



HAL
open science

Architecture des services de communication dans un contexte de convergence

Emmanuel Bertin

► **To cite this version:**

Emmanuel Bertin. Architecture des services de communication dans un contexte de convergence. Autre. Institut National des Télécommunications, 2009. Français. NNT : 2009TELE0018 . tel-00541869

HAL Id: tel-00541869

<https://theses.hal.science/tel-00541869>

Submitted on 1 Dec 2010

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

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



Ecole Doctorale EDITE

**Thèse présentée pour l'obtention du diplôme de
DOCTEUR DE L'INSTITUT NATIONAL DES TELECOMMUNICATIONS**

*Doctorat délivré conjointement par
L'Institut National des Télécommunications et l'Université Pierre et Marie Curie - Paris 6*

Spécialité Informatique et Télécommunications

Par Emmanuel Bertin

**Architecture des services de communication dans
un contexte de convergence**

Soutenue le 11 décembre 2009 devant le jury composé de :

Serge Fdida	Professeur à l'université Pierre et Marie Curie	Président du jury
Yvon Kermarrec	Professeur à Télécom Bretagne	Rapporteur
Pascal Lorenz	Professeur à l'université de Haute-Alsace	Rapporteur
Prosper Chemouil	Directeur de recherche, Orange Labs	Examineur
Noémie Simoni	Professeur à Télécom ParisTech	Examineur
Jukka Manner	Professeur à Helsinki University of Technology	Invité
Djamal Zeghlache	Professeur à Télécom SudParis	Directeur de thèse
Noël Crespi	Directeur d'études à Télécom SudParis	Encadrant

Thèse préparée au sein des Orange Labs et du Département RS2M de Télécom SudParis

Thèse n° 2009TELE0018

A Agnès

Remerciements

Au terme de ce travail, je souhaite tout d'abord remercier l'organisation qui m'a donné l'opportunité de le réaliser : les Orange Labs, incarnés notamment par mes responsables d'unité Philippe Michon puis Pascal Lesieur, ainsi que par mes responsables de laboratoire Olivier Bouillant puis Frédéric Delmond. Qu'ils soient remerciés de leur confiance.

Je remercie ceux qui ont accepté d'évaluer ce travail et de m'aider à l'améliorer : Pascal Lorenz et Yvon Kermarrec, qui a bien voulu œuvré dans des délais plus que serrés. Merci aux examinateurs, à Noémie Simoni, à Serge Fdida et à Prosper Chemouil, dont les avis m'ont toujours éclairé. Merci à Djamal Zeghlache et à Noël Crespi qui m'ont accompagné tout au long de cette thèse. Je remercie également les enseignants de Télécom Bretagne qui m'ont transmis le goût de l'architecture des services télécoms lors de ma formation d'ingénieur, en particulier Simon Znaty.

Ce travail a été principalement porté au sein des Orange Labs par les projets Urbans et Servery. Je remercie les contributeurs présents ou passés de ces deux projets. Nos nombreux échanges et débats ont contribué à l'aboutissement de cette thèse. Merci notamment à Hervé Le Corre, à Rémi Bars, à Jean-Pierre Deschrevel, à Alexis Bafcop, à Simon Bécot, à Stéphanie Fodor, à Ivan Bedini, à Mariano Belaunde, à Philippe Deschaseaux, à Francis Alizon, à Thibaut Feuillet, à Yann Sendra, à Frédéric Klamm, à Philippe Fradet, à Marc Garagnon. Je remercie également Laurence Kaufmann, professeure à l'université de Lausanne, dont les remarques m'ont aidé à cerner la notion d'invariant langagier, et Luigi Lancieri, ancien collègue et maintenant professeur à l'université de Lille 1, dont les encouragements et le soutien méthodologique ont été précieux. Un grand merci à Jacques Simonin pour son rôle clé dans l'élaboration de ce travail, pour sa disponibilité constante et la pertinence sans faille de ses remarques.

