

STRUCTURATION DE LA COMMUNICATION EN MODE SESSION ET NOUVELLES VUES POUR LA SIGNALISATION

Claude Rigault

ENST, 46 rue Barrault, 75 013 Paris, France

GET-Télécom Paris

Cofortic

48 Blancharde, 91190 Gif, France

Claude.rigault@wanadoo.fr

Résumé

Il s'agit, dans ce papier de tirer les conséquences du passage des services de télécommunications du mode appel au mode session. Ce changement nécessite un nouveau cadre théorique pour refonder les concepts du plan contrôle. Les fondations de ce nouveau cadre théorique sont abordées dans ce papier, aboutissant entre autre à de nouvelles conceptions pour la signalisation : les concepts actuels aboutissent à une multitude de protocoles spécialisés pour les signalisations de chaque domaine du plan contrôle. Nous montrons qu'en changeant le paradigme de la signalisation, nous aboutissons à la possibilité d'un protocole de signalisation unique pour tous les domaines et nous posons les bases de ce protocole nommé GCSP (Generic Context Sharing Protocol)

I- Introduction : de la communication orientée appel à la communication orientée session

Nous constatons depuis l'origine des télécommunications, une rupture technologique tous les 25 ans environ. Après le Réseau Numérique à Intégration de Service RNIS des années 80 nous en sommes maintenant à la convergence des réseaux voix-data (double play), puis interactif-diffusion (Triple play), puis fixe-mobiles (quadruple play). Les travaux du comité de standardisation TISPAN suivant ceux de l'ETSI et de 3GPP aboutissent à une nouvelle technologie pour les services de communication basée sur l'architecture de contrôle IMS (IP MultiMedia Services) [1]. Cette technologie est extrêmement innovante dans la mesure où elle bouleverse un grand nombre de paradigmes auxquels nous étions habitués. L'une des grandes nouveautés, *imposée par la notion de convergence*, consiste à baser la communication sur le mode session en opposition au mode appel précédent, c'est-à-dire que dorénavant plusieurs sessions peuvent être ouvertes au cours d'une même communication. Donnons un exemple : Vous vous déplacez chez l'un de vos sous traitants pour proposer la mise en route d'une nouvelle production. Pour mieux expliquer la finalité et les caractéristiques du nouveau produit vous démarrez depuis votre PC portable, connecté en WIMAX ou en UMTS ou au LAN ethernet de l'entreprise une communication vocale en conférence avec les commerciaux de votre société. L'image étant plus parlante, vous ajoutez le mode vidéo pour leur montrer le prototype du nouveau produit, tout en continuant votre exposé. Parallèlement, dans une seconde fenêtre, le service financier vous montre le business plan de lancement dans un tableau Excel, tandis que le service marketing vous envoie le clip publicitaire prévu. Comme vous constatez avec le sous traitant que certaines des données du business-plan doivent être quelque peu modifiées, vous vous connectez à l'intranet de votre entreprise et, une fois dans le Programme de Gestion Intégrée (PGI ou ERP), vous modifiez les données à la volée. Ce scénario qui n'a rien d'extrême montre que l'on peut envisager comme relativement proche le moment où le téléphone existera surtout couplé à d'autres services.

Cette évolution, déjà actuelle, remet en cause bon nombre des concepts établis, tels que les notions d'appels, de session, de connexion. La notion de service elle-même doit être profondément repensée. C'est un tel travail de refondation des notions classiques que nous résumons d'abord ici. Puis partant du constat que la coopération entre des entités multiples,

pour être envisageable, nécessite une certaine unité d'interfonctionnement (c'est la raison du succès de l'internet), nous proposons une nouvelle méthode unifiée pour la signalisation implémentée sous le nom de signalisation GCSP (Generic Context Sharing protocol). Ce travail est basé sur les travaux de recherche de l'auteur depuis de nombreuses années ainsi que sur les travaux de la thèse de Rony Chahine [2]

II- Structuration des fonctions de télécommunications

Pendant longtemps, les ingénieurs ont été convaincus qu'il n'existait qu'une façon de communiquer : on donnait au réseau une identification d'entité appelée, un environnement de communication était établi et les points d'extrémité échangeaient des informations. On sait aujourd'hui que cette vue est beaucoup trop simpliste et qu'il y a d'autres façons de communiquer.

II.1 Les paradigmes de communication

Un *service*, en général, est un ensemble coordonné de fonctions qu'un système offre à des personnes ou à des applications (les utilisateurs). Un *service de télécommunications* est donc un ensemble coordonné de fonctions qui permettent à des utilisateurs dits *partenaires* d'échanger des informations.

On considère que deux grandes sortes de service de communication s'opposent : Les services interactifs (entre points bien précis) et les services diffusés.

Parmi les services interactifs, nous trouvons :

- les services conversationnels (téléphonie, visiophonie, transmission de fichiers)
- les services de messagerie (messagerie vocale, email, messageries multimédia)
- les services d'information (Web, Archives multimédia)

Parmi les services diffusés, nous trouvons :

- les services diffusés sans contrôle de l'utilisateur (télévision, radio, flux RSS)
- les services diffusés avec contrôle de l'utilisateur (Vidéo à la demande)

Une *instance* d'un service est une exécution unitaire du service pour des partenaires donnés. Nous nommerons « *Communication* » une instance d'un service de télécommunication (L'abréviation « la com. » est ainsi d'un usage très courant). Une communication est donc caractérisée par les identifiants des partenaires. Dans le cas d'applications impliquant une communication entre machines multitâches, les partenaires sont des tâches. Ainsi, si deux fenêtres d'un même navigateur sont ouvertes sur le même PC vers deux sites web différents nous avons deux instances (ou deux communications simultanées) pour le même service.

Un *media* est un type d'information que deux partenaires échangent. Les Data, la vidéo, le texte, la voix sont des médias. Ils sont différents parce que les partenaires les appréhendent de manières différentes. Depuis l'avènement du numérique les ingénieurs ont donné l'illusion d'avoir uniformisé tous les médias en les représentant par des octets. Même s'il s'agit bien là du point de départ vers la convergence numérique, l'essentiel reste à faire car, à de nombreux points de vue, un octet data ou image, n'est pas équivalent à un octet voix par exemple. En particulier leur transmission, et le contrôle de leur transmission nécessitent des moyens bien différents : *l'expérience n'est pas réductible à l'octet*. Le but de ce travail est de contribuer à l'effort nécessaire pour réaliser la convergence numérique en classant et structurant les types d'activités requises pour contrôler la transmission des médias et en proposant une méthode unifiée d'interaction pour tous ces types d'activités

Nous devons d'abord remarquer qu'un service de communication transmet des octets selon un « *paradigme de communication* ». Pour les services interactifs, plusieurs paradigmes possibles

ont été développés jusqu'ici [3]. Il y a d'abord 3 paradigmes asynchrones : *Message passing*, *Message Queuing*, *Publication-Abonnement*. Il y a ensuite 2 paradigmes synchrones : *Requête-réponse* et *Conversationnel*. Nous considérerons ici essentiellement les deux paradigmes synchrones.

Tout d'abord le « *requête-réponse* ». Dans ce paradigme, la communication est limitée à un seul échange constitué d'une requête et de sa réponse. Le fonctionnement de ce paradigme correspond à un automate combinatoire, il n'y a aucune mémorisation au delà de la réponse à une requête. On dit que ce paradigme est sans état (*Stateless*) ou sans mémorisation. Un requête-réponse particulier est le « *client-serveur* ». Le client-serveur est un requête-réponse au fonctionnement dissymétrique ou monodirectionnel : c'est toujours le même partenaire, dénommé *client* qui fait les requêtes et c'est toujours le même partenaire, dénommé *serveur* qui fait les réponses. L'internet est un réseau conçu et optimisé pour le paradigme stateless de communication client-serveur. Les mécanismes de l'internet se comprennent dès lors que l'on sait qu'il a été conçu pour cette façon de communiquer.

Le deuxième paradigme synchrone développé pour les services interactifs est le paradigme *conversationnel*. Nous en donnerons la définition formelle suivante :

« *Dans le paradigme de communication conversationnel, un environnement de communication est explicitement demandé et établi avant que les partenaires puissent échanger des médias et cet environnement reste établi, que des médias soient échangés ou non, jusqu'à ce qu'une demande de relâchement soit explicitement émise pour le libérer* »

Dans le paradigme conversationnel, la communication n'est pas limitée au seul échange d'un message dans chaque sens. De plus ou moins nombreux messages se suivent dans chaque sens pour constituer un dialogue.

