

Mécanismes coopératifs de plan contrôle global pour des services de communications multi-fournisseurs et trans-réseaux

Rony Chahine

Département informatique et réseaux,
ENST, 46 rue Barrault, 75 013 Paris, France, GET-Télécom Paris
CoreBridge, 3 rue Saint Philippe du Roule, 75 008 Paris, France
rony.chahine@enst.fr

Claude Rigault

Département informatique et réseaux,
ENST, 46 rue Barrault, 75 013 Paris, France
GET-Télécom Paris ; LTCI-UMR 5141 CNRS
claude.rigault@enst.fr

Résumé : cet article décrit des mécanismes à mettre en œuvre pour réaliser un service par la coopération en temps réel de composants de plusieurs fournisseurs raccordés sur divers réseaux hétérogènes. Notre contribution porte sur les points suivants :

- a) une analyse des modèles d'entreprise structurant les divers rôles de la fourniture de service. Deux dimensions de dégroupage sont définies en utilisant le modèle SIMPSON
- b) une étude du concept de création d'un contexte global par l'association de contextes locaux situés dans divers réseaux hétérogènes. Un mécanisme d'association que nous appelons CAT est proposé ainsi qu'un mécanisme de construction du contexte global
- c) une nouvelle conception de la signalisation permettant l'établissement d'un plan contrôle coopératif généralisé.

mots clés : modèle d'entreprise, plan contrôle coopératif, meta-signalisation, communication poste à poste.

1) Introduction

De nouvelles perspectives s'ouvriraient aux réseaux et à l'industrie informatique s'il était possible de réaliser des services en faisant coopérer en temps réel des composants exécutés par plusieurs fournisseurs différents, éventuellement atteignables par des réseaux différents. Un exemple d'un tel service serait un service de présentation de fiche, que l'on sait obtenir aujourd'hui dans un réseau privé. Dans cet exemple, un employé d'une entreprise reçoit la fiche du client appelant, sur son PC, en même temps que son téléphone sonne. Ce même

employé en déplacement, s'il est appelé à son bureau, reçoit la fiche du client sur son PDA en même temps que son téléphone mobile sonne.

Pour le moment et pour ce type de service faisant intervenir plusieurs fournisseurs et plusieurs réseaux les chemins de signalisation manquent. Il y a des solutions partielles. A partir des Web services [1] ou d'autres sortes de middleware [2], on peut invoquer des composants de plusieurs fournisseurs pour les intégrer dans un service plus riche, avec un interface utilisateur unique. Cependant ces solutions ne s'appliquent pas à des composants situés dans des réseaux différents. Les services trans-réseaux nécessitent des passerelles de signalisation pour faire les traductions entre protocoles de signalisation. Si l'on a N signalisations différentes, le nombre de passerelles croît excessivement comme N^2 . Le problème des services trans-réseau n'a été étudié jusqu'à présent que dans des cas très particuliers dans le cas des services PINT [3] ou par des solutions très centralisées dans le cas de PARLAY [4].

Notre approche consiste à rechercher des solutions coopératives permettant à de multiples plates-formes de service, y compris des plates-formes PARLAY de travailler ensemble. Notre papier analyse ce problème et contribue à l'identification des mécanismes requis en abordant les points suivants :

a) Nous structurons cette intégration généralisée d'acteurs, de services et de fonctions de communications selon le modèle SIMPSON [5], modèle très efficace pour définir les niveaux de services et les interfaces nécessaires dans le cas des services multi fournisseurs et trans-réseaux. Il permet notamment d'identifier et de caractériser les divers types de signalisations requises et les chemins de signalisation nécessaires pour échanger les informations de contrôle entre les diverses entités. Cette structuration doit rester bien entendu compatible avec les architectures existantes des services IN [6], CAMEL [7], PARLAY, et OSA [8]. Comme illustration de cette structuration, notre papier examine les services Parlay au dessus d'une architecture softswitch [9].

b) Nous proposons un mécanisme d'association appelé CAT (Control Allocation Table) permettant la constitution d'un plan contrôle global capable d'exécuter des fonctions de contrôle au dessus de plusieurs réseaux hétérogènes. Le mécanisme CAT établit, maintient et relâche les associations entre les diverses entités qui coopèrent pour fournir un service à travers des réseaux hétérogènes. Le principe de la CAT est basé sur la construction d'un contexte générique permettant à des processus hétérogènes de coopérer directement sans nécessiter de passerelles de signalisation.

c) Nous proposons enfin un nouveau paradigme pour la signalisation, ou une redéfinition du concept de signalisation, applicable à tous les niveaux de services

Cette recherche se différencie du projet actuel "NSIS" [10] de l'IETF qui est consacré à la signalisation pour les fonctions de transport. Elle contribue par contre au projet TISPAN [11] de l'ETSI et aux recherches concernant le « metasignaling »

2) Structuration de l'offre de services de communication par la modélisation SIMPSON

2.1) La modélisation Simpson

Nous recherchons d'abord comment structurer les divers services de communication. La notion de service est trop générale, tout est service. Il s'agit de différencier la notion de service de manière opératoire. Le modèle SIMPSON, présenté sur la figure 1, structure les services de communication en 5 niveaux fondamentaux de services. SIMPSON est un acronyme pour Signaling Model for Programmable Services Over Networks.

