'exécution de Bob.

15.2.6 Fermeture d'une session

À l'issue de la session du service numérique, les informations la concernant sont supprimées : individus représentant la session et ceux représentant les capteurs (Figure 15.7)

15.3 Conclusion

La validation du scénario de référence a été réalisée avec un échantillon de données assez faible. Ceci nous a permis de mettre à l'épreuve notre modèle, mais pas de tester un passage à l'échelle.

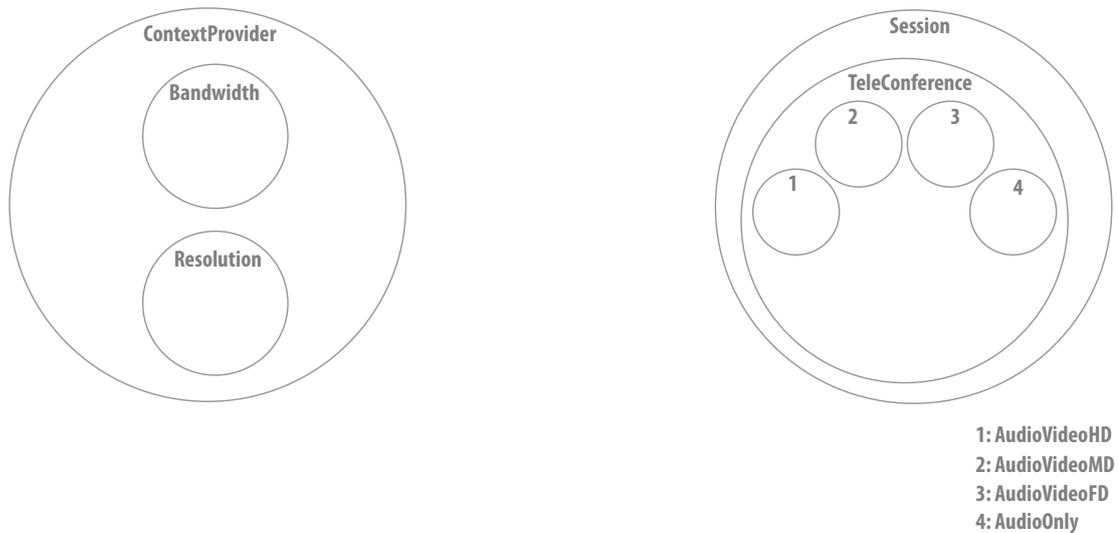


FIGURE 15.7: Ontologie après les terminaisons des sessions

Le scénario de référence a d'abord été mis en oeuvre sous la forme d'une simulation avec le logiciel Protégé-OWL pour évaluer le comportement de l'ontologie. Les règles métiers et les valeurs seuils utilisées sont identiques à celles présentées dans ce chapitre. Cette étude de faisabilité nous a permis de mieux comprendre les possibilités et les limitations des ontologies en informatique, en particulier sur le principe de déduction de la modalité d'exécution la plus adaptée. Il n'existe pas de gestion de priorité pour les règles SWRL, il devient donc très complexe de proposer une signature de contexte qui permette d'élire la modalité d'exécution la plus adaptée. Pour pallier ce problème, nous effectuons avec l'ontologie, une sélection des modalités d'exécution possibles. Le choix de la modalité d'exécution la plus adaptée est décidé par un module complémentaire à la base de connaissance, le *Context Querier*.

Cinquième partie

Conclusion

Chapitre 16

Conclusion générale

Le travail de recherche présenté dans ce manuscrit a pour objectif d'adresser l'enjeu de l'adaptation des services numériques dans les réseaux du futur (ou NGN). Cette problématique est née de la nécessité pour les fournisseurs de services numériques de pouvoir proposer le même service, quel que soit le couple technologie d'accès et terminal utilisateur qui est utilisé pour consommer le service. En effet, un NGN est un réseau convergent en terme de réseau d'accès, de terminaux et de services. Les utilisateurs ont la possibilité de se connecter de différentes façons aux services proposés. Nous parlons de multimodalité d'accès. Par ailleurs, les consommateurs ont la possibilité de changer de modalité au cours de l'exécution, il faut donc pouvoir conserver l'exécution du service. Pour *offrir une continuité de service*, il faut donc adapter les services numériques à leur environnement de consommation.

Une problématique similaire a été soulevée lors de la démocratisation des réseaux mobiles de données. Il s'agissait d'adapter le contenu (images) aux conditions du réseau. Dix ans plus tard, la problématique centrale reste identique (c'est-à-dire adapter le contenu à son environnement de consommation), mais les conditions se sont étendues. Il ne s'agit plus de prendre en compte uniquement la variation du réseau, mais aussi celle du terminal utilisateur. L'adaptation des services numériques dans les réseaux du futur fait l'objet de plusieurs travaux scientifiques [10]. Les approches retenues se focalisent sur la lecture de l'environnement informatique et la capacité à intervenir sur les services numériques. Il existe très peu de travaux sur le processus de décision et d'adaptation.

C'est donc tout naturellement que nous avons centré nos travaux sur cet aspect de la problématique. Nous avons retenu une approche transversale, qui fait référence aux applications sensibles au contexte, aux applications autonomiques et aux outils du Web sémantique.

La solution proposée est la création d'une nouvelle fonction dans les serveurs d'applications : la gestion de la sensibilité au contexte des services numériques qu'il héberge. Ce type de serveur propose des fonctions de gestion du contexte au travers d'un in-

tergiciel. Chaque service numérique s'appuie sur les fonctions du serveur pour devenir sensible au contexte et ainsi pouvoir adapter sa logique métier. L'originalité de notre approche repose sur trois éléments caractéristiques :

- La séparation des rôles.
- La signature du contexte.
- Le traitement du contexte.

Nous souhaitons conserver une distinction entre les deux logiques qui composent un service numérique sensible au contexte. C'est pourquoi la gestion du contexte a été externalisée et mutualisée au sein du serveur d'applications. Ceci permet aux concepteurs de services numériques de ne pas se soucier de la réalisation de la logique de gestion de contexte pour se concentrer sur la logique métier et son adaptation. L'adaptation de la logique métier nécessite la création de modalités d'exécution de la logique métier du service numérique. Un service numérique est un service offrant plusieurs possibilités pour réaliser son objectif, chaque modalité d'exécution correspondant à une ou plusieurs situations d'usage. Le choix d'une modalité d'exécution par rapport à une autre est fait en fonction du contexte de l'utilisateur.

Pour identifier ce contexte, nous nous appuyons sur la signature de contexte [9]. Lors du déploiement elle est séparée de la logique métier du service numérique pour être déployée sur l'intergiciel. Le service numérique est déployé sur le serveur d'applications [8]. La signature de contexte fait partie intégrante du service numérique sensible au contexte puisqu'elle définit les situations d'usage et les modalités d'exécution liées. En plus de permettre l'identification du contexte, la signature joue un rôle pivot puisqu'elle fait le lien entre les deux logiques du service numérique et permet le traitement du contexte.

Le traitement du contexte concerne les opérations de collecte, de normalisation des données et leur agrégation en vue d'une analyse pour déduire la modalité d'exécution adaptée à la situation de l'utilisateur.

Cette chaîne de traitement est réalisée par l'agencement des cinq modules clés de l'intergiciel dans la séquence suivante :

- *Harvester* : Collecte des données sur l'environnement des terminaux.
- *Semantic Formalizer* : Transforme les données brutes des capteurs en information.
- *Knowledge Base* : Déduit, à partir des informations de contexte et des services numériques, les situations d'usage des utilisateurs.
- *Context Querier* : Élit la modalité d'exécution la plus adaptée pour chaque session des services numériques hébergés.
- *Context Broker* : Communique les modalités d'exécution aux services numériques.

La déduction utilise :

- Les connaissances sur l'environnement, fournies par des éléments tiers (réseaux de capteurs, etc.)

- Les connaissances sur le service numérique (signature de contexte).

L'objectif d'adaptation des services numériques dans les réseaux du futur est réalisé puisque nous arrivons à identifier la situation d'exécution et proposer la modalité d'exécution adaptée à l'utilisateur, en fonction des propositions des concepteurs du service numérique.