De la définition, nous déduisons que l'environnement de communication est persistant, qu'il reste en place tant qu'une demande explicite de relâchement n'est pas émise. Il est donc mémorisé. Le fonctionnement de ce paradigme correspond donc à un automate séquentiel : il y a mémorisation. On dit qu'il s'agit d'un paradigme à état (*Statefull*) qui correspond au mode orienté connexion au sens de l'OSI. Le service emblématique qui utilise le paradigme conversationnel est le service POTS (Plain Old Telephone Service) où l'on réserve des circuits et des connexions entre ces circuits dans tous les commutateurs participant à la communication. Ces ressources restent réservées, que l'on parle ou non, jusqu'à ce qu'une demande explicite de relâchement soit effectuée. Le réseau téléphonique est en effet un réseau qui a été conçu et optimisé pour le paradigme conversationnel.

Nous déduisons de la définition du paradigme conversationnel qu'à tout service de communication qui fait appel à ce paradigme, il doit être nécessairement adjoint une application dont le rôle est de mettre en place puis de relâcher l'environnement de communication. *Par définition nous appelons « Application de Contrôle » l'application adjointe à tout service utilisant le paradigme conversationnel dont le rôle est de mettre en place, éventuellement modifier, puis relâcher l'environnement de communication.*

Le contrôle a donc une signification « *communication par communication* » : son action se termine et est effacée à la fin d'une communication. La limitation des activités de contrôle à la durée de la communication constitue une différence fondamentale entre le contrôle et la gestion. De manière générale la gestion est l'ajustement des paramètres d'un service. Les effets de la gestion ont une durée qui n'a pas de relation avec celle de la communication. (Normalement, les effets d'une action de gestion durent bien au-delà de la durée d'une simple communication). Le contrôle agit sur des données temporaires effacées à la fin de la

communication que l'on nomme Session Instance Data (SID) alors que la gestion agit sur des données de support du Service (SSD Service Support Data). Tout type de service doit être géré, indépendamment du paradigme de communication qu'il utilise. *Seul le paradigme conversationnel nécessite des fonctions de contrôle.*

II.2 Les activités du plan contrôle

Dans une communication utilisant le paradigme conversationnel, chaque entité participante au service (utilisateurs, nœuds de réseau, serveurs divers) démarre un processus ou une tâche de contrôle (représentée par les petits cercles sur la figure 1). L'ensemble de tous ces processus de contrôle constitue le *plan contrôle*.

Comment peut-on classer les activités des fonctions de contrôle c'est-à-dire les fonctions à mettre en œuvre pour établir un environnement de communication ?

Dans une première approche, on peut considérer que l'ensemble des activités du plan contrôle peuvent être classées en 4 types principaux : *l'accès, l'intelligence, l'appel et la connexion.*

- *Les fonctions d'accès* peuvent elles mêmes se séparer en deux groupes : l'accès demandeur et l'accès demandé. *L'accès demandeur* est exécuté au moment où l'utilisateur du réseau se logue (se porte présent). Il s'agit de l'enregistrement (authentification et localisation) dans le sens montant et de l'acquisition du profil (droits, environnement de travail) dans le sens descendant ainsi que lui renvoyer les pointeurs de ses divers services asynchrones (messageries vocales, courriels, messageries MMS). *L'accès demandé* est exécuté à chaque nouvelle communication dont cet utilisateur est destinataire. Il consiste en une traduction nom/adresse, du fait que les utilisateurs sont identifiés par un nom et que la correspondance nom-adresse varie en permanence (mobilité, DHCP, etc.)

- *Les fonctions d'intelligence* [4]. sont mise en œuvre à chaque fois qu'un service différent du service par défaut est demandé. Par définition un *réseau est intelligent est un réseau qui permet la substitution de traitement* et les technologies de Réseau Intelligent RI, (CAMEL en réseau Mobile) sont des techniques normalisées pour implémenter le remplacement du service par défaut par un autre service, normalement localisé dans une plateforme de service spécialisée (Service Control Point SCP en RI ou Application Server en ToIP)

- *Les fonctions d'appel* : qu'est-ce donc qu'un appel ? Au risque de surprendre, nous affirmons que les fonctions d'appel n'existaient pas en téléphonie classique mais qu'elles deviennent nécessaires dès lors que nous sommes en communication multimédia. Pour définir formellement la fonction d'appel, il faut avoir recours à la notion informatique d'*association*. Le paradigme conversationnel étant Statefull, chaque partenaire d'une communication doit ouvrir une page mémoire pour mémoriser les Call Instance Data CID de cette communication. Nous appelons *Contexte Local* cette page mémoire (représenté par les rectangles sur la figure 1). Le contexte local est la vue locale à chaque partenaire de la communication. L'ensemble de tous les contextes locaux de tous les partenaires constitue le Contexte Global de la communication [Figure 1]. Il constitue une mémoire globale réalisée par un chaînage entre tous les contextes locaux (de la même manière qu'un fichier sur un disque est constitué par un chaînage de secteurs) : *le contexte global est une structure de liste*. Pour une communication donnée chaque partenaire doit donc référencier cette communication pour lui-même et connaître quelle référence ses voisins ont affecté à la même communication. *Cet inter-référencement ou cette acquisition et mémorisation des références des partenaires pour une communication donnée est appelée l'association des partenaires*. (Les associations sont les zones d'ancrage ou de chaînage dans une structure de liste). Une association d'importance particulière est l'association des points d'extrémité d'un réseau. Par exemple, sur la figure 1, Alice sait que sa conversation n°3 est la conversation n°8 de Bob. Plusieurs groupes de recherche ont trouvé particulièrement utile de nommer « *Appel* » cette association particulière

des points d'extrémité. Il se trouve que c'est la définition la plus opératoire et pour cette raison elle a été adoptée pour les dernières recommandations de l'UIT [5], [6] en particulier pour les normes du RNIS Large Bande et pour les normes de l'IMT2000. Plus généralement pour tenir compte des appels multi-parties, nous définirons que *l'appel est le graphe d'associations (d'inter-référencement) entre les points d'extrémités d'une même communication.*

L'appel est caractéristique du paradigme conversationnel : c'est l'association de mémoire d'extrémités qui persiste pendant toute la durée d'un dialogue.

Considérons le petit dialogue suivant :

« Allo !, ici Alice, je voudrais parler à Bob »

« Bonjour Alice ! Bob à l'appareil »

Ce petit dialogue est un protocole. C'est un protocole d'appel puisqu'il a permis à Bob d'acquérir la référence Alice et à Alice d'acquérir la référence Bob.

De par sa définition nous constatons que *l'appel est une notion de bout en bout.* De nouveau au risque de surprendre nous affirmons que contrairement au sens commun, le réseau téléphonique ne traite pas d'appel : l'appel est en réalité réalisé par les entités d'extrémités qui communiquent (en général des humains). Le réseau téléphonique serait d'ailleurs incapable de le faire puisque son plan contrôle ne comporte aucune fonction de bout en bout. *Nous allons voir que ce que le réseau téléphonique traite ce sont des établissements de connexion, la connexion étant, elle, une notion de proche en proche.*

Au contraire du réseau téléphonique qui ne traite pas les appels nous verrons que le réseau multimédia est obligé, lui, de traiter cette fonction. Pour comprendre ce point nous devons d'abord préciser la nature d'une connexion.

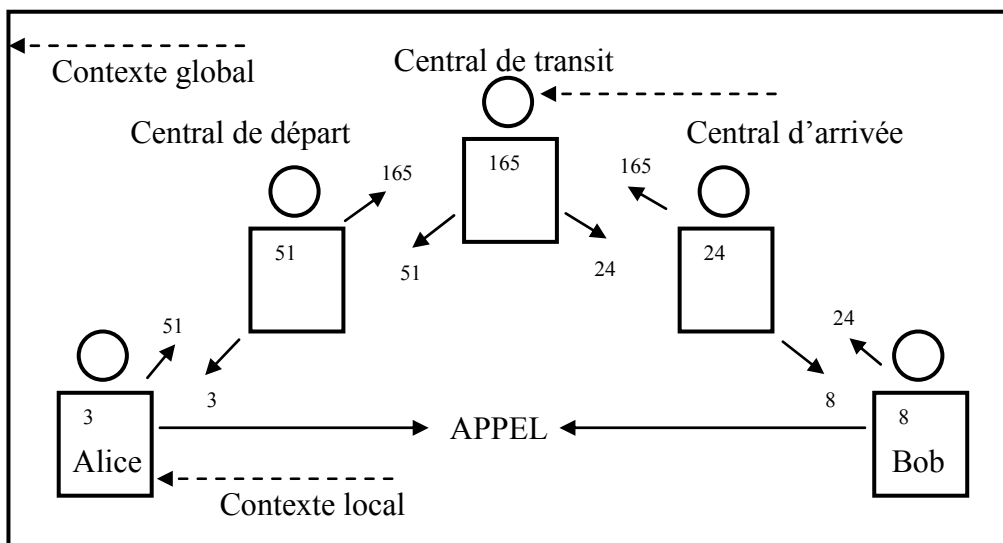


Figure 1
Contextes locaux, Associations, Contexte Global, Appel

- *Les fonctions de connexion* : qu'est-ce donc qu'une connexion ? Pour comprendre la notion de connexion nous devons préciser la nature de l'environnement de communication mis en place lorsque l'on utilise le paradigme conversationnel. Cet environnement est en premier lieu constitué de mémoire : chaque entité participant a ouvert, en effet, un contexte local. L'environnement peut n'être constitué que de mémoire, on dit alors que l'appel est sans connexion. Il se peut aussi, qu'en plus de la mémoire, l'environnement comporte des ressources physiques ou logiques (circuits à 64 Kbit/S dans le cas du Réseau Téléphonique Commuté RTC, réservation d'un route en commutation de paquets, bande passante réservée

en IntServ, agrégats de trafic en MPLS, etc.) Par définition nous dirons que *l'appel est orienté connexion si on lui réserve des ressources physiques ou logiques en plus de la mémoire préalablement au déroulement du dialogue*. C'est comme au restaurant : si vous allez au restaurant sans avoir réservé de table, vous pouvez dîner tout de suite ou vous pouvez attendre. Votre dîner est sans connexion. Si au contraire vous avez préalablement réservé votre table, vous n'attendez pas, votre dîner est orienté connexion. Le choix de l'orienté connexion résulte de la volonté d'éviter la prise de risque d'un éventuel délai. Notons ici que le terme connexion a comme synonyme le terme de « *service support* » et que l'ensemble des caractéristiques des ressources réservées est appelé « *Qualité de Service* » QoS.