Je remercie les thésards dont j'ai encadré le travail, Imen Grida Ben Yahia et Nassim Laga ; leurs questionnements m'ont aidé à avancer. Merci enfin à Agnès, pour sa relecture attentive et patiente.

Résumé

Les services de communication de nouvelle génération doivent pouvoir coopérer pour répondre à des besoins spécifiques tout en gardant leur autonomie. Ceci nécessite de maîtriser leurs architectures et de partager ces architectures au sein de l'entreprise. Des cadres architecturaux communs sont alors indispensables. Après avoir fait le point des travaux sur ce sujet dans les domaines télécoms, web et IT, et après avoir discuté des enjeux de la convergence des services télécoms, nous introduisons ici quatre angles de vue (métier, fonctionnel, technique et applicatif), ainsi qu'une méthodologie pour construire des vues de référence partagées au sein d'une entreprise et des vues d'architecture propres à chaque service. Nous illustrons cette démarche à l'aide d'exemples et montrons ses applications pour construire des offres de service, pour rationaliser les services existants et pour réaliser une convergence entre différents services.

Abstract: Towards a converged environment for communication services

Next-generation communication services should be able to cooperate in order to meet specific needs, while keeping their autonomy. This implies to master their architectures and to share these architectures within the enterprise. Architectural frameworks are therefore essential. After a review of the state of the art in the telecom, web and IT fields, and after having discussed the stakes of the telecom service convergence, we introduce here four viewpoints (business, functional, technical and applicative), as well as a methodology to build reference views that will be shared across the enterprise, and architectural views that are dedicated to each service. We illustrate this approach through case studies and we point out its applications to build service offers, to rationalize existing services and to realize a convergence between various services.

TABLE DES MATIERES

Introduction	13
1 Contexte	13
2 Problématique.....	14
3 Contribution	15
4 Cadre de travail	16
5 Organisation du document	16
Chapitre I Décrire les architectures de services	19
I.1 Dans la communauté télécom	20
I.1.1 Les plans service et fonctionnel global du Réseau Intelligent	20
I.1.2 De TINA au NGN	25
I.1.3 OMA et les enablers	28
I.2 Dans la communauté Web.....	32
I.2.1 Les services web comme unité architecturale de base	32
I.2.2 La description sémantique de ressources	34
I.2.3 La description sémantique de services web	35
I.3 Dans la communauté IT	39
I.3.1 Les architectures orientées service	39
I.3.2 La notion de vue	43
I.3.3 L'architecture d'entreprise et l'urbanisation	45
Synthèse	51
Chapitre II La convergence de service	53
II.1 Panorama des services de communication	53
II.1.1 Les services télécoms et le NGN.....	53
II.1.2 Vers la convergence	57
II.1.3 Implémentation de services convergents.....	60
II.2 Evolutions communes du secteur des services.....	61
II.2.1 Le point de vue de la production du service : du besoin au service.....	63
II.2.2 Le point de vue de l'utilisation du service : client, consommateur, utilisateur	67
II.2.3 Du service au système de service	71
II.3 Application aux services télécoms	73