Parmi les pistes d'amélioration de ce travail, nous retrouvons les processus liés au choix de la modalité d'exécution (*Context Querier*). Dans la proposition actuelle, nous ne prenons pas en compte l'analyse des adaptations précédentes. Le rajout d'un module qui prend en charge cette fonction d'analyse permettrait de proposer, dans un premier temps, un système gestion des modifications intempestives dues à un environnement trop variable. En effet lorsque l'environnement est volatile, le système proposera les adaptations correspondantes, ce qui aura pour effet de rendre le service numérique instable en adaptant en permanence celui-ci.

Dans un second temps, cet ajout permettrait de rendre les adaptations proactives avec un système d'apprentissage automatisé. Pour mettre en place un tel système, il faudrait pouvoir prendre en compte une évaluation basée sur le retour de l'utilisateur, rapprochant ainsi ces travaux de la notion de qualité d'expérience. Nous pouvons même imaginer compléter l'analyse avec la personnalisation des modalités d'exécution à utiliser selon les situations d'usage vécues par l'utilisateur.

Mais l'amélioration la plus importante à mes yeux reste un travail autour de la signature de contexte. Actuellement, le développement de la signature de contexte nécessite d'énumérer les sources de contexte déclarées sur le serveur d'applications puis de lister l'ensemble des situations d'usage. Dans cette situation, nous atteignons très rapidement les limites de cette proposition puisque la multiplication des paramètres d'évaluation complexifie énormément la création des règles composant la signature de contexte. Il devient alors fastidieux de modéliser l'ensemble des règles qui composent la signature de contexte sans faire d'oubli ou de règles contradictoires. Un outil de développement assisté de signature de contexte permettrait de rendre ces travaux beaucoup plus utilisables.

Sixième partie

Annexes

Annexe A

Le contrôle de session et le réseau coeur

Les fonctions de contrôle de session (section 4.6.1) sont réparties sur trois entités :

- Proxy CSCF (P-CSCF)
- Interrogating CSCF (I-CSCF)
- Serving CSCF (S-CSCF)

Chaque entité a une fonction déterminée au sein du réseau : respectivement serveur proxy, serveur d'interrogation ou serveur de service. Associé à ces trois équipements, on retrouve dans le coeur de réseau IMS, un quatrième équipement : le HSS (Home Subscriber Server).

A.1 La base utilisateur

Le HSS est la base de données utilisateur du réseau. Cette entité est l'équivalente du *Home Location Register (HLR)* présent dans les réseaux GSM. Elle contient toutes les données nécessaires à l'accès au réseau et aux services. Ces informations sont utilisées par les utilisateurs et les équipements (informations d'authentification, d'autorisation, profils d'accès...). Les informations indispensables pour IMS sont :

- Les abonnements de l'utilisateur en terme de sessions autorisées,
- La localisation,
- L'authentification et l'autorisation,
- Les abonnements aux services du réseau,
- Le serveur de service S-CSCF alloué à l'utilisateur.

Le schéma ci-dessous présente le HSS et son écosystème.

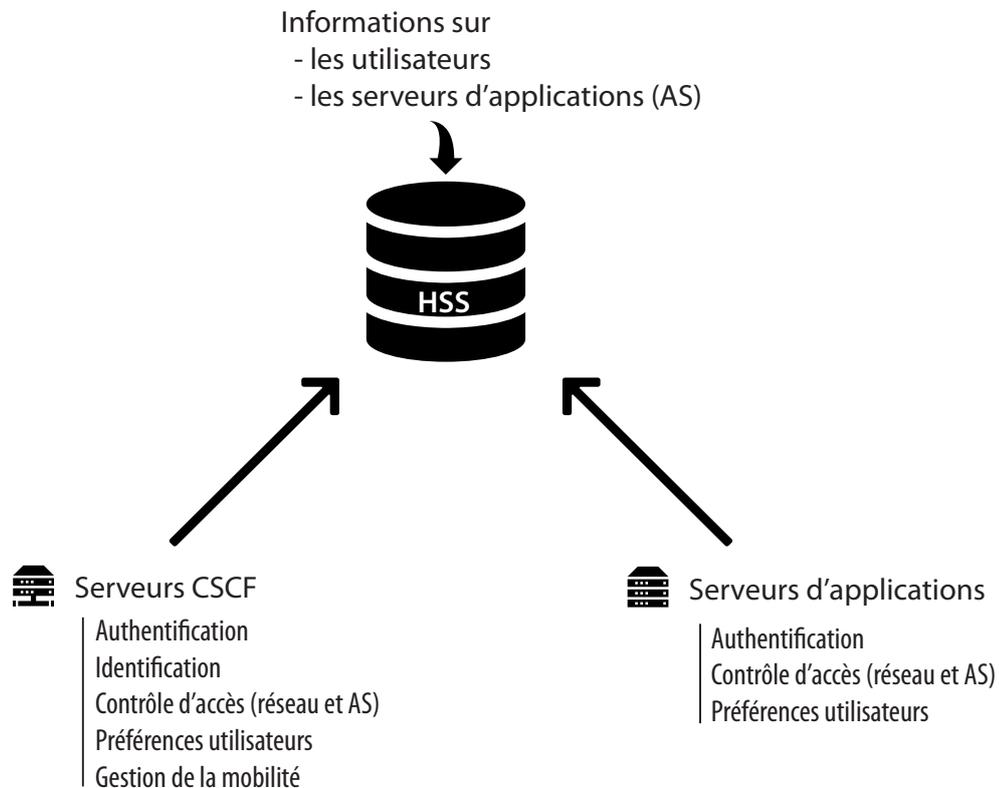


FIGURE A.1: IMS : fonctions du HSS et son environnement [81]

Cette base de données utilisateur est accessible uniquement par les entités de contrôle du réseau et dans certaines situations par les serveurs d'applications. Bien que le HSS soit moins mis en avant dans les descriptions d'IMS, il n'en reste pas moins un élément clé d'IMS. Il sert de support pour les fonctions d'authentification, de mobilité, d'identification et d'accès aux services.

A.2 P-CSCF

Le *P-CSCF* est un serveur proxy, il sert d'intermédiaire entre le terminal utilisateur et le réseau IMS. C'est le premier point de contact de l'utilisateur avec le réseau IMS. Il reste attribué au même équipement utilisateur tout au long de la session. C'est l'interlocuteur privilégié du terminal utilisateur dans le plan de contrôle.

La procédure de contact du *P-CSCF* commence par l'autorisation du terminal sur le

réseau. Cette identification est réalisée par la couche de transport. Dès que l'équipement a la possibilité d'échanger des données sur le réseau (connectivité IP¹), il recherche et contact le P-CSCF. Cette phase de prise de contact permet l'authentification de l'utilisateur. Une fois l'utilisateur authentifié, le *P-CSSF* certifie l'identité de l'utilisateur pour les autres noeuds du réseau. Il conserve un lien constant entre l'utilisateur et le terminal.

Au-delà d'une simple table de correspondance, il joue le rôle de proxy de sécurité en s'assurant que les messages SIP sont bien formés. Ce proxy est sollicité tout au long de la session de l'utilisateur dans le réseau IMS, qu'il existe ou non du trafic. Le passage à l'échelle est assuré par un groupe de *P-CSCF* disséminé dans le réseau. Malgré la présence de plusieurs *P-CSCF*, il n'est pas possible pour l'utilisateur d'en changer au cours d'une session réseau.

Les messages SIP en provenance de l'utilisateur sont relayés par le *P-CSCF*. Ils sont ensuite transférés à une des deux autres entités (I-CSCF ou S-CSCF). Ce sont des unités de traitement des messages de signalisation.

A.3 I-CSCF

L'*I-CSCF* est un serveur de routage de messages. Il prend en charge les requêtes en provenance du P-CSCF et les transfère :

- Si c'est une demande d'authentification, il route le message vers un S-CSCF sans distinction préalable.
- Si l'utilisateur est déjà enregistré, il route les messages vers le S-CSCF qui a authentifié l'utilisateur, pour les raisons évoquées ci-dessus.