Nous voyons donc que les activités du plan contrôle peuvent être classées en 4 catégories, ou *4 domaines fonctionnels du plan contrôle* : Accès, Intelligence, Appel, Connexion. Du fait qu'il ne peut pas y avoir de connexion s'il n'y a pas d'appel, nous prendrons l'habitude de regrouper appel et connexion dans un même sous ensemble que nous nommons « *transport* »

Munis de ces notions nous pouvons maintenant comprendre pourquoi un réseau multimédia doit nécessairement inclure une fonction de traitement d'appel alors que le réseau téléphonique n'incluait qu'une fonction de traitement de connexion : dès lors que la communication va mettre en œuvre plusieurs médias, il devient obligatoire de négocier de bout en bout entre les partenaires quels médias ils sont désireux d'accepter et surtout quelle type de connexion (QoS) ils veulent associer à chacun de ces médias. Evidemment cette négociation ne peut se faire que dans le protocole de bout en bout préalable qui sert à l'association des points d'extrémité c'est-à-dire dans le protocole d'appel. C'est ainsi que les protocoles de Téléphonie sur IP (ToIP) tels que SIP [7] ou H225-Q931 [8] où le protocole BICC de l'ATM [9] sont en réalité des protocoles d'appels (alors que Q931 du RNIS était un protocole de connexion, la différence est sémantiquement très importante) et qu'ils interviennent en préalable à la négociation des médias (par SDP pour SIP et par H245 pour H323). Il n'y avait pas besoin de protocole d'appel dans le réseau téléphonique du fait que la connexion n'était pas négociable.

II.3 La fin du modèle unifié du commutateur : le séquençement des activités de contrôle et le principe de précedence

La communication multimédia en mode session impose que les fonctions d'accès, d'intelligence, d'appel et de connexion puissent être exécutées dans des processus logiciels indépendants. C'est le contraire du fonctionnement des commutateurs classiques où l'ensemble des fonctions étaient traitées ensemble dans un processus logiciel unique appelé par abus de langage « traitement d'appel ». Il s'agissait en fait d'une fusion entre le traitement d'accès, le traitement de connexion et partiellement le traitement de l'intelligence.

Nous appelons cette fusion dans un processus logiciel unique de l'ensemble des fonctions, le « modèle unifié du commutateur ».

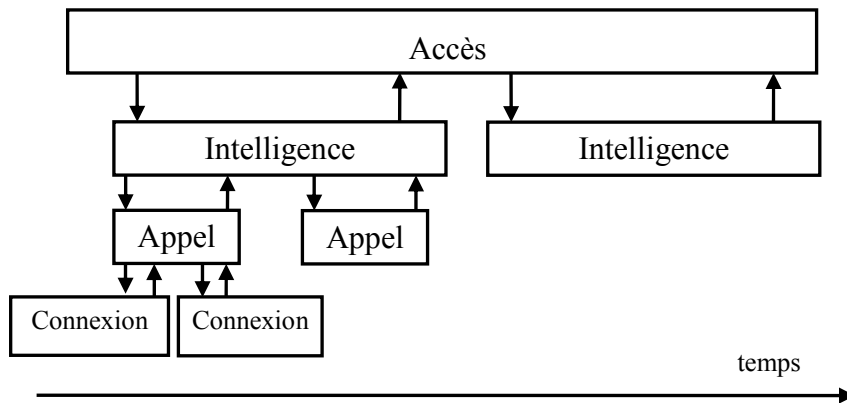


Figure 2
Le principe de précedence

Nous allons voir pourquoi ce mélange était obligatoire. Toujours est-il qu'il est incompatible avec le modèle multisession vers lequel nous nous orientons : le modèle unifié des commutateurs que nous avons connu est destiné à disparaître comme incompatible avec le fonctionnement multisessions. Comment peut-on rendre ces fonctions indépendantes c'est-à-dire réalisables dans des processus logiciels séparés ? Une condition nécessaire et suffisante d'indépendance des activités de contrôle est le *principe de précedence*.

Selon ce principe, l'accès doit être réalisé en premier. Pendant toute la durée d'une session d'accès, nous pouvons avoir une ou plusieurs sessions d'intelligence, pendant une session d'intelligence, nous pouvons avoir un ou plusieurs appels, pendant un appel, nous pouvons avoir une ou plusieurs connexions. Si cet ordre n'est pas respecté, il ne peut pas y avoir d'indépendance intelligence/appel ni d'indépendance appel/connexion.

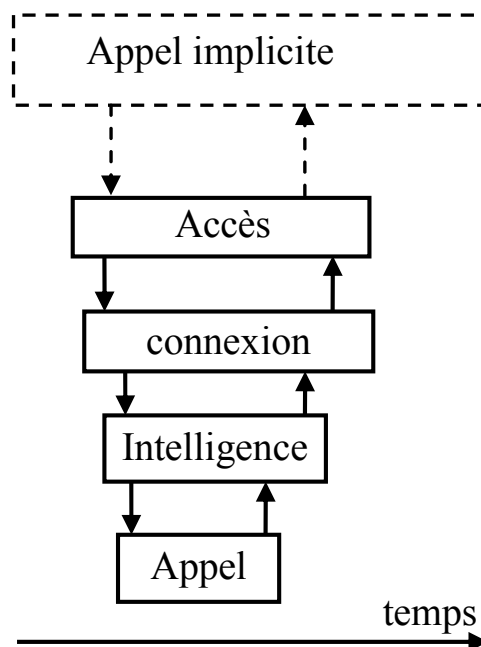


Figure 3
Le modèle unifié des commutateurs du RTC est en infraction du principe de précedence

Or il se trouve que le fonctionnement des commutateurs du RTC [Figure 3] est en infraction du principe de précedence : il y a au départ un appel implicite (qui n'existe pas encore). Pour

cet appel implicite, le commutateur établissait une session d'accès puis une connexion. Au cours de l'établissement de la connexion un service d'intelligence pouvait être invoqué. Enfin l'appel comme toujours dans le RTC est effectué par les utilisateurs et pas par le réseau. La condition nécessaire et suffisante d'indépendance des processus (la précedence) n'étant pas réalisé, les processus ne sont pas séparables et c'est pour cela que nous avons le modèle unifié du commutateur.

Le NGN et l'IMS mettent fin à cette situation en implémentant le principe de précedence.

II.4 La structuration du partitionnement des fonctions de contrôle : le modèle SIMPSON étendu

Dès lors que le principe de précedence est respecté, les fonctions du plan contrôle deviennent séparables et le plan contrôle devient le champ de la coopération entre de multiples processus indépendants, pouvant appartenir à des entreprises indépendantes. Peut-on classer les divers processus dans un modèle d'entreprise ? Une première réponse est immédiate et avait été développée dans l'effort de recherche TINA [10]: Aux fonctions d'accès correspondent des opérateurs d'accès, aux fonctions de transport correspondent les fournisseurs de connectivités ou transporteurs (carriers), aux fonctions d'intelligence correspondent les fournisseurs de services (de réseau intelligent). Nous qualifions cette première approche de *partitionnement horizontal* des fonctions de contrôle. La Figure 4 montre dans le cas d'un réseau mobile de deuxième génération quelles parties appartiennent à l'accès, quelle parties appartiennent au transport, quelles parties appartiennent à l'intelligence

Il se trouve cependant qu'une analyse plus fine permet pousser plus loin le partitionnement des fonctions. Le modèle SIMPSON [11], [12] présenté sur la figure 5, structure les services de communication en 5 niveaux fondamentaux de services. SIMPSON est un acronyme pour Signaling Model for Programmable Services Over Networks.