II.3.1	Les services télécoms du point de vue de la production ?	73
II.3.2	Les services télécoms du point de vue de l'utilisation ?.....	75
II.3.3	Comment architecturer les services télécoms ?.....	77
	Synthèse	85
Chapitre III	Construire un référentiel d'architecture pour les services télécoms	87
III.1	Une vue métier de référence pour les services télécoms.....	88
III.1.1	Comment définir les activités de la vue métier ?	88
III.1.2	Comment construire une vue métier de référence ?.....	93
III.1.3	Evaluation et résultats	100
III.2	Une vue fonctionnelle de référence pour les services télécoms.....	110
III.2.1	Comment définir les composants de la vue fonctionnelle ?.....	110
III.2.2	Comment construire une vue fonctionnelle de référence ?	116
III.2.3	Evaluation et résultats	130
III.3	Une vue technique de référence pour les services télécoms	148
III.3.1	Comment définir les éléments de la vue technique ?	148
III.3.2	Comment construire une vue technique de référence ?.....	152
III.3.3	Evaluation et résultats	154
	Synthèse	171
Chapitre IV	Utiliser le référentiel pour décrire l'architecture des services télécoms	173
IV.1	La vue fonctionnelle d'un service.....	174
IV.1.1	Méthode de description	174
IV.1.2	Exemples de services simples	181
IV.1.2.1	Exemple d'un service de téléphonie	182
IV.1.2.2	Exemple d'un service de vidéo.....	186
IV.1.2.3	Exemple d'un service de messagerie	189
IV.1.3	Cas d'un service composé : VideoMe	192
IV.2	La vue technique d'un service	196
IV.2.1	Méthode de description	196
IV.2.2	Exemples de services simples	199
IV.2.2.1	Exemple d'un service de téléphonie	199
IV.2.2.2	Exemple d'un service de vidéo.....	201
IV.2.2.3	Exemple d'un service de messagerie	203
IV.2.3	Cas d'un service composé : VideoMe	205

IV.3	La vue applicative d'un service	207
IV.3.1	Comment identifier et décrire les composants applicatifs d'un service ?.....	208
IV.3.2	Exemples de services simples	215
IV.3.2.1	Exemple d'un service de téléphonie	215
IV.3.2.2	Exemple d'un service de vidéo	220
IV.3.2.3	Exemple d'un service de messagerie	223
IV.3.3	Cas d'un service composé.....	227
	Synthèse	231
Chapitre V	Applications et perspectives organisationnelles et logicielles	233
V.1	Une aide pour la construction des offres de service.....	233
V.1.1	La conception de services	233
V.1.2	L'évaluation et la comparaison de services	236
V.1.3	La gestion de services	240
V.2	Une aide pour la rationalisation des services	242
V.2.1	Le cas des enableurs	242
V.2.2	Le cas des services logiciels.....	244
V.2.3	Le cas des services sémantiques.....	246
V.3	Une aide pour réaliser la convergence de service	248
V.3.1	Un motif technique pour la convergence web/IMS	249
V.3.2	Un accès unifié aux services	256
V.3.3	La communication entre services	261
	Synthèse	266
	Conclusion.....	267
	Bibliographie.....	273
	Publications	273
	Références	278
	Annexes.....	291
	Acronymes des vues fonctionnelle, technique et applicative.....	291
	Autres acronymes	292
	Table des figures	295
	Table des tableaux.....	299

Introduction

Les architectures de service de communication ont fait l'objet de nombreux travaux de modélisation. Toutefois, des ruptures en termes de technologie, de marché, d'usages, nous obligent à repenser nos façons de décrire ces architectures de services.

1 Contexte

Le NGN (pour Next Generation Network) désigne d'une façon générique les nouvelles architectures de réseau de communication, qui reposent sur une distinction et une indépendance entre les niveaux transport, contrôle et service, ainsi que sur un réseau de transport IP mutualisé pour tous les services. Le NGN doit être ouvert à une large gamme de services multimédia et est vu comme un moyen de réaliser plus facilement la convergence des services conversationnels et des services de données.

Mais les services de télécommunications ne sont plus l'apanage des opérateurs. Des fournisseurs de service sans réseau utilisent le réseau universel Internet pour rendre leurs services, comme par exemple Skype, Google ou YouTube. Ils ne s'appuient pas sur une architecture réseau pour offrir leurs services, mais sur des architectures techniques issues du monde web et du monde IT (*Information Technology*). Ces acteurs, qui ont généralement un modèle de revenus basé sur la publicité, ont profondément modifié les services télécoms en amenant une plus grande prise en compte de l'utilisateur d'un service par rapport au client d'une offre d'accès.