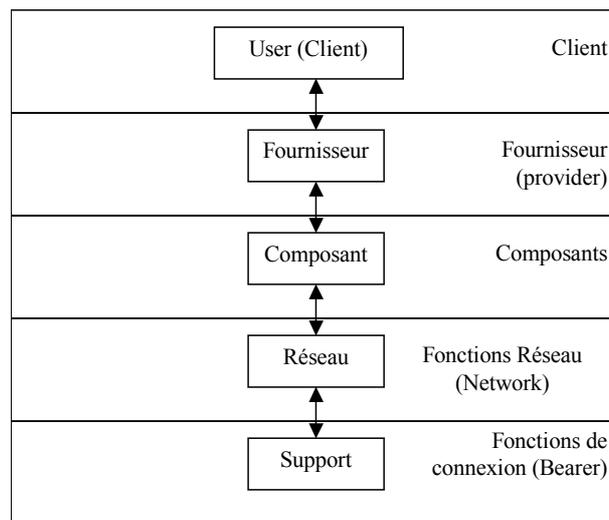


Figure 1 –Rôles fonctionnels définis par le modèle SIMPSON

Au premier niveau, appelé niveau « client » l'utilisateur met en œuvre un logiciel client, de préférence un navigateur classique, pour communiquer avec son service multi fournisseur et trans-réseaux localisé au deuxième niveau.

Ce deuxième niveau est appelé « niveau fournisseur ». Le service y est fourni par un serveur ou « plate-forme fournisseur » qui exécute une logique de service constituée par un graphe d'abstractions de composants.

Ces composants sont eux mêmes exécutés par des plate-formes 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 des fonctions réseaux. Ces fonctions réseaux sont prises en charge par les plate-formes du niveau « fournisseur de services réseaux » 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 plate-formes de services réseaux sont les Média Gateway Controlers MGC de la téléphonie sur IP en architecture softswitch.

Dans certains cas, en particulier dans le cas de la communication humaine, certains services réseaux nécessitent qu'une QoS bien spécifique leur soit affectée. C'est le rôle des « fournisseur de services supports » au 5^{ème} et dernier niveau du modèle SIMPSON d'affecter les supports adaptés. Il est possible de définir la notion de connexion comme l'affectation de ressources à un service réseau. En ce sens une connexion peut être une réservation de bande passante, comme en téléphonie ou en INTSERV, ce peut être aussi une réservation de route comme en commutation de paquets orientée connexion, ce peut être encore une réservation de priorité d'ordonnement comme en DIFFSERV ou la réservation d'un agrégat de trafic comme en MPLS.

Si l'on compare, comme dans la figure 2, le modèle SIMPSON au modèle du NGN tel que définit par l'ART [12] les niveaux clients, fournisseurs et composants appartiennent à la couche service du modèle NGN, le niveau services réseaux appartient à la couche contrôle du modèle NGN et le niveau support appartient à la couche transport du modèle NGN.

Modèle du NGN	Modèle SIMPSON
Couche Service (Opérateurs et tiers)	Client
	Fournisseur
	Composants
Couche contrôle	Services réseaux
Couche Transport (mode paquets)	Services supports

Figure 2 – Position des niveaux SIMPSON dans le modèle 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 figure 3, 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 Utilisateur-Fournisseur peut être implémenté par des Web services. L'interface Fournisseur-Composants est réalisée par l'API Parlay. Par cet API, un fournisseur de service peut invoquer des composants dans une plate-forme Parlay d'un opérateur de réseau téléphonique. Un exemple de plate-forme Parlay est la Plate-forme Jambala proposée par la société Ericsson [13].

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

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

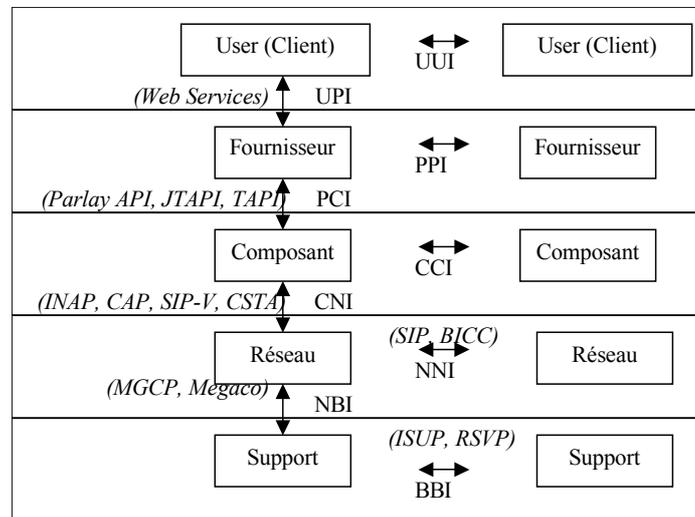


Figure 3 – Signalisations horizontales et verticales dans le modèle SIMPSON

Comme signalisations horizontales, nous trouvons l'interface d'utilisateur à utilisateur UUI (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 medias, 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 est la signalisation MAP entre plusieurs réseaux mobiles. Un autre

exemple de signalisation CCI serait une éventuelle communication de SCP à SCP telle qu'elle était prévue dans les futurs CS du réseau intelligent.

Au niveau des services réseaux, nous avons la signalisation NNI de fonction réseau à fonction réseau. Un cas particulier de signalisation NNI est la signalisation d'appel. Il convient ici d'approfondir quelque peu la notion d'appel. Suivant en cela la définition donnée par l'UIT-T pour les normes du RNIS-LB et de l'IMT-2000 [14], nous définissons l'appel comme une association (de références de contextes) ou lien logique entre points d'extrémités d'un réseau. Selon cette définition l'appel est une fonction de bout en bout et le réseau téléphonique, contrairement à la terminologie courante n'exécute pas de fonction d'établissement d'appel. Ce sont les utilisateurs « humains » du réseau téléphonique qui associent eux-mêmes leurs références et les utilisateurs « machines » le font par un protocole d'appel dans ces machines d'extrémités. La fonction des centraux téléphonique est en fait un « traitement de connexion » de proche en proche qui trouve sa place au niveau « services supports » du modèle SIMPSON. Un exemple de signalisation d'appel est la partie dialogue de TCAP [15] pour les appels entre points d'extrémités du réseau sémaphore. C'est ainsi qu'une VLR établit un appel avec une HLR par la fonction d'association de TCAP. D'autres exemples de signalisation d'appel sont le « Session Initiation Protocol » SIP [16] et la signalisation « Bearer Independent Call Control » BICC [17]. 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 fonction support à fonction support BBI. Les exemples de signalisation BBI sont nombreux, puisque la fonction de connexion du réseau téléphonique est une fonction de service support. Toutes les signalisations « associées circuit » ou signalisations CAS (Circuit Associated Signaling) sont des signalisations de connexion et donc des BBI dans la modélisation SIMPSON (et pas des NNI comme dans la terminologie courante). La signalisation BBI la plus utilisée est la signalisation ISUP pour établir les connexions entre centraux téléphoniques.