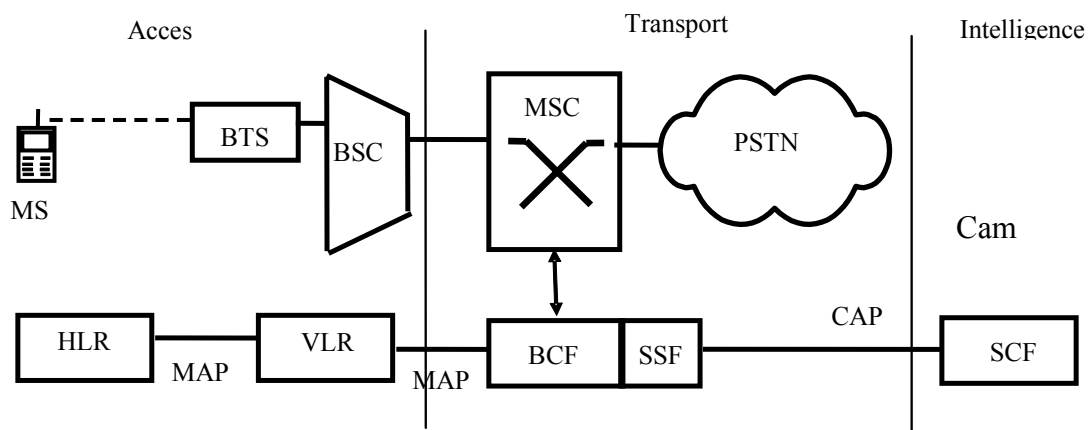


Figure 4
Dégroupage horizontal

Au premier niveau, appelé niveau « *client* » l'utilisateur met en œuvre un logiciel client, de préférence un navigateur classique, pour communiquer avec un service qui peut faire intervenir plusieurs fournisseurs et plusieurs réseaux. Le service est fourni par une plateforme située au 2^{ème} niveau. Ce deuxième niveau est appelé « niveau fournisseur ».

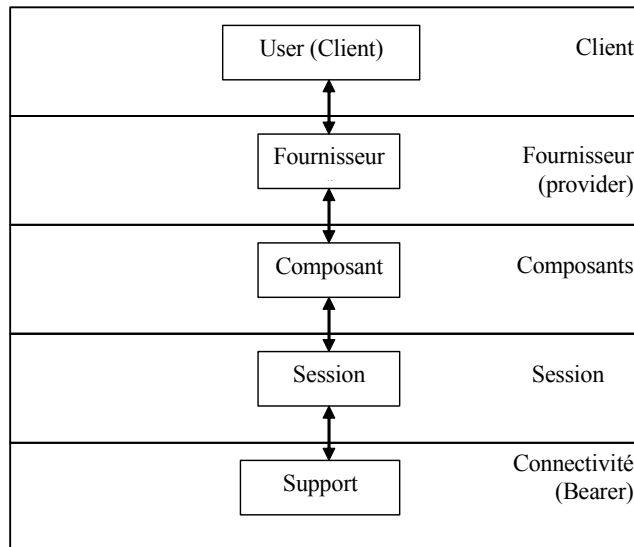


Figure 5 –Rôles fonctionnels définis par le modèle SIMPSON

La plateforme du fournisseur exécute une logique de service constituée par un graphe d'abstractions de composants. Nous appelons « *service généralisé* » un service résultant d'une telle composition de composants :

Les composants sont eux mêmes exécutés par des plateformes de « *fournisseurs de composants* » situées dans le 3^{ème} niveau. Ces composants peuvent être de toutes natures : composants financiers, composants de gestion de stock, composants de gestion de la relation client, de gestion de personnel, etc.

Certains composants peuvent nécessiter l'établissement de sessions de communication. Ces sessions sont prises en charge par les plateformes du niveau « *session* » au 4^{ème} niveau du modèle SIMPSON. Elles accomplissent les services de contrôle d'appels, de routage, de réseaux privés virtuels, de Services Vocaux Interactifs SVI, de distribution, de diffusion, de conférences, etc. Des exemples de telles plateformes de sessions sont les Média Gateway Controlers MGC de la téléphonie sur IP en architecture softswitch ou les proxys SIP ou des serveurs propriétaires tels que les serveurs de Skype.

Dans tous les cas, les services nécessitent des moyens de transmission ou supports de communication, éventuellement affectés d'une QoS bien spécifique. C'est le rôle des « *fournisseur service supports ou de connectivité* » (en Anglais : Bearer) au 5^{ème} et dernier niveau du modèle SIMPSON d'affecter ces supports adaptés.

Si l'on compare, comme dans la figure 6, le modèle SIMPSON au modèle du NGN tel que définit par l'ARCEP les niveaux clients, fournisseurs et composants appartiennent à la couche service du modèle NGN, le niveau session appartient à la couche contrôle du modèle NGN et le niveau connectivité appartient à la couche transport du modèle NGN.

Modèle en plans	Modèle ETSI	Modèle SIMPSON
Plan Services	Services	Client
		Fournisseur
		Composants
Plan Contrôle	Contrôle d'appel	Sessions
	Contrôle de support	Support
	Contrôle des Medias	Media
Plan Transport	Transport des services	
	Transport du Contrôle	
	Transport des medias	

Figure 6 – Position des niveaux SIMPSON dans les modèles du NGN

Le modèle SIMPSON identifie deux types différents de coopérations entre entités du plan contrôle : Soit les entités qui coopèrent appartiennent à un même niveau SIMPSON, soit elles appartiennent à deux niveaux adjacents. Nous appelons signalisation horizontale la communication entre entités de contrôle d'un même niveau. Nous appelons signalisation verticale la communication entre entités de contrôle de niveaux adjacents. Il est fréquent que la signalisation verticale soit aussi appelée « API » (Application Programming Interface). La signalisation est une communication pair à pair, l'API est une communication d'utilisateur à fournisseur. La figure 7, montre comment le modèle SIMPSON permet de caractériser les divers types de signalisations horizontales et verticales requises dans une architecture de plan contrôle. Nous avons identifié les diverses signalisations par des acronymes génériques. Pour chaque type, nous avons indiqué entre parenthèses des exemples de ces protocoles.

Comme API, l'interface Client-Fournisseur peut être implémentée par des Web services. L'interface Fournisseur-Composants est réalisée par l'API Parlay [13], [14]. Par cet API, un fournisseur de service peut invoquer des composants dans une plate-forme Parlay d'un opérateur de réseau. Un exemple de plate-forme Parlay est la Plate-forme Jambala proposée par la société Ericsson [15].

A son tour, la plate-forme Parlay peut invoquer les service d'une fonction SSP dans une unité de contrôle MGC (Média Gateway Controler) d'un softswitch en utilisant la signalisation INAP de l'IN ou CAP de CAMEL.

Enfin, les fonctions réseaux dans le MGC [16] peuvent mettre en œuvre des fonctions de connexion dans des Media Gateway par les signalisations MGCP [16] ou MEGACO [17] de l'architecture Softswitch.

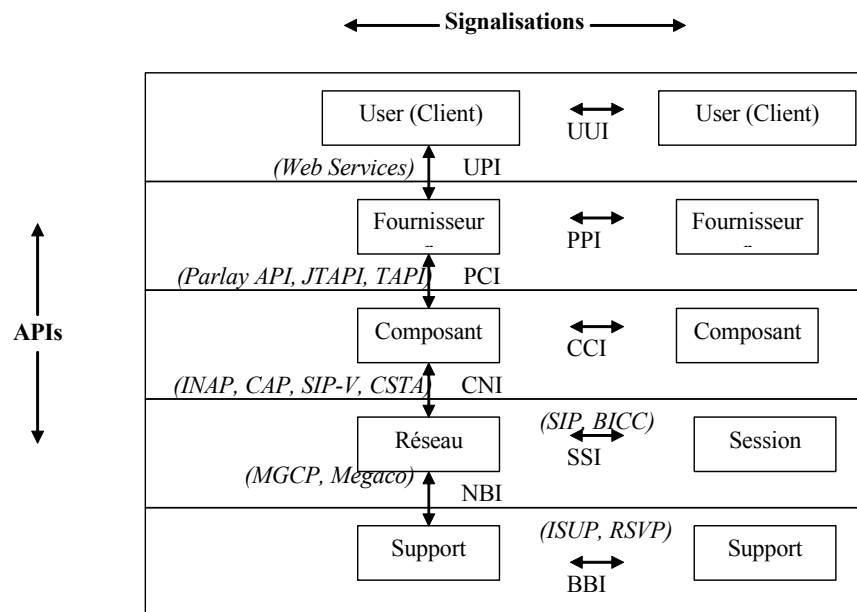


Figure 7 – Signalisations et APIs dans le modèle SIMPSON

Comme signalisations horizontales, nous trouvons l'interface d'utilisateur à utilisateur UII (ou client à client). Il s'agit de messages de signalisation dans le cadre d'un service. Les SMS, que l'on doit considérer comme des médias, n'entrent pas dans cette catégorie. Bien que très souvent invoquées, on ne peut pas dire à l'heure actuelle qu'il existe vraiment des exemples de telles signalisations. Au niveau fournisseur, nous avons l'interface de fournisseur à fournisseur PPI. Là encore on ne peut pas dire à l'heure actuelle qu'il existe des exemples de telles signalisations. Au niveau des composants, nous avons l'interface de composant à composants CCI. Un exemple d'interface CCI serait une éventuelle communication de SCP à SCP telle qu'elle était prévue dans les futurs CS du réseau intelligent.