A.4 S-CSCF

Le *S-CSCF* est à la fois une entité SIP standard (Registrar) et un serveur SIP avec sa propre logique métier. Un serveur d'enregistrement SIP (Registrar) authentifie et enregistre les utilisateurs. Chaque utilisateur est identifié par une URI SIP (Tableau B.1). L'enregistrement associe la ou les adresses IP utilisées par l'utilisateur avec son URI. Autrement dit, un utilisateur peut être connecté simultanément sur plusieurs terminaux.

Concrètement, un *S-CSCF* :

- Récupère dans la base HSS les informations relatives à l'utilisateur pour les mettre en cache,
- Associe l'adresse IP utilisée par les terminaux de l'utilisateur avec l'URI de ces derniers,

1. Interface avec une adresse IP reconnue par le réseau

- Effectue les requêtes de bande passante auprès des éléments de la couche transport,
- Appelle les fonctions de facturation,
- Contrôle l'accès aux ressources en utilisant le profil de l'utilisateur rapatrié depuis le HSS (Figure A.1),
- Route les messages SIP vers les autres entités du réseau.

Le routage des messages SIP dépend de l'opération à effectuer :

- Une consommation de services transfère le message vers le serveur d'applications correspondant,
- Une communication vers un autre utilisateur du réseau route le message SIP vers l'UE ou un autre S-CSCF,
- Une communication vers un utilisateur IMS extérieur au réseau route le message vers le I-CSCF,
- Une communication vers un utilisateur de réseau traditionnel route la requête vers les passerelles correspondantes.

Comme pour le P-CSCF, il n'est pas possible pour l'utilisateur de changer de serveur de service au cours d'une session réseau.

A.5 Conclusion

Les entités de gestion et de contrôle sont au nombre de trois : un *P-CSCF*, un *I-CSCF* et un *S-CSCF*. D'un point de vue SIP, leurs fonctions sont respectivement : proxy, routeur et registrar. Leur travail ne concerne pas uniquement le contrôle d'accès et l'authentification des utilisateurs sur le réseau. Les *CSCF* supervisent l'ensemble des sessions d'un réseau IMS :

- L'ouverture de session entre utilisateurs d'un même réseau
- L'ouverture de session entre utilisateurs d'un autre réseau IMS.
- L'ouverture de session entre utilisateurs et serveurs d'applications.

Ces trois équipements (*CSCF*) s'appuient sur le *HSS* qui leur sert de support de stockage.

Annexe B

Technologies du web sémantique : les URI et RDF

B.1 URI

L'URI (Uniform Resource Identifier) est un système d'identification normalisé de ressources réseau dont la syntaxe est normalisée dans la RFC 3986. Elle a été créée pour proposer un système d'identification à la fois compréhensible par les programmes informatiques, ainsi que lisible et mémorisable par les humains. La syntaxe générique définie dans la RFC précise que l'URI est composé des éléments présentés ci-dessous.

TABLEAU B.1: URI

Élément	Remarque
Plan	Définit la syntaxe du Chemin
:	
Autorité	Définit qui gère l'URI (optionnel)
Chemin	Précise la ressource
Requête	Se situe après le "?" (optionnel)
Fragment	Se situe après le "#" (optionnel)

Par exemple, si l'on décompose l'URI **http ://www.google.fr/?hl=fr&q=test** en suivant le schéma proposé plus haut, nous obtenons :

- ***http*** est le plan qui définit la syntaxe du chemin.
- ***www.google.fr*** est l'autorité.
- **/*** est le chemin d'accès à la ressource.
- ***hl=fr&q=test*** représente la requête sur cette ressource.
- Aucun fragment ne compose cette requête.

Bien qu'HTTP soit l'utilisation la plus courante de ce système d'URI, il en existe bien d'autres :

- `ftp://ftp.is.co.za/rfc/rfc1808.txt`
- `ldap://[2001:db8::7]/c=GB?objectClass?one`
- `mailto:John.Doe@example.com`
- `news:comp.infosystems.www.servers.unix`
- `tel:+1-816-555-1212`
- `telnet://192.0.2.16:80/`
- `urn:oasis:names:specification:docbook:dtd:xml:4.1.2`

Nous retrouvons, dans ces exemples, deux catégories d'éléments sous-jacents au système URI :

- URL (Uniform Resource Locator)
- URN (Uniform Resource Name).

Le premier système permet d'identifier, mais aussi de localiser des ressources. Il indique comment on récupère la donnée, on parle d'URI déréréférençable¹. Le second système sert uniquement à identifier.

B.2 RDF

RDF est un format de données utilisé pour décrire des ressources (identifiés par des URI). Cette description est réalisée par l'ajout de métadonnées qui suivent le principe du modèle entité-relations. Ainsi chaque ressource (livre, personne, page web...) est décrite au travers de ses relations avec d'autres ressources. Pour cela, le modèle RDF s'appuie sur deux principes :

- Le modèle entité-association, où chaque entité possède des relations avec d'autres entités, sous la forme de triplets.
- La théorie des graphes.

1. <http://www.w3.org/2001/tag/awwsw/issue57/20110327/#ir-ref>

B.2.1 Modèle entité-association

Les relations sont décrites sous la forme d'un graphe, dont les noeuds sont soit une ressource, soit une valeur reliée par un arc nommé. La description d'une ressource sous la forme "Sujet-Prédicat-Objet" fait référence à la forme grammaticale d'une phrase, Sujet-Verbe-Complément. Autrement dit, un triplet RDF est composé de :

- Une ressource à décrire (sujet),
- Une propriété de la ressource (prédicat),
- La valeur de cette propriété (objet).

Chaque triplet exprime une assertion. Si nous prenons l'exemple de l'assertion suivante

L'article <http://www.w3.org/TR/rdf-syntax-grammar> a pour titre "RDF/XML Syntax Specification (Revised)"

Cette assertion se traduit en triplet :

- Sujet : <http://www.w3.org/TR/rdf-syntax-grammar>
- Prédicat : titre
- Objet : RDF/XML Syntax Specification (Revised)

Pour pouvoir offrir un traitement automatisé des données comme cela doit être le cas avec RDF, il faut un formalisme et un référentiel commun (vocabulaire). Ce dernier permet de définir la propriété (prédicat). Il est défini par les schémas RDF, une autre technologie proposée par le W3C. Les schémas sont identifiés et accessibles par une URI. Ainsi en définissant le rôle *titre*, l'exemple présenté ci-dessus devient :

- Sujet : <http://www.w3.org/TR/rdf-syntax-grammar>
- Prédicat : <http://purl.org/dc/elements/1.1/title>
- Objet : "RDF/XML Syntax Specification (Revised)"

B.2.2 Graphe

Chaque assertion permet de construire un graphe RDF, qui utilise les conventions suivantes :

- Les *sujets* et *objets* sont les noeuds de ces graphes, représentés soit par un rectangle (*littérale*), soit par un ovale (*ressource*).
- Les *prédicats* sont des flèches orientées.

Cette représentation graphique (Figure B.1) donne tout son sens à la description des ressources par RDF.