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.

2.2) Exemple d'application du modèle SIMPSON : Services Parlay sur une architecture Softswitch

La modélisation SIMPSON est un outil très puissant pour identifier les fonctions des architectures de services et examiner leur possible dégroupage. Pour illustrer ce point, nous prenons l'exemple, décrit sur la figure 4, de Services Parlay sur une architecture Softswitch.

Normalement dans les réseaux actuels les fonctions réseaux et les fonctions supports sont implémentées dans un même équipement, le commutateur téléphonique, et sont donc

nécessairement réalisées par le même opérateur. L'architecture softswitch est très intéressante dans la mesure où elle permet de réaliser un dégroupage des fonctions réseaux (établissement d'appel, routage...) exécutées par un Media Gateway Controller MGC, appartenant à un opérateur de services réseaux, et des fonctions supports (connexion) exécutées par des Media Gateways MG appartenant à un opérateur de services supports. Il existe déjà de tels exemples de dégroupages opérateurs-réseaux / opérateurs-supports. D'un côté, des opérateurs de connexion IP au niveau services supports fournissent à leurs clients des MG ou IAD (Integrated Access Device) et se chargent du transport IP. De l'autre côté, des opérateurs de services de contrôle d'appel au niveau services réseaux possédant des MGC externalisent les fonctions de traitement d'appel ou les fonctions d'IP Centrex auparavant réalisées par des PABX.

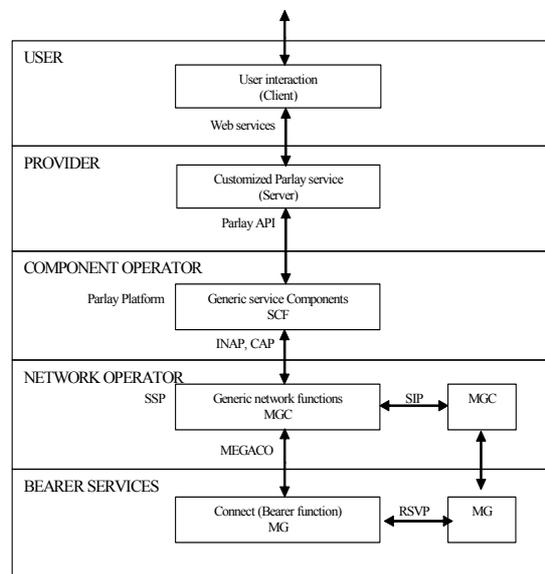


Figure 4 – Vue SIMPSON de Services Parlay sur une architecture Softswitch

Aux niveaux fournisseurs et composants le consortium Parlay propose une nouvelle architecture de services qui diffère de l'architecture du réseau intelligent par un niveau supplémentaire de personnalisation des services. Plutôt que de souscrire un service standard d'un opérateur de service réseau intelligent, une organisation peut demander une personnalisation à ses besoins propres de ce service. Dans ce cas la plate-forme de l'opérateur d'IN n'est plus qu'un composant pour une autre plate-forme fournissant le service personnalisé au niveau fournisseur, l'interface entre les deux étant réalisée par les API Parlay.

2.3) Dégroupage horizontal et dégroupage vertical

Le groupe de recherche TINA-C [18] avait proposé une structuration de l'offre de service différente du modèle SIMPSON, dans laquelle les services mettaient en œuvre des fonctions de trois types : les fonctions d'accès (Fonctions d'accès demandeur : localisation, authentification, acquisition du profil. Fonctions d'accès demandé : traduction nom adresse), les fonctions de transport et les fonctions d'intelligence (Nous appelons fonctions d'intelligence les services de réseau intelligent réalisés hors des commutateurs).

Accès	Transport	Intelligence
Client d'accès		Client d'intelligence
Fournisseur d'accès		Fournisseur d'intelligence
Composants d'accès		Composants d'intelligence
Réseau d'accès	Réseau	
Supports d'accès	Supports	

Figure 5 – Dégroupage horizontal et dégroupage vertical

Nous appelons cette séparation fonctionnelle de l'offre de service une « séparation horizontale » par opposition à la séparation fonctionnelle de SIMPSON que nous qualifierons de « séparation verticale ».

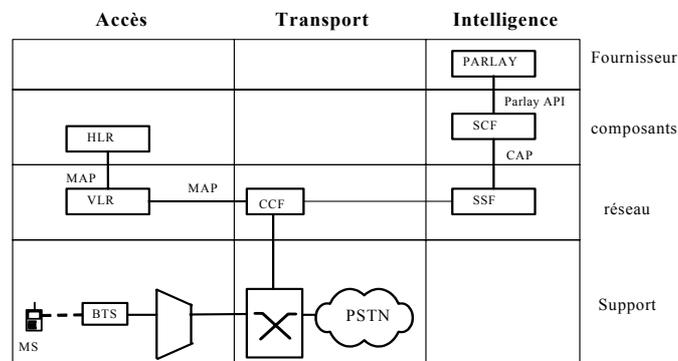


Figure 6 – Séparation fonctionnelle horizontale et verticale dans le cas des réseaux mobiles

En réalité dans chacun de ces types de fonctions nous pouvons identifier nos 5 niveaux SIMPSON, ce qui veut dire que nous avons en fait 3 « colonnes » SIMPSON correspondants aux services d'accès, aux services de transport et aux services d'intelligence. Si l'on recherche

une découpe complète de l'offre de services en fonctions indépendantes on obtient donc le dégroupage à deux dimensions de la figure 5.