Les services de communication sont en évolution rapide. Didier Lombard, président directeur général de France Telecom parle ainsi d'évolution des "*réseaux vers un melting-pot de services et de contenus*", dans son ouvrage [Lombard, 2008] sous-titré "*la deuxième vie des réseaux*". Les services de communication ne se limitent plus aux communications vocales interpersonnelles, mais intègrent des fonctionnalités comme email, carnet d'adresses, messagerie instantanée. Les services de contenu deviennent également des services phares, et des prototypes permettent de les coupler avec des services conversationnels. Pour permettre aux utilisateurs de gérer au mieux ce foisonnement de services, certains acteurs tentent de leur

fournir la possibilité de composer des services, c'est-à-dire d'assembler eux-mêmes des éléments de service divers pour répondre à un besoin spécifique. Les frontières entre les différents services deviennent plus ténues. Par exemple, les contacts d'un service de carnet d'adresses peuvent être utilisés aussi bien pour envoyer un email que pour émettre un appel téléphonique. L'utilisateur peut ainsi accéder à un environnement de services capables d'interagir pour répondre à un besoin. C'est ce que Noémie Simoni appelle, dans son ouvrage [Simoni, 2007], la nouvelle génération de services.

Les différents services offerts par un opérateur télécom à ses clients ne peuvent plus être indépendants. Toutefois, construire une application monolithique regroupant tous les services est un leurre, ne serait-ce que pour des raisons de faisabilité technique, et d'agilité nécessaire dans l'introduction de nouveaux services. Les architectures de service deviennent ainsi nécessairement complexes, composées d'une multitude de modules en interaction.

2 Problématique

Si la description des architectures de réseau est relativement claire, reposant sur les acquis de décennies de standardisation, la description des architectures de service de communication est moins étudiée. On utilise souvent des termes désignant un environnement technique d'exécution d'applications informatiques, comme AS (*Application Server*), plate-forme de service ou SDP (*Service Delivery Platform*), pour indiquer le service rendu, en parlant par exemple d'AS de téléphonie ou d'AS de présence. Par ailleurs, on tend à standardiser des *enablers*, des applications réutilisables dans le cadre de divers services, pour fournir des fonctions fréquemment utilisées, sans que les méthodes de construction de ces *enablers* soient clarifiées et sans que la distinction soit faite entre des *enablers* rendant un service aux utilisateurs et des *enablers* fournissant une facilité technique.

Cette thèse vise à proposer un cadre général pour mieux décrire les architectures de service de communication. Cela permettra d'une part de formaliser ces architectures en utilisant une syntaxe et une sémantique communes aux différents services, et d'autre part de répondre à des questions diverses, comme l'identification du périmètre fonctionnel des *enablers* ou des services SOA, la pertinence de compositions de service pour l'utilisateur ou la communication entre services dans un environnement convergent.

3 Contribution

Le point de départ de cette thèse a été d'étudier si les principes de l'urbanisme des systèmes d'information pouvaient être déclinés pour les services de communication. En effet, ces principes permettent dans le monde IT de décrire, de façon univoque et partageable entre différents métiers, l'architecture des systèmes d'information d'entreprise dont la complexité peut dépasser celle des services télécoms, du moins pour de grandes entreprises telles les banques ou les compagnies d'assurance. Les systèmes d'information d'entreprise y sont décrits selon quatre vues : métier, fonctionnelle, technique et applicative. Si les vues applicatives et techniques se transposent relativement intuitivement, il n'en est pas de même des vues métier et fonctionnelle. Nous nous heurtons en effet à l'absence de processus métier formalisés pour l'usage des services de communication, et donc à l'absence d'une vue métier. Le remplacement des processus métier par des cas d'utilisation de services a été étudié, mais les cas d'utilisation sont trop volatiles, trop spécifiques à un service particulier pour fournir une base stable.