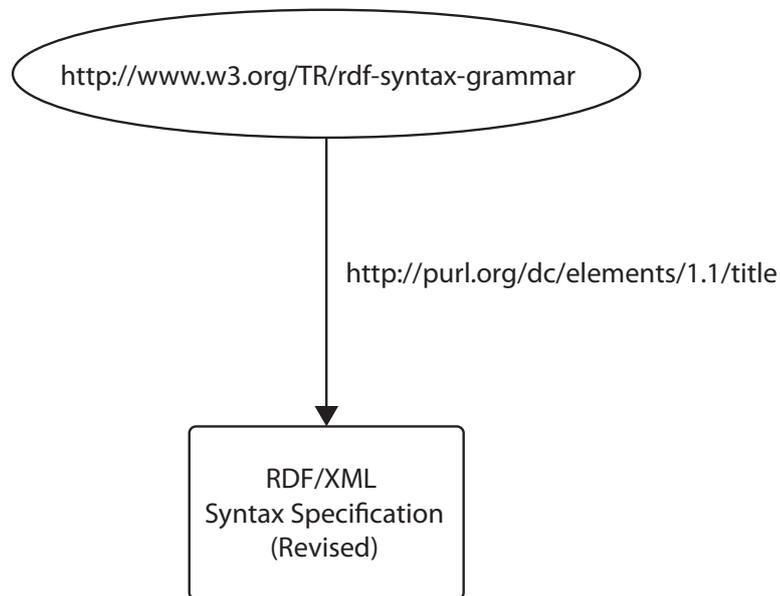


FIGURE B.1: Exemple d'un graphe RDF orienté

B.2.3 Sérialisation

L'échange et le traitement des données par des logiciels ou des humains passent par la sérialisation des données. Nous ne rentrerons pas dans le détail, mais rappellerons qu'il existe plusieurs formats de sérialisation pour les outils sémantiques :

- RDF/XML (sérialisation XML)
- Notation3 (sérialisation non-XML)
- RDFa (sérialisation HTML)

B.2.4 RDF Schema

RDF Schema étend RDF à l'aide d'un métamodèle qui propose de nouvelles primitives (Tableau B.2).

Cette extension permet de définir le vocabulaire utilisé pour décrire les ressources. Il existe déjà plusieurs RDF-Schema universels, comme le *Dublin Core*² qui souhaite s'imposer comme norme pour l'exploitation de ressources en permettant d'avoir un système de métadonnées (15 termes) partagé entre tous et compréhensible par tous (humains,

2. <http://purl.org/dc/elements/1.1/>

TABLEAU B.2: Primitives RDF Schema

Primitive	Traduction	Description
Class	Classe	Déclare une ressource comme une classe
subClassOf	Sous-classe	Déclare une 1 relation de hiérarchie pour permettre la création de hiérarchie (spécialisation)
Type	Type	Définit le type (Class, Property...)
domain	Domaine de définition	Définit la classe du sujet (type de sujet autorisé pour une propriété)
range	Ensemble d'arrivée	Définit la classe de l'objet (type de valeur autorisé pour une propriété)
subPropertyOf	Sous-propriété de	Spécialisation des propriétés

machines). Un second RDF-Schema est très utilisé, celui proposé par le projet FOAF³. C'est un vocabulaire utilisé pour exprimer les informations concernant des personnes.

B.2.5 Web de données (Linked Data)

Linked Data est le terme retenu par Tim Berners-Lee pour définir la *web des données*. Derrière ce terme se cache la réalisation d'une vision née en 1998 et mise en pratique en 2006. Il s'agit de publier des données structurées et reliées entre elles.

Pour réaliser cette vision, Tim Berners-Lee propose quatre règles fondamentales⁴ :

- Utiliser les URI pour identifier les choses,
- Utiliser les URI HTTP pour récupérer les ressources,
- Fournir dans les URI des renseignements intelligibles par les machines et les hommes,
- Inclure d'autres URI pour améliorer la découverte d'autres informations (données liées).

C'est autour du projet du W3C *Linking Open Data*⁵ que se réalise cette vision proposée par Tim Berners-Lee⁶, dans lequel on retrouve la notion de *Linked Data* et d'*Open Data*.

Le principe du mouvement *Open Data* est de rendre accessibles les données, des entreprises ou états, sur le web et créer ainsi un web de données. Cependant, ces initiatives sont locales et peu coordonnées. Autrement dit, les *données ouvertes (Open Data)* sont des données exposées sur le web, qui ne sont pas forcément reliées entre elles et avec des formats parfois exotiques. L'objectif du projet *linking Open Data* est donc de pouvoir proposer des données ouvertes structurées et liées entre elles. Ce projet est un retour

3. <http://www.foaf-project.org/>

4. <http://www.w3.org/DesignIssues/LinkedData.html>

5. <http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

6. http://www.ted.com/talks/lang/en/tim_berniers_lee_on_the_next_web.html

aux sources avec une mise en place d'un web de données structurées qui est consommé par des machines. Une étape indispensable pour le traitement de cette masse croissante de données.

Annexe C

Composition d'un message SIP

Session Initiation Protocol (SIP) est le protocole choisi pour le contrôle de session (ou signalisation) dans IMS. Il a été créé pour établir et contrôler des sessions multimédia sur les réseaux IP. On le retrouve principalement dans les applications de VoIP. Sa simplicité (encodage au format texte) et son fonctionnement basé sur SMTP/HTTP sont les clés de son succès. L'encodage au format texte est, à la fois, une de ses grandes forces et une faiblesse. Cela lui permet d'être lisible par les machines et les humains tout en permettant d'être étendu à souhait. Mais l'échange d'un grand nombre de messages SIP peut mener à l'encombrement du réseau. Pour pallier à ce problème, IMS propose une compression des messages. Le protocole SIP se base sur le modèle requêtes/réponses. Chaque message est donc soit une requête, soit une réponse.

La structure d'un message SIP est découpée en 3 parties :

- La Ligne de début
- L'entête
- Le corps du message

C.1 La ligne de début

Le type de message (requête ou réponse) détermine le format de la première ligne. Une requête est composée d'une méthode, d'un destinataire et de la version du protocole. Il existe 6 méthodes principales pour une requête SIP : INVITE, ACK, BYE, CANCEL, OPTIONS et REGISTER. Une réponse est composée de la version du protocole, d'un code d'état et de sa traduction textuelle. Nous retrouvons une classification identique à celle des codes d'état HTTP. Ainsi, ils s'étendent de 100 à 699 (Tableau C.1) et répartis en 6 catégories.

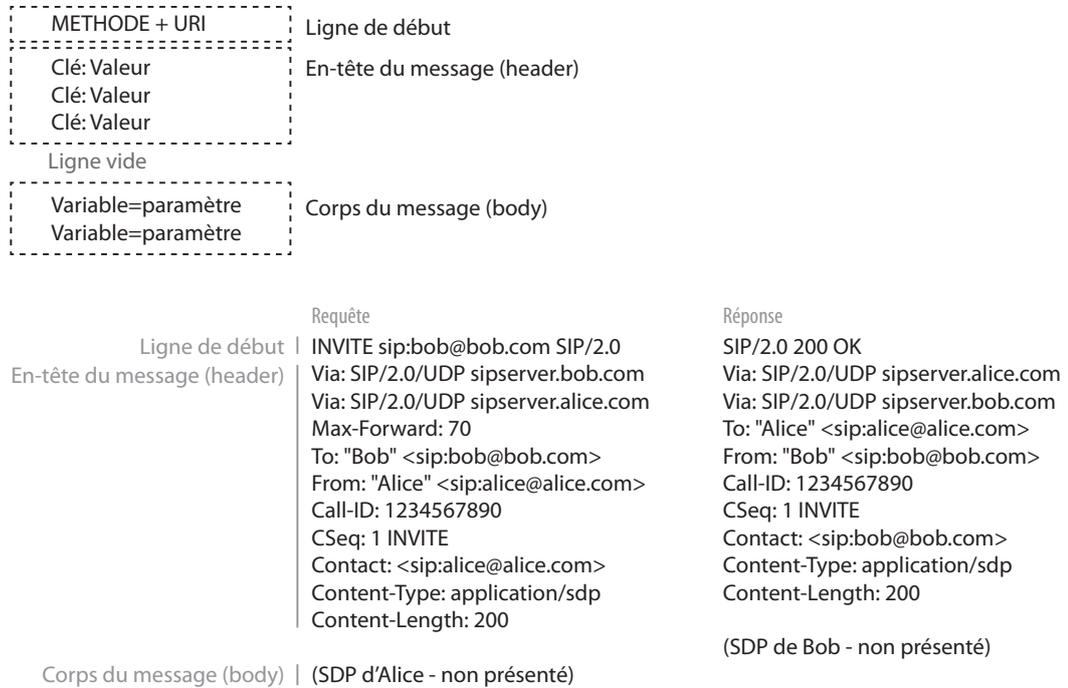


FIGURE C.1: Composition d'un message SIP

TABLEAU C.1: Code d'état SIP

Code d'état	Catégorie
1xx	Demande d'informations complémentaires
2xx	Réussite
3xx	Redirection
4xx	Erreur du client
5xx	Erreur du serveur
6xx	Erreur généralisée

C.2 L'entête

C'est dans cette partie que sont regroupées les informations relatives au message (expéditeur, destinataire...). Six champs sont indispensables (Tableau C.2) pour que l'entête de message soit considéré comme bien formé.