Ce modèle complet de dégroupage des services de communication permet de raffiner de manière beaucoup plus complète les divers chemins de signalisations horizontales ou verticales. Cet aspect bidirectionnel de la séparation fonctionnelle des services de communication devient tout à fait évident lorsque l'on prend comme exemple les services CAMEL sur un réseau GSM comme indiqué sur la Figure 6.

3) Contexte global, associations trans-réseaux, Control Allocation Table

3.1 Contexte global et contextes locaux. Association

Les services que nous considérons sont des exemples d'informatique coopérative. Ils nécessitent la coopération de beaucoup de processus partenaires et une mémorisation des données d'instance pendant la durée de la session. A l'initialisation d'une session de service, une page mémoire se trouve donc ouverte dans chacun des processus qui coopèrent et sont des partenaires dans cette instance de service.

Nous appelons « contexte local » chacune de ces pages mémoire contenant les données d'instance des processus partenaires.

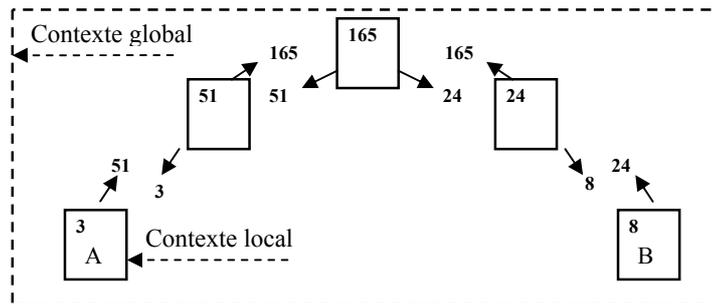


Figure 7 – Construction d'un contexte global par association des contextes locaux

Cependant ces partenaires, pour coopérer, doivent pouvoir partager leurs informations. Si Alice demande à son central de la connecter à Bob, il faut que le central puisse connaître l'identité de Bob qui se trouve dans le contexte local d'Alice.

Il faut que chaque processus puisse accéder aux informations des partenaires d'une même instance de service. Nous définissons que *des instances de services locales sont associées, si elles peuvent s'adresser mutuellement parmi de nombreuses autres instances de service dans des machines multitâches distantes*. Pour que cela soit possible, il faut qu'il y ait une table (forcément répartie) qui identifie les contextes locaux des processus partenaires.

La vue globale de l'instance de service, c'est à dire l'ensemble des données de cette instance, est donc répartie sur l'ensemble de tous les contextes locaux des processus partenaires. Nous appelons « contexte global » d'une instance de service la réunion de l'ensemble des contextes locaux des processus partenaires associés de cette instance qui donne ainsi une vue globale de l'instance de service. Cette réunion est réalisée par un mécanisme d'association qui relie ensemble, par une inter-référenciation, tous les processus locaux impliqués dans une même instance de service. Par ce mécanisme d'association, chaque contexte local possède un pointeur vers les contextes partenaires comme indiqué sur la figure 7.

3.2 Exemple d'association et difficultés des mécanismes clients serveurs

La figure 8 nous donne l'exemple d'un fournisseur réalisant une fonction de présentation du coût de communication avec une passerelle Parlay.

Chaque contexte local est identifié par sa référence locale : Le contexte du processus client est identifié par la référence Cl_k , le contexte du processus Fournisseur est identifié par la référence F_j , le contexte du processus Composant dans la passerelle Parlay est identifié par la référence Co_i .

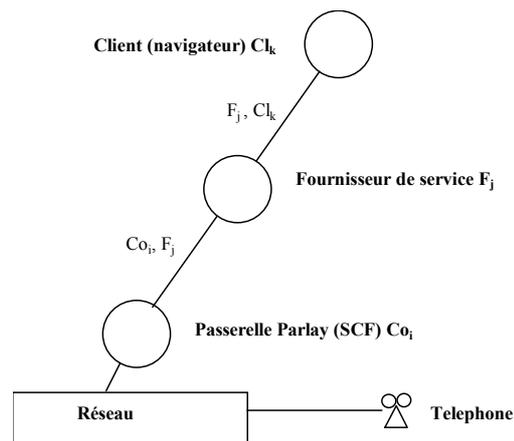


Figure 8: Exemple d'un appel déclenché par un fournisseur de service Parlay

Dans chaque contexte des couples de références locales sont maintenus et constituent des associations. Dans les réseaux téléphoniques actuels les associations sont réalisées par le protocole TCAP qui dans chaque message inclue les références de transaction et de dialogues d'origine et de destination.

Quand l'un des utilisateurs raccroche, la passerelle Parlay en reçoit une notification depuis le réseau et calcule le ticket de taxe. Elle envoie ce ticket de taxe avec l'association $[Co_i, F_j]$ au

fournisseur. L'association $[Co_i, F_j]$ permet au fournisseur de trouver le bon processus fournisseur F_j . Dans son contexte le processus fournisseur F_j trouvera la référence Cl_k du processus client pour lui envoyer l'information. Quand l'appel est libéré tous les contextes locaux sont libérés.