Nous nous sommes en effet rendu compte que l'urbanisation des systèmes d'information d'entreprise reposait implicitement sur l'identification d'invariants dans ces systèmes d'information. Christophe Longépé identifie ainsi dans son ouvrage de référence [Longépé, 2006] des invariants issus des processus métier, des invariants issus de la définition des objectifs d'un système d'information d'entreprise, et des invariants techniques. Nous avons alors resitué cette recherche d'invariants dans les travaux des communautés télécom, web et IT concernant les architectures de services. Puis nous avons cherché à caractériser précisément les différentes dimensions des services et de leurs utilisateurs, le sens de ces termes variant sensiblement selon les communautés de recherche et les métiers. Ceci nous a conduit à préciser les caractéristiques d'un service en général, afin de mieux cerner les services de communication. Nous nous sommes alors aperçu que la nouvelle génération de services de communication, convergents, centrés sur l'utilisateur (*user-centric*), s'inscrivait dans les évolutions générales du secteur des services. Ce faisant, en distinguant entre le point de vue de la production du service et le point de vue de l'utilisation du service, nous avons pu dégager les prémices d'une distinction entre vue technique et vue fonctionnelle d'un service de communication. Nous avons ensuite formalisé et systématisé ces intuitions en identifiant les invariants recherchés dans une vue métier de référence, une vue fonctionnelle de référence et une vue technique de référence, puis en montrant comment les utiliser pour décrire un service de communication, à travers une vue fonctionnelle, une vue technique et une vue applicative

du service. Nous avons ainsi montré que les principes de l'urbanisme des systèmes d'information pouvaient bien être transposés aux services de communication, mais au prix d'une redéfinition complète du contenu des vues métier et fonctionnelle et d'une définition plus ciblée de la vue technique afin de prendre en compte à la fois les composants télécoms et informatiques. Nous obtenons alors une vue applicative qui permet de décrire simplement l'architecture des services de communication de nouvelle génération.

Nous avons appliqué ces résultats à des problématiques diverses, comme la conception et la validation d'offres de service, l'identification d'enablers, de services SOA, de widgets, ou la composition de services et la communication entre services dans un environnement de services convergents, montrant ainsi l'intérêt d'une approche dérivée de l'urbanisation des systèmes d'information et reposant sur l'identification d'invariants pour répondre aux problématiques architecturales liées au contexte de convergence des services.

4 Cadre de travail

Cette thèse s'est déroulée dans le cadre de l'équipe RS2M de l'institut Télécom SudParis. Les travaux réalisés se sont également inscrits au sein du projet Urbans d'urbanisation des services télécoms au sein des Orange Labs, ainsi que de projets de recherche sur la composition de services de communication, notamment le projet collaboratif CELTIC Server¹.

5 Organisation du document

Le chapitre I expose l'état de l'art sur la description des architectures de service dans les communautés télécoms, web et IT. Le chapitre II introduit en détail les ruptures qui conduisent les opérateurs à construire des services centrés sur l'utilisateur et présente les travaux académiques pertinents à prendre en compte pour définir un service. Les chapitres III et IV constituent la principale contribution apportée par cette thèse à la description des architectures de service télécom. Le chapitre III étudie la possibilité d'identifier des invariants dans les services télécoms pour construire une vue métier, une vue fonctionnelle et une vue technique de référence. Le chapitre IV utilise les invariants des vues fonctionnelle et technique de référence pour décrire un service télécom (que ce soit un service unitaire, un *enabler*, un service composé) à travers une vue fonctionnelle, une vue technique et une vue

¹ <http://projects.celtic-initiative.org/server/>

applicatives du service. A chaque étape, nous élaborons un méta-modèle afin de formaliser les relations entre les concepts mis en œuvre. Au chapitre III, nous proposons un méta-modèle d'urbanisme de service pour construire le référentiel d'architecture des services télécoms. Au chapitre IV, nous introduisons un méta-modèle d'architecture de service pour décrire un service télécom à partir de ce référentiel. Le chapitre V présente les applications réalisées dans le cadre de cette thèse et leurs perspectives.