TABLEAU C.2: Les 6 champs d'entête obligatoire

Champs	Description
To	Destinataire du message.
From	Expéditeur du message
Cseq	Numéro de séquence du message dans l'échange
Call-ID	Identifiant unique pour l'échange de message
Max-forwards	Nombre maximum de noeuds de routage.
Via	Indique les proxy utilisés

C.3 Le corps du message

Le corps du message est séparé de l'entête par une ligne vide. Il n'existe pas de format spécifique pour le corps du message. Cette absence de formalisme lui permet d'être étendu et modifié à souhait. À l'instar des courriels, il utilise le codage MIME (Multipurpose Internet Mail Extensions) pour embarquer des documents binaires dans son corps de message. Mais il reste principalement utilisé comme protocole de signalisation et non de transport.

Le corps d'un message SIP peut être étendu avec le protocole Session Description Protocole (SDP). Ce dernier est un formalisme de caractérisation de médias, normalisé par l'IETF. Les informations sont codées textuellement et relatent les éléments tels que l'adresse IP et les ports, le type de média, l'encodage, les protocoles de transport.

Annexe D

Installation d'OpenIMS

D.1 Préparation du système

Le système d'exploitation recommandé pour le déploiement de l'architecture OpenIMS¹ est une version Linux Debian ou Ubuntu avec un noyau 2.6. En effet parmi les méthodes d'installation disponibles pour OpenIMS il existe des paquets “non-supportés”² ainsi qu'un tutoriel permettant de construire ses packages Debian à l'aide des sources. Le système de base doit avoir les pré-requis suivant :

- GCC3/4, JDK 1.6, ant et make
- MySQL
- Bison, Flex
- libxml2 (>2.6), libmysql (development version)
- bind9
- ipsec-tools et openssl pour l'utilisation des système IPsec et TLS Security

L'ensemble de ces outils peut être installé à l'aide des commandes suivantes³ :

```
sudo apt-get update -y
apt-get install linux-headers-2.6.28-13-server
apt-get install ssh subversion ant sun-java6-jdk bison flex
apt-get install mysql-server libmysql++-dev make gcc
apt-get install bind9 pure-ftpd ipsec-tools
apt-get install libmysqlclient16-dev libxml2 libxml2-de
apt-get remove avahi-daemon -y
```

1. <http://www.openimscore.org/>

2. non-supportés signifie que ces packages ne sont pas officiel et qu'aucun support n'est offert. Cependant ils peuvent-être totalement fonctionnels

3. <http://www.forum-ims.org/viewtopic.php?f=19&t=26>

D.2 Installation d'un environnement graphique persistant accessible via VNC

Installation de l'ensemble des éléments (environnement X11, Window Manager et VNCServer)

```
sudo apt-get install xorg openbox vnc4server
```

Configuration de VNCServer (ajout du mot de passe, configuration de l'environnement persistant)

```
vncserver -geometry 1280x1024 -depth 24
vncserver -kill :1
vim .vnc/xstartup
```

Décommenter les lignes 3 et 4. Sur la ligne 4 ajouter sh entre exec et /etc. Modifier la valeur xterm-geometry par 1280x1024+10+10.

```
.vnc/xstartup
#!/bin/sh
#Uncomment the following two lines for normal desktop:
unset SESSION_MANAGER
exec sh /etc/X11/xinit/xinitrc
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdp $HOME/.Xresources
xsetroot - solid grey
vncconfig -iconic &
xterm -geometry 1280x1024+10+10 -ls -title "$VNCDESKTOP Desktop" &
twm &
```

Lancement de VNCServer

```
vncserver -geometry 1280x1024 -depth 24
```

Connexion sur le serveur

La connexion se fait sur ip:5901.

D.3 Installation d'OpenIMS

L'installation d'OpenIMS peut se faire à partir de 3 formes :

- Téléchargement et compilation des sources.
- Téléchargement et création des package Debian.
- Installation à partir des packages Debian “non-supportés”.

D.3.1 Pré-requis à toute installation

Quelle que soit la version d'OpenIMS, il est pré-configuré pour fonctionner dans les conditions suivantes :

- Répertoire de base : */opt/OpenIMSCore*
- Adresse IP : *127.0.0.1*

Création du répertoire de base d'OpenIMSCore

```
mkdir /opt/OpenIMSCore
```

Le HSS ne peut pas fonctionner sans Java JDK ≥ 1.5 .

```
java -version
```

D.3.2 Récupération des sources

Le code est disponible sur <http://svn.berlios.de/svnroot/repos/openimscore> (sub-version doit être installé).

Les deux éléments à installer sont les éléments coeur (CSCF) et la base utilisateur (HSS) :

- Les CSCFs : http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk
- HSS : <http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk>

Il faut se rendre dans le répertoire d'OpenIMSCore puis télécharger les sources via SVN. Dans un premier temps, les éléments coeur. Puis dans un second temps la base utilisateurs.

```
cd /opt/OpenIMSCore
mkdir ser_ims
svn checkout http://svn.berlios.de/svnroot/repos/openimscore/ser_ims/trunk ser_ims

cd /opt/OpenIMSCore
mkdir FHoSS
svn checkout http://svn.berlios.de/svnroot/repos/openimscore/FHoSS/trunk FHoSS
```

D.3.3 Compilation des sources

```
cd /opt/OpenIMSCore/ser_ims
make install-libs all
cd /opt/OpenIMSCore/FHoSS
ant compile
ant deploy
```

D.4 Configuration de l'environnement

D.4.1 Serveur DNS

La résolution de nom permet entre autre l'utilisation des URI SIP aussi bien sur les clients qu'entre les éléments d'IMS. Deux possibilités nous sont offertes :

- Utilisation d'un serveur DNS indépendant -> Dans le cas d'une installation sur un nouveau réseau

- Utilisation d'un serveur DNS maître - Intégration dans un réseau. Le cas de EU-ROP.

DNS indépendant : Installation sur un nouveau réseau

Copie du fichier zone proposé dans l'installation puis modification du fichier de configuration de Bind /etc/bind/named.conf.local

```
sudo cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone /etc/bind/  
sudo vi /etc/bind/named.conf.local
```

Edition du fichier *named.conf.local*

```
Zone "open-ims.test" IN {  
    Type master;  
    File "etc/bind/open-ims.dnszone";  
    Notify no;  
};
```

Redémarrage du serveur DNS.

```
sudo /etc/init.d/bind9 restart
```

DNS existant : Installation sur un réseau existant - type EUROP

Dans cette architecture, le réseau existant dispose de :

- 1x serveur DHCP qui offre les adresses IP
- 1x server DNS qui offre la résolution de nom
- 1x passerelle pour l'accès internet

Il est recommandé d'utiliser une architecture maître-esclave pour la résolution de nom. Dans cette configuration, un serveur DNS est assigné à un domaine (il est le maître). Le serveur esclave s'appuie sur le maître pour répondre aux requêtes

Dans cet exemple :

- Le serveur DNS du réseau local est : 192.168.8.101 - Il est le serveur esclave pour le domaine open-ims.test
- Le serveur DNS pour le domaine OpenIMS est : 192.168.8.131 - Il est le serveur maître pour le domaine open-ims.test

Configuration du serveur DNS du réseau local

```
sudo vi /etc/bind/named.conf.local
```

```
named.conf.local
Zone "open-ims.test" IN {
    Type slave;
    File "open-ims.dnszone";
    masters { 192.168.8.131; };
};
```

```
rm /var/cache/bind/open-ims.dnszone
sudo /etc/init.d/bind9 restart
```

Configuration du serveur DNS OpenIMS. Copier puis modifier le fichier de zone.

```
sudo cp /opt/OpenIMSCore/ser_ims/cfg/open-ims.dnszone /etc/bind/
```