Supposons maintenant que l'un des utilisateurs termine la communication en cliquant sur son navigateur plutôt qu'en raccrochant. Une notification est envoyée par les web services au fournisseur. Le fournisseur peut à son tour informer le bon processus composant dans passerelle Parlay en utilisant l'API Parlay qui est transmis par l'intermédiaire de Corba [19]. Corba est un ORB persistant qui transmet l'association. La passerelle Parlay peut donc trouver la bonne instance du composant.

Il en serait tout autrement si le fournisseur utilisait, à la place de Corba, des web services localisés dans la passerelle Parlay pour communiquer avec cette passerelle. Les web services sont un mécanisme purement client-serveur et dans ce cas le client serait le fournisseur et la passerelle serait serveur. Si la communication est interrompue depuis le navigateur du client tout fonctionne comme précédemment. Mais si la communication est interrompue par un raccrochage, la passerelle Parlay ne peut pas transmettre la notification au fournisseur puisqu'un serveur ne maintient pas de pointeurs vers un client avec qui il a auparavant communiqué : un serveur (passerelle) ne peut pas faire un « push » chez un client (fournisseur). Nous découvrons là une difficulté des mécanismes client-serveurs pour la réalisation de services de communication.

Pour résoudre ce problème, le fournisseur doit pouvoir envoyer des événements à la passerelle Parlay et la passerelle Parlay doit pouvoir « pousser » des événements chez le fournisseur. Les solutions client-serveurs classiques ne conviennent pas. La solution est de maintenir une association entre ces deux processus qui persiste pendant toute la durée de la session de service, c'est à dire de mémoriser dans chaque contexte les inter-références avec les processus associés. Il existe actuellement plusieurs protocoles qui permettent d'implémenter de telles inter-références persistantes : le socket TCP en est une, il y a aussi les dialogues et les transactions TCAP, il y a enfin les associations Corba. L'inconvénient de ces protocoles est qu'ils ne permettent pas un fonctionnement trans-réseaux.

3.3 Associations trans-réseaux, Control Allocation Table CAT

Nous proposons maintenant de généraliser les mécanismes possibles d'associations persistantes pour pouvoir établir ces associations indépendamment des réseaux sous jacents.

Nous avons vu que toute la chaîne de processus qui coopèrent doivent être associés et qu'en conséquence l'ensemble des contextes locaux se trouvent liés dans un même contexte global et leurs données d'instances peuvent être partagées. Nous proposons donc qu'il existe dans chaque contexte local une zone de chaînage contenant les références des processus associés. De la même manière qu'une File Allocation Table FAT [20] chaîne ensemble les divers secteurs d'un disque pour construire une mémoire globale, ces zones de chaînage dans les contextes locaux forment une structure de liste ou une table répartie permettant de chaîner entre eux les contextes locaux pour former un contexte global. Par analogie avec la FAT, nous appelons cette table répartie Control Allocation Table : CAT. Ce mécanisme basé sur une CAT est maintenant applicable à tout type d'association, y compris pour associer des processus qui ne peuvent être atteints qu'en traversant des réseaux hétérogènes.

Lorsqu'un processus a besoin de partager des données d'instance avec un partenaire, il recherche dans la CAT ses références d'association. Comme indiqué sur la figure 9, la CAT est un graphe de références d'associations réparti sur tous les contextes locaux des processus associés. Il s'agit d'une structure de données qui persiste pendant toute la durée de la session de service. Elle est créée à l'établissement de la session et elle se trouve détruite au la fin de la session lorsque toutes les pages mémoires constituant les contextes locaux sont libérées.

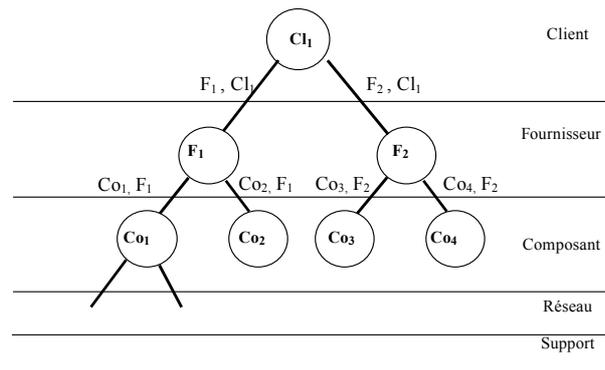


Figure 9: vue SIMPSON d'une CAT verticale

Il est utile d'attirer l'attention sur la différence fondamentale qui existe entre le mécanisme de CAT que nous proposons et les références RMI. Contrairement à ces références, les pointeurs de la CAT sont dynamiques et ne persistent que pendant la durée de la session. Ils dépendent de l'instance de service. Les références RMI d'un client vers un objet distant sont statiques et prédéfinies par le programmeur avant toute exécution de service. Elles ne dépendent pas de l'instance de service et ne caractérisent pas cette instance.

4) Le concept de plan control coopératif global

Nous avons vu que les services trans-réseaux nécessitent des passerelles de signalisation pour faire les traductions entre protocoles de signalisation. Si l'on a N signalisations différentes, le nombre de passerelles croît excessivement comme N^2 . Cette complexité se trouve aggravée par le fait qu'une traduction directe est généralement difficile, voire impossible à cause des différences comportementales entre les modèles d'appels dans des réseaux différents qui impliquent des séquencements différents des messages de signalisation.

Pour affronter cette difficulté, nous proposons une architecture décentralisée de plan contrôle permettant à des processus reliés par des réseaux hétérogènes de communiquer entre eux pour coopérer au moment de la session de service. La notion qui permet cette communication entre processus de réseaux hétérogènes est le concept de contexte générique.

Nous allons voir que ce concept de contexte générique conduit à une nouvelle définition de la signalisation ou un nouveau paradigme de la signalisation. Ce nouveau paradigme résout le problème des traductions de signalisation et en ce sens peut constituer une solution au problème de meta-signalisation posé par le groupe TISPAN de l'ETSI.