Au niveau sessions, nous avons la signalisation SSI de session ou d'appel, par exemple la signalisation SIP « Session Initiation Protocol » [18]. On peut aussi remarquer que les services peer to peer mettent en œuvre des signalisations d'appel pour établir l'association entre les utilisateurs qui échangent des fichiers.

Finalement au niveau des services supports, nous trouvons la signalisation de connexion ou de service support BBI. Les exemples de signalisation BBI sont nombreux puisque la fonction de connexion du réseau téléphonique est une fonction de service support et fait intervenir les signalisations Q931 et ISUP [19]. RSVP [20] est un autre exemple de signalisation de connexion BBI dans le cas des réseaux IP.

Nous remarquons que les principales signalisations horizontales développées jusqu'à présent l'ont été pour les niveaux bas du modèle SIMPSON. Ceci tient à la conception très centralisée de la fourniture de service jusqu'à maintenant. Il y a un grand contraste entre l'informatique coopérative des centraux téléphoniques et l'informatique centralisée utilisée jusqu'à maintenant pour l'ensemble des couches de service. Une importante voie de recherche est de rendre les couches de service coopératives, ce qui nécessitera de développer les signalisations horizontales dans les niveaux hauts du modèle SIMPSON. (C'est le sens des recherches actuelles sur les services peer to peer)

Nous voyons donc que la modélisation SIMPSON nous mène à un nouveau partitionnement des activités de services de télécommunications que nous dénommons « partitionnement vertical ».

En réalité, à chacune des 3 composantes horizontales de service que nous avons déjà identifiées (Accès, Transport, Intelligence), nous pouvons faire correspondre une « colonne » SIMPSON, ce qui nous fait aboutir à un partitionnement à deux dimensions que nous nommons « *Modèle SIMPSON étendu* » indiqué sur la figure 8.

Accès	Transport	Intelligence	
Client d'accès	Client de transport	Client d'Intelligence	Client
Fournisseur de services d'accès (Services géographiques, services de portabilité...)	Fournisseur de services d'appels	Fournisseur de services généralisés	Fournisseur
Composants de services d'accès (Authentification, localisation, enregistrement, messageries, présence, traduction nom/adresse...)	Composants de transport : Fonctionnalités de services d'appel	Serveurs d'Application et Parlay gateways Distribution, filtrage, files d'attente, annonces traduction, notifications...	Composants
Sessions d'accès Supervision de la localisation I-CSCF	Sessions de transport : MGC, Proxies S-CSCF	SSF, MRFC	Session
Connectivité d'accès (voies balises)	Connectivité de transport : MGW, PEP	Serveurs vocaux, video, MRFP	Support

Figure 8 – Partitionnement horizontal et vertical des activités de service par le modèle SIMPSON étendu

Le célèbre service Skype [21] est un excellent exemple de la justesse de la modélisation SIMPSON : Skype est un opérateur de Session de transport, totalement indépendant des opérateurs de connectivité de transport.

Il est également intéressant de voir comment l'architecture IMS (IP Multimedia Services) constituant la nouvelle technologie en cours de déploiement par les opérateurs se conforme à cette découpe fonctionnelle. La figure 9 montre en effet l'architecture IMS et sa répartition sur les diverses partitions du modèle SIMSON étendu

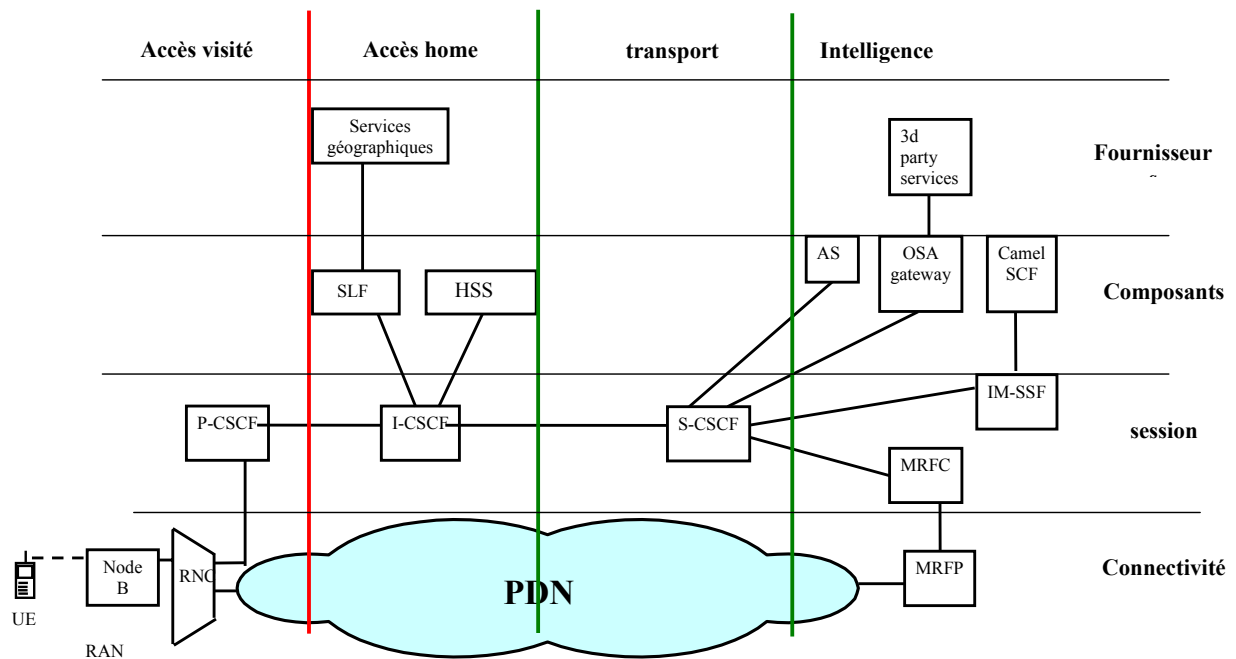


Figure 9 – Partitionnement de l’IMS selon le modèle SIMPSON étendu

III- NECESSITE D’INTERFACES UNIVERSELLES

La modélisation SIMPSON étendue nous a permis de caractériser les rôles des diverses entités fonctionnelles du plan contrôle. Il nous reste à comprendre l’interfonctionnement de ces diverses entités fonctionnelles. Il nous faut remarquer, pour cela, que ces entités fonctionnelles travaillent selon un type d’informatique tout à fait particulier : *l’informatique coopérative*. Dans une telle informatique, il n’y a pas de position centrale, tous les partenaires sont égaux et personne ne se trouve dans une position permanente pour donner des ordres aux autres. Il y a beaucoup d’efforts de recherche pour constituer une théorie de l’informatique coopérative (grilles, applications pairs à pairs...). À ce jour cependant aucune solution générale n’a fait l’unanimité. Néanmoins des applications coopératives existent. La plus emblématique en ce qui nous concerne est le plan contrôle du Réseau Téléphonique Commuté RTC. Dans le RTC en effet, il y a une hiérarchie de routage, mais pas de hiérarchie de traitement : tous les centraux téléphoniques sont égaux, il n’y a pas un ordinateur quelque part pour dire aux autres ce qu’ils doivent faire, chacun prend ses décisions en complète autonomie. C’est cette nature particulière du logiciel du plan contrôle et l’absence de base théorique forte pour le constituer qui a mené les ingénieurs à le réaliser de manière AD HOC et particulièrement compliquée. Rappelons en effet que l’écriture du logiciel des générations actuelles de commutateurs numériques a constitué un effort colossal et totalement ruineux de l’ordre de 5000 homme ans.

Cependant nous pouvons entreprendre de lister quelques conditions nécessaires sinon suffisantes, qui doivent obligatoirement être respectées pour réaliser tout projet d’informatique coopérative.