Chapitre I Décrire les architectures de services

Cette thématique a déjà donné lieu à de nombreux travaux dans les communautés télécom, web et IT. Dans ce chapitre d'état de l'art, notre objectif n'est pas de les détailler de façon exhaustive, mais d'en faire ressortir les résultats clés et les points marquants, qui nous serviront par la suite de base de travail.

A quoi sert une architecture de service ? A travers les diverses sections de ce chapitre, nous verrons qu'elle permet de répondre aux préoccupations des parties-prenantes de la conception d'un service. En amont, en aidant les décideurs à identifier les enjeux majeurs des services (enjeux techniques, enjeux fonctionnels) et leur positionnement les uns vis-à-vis des autres, y compris dans une optique de mutualisation et de contrôle des coûts. En aval, en fournissant aux équipes de développement et de déploiement des services une expression de besoin formelle et non équivoque sur les différents éléments à développer/déployer et sur les relations entre eux. Un chef de projet de développement/déploiement peut alors avoir une vision claire des tâches à réaliser. Une architecture de service peut ainsi être considérée comme un lieu de délibération entre les diverses parties-prenantes dans la conception d'un service, et notamment entre les multiples décideurs (décideurs marketing, décideurs techniques, décideurs financiers...) et les multiples équipes de développement ou de déploiement (réseau, plates-formes de service...). Chaque partie-prenante ayant son propre vocabulaire, son propre métier, ses propres contraintes, cette délibération est médiatisée par des représentations (l'architecture de service), et par des personnes (les architectes de service).

Précisons maintenant le contenu de ces architectures de service, que nous retrouverons sous différentes formes dans la suite de ce chapitre. Le terme architecture est défini dans le dictionnaire Le Robert comme *"organisation des éléments composant un système"*. Organisation est définie dans ce même dictionnaire comme la *"façon dont un ensemble est constitué en vue de son fonctionnement"*. Nous pourrions ainsi définir l'architecture comme la façon dont un système est constitué par des éléments, en vue de son fonctionnement. L'architecture est la réponse à des exigences (notamment le bon fonctionnement du système), ces exigences étant satisfaites grâce à l'identification d'éléments constitutifs puis à la mise en relation de ces éléments entre eux. Dans chacune des sections de ce chapitre, nous

retrouverons ainsi cette recherche d'éléments constitutifs en relation pour répondre à des exigences (*requirements*).

Au cours de ce chapitre, nous n'approfondirons volontairement pas le terme de service. Nous indiquerons la signification de ce terme au sein de chaque communauté de recherche, mais ne chercherons pas à l'analyser d'une façon globale ; cela sera l'objet du chapitre suivant.

I.1 Dans la communauté télécom

I.1.1 Les plans service et fonctionnel global du Réseau Intelligent

Le monde télécom a une longue expérience de la conception de service. On peut y faire remonter la notion de service aux années 70 avec les commutateurs du RTC (pour Réseau Téléphonique Commuté). Le comportement de ces commutateurs est défini par un automate à états, définissant le service téléphonique de base. Petit à petit, les opérateurs et les équipementiers télécoms ont rendu possibles des modifications de ce comportement de base, déclenchées selon divers critères. Ces modifications se sont appelées services supplémentaires. Ils ont été initialement implémentés dans le code des commutateurs, puis à partir de la fin des années 80, afin de raccourcir les délais et de diminuer les coûts de développement, dans des entités externes aux commutateurs, à travers ce que l'on a appelé le réseau intelligent (abrégié par RI). Le principe de base du RI est de demander au commutateur d'interrompre son traitement par défaut en un point particulier de son automate à état, et sous certaines conditions, pour faire appel à une plate-forme de service, puis d'interpréter les commandes reçues de cette plate-forme. Le RI a ainsi introduit une dissociation entre le réseau et les services, comme illustré par le schéma suivant extrait de la norme Q.1201 [ITU, 1992a].