4.1) Le concept de contexte générique. Un nouveau paradigme de signalisation

On comprend généralement la signalisation comme « l'invocation d'opérations distantes ou l'échange de notifications entre processus locaux d'un processus global de contrôle ». Le concept de CAT liant les contextes nous permet de proposer une nouvelle vision ou un nouveau paradigme selon laquelle la **signalisation est l'écriture ou la lecture de données d'instance par un processus de contrôle dans une partie « non locale » du contexte global.**

Cette nouvelle vision n'est possible que parce que les contextes sont chaînés entre eux par la CAT. Selon ce nouveau paradigme les opérations distantes peuvent être invoquées par des opérations « set » sur les attributs appropriés du contexte éloigné et plus généralement la signalisation peut se réduire à des opérations get/set/notify sur les attributs appropriés du contexte éloigné.

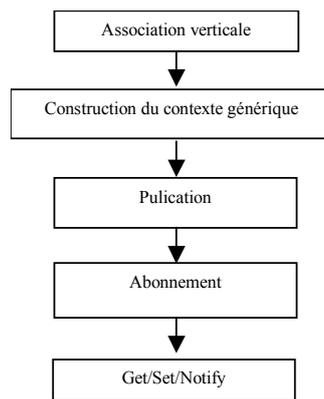


Figure 10: phases successives d'un mécanisme générique de signalisation

Ceci suppose cependant qu'un processus de contrôle puisse comprendre la syntaxe et la sémantique des données du contexte distant. Il y a donc nécessité d'une structuration des contextes qui doivent inclure une partie générique, commune à tout service, que nous appelons « contexte générique » suivie d'une partie dépendante des services. Comme indiqué sur la figure 10, à l'établissement du service, la phase d'association est suivie d'une phase de publication où le contexte éloigné informe le contexte local du contenu syntaxique et sémantique de sa partie dépendante du service.

Cette phase de publication est suivie d'une phase d'abonnement aux fonctions *get/set/notify*. Cette phase d'abonnement doit bien sûr respecter des procédures standard de sécurité. La signalisation peut alors se dérouler sous la forme de commandes *get/set/notify*.

4.2) *VCAT et HCAT*

Le graphe de la CAT se construit au fur et à mesure que les processus déjà initialisés découvrent la nécessité de démarrer des processus distant et qu'ils doivent échanger leurs premières signalisations avec ces processus (comme le TC Begin en IN). La difficulté apparaît quand le processus distant est un processus de même niveau SIMPSON dans un réseau différent sans aucune signalisation commune.

Une façon de contourner cette difficulté consiste à faire relayer la signalisation par un processus du niveau SIMPSON supérieur vers lequel nos deux processus de réseaux hétérogènes disposent chacun d'un API. Le graphe d'association devient alors une arborescence comme indiqué sur la figure 9. Nous appelons cette arborescence une « Vertical Control Allocation Table » ou VCAT. Sur la figure 9 les contextes locaux du niveau N sont représentés par un cercle et les références permettent de construire les associations entre le niveau N et les niveaux N-1 et N+1.

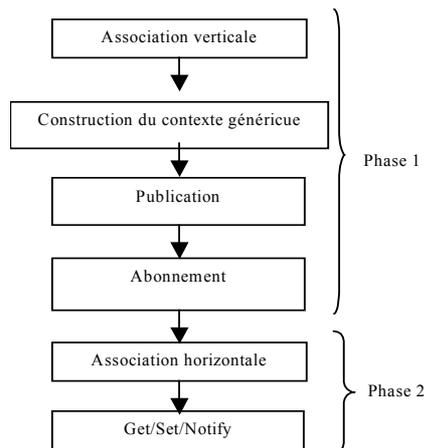


Figure 11: construction d'une HCAT dans des réseaux hétérogènes

La VCAT permet la signalisation entre processus de même niveaux situés dans plusieurs réseaux hétérogènes. Toutefois ce relayage systématique de signalisation par un processus de niveau supérieur qui occupe une position centralisée risque d'introduire une certaine latence de signalisation. Il paraît préférable d'ajouter un mécanisme permettant une interconnexion directe entre processus de même niveau SIMPSON mêmes situés dans des réseaux

hétérogènes. Nous appelons Horizontal Control Allocation Table ou HCAT un graphe d'association comprenant des liens de signalisation directs entre processus d'un même niveau SIMPSON. Comment dans ce cas pouvons nous résoudre le problème des signalisations et des modèles comportementaux incompatibles entre réseaux différents ? Une nouvelle approche consiste, comme indiqué sur la figure 11 à mettre en œuvre un mécanisme à 2 phases. Dans une première phase le contrôle inter-réseau est initialement assuré par une fonction de niveau supérieur dans le modèle SIMPSON. La CAT est alors considérée comme une V-CAT (Vertical CAT). Elle permet aux contextes locaux d'acquérir la structure, la syntaxe et la sémantique des contextes distants de même niveau. Il est alors possible de démarrer une deuxième phase où l'on associe directement entre eux les contextes d'un même niveau SIMPSON qui peuvent communiquer entre eux par l'intermédiaire de notre nouveau paradigme de signalisation, sans passer par un contexte de niveau supérieur et même s'ils sont situés dans des réseaux différents. Nous aboutissons ainsi à une réorganisation de la CAT qui devient une HCAT.