Modifier la configuration de Bind9 pour prendre en compte cette nouvelle zone

```
sudo vi /etc/bind/named.conf.local
```

```
named.conf.local
Zone "open-ims.test" IN {
    Type master;
    file "/etc/bind/open-ims.dnszone";
    notify yes;
};
```

```
sudo vi /etc/bind/named.conf.options

named.conf.options
    allow-transfer { 192.168.8.101; };

sudo /etc/init.d/bind9 restart
```

D.4.2 MySQL

L'ensemble de l'architecture OpenIMS s'appuie sur une base de données de type MySQL. Les configurations initiales sont fournies.

```
mysql -u root -p -h localhost < /opt/OpenIMSCore/ser_ims/cfg/icscf.sql
mysql -u root -p -h localhost < /opt/OpenIMSCore/FHoSS/scripts/hss_db.sql
mysql -u root -p -h localhost < /opt/OpenIMSCore/FHoSS/scripts/userdata.sql
```

D.4.3 Fichiers de configuration et scripts

Copier l'ensemble des fichiers de configuration et des scripts de lancement dans le répertoire de base d'OpenIMS.

```
cd /opt/OpenIMSCore
cp ser_ims/cfg/*.cfg .
cp ser_ims/cfg/*.xml .
cp ser_ims/cfg/*.sh .
```

Pour un fonctionnement avec une adresse IP différente de 127.0.0.1 il est nécessaire d'éditer des fichiers de configuration. Dans ces fichiers il faut spécifier l'adresse IP sur laquelle le service sera en écoute. Par défaut, cette adresse est 127.0.0.1.

Les fichiers à modifier sont ceux présents dans /opt/OpenIMSCore :

```
icscf.cfg & icscf.xml
pcscf.cfg & pcscf.xml
```

```
scscf.cfg & scscf.xml
```

Ainsi que les fichiers présents dans */opt/OpenIMSCore/FHoSS/deploy* :

```
DiameterPeerHSS.xml (FHoSS/deploy)  
hss.properties (FHoSS/deploy)
```

D.4.4 Utilisation de l'authentification MD5 (plus grande compatibilité)

Editer le fichier de configuration *scscf.cfg* et activer l'authentification par défaut en MD5 (au lieu de AKAv1-MD5)

D.5 Utilisation d'OpenIMS

D.5.1 Démarrage des noeuds

```
cd /opt/OpenIMSCore  
./pcscf.sh  
./scscf.sh  
./icscf.sh  
cd /opt/OpenIMSCore FHoSS/deploy  
sh startup.sh
```

D.5.2 Ajout d'un UE

Les UE sont les terminaux utilisateurs d'un réseau IMS. Ce sont des équipements qui utilisent SIP.

Clients SIP gratuits :

- X-Lite 6
- UCT IMS Client
- FOKU'S Monster
- Mercurio IMS Client 7

Comptes de démonstration

Alice :

- Username : alice
- Password : alice
- Domain : open-ims.test
- Authorization user name : alice@open-ims.test

Bob : - Username : bob - Password : bob - Domain : open-ims.test - Authorization user name : bob@open-ims.test

Configuration

Utiliser un des deux comptes par défaut ou en créer un nouveau. Le client se connecte via un proxy *sip :pcscf.open-ims.test :4060*

D.5.3 Ajout d'un serveur d'applications

D.5.4 Installation du PDF

```
cd /opt/OpenIMSCore/  
wget http://download.berlios.de/uctimsclient/uctpcrf1.0.2.tar.gz  
sudo tar -xvzf uctpcrf1.0.2.tar.gz  
sudo apt-get install libxerces2-java libcommons-httpclient-java tomcat6  
export JAVA_HOME="/usr/lib/jvm/java-6-sun"  
  
cd /opt/OpenIMSCore/uctpcrf:  
./compile
```

Modifier les entrées DNS

```
pcrf          1D IN A          192.168.8.133  
pcef          1D IN A          192.168.8.133
```

```
cd /opt/OpenIMSCore/uctpcrf  
sudo cp -r uct_web_pcm /var/lib/tomcat6/webapps/  
cd config  
sudo cp uct_web_pcm.policy /etc/tomcat6/policy.d/
```

```
chmod uog+rw /opt/OpenIMSCore/uctpcrf/policies/domain_policies.xml
```

```
chmod uog+rw /opt/OpenIMSCore/uctpcrf/policies/default_domain_policies.xml
chmod uog+rw /opt/OpenIMSCore/uctpcrf/config/pcef.properties
chmod uog+rw /opt/OpenIMSCore/uctpcrf/config/pcrf.properties
```

```
sudo /etc/init.d/tomcat6 restart
```

```
cd /opt/OpenIMSCore/uctpcrf/config
```

```
cp pcscf.qos.cfg pcscf.qos.rtp.cfg pcscf.qos.xml pcscf.qos.sh ims.sh ims.qos.sh
```

Compatibilité descendante avec l'ancienne version de OpenIMSCore

```
mkdir /opt/OpenIMSCore/ser_ims_qos
```

```
sudo cp -R /opt/OpenIMSCore/ser_ims/* /opt/OpenIMSCore/ser_ims_qos/
```

Bibliographie

- [1] 3GPP. 3GPP TS 23.002 - Network architecture. Technical report.
- [2] 3GPP. 3GPP TS 22.228 - Service requirements for the Internet Protocol (IP) multimedia core network subsystem (IMS); Stage 1. Technical report, 2004.
- [3] A. Aamodt and M. Nygard. Different roles and mutual dependencies of data, information, and knowledge—An AI perspective on their integration. *Data & Knowledge Engineering*, 16(3) :191–222, September 1995.
- [4] C. Balakrishna and K. Al-Begain. Towards a User-Centric and Quality-Aware Multimedia Service Delivery Implementation on IP Multimedia Subsystem. *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)*, pages 36–42, September 2007.
- [5] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4) :263–277, 2007.
- [6] M. Bazire and P. Brézillon. Understanding Context Before Using It. In *5th International and Interdisciplinary Conference on Modeling and Using Context*, pages 29 – 40, 2005.
- [7] G. Bertrand. The IP Multimedia Subsystem in Next Generation Networks. *Rapport technique, ENST Bretagne*, 7(March), 2007.
- [8] Y.-G. Billet, C. Gravier, and J. Fayolle. Context-awareness for Next-Generation Applications Servers. In *CONTEXT 2011, 7th Modeling and Reasoning in Context workshop, Karlsruhe*, 2011.
- [9] Y.-G. Billet, C. Gravier, and J. Fayolle. SWRL-based context awareness for application servers hosting digital services. In *Proceedings of the 5th international conference on Rule-based modeling and computing on the semantic web*, pages 222–229, Ft. Lauderdale, FL, USA, 2011. Springer-Verlag.
- [10] Y.-G. Billet, C. Gravier, and J. Fayolle. Advances on Adaptive Systems in NGN. *International Journal of Mobile Computing and Multimedia Communications*, 4(1) :69–78, 2012.