- *L’informatique coopérative nécessite le partage d’information entre les partenaires.* Dans le cas du plan contrôle cet échange d’information, c’est la signalisation : la recherche en signalisation n’est pas juste un simple problème de recherche en téléphonie, c’est un problème d’informatique théorique fondamentale : *il n’y aura pas d’informatique coopérative sans signalisation.*

- *L'informatique coopérative nécessite une politique d'attribution de l'autorité de décision* pour chaque type de situation. En effet, du fait que tous les partenaires sont égaux et qu'il n'y a pas de point central de commandement la difficulté est de savoir comment ou qui prend des décisions pour des problèmes donnés. (Cette difficulté est d'ailleurs un problème général de tout type de coopérations, dont l'histoire récente est une illustration)
- *L'informatique coopérative nécessite la connaissance des modèles comportementaux des partenaires.* Pour prendre ses décisions chaque partenaire doit en connaître l'effet sur les autres partenaires. Ceci signifie que chacun doit connaître le *modèle comportemental des autres*. En téléphonie ces modèles comportementaux sont appelés *modèles d'appel*. C'est là la raison des modèles d'appels Basic Call State Models BCSM en technologie de réseau intelligent ou « PDP context » pour les SGSN du GPRS.
- *L'informatique coopérative nécessite des associations* : chaque partenaire doit maintenir des pointeurs vers les tâches partenaires qui travaillent avec lui, formant ainsi un arbre d'associations qui donne une vue globale de la communication
- *L'informatique coopérative nécessite la confiance* : l'authentification et le chiffrement sont évidemment obligatoires pour sécuriser la communication entre partenaires coopérants
- Enfin, dernière condition que nous identifions dans le cadre de ce travail : *il faut une unité d'interfonctionnement*, c'est-à-dire un protocole unique de communication entre tous les coopérants. Sinon, bien évidemment ce serait la tour de Babel, et la coopération d'entités qui ne parlent pas le même langage serait évidemment impossible. C'est d'ailleurs là l'une des raisons du succès de l'internet : une unité d'interfonctionnement dans le plan transport. Dans le plan contrôle nous n'avons pas à l'heure actuelle une telle unité d'interfonctionnement : les protocoles varient selon les niveaux SIMPSON : SIP comme signalisation d'appel, RSVP, LDP comme signalisations de connexions, mgcp, INAP, Parlay comme APIs entre les divers niveaux. Toutes ces manières différentes pour finalement le même problème : partager les informations ne peut que gravement compliquer les possibilités de coopérations.

Cette dernière considération pilote la suite de notre travail : il apparaît qu'une interface universelle entre toutes les entités du modèle SIMPSON étendu est une condition nécessaire de la communication en mode multisession. Le reste de ce papier est consacré à la description d'une telle interface universelle : la signalisation GCSP (Generic Context Sharing Protocol) capable de remplacer par une méthode unique tous les autres protocoles de signalisation et APIs

IV- UNE INTERFACE BASÉE DATA

IV.1 Les mécanismes possibles de signalisation

Au cours d'une communication, chaque partenaire range ses Call Instance Data CID dans son Contexte Local. Ce contexte local est une page mémoire temporaire qui est relâchée à la fin de la communication. La finalité de la signalisation est de partager ces CID avec les partenaires. Trois mécanismes possibles de partage d'informations entre processus éloignés ont été identifiés [22], [23] : les mécanismes basés commandes, les mécanismes basés data et les mécanismes basés objets.

Dans le mécanisme basé commande les données du contexte local sont privées et ne peuvent être modifiées qu'indirectement par la réception d'une commande : un processus de contrôle n'est pas autorisé à lire ou écrire des données dans le contexte d'un partenaire, il utilise au contraire un ensemble prédéfini de commandes. Le partenaire qui reçoit la commande exécute les actions correspondantes et modifie son contexte local.

Dans le mécanisme basé data, (ou basée variable), les partenaires d'une même communication peuvent au contraire lire et modifier les informations des contextes locaux des partenaires. Ce mécanisme est mis en œuvre par le protocole de gestion SNMP [24]. Pour que ce mécanisme soit possible il faut que l'organisation du contexte local soit connue de tous. Une méthode possible consiste à organiser les données du contexte locale en une structure arborescente comme dans la MIB SNMP [25], ou en une structure orientée objet comme dans la MIB OSI [25]. Dans ce cas toute signalisation peut se réduire au simple échange de commandes GET/SET/NOTIFY plutôt qu'à l'utilisation d'un grand nombre de commandes spécifiques de l'entité partenaire.

Finalemnt, le mécanisme orienté objet fonctionne en invoquant des objets distants en utilisant le mécanisme client serveur. On trouve des implémentations de ce mécanisme dans les web services [26], dans CORBA[27], et dans les RMI[28]. Ce mécanisme est adapté aux communications client-serveur et ne répond pas à notre problématique d'associations persistantes.

IV.2 Les domaines d'application des divers mécanismes

Nous comparons maintenant les deux méthodes utilisables : le mécanisme orienté commande et le mécanisme orienté variable. A priori ces deux méthodes paraissent équivalentes : il semble en effet équivalent de dire « exécute la commande toto » ou de dire je mets ta variable toto à 1 sachant que ce changement de valeur va déclencher l'action « toto » chez le partenaire. En réalité, il n'en est rien, l'efficacité de l'une ou l'autre méthode dépend du type d'environnement informatique dans lequel on se situe. Il a été démontré dans la littérature [23] que le mécanisme basé variable est plus efficace dans un environnement coopératif et le mécanisme basé commande est plus efficace dans un environnement centralisé. C'est simple à comprendre : une commande spécifique peut agréger de nombreux get/set portant sur de nombreuses variables.

Dans le cas d'une architecture centralisée l'émission de commandes du point central vers de nombreux partenaires est donc très allégé par le mécanisme orienté commandes, le prix à payer pour cet allègement étant la perte de généricité du mécanisme.

En informatique coopérative, cet avantage du mode commande n'existe plus : chaque partenaire dans ce type d'informatique ne parle qu'à un petit nombre de voisins, il est donc inutile de chercher à agréger les get/set puisqu'il n'y en a de toutes façons pas beaucoup et il est donc inopportun de délaissé le mécanisme générique en faveur de commandes spécifiques.

Notons de plus que le mécanisme basé data est plus efficace pour les fonctions dépendantes du contexte. Une commande en effet arrive sur une machine dont on ne peut que présumer l'état interne. Au contraire, dans le mécanisme basé data, il est toujours possible de collecter la situation précise du partenaire en lisant ses variables de façon à le modifier en connaissance exacte de la situation.

Il est par ailleurs évident que si l'on cherche un mécanisme universel, le mécanisme basé commande ne convient pas puisqu'une commande n'est qu'un agrégat de certaines variables, spécifiques à une action.

Nous retenons donc que le mécanisme basé data ou variable est préférable en informatique coopérative, ce qui est le cas du plan contrôle et le mécanisme orienté commande est plus efficace en informatique centralisée ce qui est très généralement le cas de la gestion.

IV.3 Le monde à l'envers

Nous venons de voir que la gestion, majoritairement centralisée, nécessiterait un échange d'information basé commande or il se trouve que l'un des principaux protocoles de gestion est SNMP dont le fonctionnement est basé data !

A l'inverse nous venons de voir que l'informatique de contrôle, intrinsèquement coopérative, nécessiterait un échange d'information basé data, et il se trouve que la totalité des protocoles de signalisation sont basés commandes. Le monde est à l'envers, deux des fonctionnalités totalement incontournables de la communication utilisent chacune à ce jour le mauvais mécanisme !

Il se trouve que les chercheurs du monde de la gestion se sont rendus compte de cette situation inappropriée et proposent maintenant des protocoles basés commandes.

Par contre nous n'avons pas encore vu de proposition basées data dans le monde du contrôle et de la signalisation. *Pour obtenir le protocole de signalisation universel nécessaire au déploiement sans limitations de la communication orientée session nous proposons donc un changement radical de paradigme et d'utiliser en signalisation le mécanisme basé data.*

Pour atteindre cet objectif nous devons d'abord définir une organisation générique des données du contexte local. Nous appelons cette structure de données le « *Contexte Générique* » GC. Nous définissons ensuite un nouveau protocole de signalisation conforme à l'impératif d'unité de fonctionnement c'est-à-dire universel pour tous les domaines de signalisation : le « *Generic Context Sharing Protocol* » (GCSP), où la signalisation est réalisée par la lecture ou la modification des données dans le contexte générique du partenaire par les opérations génériques GET/SET/NOTIFY sous la contrainte bien sûr de procédures de confiance et de sécurité.

V- LA STRUCTURATION DU CONTEXTE GÉNÉRIQUE

V.1 Description globale du contexte générique

Pour structurer et décrire le GC, nous utilisons une modélisation UML et nous en décrivons les principaux schémas. La structure de donnée du GC est orientée objet de manière semblable au Common Information Model CIM [29]. L'objectif est de pouvoir enrichir en cours de vie le schéma du GC. Cependant les données du GC ne seront pas manipulées par des RMI, mais par le protocole GCSP.

Le GC est donc de principe différent des MIB. Les MIB ont une structure de données hiérarchique en arbre et n'offrent pas les facilités d'héritage que nous avons dans le GC. Par ailleurs la MIB est une structure de données de gestion et donc persistante au-delà d'une communication, alors que le GC se trouve effacé à la fin de chaque communication.

Nous basons la modélisation sur le modèle SIMPSON étendu.