4.3) Le Cooperative Control Protocol CCP

Comme la performance est une nécessité impérative en signalisation, il importe que le protocole de signalisation soit le moins verbeux possible. Nous appelons Cooperative Control Protocol un protocole de signalisation qui implémente notre nouveau paradigme de signalisation. Le CCP, de manière assez semblable au protocole SNMP de la gestion, met en œuvre de simples commandes GET/SET pour lire ou modifier un objet distant. En utilisant la programmation orientée objets, on peut récupérer la totalité, ou une partie du contexte distant en une seule commande GET. Par exemple la commande « get contexte » renverra une instance complète du contexte distant. La commande « get contexte.media » renverra une instance de l'objet « media » du contexte distant. Le téléchargement de la totalité d'un objet est une option de certains middlewares orientés objet comme les RMI. En général cette option n'est pas utilisée et seule les invocations de méthodes distantes sont possibles.

La structure du contexte générique est orientée objet. Nous donnons dans la figure 12 un diagramme UML partiel d'un exemple d'architecture du contexte générique et nous décrivons maintenant quelques exemples de signalisation.

Soit l'attribut "session type". Cet attribut peut avoir les valeurs Multi-cast, conference, broadcast, single-party...

Pour demander la valeur de cet attribut au contexte générique distant, plusieurs solutions sont possibles.

On peut télécharger le contexte entier par la commande « get contexte », puis en utilisant localement une méthode « get contexte.session().get type » de l'objet session.

Une autre solution consiste à ne télécharger que l'objet session du contexte générique puis à utiliser localement la méthode « get type » de l'objet session.

Pour modifier la valeur d'un attribut dans un contexte distant, on télécharge la totalité (ou une partie) du contexte distant, on modifie localement l'attribut en question en utilisant « setMethodName(...) ». Une fois la modification effectuée on renvoie l'objet modifié au processus distant avec une commande SET (par exemple « setcontexte »).

Dans la méthodologie des web services (et de la plupart des middlewares orientés objet), l'objet qu'un client veut modifier reste distant et n'est pas téléchargé sur la machine du client. Celui-ci doit invoquer à distance les méthodes de l'objet distant.

Au contraire dans l'approche de notre nouveau paradigme de signalisation, nous considérons qu'il est plus efficace de télécharger d'abord l'objet (get contexte), de lire ou modifier ses attributs et en cas de modification de recharger l'objet modifié dans l'entité distante (set contexte). Le résultat étant de déclencher une opération du processus distant.

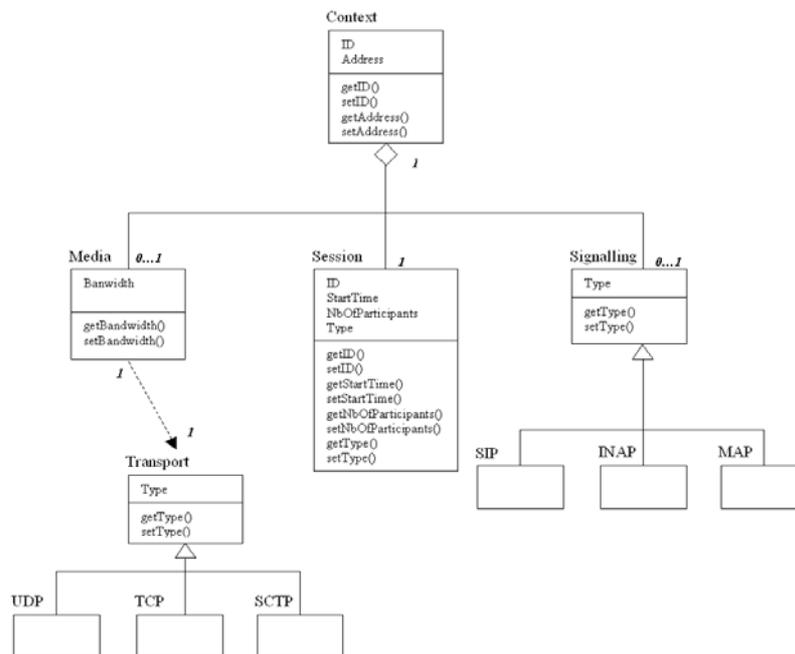


Figure 12: représentation UML partielle du contexte générique

5) Un exemple d'implémentation d'une CAT

Nous donnons maintenant un exemple de service pour illustrer notre conception du plan contrôle global. Le service que nous utilisons pour cet exemple est une vidéoconférence où la voix, pour en assurer la qualité, est transmise par le réseau téléphonique commuté et la vidéo est transmise par l'internet. La figure 13 nous montre la structure de CAT de ce service. Le service est exécuté par la plate-forme d'un fournisseur. La logique de ce service répond à la séquence suivante :

- 1- Un client (Alice) démarre le service sur son navigateur en communiquant par l'Internet avec la plate-forme du fournisseur. Celui ci informe Alice du tarif du service en supplément de la tarification téléphonique.
- 2- Le fournisseur demande à la passerelle Parlay de démarrer une session d'appel téléphonique entre Alice et le demandé Bob ainsi que de superviser l'éventuel décrochage de Bob.
- 3- Bob répond
- 4- Le fournisseur est informé par la passerelle Parlay du décrochage de Bob. Il demande à un composant serveur de vidéoconférence de démarrer une session vidéo sur l'Internet entre les ordinateurs d'Alice et de Bob. L'heure de début de cette session est enregistrée dans le contexte local du vidéo-serveur.
- 5- Bob ou Alice terminent la session soit en fermant la webcam soit en raccrochant le téléphone.