- [11] L. Boula, H. Koumaras, and A. Kourtis. An Enhanced IMS Architecture Featuring Cross-Layer Monitoring and Adaptation Mechanisms. In *Autonomic and Autonomous Systems, 2009. ICAS'09. Fifth International Conference on*, pages 283–288. IEEE, 2009.
- [12] P. Brézillon. Context-based Modeling of Operators' Practices by Contextual Graphs. In *Proceedings of HCP-2003, the 14th Mini-Euro Conference on Human Centered Processes*, pages 129–137, 2003.
- [13] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins. "What are ontologies and why do we need them?" *IEEE Intelligent Systems*, 14(1) :20–26, 1998.
- [14] G. Chen and D. Kotz. A Survey of Context-Aware Mobile Computing Research. Technical report, Dartmouth College, Computer Science, 2000.
- [15] B H C Cheng, R De Lemos, H Giese, P Inverardi, J Magee, J Andersson, B Becker, N Bencomo, Y Brun, B Cukic, G Di Marzo Serugendo, S Dustdar, A Finkelstein, C Gacek, K Geihs, V Grassi, G Karsai, H M Kienle, J Kramer, M Litoiu, S Malek, R Mirandola, H A Muller, S Park, M Shaw, M Tichy, M Tivoli, D Weyns, and J Whittle. Software Engineering for Self-Adaptive Systems : A Research Roadmap. *Software Engineering for SelfAdaptive Systems*, 5525(08031) :1–26, 2009.
- [16] European Commission. Green paper on the convergence of the telecommunications, media and information technology sectors, and the implications for regulation. Technical Report December, 1997.
- [17] J. Coutaz, J.L. Crowley, S. Dobson, and D. Garlan. Context is key. *Communications of the ACM*, 48(3) :49–53, 2005.
- [18] M. Cuevas. Admission control and resource reservation for session-based applications in next generation networks. *BT technology journal*, 23(2) :130–145, 2005.
- [19] O. Dameron. *Modélisation, représentation et partage de connaissances anatomiques sur le cortex cérébral*. PhD thesis, 2003.
- [20] A.K. Dey. *Providing architectural support for building context-aware applications*. PhD thesis, 2000.
- [21] A.K. Dey and G.D. Abowd. Towards a better understanding of context and context-awareness. In *CHI 2000 workshop on the what, who, where, when, and how of context-awareness*, volume 4, pages 1–6. Citeseer, 2000.
- [22] N. Ferry, S. Lavirotte, G. Rey, and J.Y. Tigli. Adaptation Dynamique d'Applications au Contexte en Informatique Ambiante. *Research Report I3S (Université de Nice-Sophia Antipolis/CNRS), number I3S/RR-2008-20-FR*, 36(0), 2008.
- [23] F. Gandon. *Ontologies informatiques*, 2006.

- [24] I. Georgiadis, J. Magee, and J Kramer. Self-organising software architectures for distributed systems. In *In Proceedings of the first workshop on Self-healing systems*, 2002.
- [25] T. Grgic, V. Huskic, and M. Matijasevic. Resource authorization in IMS with known multimedia service adaptation capabilities. *Intelligent Interactive Multimedia*, 2008.
- [26] B. Groszof, A. Arbor, and M. Kifer. The SILK System : Scalable and Expressive Semantic Rules. In *International RuleML Symposium on Rule Interchange and Applications - RuleML'09*, 2009.
- [27] T. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition - Special issue : Current issues in knowledge modeling*, 5(2) :199–220, 1993.
- [28] T. Gruber. Ontology. In *Encyclopedia of Database Systems*. Springer-Verlag, 2009.
- [29] N. Guarino. Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2-3) :293–310, 1997.
- [30] N. Guarino and P. Giaretta. Ontologies and Knowledge Bases : Towards a Terminological Clarification. In *Towards Very Large Knowledge Bases : Knowledge Building and Knowledge Sharing*, pages 25–32. IOS Press, Amsterdam, 1995.
- [31] F. J. Hens and J. M. Caballero. *Triple Play Building the Converged Network for IP, VoIP and IPTV*. 2008.
- [32] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retzschitzegger. Context-awareness on Leonhartsberger, G. mobile devices – the hydrogen approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003.*, 2003.
- [33] J. I. Hong, J. A. Landay, and S. Hall. An Infrastructure Approach to Context-Aware Computing. *Human-Computer Interaction*, 16(2) :287–303, 2001.
- [34] J.-Y. Hong, E.-H. Suh, and S.-J. Kim. Context-aware systems : A literature review and classification. *Expert Systems with Applications*, 36(4) :8509–8522, May 2009.
- [35] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean. SWRL : A Semantic Web Rule Language Combining OWL and RuleML, 2004.
- [36] M. C. Huebscher and J. A. McCann. A survey of autonomic computing—degrees, models, and applications. *ACM Computing Surveys*, 40(3) :1–28, August 2008.
- [37] IBM. An architectural blueprint for autonomic computing., 2003.
- [38] ITU-T. Y.2001 - General overview of NGN. Technical report, ITU-T, 2004.

- [39] ITU-T. Functional requirements and architecture of the NGN release 1 (Y.2012). June 2006.
- [40] H. Khlifi and J.-C. Gr. IMS Application Servers : Roles, Requirements, and Implementation Technologies. *IEEE Internet Computing*, 12(3) :40–51, May 2008.
- [41] C.C. Kim, S.C. Shin, S.Y. Ha, S.Y. Han, and Y.J. Kim. End-to-end qos monitoring tool development and performance analysis for NGN. *Management of Convergence Networks and Services*, pages 332–341, 2006.
- [42] G. Kleiber. Contexte, interprétation et mémoire : approche standard vs approche cognitive. *Langue française*, 103(1) :9–22, 1994.
- [43] M.M. Kokar, K. Baclawski, and Y.A. Eracar. Control theory-based foundations of self-controlling software. *Intelligent Systems and their Applications, IEEE*, 14(3) :37–45, 1999.
- [44] P. Korpipää and J. Mäntyjärvi. An ontology for mobile device sensor-based context awareness. In *CONTEXT'03 Proceedings of the 4th international and interdisciplinary conference on Modeling and using context*, pages 451–458, 2003.
- [45] A. Kos, M. Volk, and J. Bester. Quality assurance in the IMS-based NGN environment. *Handbook of Research on Wireless Multimedia*, N. Cranley and L. Murphy (Eds.), *Information Science Reference*, pages 240–57, 2009.
- [46] H. Koumaras, A. Kourtis, D. Martakos, and J. Lauterjung. Quantified PQoS assessment based on fast estimation of the spatial and temporal activity level. *Multimedia Tools and Applications*, 34(3) :355–374, March 2007.
- [47] H. Koumaras, D. Negrou, F. Liberal, J. Arauz, and A. Kourtis. ADAMANTIUM project : Enhancing IMS with a PQoS-aware multimedia content management system. *The Journal of Control Engineering and Applied Informatics, Romanian Society of Automation and Technical Informatics*, 10(2) :24–32, 2008.
- [48] H. Koumaras, L. Sun, and A. Kourtis. The ADAMANTIUM Multimedia Content Management System for Real Time Cross-Layer Adaptation of IPTV and VoIP Services over IMS. In *TEMU2008, Int. Conf. on Telecommunications and Multimedia*, 2008.
- [49] R. Laddaga. Creating robust software through self-adaptation. *IEEE Intelligent System*, 14(3) :26–29, 1997.
- [50] J. Lardon. *Proxy d'Interface Homme Machine : Apport des algorithmes génétiques pour l'adaptation automatique de la présentation des documents Web*. PhD thesis, 2010.

- [51] B. Li and K. Nahrstedt. A control theoretical model for quality of service adaptations. *1998 Sixth International Workshop on Quality of Service (IWQoS'98) (Cat. No.98EX136)*, pages 145–153, 1998.
- [52] I. MacLarty, L. Langevine, M. V. Bossche, and P. Ross. Using SWRL for Rule-Driven Applications. Technical report, 2009.
- [53] M. Marchese. *QoS over heterogeneous networks*. 2007.
- [54] P.K. McKinley, S.M. Sadjadi, E.P. Kasten, and B.H.C. Cheng. Composing adaptive software. *IEEE Computer*, 37(7) :56–64, 2004.
- [55] A. Minaburo and J.-C. Point. Signalisation pour la QoS de Bout en Bout Dans les Reseaux NGN. *Techniques de l'ingénieur. Télécoms*, pages 1–8, 2007.
- [56] E. Mingozzi, G. Stea, M.A. Callejo-Rodríguez, J. Enríquez-Gabeiras, G. García-de Blas, F.J. Ramón-Salquero, W. Burakowski, A. Beben, J. Sliwinski, and H. Tarasiuk. EuQoS : End-to-End Quality of Service over Heterogeneous Networks. *Computer Communications*, 32(12) :1355–1370, July 2009.
- [57] M. J. O'Connor and A. K. Das. SQWRL : a Query Language for OWL. In *OWL : Experiences and Directions (OWLED)*, 6th International Workshop, 2009.
- [58] P. Oreizy, M. Gorlick, R. Taylor, D. Heimbigner, G. Johnson, N. Medvidovic, A. Quilici, D. Rosenblum, and A. Wolf. An architecture-based approach to self-adaptive software. *IEEE Intelligent System*, 1(3) :54–62, 1999.
- [59] A. Outtagarts and O. Martinot. iSSEE : IMS Sensors Search Engine Enabler for Sensors Mashups Convergent Application. *Arxiv preprint arXiv :0912.1805*, 6 :1–7, 2009.
- [60] T. Özçelebi, I. Radovanović, and D. Sengupta. Real-Time Resource Availability Signaling in IP Multimedia Subsystem Networks. *Journal of Service Science and Management*, (June) :128–134, 2008.
- [61] S. Park and S. Lee. Dynamic Session Control for Scalable Video Coding over IMS. *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2779–2783, May 2008.
- [62] M. Pirhadi, S.M.S. Hemami, and A. Khademzadeh. Resource and Admission Control Architecture and QoS Signaling Scenarios in Next Generation Networks. *World Applied Sciences Journal*, 7 :87–97, 2009.
- [63] M. Poikselkä and G. Mayer. *The IMS : IP Multimedia Concepts and Services*. 3rd editio edition, 2009.
- [64] G. Pujolle. *Les réseaux - Edition 2008*. 6e édition edition, 2007.