La figure 10 montre l'application du modèle SIMPSON étendu au schéma UML du GC.

Dans ce schéma, nous voyons que le GC est la somme de l'ensemble des services du modèle SIMPSON étendu : il y a les services de chacun des composant horizontaux : Accès, Transport, Intelligence et dans chacune de ces rubriques nous trouvons les services de toutes les composantes verticales : Client, Fournisseur, Composant, Session, Support, Média.

Pendant une session, seuls les objets pertinents pour le type de session en cours sont instanciés

Nous voyons également sur la figure 10 que tous les services constituant le contexte générique héritent tous de la classe service.

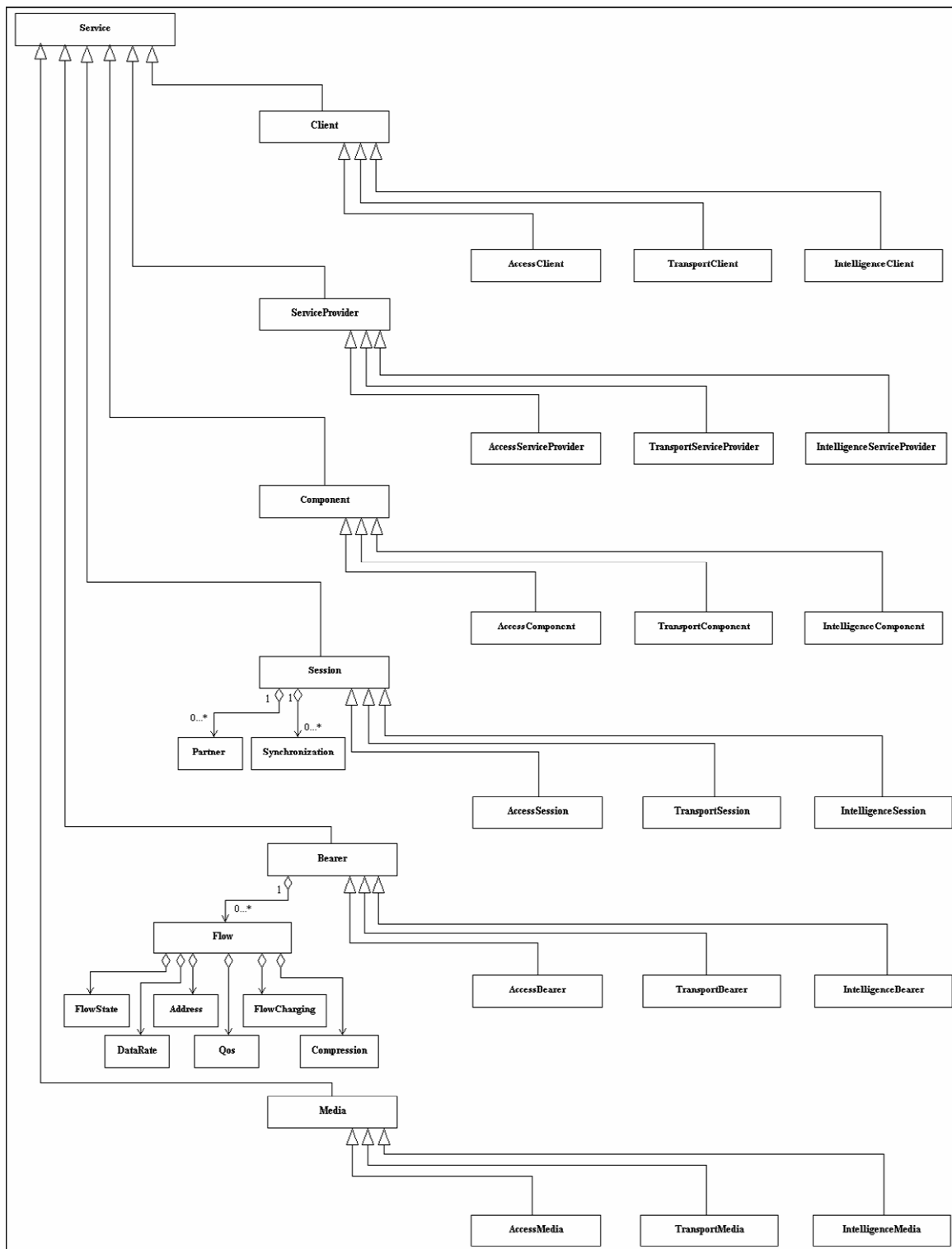


Figure 10 : structure d'héritage du GC et application du modèle SIMPSON étendu

V.2 Description détaillée du contexte générique

La Classe de service

Tous les objets du GC héritent de cette classe. Elle est représentée sur la figure 11. Nous listons sans les décrire les classes qui la composent

- Classe d'association
- Classe de sécurité
- Classe de taxation
- Classe de modèle comportemental
- Classe d'état en cours
- Classe de profile de l'utilisateur

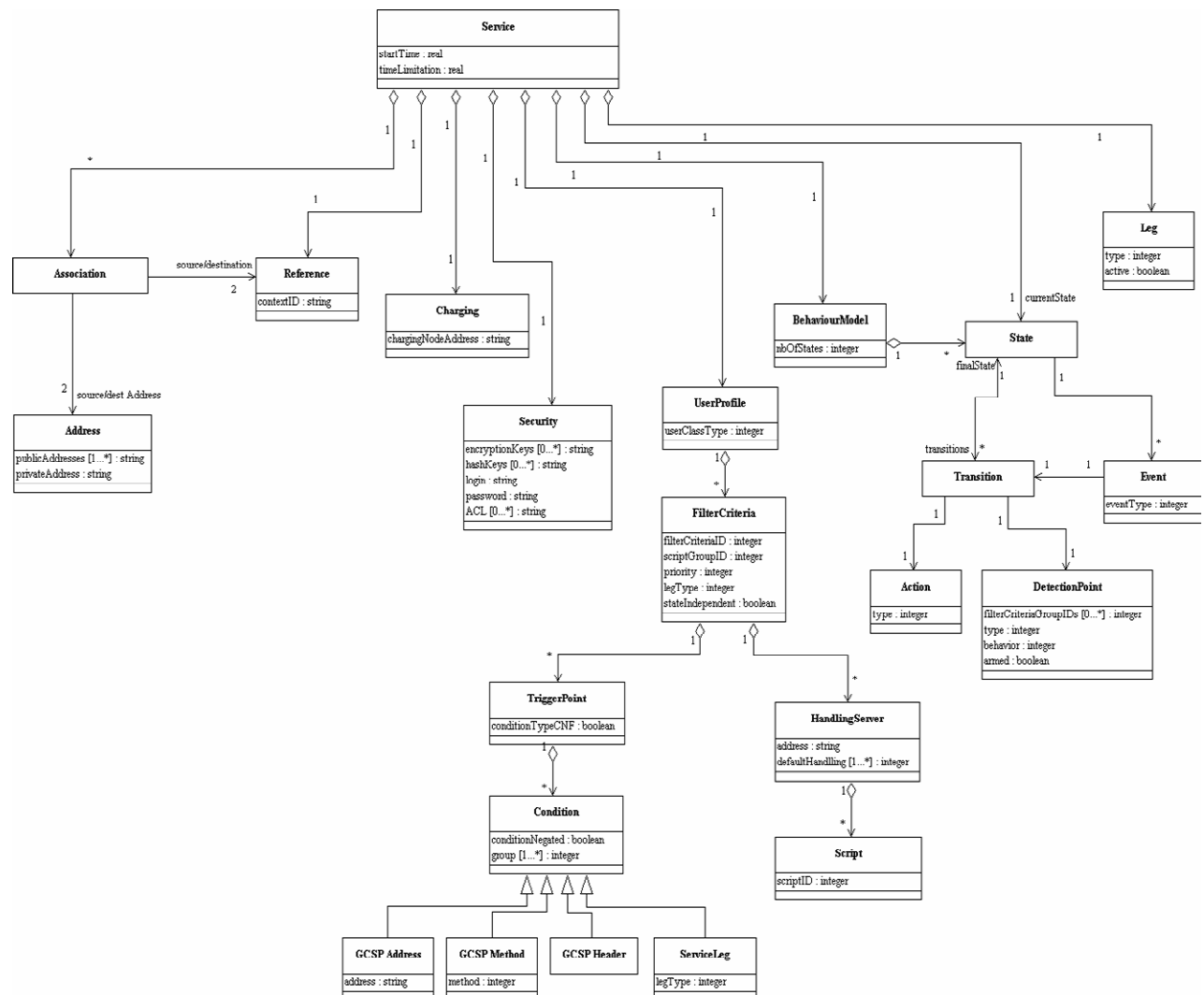


Figure 11 : Classe service

VI- DESCRIPTION GLOBALE DE GCSP : « GENERIC CONTEXT SHARING PROTOCOL »

Le partage, par tous les partenaires d'une communication, dans leur contexte local, d'une même base d'informations de contrôle, portant sur tous les acteurs pouvant intervenir globalement permet de changer la façon de signaler en remplaçant les commandes spécifiques des protocoles de signalisation par des GET/SET universels. Nous avons défini et implémenté un protocole qui réalise cet objectif de signalisation universelle [2]. Il s'agit du protocole GCSP ou Generic Context Sharing protocol. GCSP déclenche l'action distante TOTO par une commande SET sur la variable TOTO du GC distant. Lorsque le partenaire distant détecte le changement de valeur de sa variable TOTO, il exécute l'action TOTO. Un GET préalable peut être effectué pour vérifier d'abord l'état courant de la variable TOTO, mais cela n'est pas indispensable sauf pour réaliser des actions sensibles au contexte. Avec GCSP on peut lire l'état courant du partenaire et donc prédire avec certitude le résultat d'un ordre. Pour éviter que le protocole soit verbeux, et pour réduire la taille des messages, les noms des objets du GC peuvent être remplacés par de chiffres comme dans une MIB SNMP.