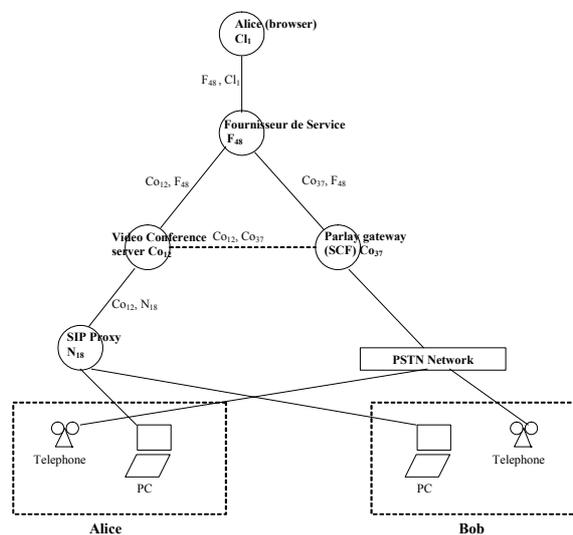


Figure 13 : exemple d'implémentation d'un service de vidéoconférence simple

Pour réaliser ce service, des contextes locaux sont ouverts chez le fournisseur, dans la passerelle Parlay et dans le vidéo-serveur. A l'établissement de la session, ces divers contextes sont associés par une VCAT. Une fois cette VCAT construite, le fournisseur, la passerelle Parlay et le vidéo-serveur peuvent se construire un contexte générique et le mettre en commun par publication comme dans les protocoles Peer to Peer. Alors la passerelle Parlay et le vidéo-serveur peuvent associer directement leurs contextes en créant ainsi une HCAT. Dès lors les signalisations peuvent gagner en performance en évitant de passer par le point central du fournisseur. Le contexte du fournisseur, n'étant plus vraiment utile, peut éventuellement être libéré. Si Bob termine la session en fermant sa webcam le vidéo-serveur est prévenu et notifie directement la passerelle Parlay pour qu'elle relâche l'appel téléphonique. Si le contexte du

fournisseur avait été libéré, le vidéo-serveur ou la passerelle Parlay peuvent envoyer une notification de fin par un mécanisme client-serveur stateless.

6) conclusions

Le problème des services impliquant plusieurs réseaux et réalisés par la coopération de plusieurs composants fournis par des acteurs différents est rarement abordé. Ce problème se heurte à la grande difficulté de l'inter fonctionnement de protocoles de signalisation différents. Notre étude donne une première analyse de ce problème ainsi qu'une nouvelle approche pour la création d'une meta-signalisation générique. Pour atteindre ce but, nous avons utilisé une nouvelle modélisation des service (modèle SIMPSON) très structurante pour étudier les services de télécommunication, pour découvrir les chemins manquants de signalisation et pour identifier de nouvelles fonctions de contrôle. Nous avons utilisé ce modèle pour proposer de nouveaux mécanismes pour réaliser les associations nécessaires entre les multiples partenaires d'un service. Ces nouveaux mécanismes d'associations nous ont finalement conduit à repenser le concept même de signalisation et à proposer une nouvelle approche pour le problème de la meta-signalisation au sens de TISpan et à la constitution d'un plan contrôle global.

Les travaux ultérieurs devraient maintenant porter sur le remplacement de chacune des signalisations verticales et horizontales actuelles par notre protocole générique CCP, ainsi que sur l'étude des modèles comportementaux liés aux signalisations horizontales permettant le déclenchement des signalisations verticales.

REFERENCES

- [1] W3C : "Web Services Description Language (WSDL) 1.1", 15 march 2001
<http://www.w3.org/TR/wSDL>
- [2] Michael Stal: "Effective Architectures for Distributed Object Computing", Siemens :
<http://www.stal.de/Downloads/docpatterns.pdf>
- [3] IETF RFC 2458: "Toward the PSTN/Internet Inter-Networking--Pre-PINT Implementations", November 1998
IETF RFC 2848: "The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services", June 2000
- [4] Parlay : <http://www.parlay.org>
- [5] Astronefs: "Network and Telecommunication Global Service Convergence: White paper",
<http://www.infres.enst.fr/~rigault/white-paper.pdf>
- [6] Global Functional Plane for Intelligent Network CS1. ITU-T Recommendation Q1213

- [7] CAMEL: ETSI recommendation TS 101 046 (V7.0.0)
- [8] ETSI ES 201 915 Phase 1: "Open Service Access (OSA); Application Programming Interface (API); all parts", 02/2002
ETSI ES 202 915 Phase 2: "Open Service Access (OSA); Application Programming Interface (API); all parts", 07/2002
ETSI ES 203 915 Phase 3: "Open Service Access (OSA); Application Programming Interface (API); all parts", 10/2002
- [9] Soft-switch architecture : IETF RFC 2705: Media Gateway Control Protocol (MGCP) Version 1.0 Oct. 1999, IETF RFC 3015: Megaco Protocol Version 1.0, Nov. 2000
- [10] IETF Internet-Draft: "Next Steps in Signalling: Framework ", 5 July 2004
IETF RFC 3726: "Requirements for Signalling Protocols", April 2004
- [11] TISPAN: http://www.item.ntnu.no/fag/ttm4130/NGN_Workshop_Protocol.ppt
- [12] "Principe général d'architecture d'un réseau NGN", Etude technique, économique et réglementaire de l'évolution vers les réseaux de nouvelle génération, Etude réalisée par le cabinet Arcome pour le compte ART, Septembre 2003, <http://www.art-telecom.fr/>
- [13] Ericsson's Service Capability Server Parlay Gateway
http://www.ericsson.com/mobilityworld/sub/open/technologies/parlay/about/parlay_about_gs1
- [14] ITU-T recommendation Q 1701 Signalling requirements for IMT-2000 networks
- [15] ITU-T Q.771_Q775 : TCAP : Transaction Capability
- [16] IETF RFC 3261: "Session Initiation Protocol (SIP)", June 2002
- [17] ITU-T Recommendation Q1901: BICC: Bearer Independent Call Control
- [18] TINA-C Consortium (Telecommunication Information Networking Architecture).
www.tinac.com/
- [19] CORBA : www.corba.org
- [20] Silberschatz and Galvin: "Operating System Concepts", 2004:
<http://www.cs.biu.ac.il/~wiseman/os/os/os12.pdf>