- [65] F. Racaru, M. Diaz, and C. Chassot. Quality of Service Management in Heterogeneous Networks. In *Communication Theory, Reliability, and Quality of Service, 2008. CTRQ '08. International Conference on*, Bucharest, 2008.
- [66] S. M. Sadjadi and P. K. McKinley. ACT : An adaptive CORBA template to support unanticipated adaptation. In *In Proceedings of International Conference on Distributing Computing System*, pages 74–83, 2004.
- [67] M. Salehie and L. Tahvildari. Self-Adaptive Software : Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems*, 4(May) :1–40, 2009.
- [68] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. *Workshop on Mobile Computing Systems and Applications*, pages 85–90, 1994.
- [69] L. Skorin-kapov and M. Matijasevic. End-to-End QoS Signaling for Future Multimedia Services in the NGN. *6th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, (3) :408–419, 2006.
- [70] L. Skorin-kapov and M. Matijasevic. Modeling of a QoS Matching and Optimization Function for Multimedia Services in the NGN. *12th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services*, pages 55–68, 2009.
- [71] L. Skorin-Kapov and M. Matijašević. Dynamic QoS Negotiation and Adaptation for Networked Virtual Reality Services. In *proceedings of IEEE International Conference on a World of Wireless and Mobile Multimedia (WoWMoM 2005)*, pages 344–351, 2005.
- [72] L. Skorin-Kapov, M. Mosmondor, O. Dobrijevic, and M. Matijasevic. Application-level QoS negotiation and signaling for advanced multimedia services in the IMS. *Communications Magazine, IEEE*, 45(7) :108–116, 2007.
- [73] J. Song, M. Chang, S. Lee, and J. Joung. Overview of ITU-T NGN QoS Control. *IEEE Communications Magazine*, 45(9) :116–123, September 2007.
- [74] France Telecom. Rapport financier annuel 2010. Technical report, 2010.
- [75] I. Vidal, J. Garcia, F. Valera, I. Soto, and A. Azcorra. Integration of a QoS aware end user network within the TISPAN NGN solutions. In *IEEE 4th European Conference on Universal Multiservice Networks (ECUMN'2007)*, pages 152–160, 2007.
- [76] I. Vidal, I. Soto, F. Valera, and J. Garcia. IMS signalling for multiparty services based on network level multicast. In *3rd EURO-NGI Conference on Next Generation Internet Networks*, pages 103–110, 2007.

- [77] I. Vidal, F. Valera, and A. Azcorra. Experiences with a flexible QoS capable residential gateway within a multiservice framework. *Journal of Internet Engineering*, 2(1), 2008.
- [78] I. Vidal, F. Valera, and I. Soto. Adaptive quality of service management for next generation residential gateways. In *Management of Multimedia and Mobile Networks and Services*, pages 183–194, 2006.
- [79] F. Weber, W. Fuhrmann, U. Trick, U. Bleimann, and B. Ghita. QoS in SIP-based NGN—state of the art and new requirements. In *SEIN 2007 : Proceedings of the Third Collaborative Research Symposium on Security, E-Learning, Internet and Networking*, page 201. Lulu. com, 2007.
- [80] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3) :94–104, 1991.
- [81] M. Wuthnow, J. Shih, and M. Stafford. *IMS : A New Model for Blending Applications*. Informa Telecoms & Media, 2009.
- [82] A. Zimmermann, A. Lorenz, and R. Oppermann. An operational definition of context. In *Proceedings of the 6th international and interdisciplinary conference on Modeling and using context*, pages 558–571. Springer-Verlag, 2007.
- [83] C. Zins. Conceptual Approaches for Defining Data, Information and Knowledge. *Journal of the American Society for Information*, 58(January) :479–493, 2007.

Résumé de thèse

On appelle *réseaux du futur* (NGN) la nouvelle architecture des réseaux de télécommunications qui se caractérise par une convergence au niveau du réseau permettant de lier de bout en bout des réseaux hétérogènes. Ces réseaux convergents, centrés autour des services, permettent de développer de nouveaux usages, par exemple la possibilité de changer de terminal utilisateur ou de technologie d'accès sans entraîner de coupure au niveau de la connexion du réseau. Cependant, une continuité de la connexion au réseau n'entraîne pas forcément une continuité de la consommation des services numériques qui peupleront ces réseaux. En effet, les caractéristiques et la qualité de service qui sont appliquées sur les différents réseaux traversés ne sont pas homogènes, cela entraîne donc des perturbations sur la fourniture du service. Dans ces conditions, il devient indispensable d'adapter, automatiquement et tout au long de la session, le service numérique à ses conditions d'exploitation.

L'objectif de cette thèse est d'apporter une réponse à cette problématique d'adaptation automatique et dynamique des services numériques au sein des réseaux du futur. Pour cela, notre contribution s'articule autour d'un intergiciel qui prend en charge les fonctions de gestion du contexte pour les services numériques ; ceci pour permettre l'émergence de services numériques sensibles au contexte dans les réseaux du futur.

Dans nos travaux, nous définissons un service numérique sensible au contexte comme un service numérique qui propose plusieurs modalités d'exécution et qui s'adapte à son environnement de consommation. Chaque modalité correspond à une ou plusieurs situations d'usage. L'intergiciel, au cours de l'exécution d'une session d'un tel service, collecte les données sur l'environnement pour déduire la situation d'usage et proposer la modalité d'exécution la plus performante dans cette session. Une situation d'usage est caractérisée par des paramètres informatiques, notamment la performance du terminal utilisateur et du réseau d'accès. L'originalité de ces travaux dans les NGN se situe dans le modèle de conception pour des services numériques sensibles au contexte où l'on découple la logique métier du service numérique de la logique de gestion du contexte. Plus spécifiquement nous définissons la notion de *signature de contexte*, propre à chaque service numérique, qui permet d'identifier les situations d'usage du service numérique et accompagne l'intergiciel dans le choix de la modalité d'exécution la plus performante. Elle est liée au service numérique sensible au contexte dès sa conception. Lors du déploiement du service dans le serveur d'applications, la signature de contexte est désolidarisée de la logique métier et se voit insérée dans la base de connaissance de l'intergiciel. La signature fait ainsi le lien entre l'intergiciel et le service numérique. Elle permet d'activer l'ensemble des fonctions de traitement du contexte que nous avons modélisé sous la forme d'une *chaîne de traitement du contexte*. Cette approche assure une séparation des deux logiques qui composent un service numérique sensible au contexte. Les concepteurs de tels services peuvent alors se concentrer sur le développement de la logique métier sans se soucier des fonctions de gestion du contexte.

La mise en oeuvre de ces travaux s'articule autour des technologies du web sémantique pour la réalisation du mécanisme de décision d'adaptation des services numériques. Le modèle proposé dans cette thèse a été mis en place dans un serveur d'application J2E Mobicents. Pour évaluer ce travail, nous avons mis en place un environnement NGN complet au sein de la plateforme EUROP et y avons inséré notre proposition.