GCSP est un protocole basé texte dont la syntaxe est très proche de celle de http ou de SIP. Un message GCSP est constitué de headers et d'un corps de message comme indiqué sur la figure 12. Le corps de message contient les objets du contexte générique à lire ou à modifier

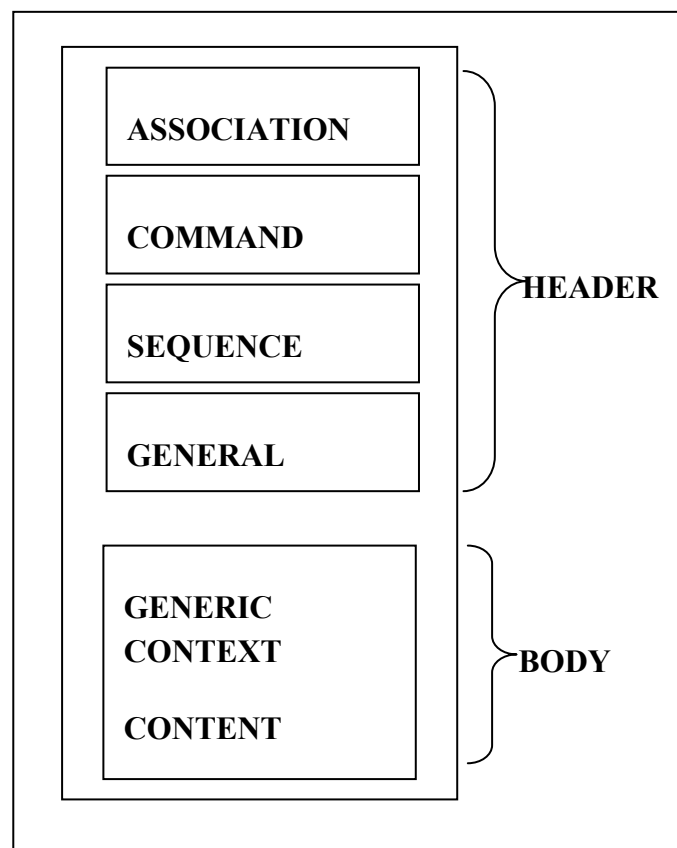


Figure 12 – Un message GCSP

Normalement, les messages GCSP sont transportés par UDP, mais ils peuvent être transportés par la couche transport de NSIS [30] ou par TCP pour supporter un chiffrement SSL ou la traversée de firewalls. La sûreté de fonctionnement est assurée par des temporisations dans les machines à état du protocole.

Nous décrivons brièvement les headers :

Association : Champs sources et destinations et références des contextes sources et destinations.

Commande : Ce champ décrit la commande invoquée : get, set, notify, open context, close context, Describe. Des réponses aux commandes sont attendues caractérisé par un code de réponse comme en http.

Sequence number, transaction number : ces <sequence_number> <transaction_number> permettent d'ordonner le séquençement d'un message dans une communication

VII- CONCLUSIONS

Nous avons donné dans notre introduction un exemple qui montrait que l'on peut envisager comme relativement proche le moment où le téléphone existera surtout couplé à d'autres services. Notre redéfinition des concepts de la communication en mode appel pour les rendre compatibles avec le nouveau mode multisessions nous a permis de structurer les natures différentes de services de télécommunication et d'en apporter une modélisation nouvelle et puissante sous la forme du modèle SIMPSON étendu. En outre ce modèle SIMPSON étendu s'est révélé être un outil pour identifier les divers domaines et natures de signalisations et d'API. Avec ce modèle, nous n'avons pu que constater la très grande variété des types de signalisations, se traduisant par une multitude de protocoles ce qui ne peut être qu'un obstacle à l'élaboration de services véritablement innovants, faisant intervenir de nombreux services à la fois. Une analyse précise de la nature coopérative des services de télécommunication nous a démontré la nécessité absolue d'une unité d'interfonctionnement entre les multiples services. Il nous faut donc un nouveau paradigme de signalisation menant à un protocole unifié pour la signalisation entre services de tous les domaines. Cette unité d'interfonctionnement permise par la signalisation GCSP est l'une des clés de l'interopérabilité des multiples services du plan contrôle pour permettre la création de services innovants.

L'élément clé qui a rendu le mécanisme de GCSP possible, c'est la possibilité de définir la même structure de donnée (le Generic Context GC), utilisable par tous les services, grâce à l'outil très puissant que constitue le modèle SIMPSON étendu.

Dans une première application pratique GCSP a été mis en œuvre [2] avec succès dans les parties hautes du modèle SIMPSON étendu pour une réalisation industrielle de services de couplage téléphonie informatique. Nous souhaitons maintenant mettre en œuvre cette signalisation à tous les niveaux du modèle SIMPSON ce qui aboutirait à une plateforme générique et très simplifiée pour des services de télécommunications beaucoup plus élaborés.

REFERENCES

- [1] 3GPP, "[TS 23.228: IP Multimedia Subsystem \(IMS\) \(Stage 2\) - Release 5](#)", September 2002
- [2] Rony Chahine, "Multi-provider and Cross-network services : mechanisms for a global control plane : Application to a new signalling paradigm end to signalling mediators", PHD thesis, ENST, 2006
- [3] Christophe Bezault, "Les middlewares orientés messages: Parnorama de l'offre et exemples avec MQSeries", Dunod, Paris, 2001
- [4] Intelligent Network Application Part. Normes Q.1200 – Q.1999
- [5] ITU-T recommendation Q 1701 Signalling requirements for IMT-2000 networks
- [6] B-ISDN Signalling Capability Set 3, UIT-T
- [7] IETF RFC 3261: "Session Initiation Protocol (SIP)", June 2002
- [8] ITU-T Recommendation H. 323
- [9] ITU-T Recommendation Q1901: BICC: Bearer Independent Call Control
- [10] TINA-C Consortium <http://www.tinac.com>
- [11] Astronefs: "Network and Telecommunication Global Service Convergence: White paper", <http://www.infres.enst.fr/~rigault/white-paper.pdf>
- [12] Guillaume Buridant, "Un modèle de signalisation générique pour les réseaux IP de nouvelle génération : unification par RSVP", PHD these, L'Université de Versailles Saint Quentin en Yvelines, 2004
- [13] www.parlay.org
- [14] ETSI ES 201 915 Phase 1: "Open Service Access (OSA); Application Programming Interface (API); all parts", 02/2002
- [15] Ericsson's Service Capability Server Parlay Gateway http://www.ericsson.com/mobilityworld/sub/open/technologies/parlay/about/parlay_about_gs1
- [16] Soft-switch architecture : IETF RFC 2705: Media Gateway Control Protocol (MGCP), Version 1.0 Oct.
- [17] IETF RFC 3015: Megaco Protocol Version 1.0, Nov. 2000
- [18] IETF RFC 3261 SIP: Session Initiation Protocol
- [19] ITU-T Recommendation Q.764, "Signalling system No. 7 – ISDN user part signalling procedures", 12/1999
- [20] IETF RFC 2205: "Resource ReSerVation Protocol (RSVP)", Version 1 Functional Specification
- [21] Skype : <http://www.skype.com>
- [22] Cherkaoui, O, presentation à MAN'2005 Ho Chi Minh ville, Avril 2005
- [23] Aiko Pras, PHD Thesis: "Network Management Architectures", 17 February 1995
- [24] IETF RFC 1157, "A Simple Network Management Protocol (SNMP)", May 1990
- [25] Cherkaoui, O, and Agoulmine, N (2003). [La gestion de réseau](#) Eyrolles, France
- [26] Web services: <http://www.w3.org/2002/ws/>
- [27] CORBA : www.corba.org
- [28] Java RMI : <http://java.sun.com/docs/books/tutorial/rmi/index.html>
- [29] Common Information Model CIM : <http://www.dmtf.org/standards/cim/>
- [30] IETF RFC 4080, "Next Steps in Signalling (NSIS): Framework", June 2005