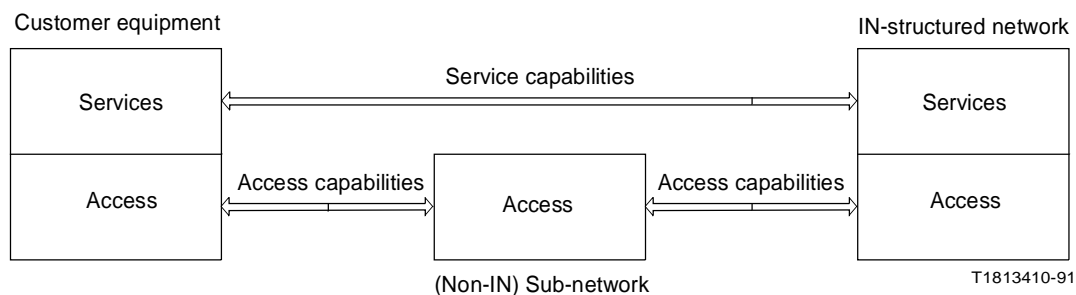


Figure 1: Séparation entre service et accès

Pour mettre en œuvre ce principe de façon normalisée et pour faciliter la conception des services selon ce mode, la communauté RI a défini un "modèle conceptuel" (appelé INCM pour *Intelligent Network Conceptual Model* dans la norme ITU-T Q.1201). Ce modèle est constitué de 4 plans, chaque plan correspondant à une vue architecturale différente (au sens de la section 3.1 de ce chapitre).

- Le plan service [ITU, 1997a] décrit un service RI tel qu'il peut être vu par un usager du service, par exemple un service de numéro vert, de transfert d'appel, de numérotation abrégée ou d'appel par carte de crédit.
- Le plan fonctionnel global [ITU, 1997b] décrit le déroulement d'un service, selon une méthode formelle, à travers un enchaînement de composants formels appelés SIB pour *Service Independent Building Block*.
- Le plan fonctionnel réparti [ITU, 1993a] ne décrit plus le service rendu, mais l'implémentation logicielle du service. Une architecture d'exécution du service définit les entités mises en œuvre, reliées entre elles par des protocoles, et leur comportement.
- Le plan physique [ITU, 1993b] décrit l'implémentation du plan fonctionnel réparti sur des machines physiques.

Les plans service et fonctionnel global permettent de définir les services ; les plans fonctionnel réparti et physique permettent de définir leur réalisation logique et physique dans un environnement particulier. Détaillons maintenant les plans consacrés spécifiquement à l'architecture de service : les plans service et fonctionnel global.

Sur le plan service, un service est décrit comme une composition d'éléments de service (*service feature*). Par exemple, un service de numéro vert pourra être composé d'un élément de service "répartition des appels" et d'un élément de service "mise en file d'attente". Le plan service est présenté de la façon suivante dans la norme Q.1202 le définissant :

" Le plan des services illustre le fait que des services assurés par le RI peuvent être décrits à l'utilisateur final ou à l'abonné à l'aide d'un ensemble de blocs génériques appelés "éléments de service". (...) Un élément de service est un aspect spécifique d'un service qui peut être également utilisé conjointement avec d'autres services ou éléments de service dans le cadre d'une offre commerciale. Il s'agit soit d'une partie essentielle d'un service soit d'une partie facultative offerte pour améliorer un service. Le plan des services représente une vue exclusivement orientée vers les services. Cette vue ne contient aucune information quelconque concernant la mise en œuvre des services dans les réseaux (par exemple, un type

RI de mise en œuvre est invisible). Tout ce que l'on perçoit est le comportement du réseau lié au service, tel que ce comportement apparaît, par exemple, à un usager du service. [ITU, 1997a – nous soulignons en gras]

Comme l'indique la dernière phrase, l'architecture d'un service RI est constituée dans le plan service à partir d'une vision externe de ce service, et non à partir du fonctionnement interne des réseaux ou des plates-formes. Sur ce plan premier, c'est la perception du service par ses utilisateurs qui est fondamentale. La structure proposée peut être synthétisée dans la figure suivante.

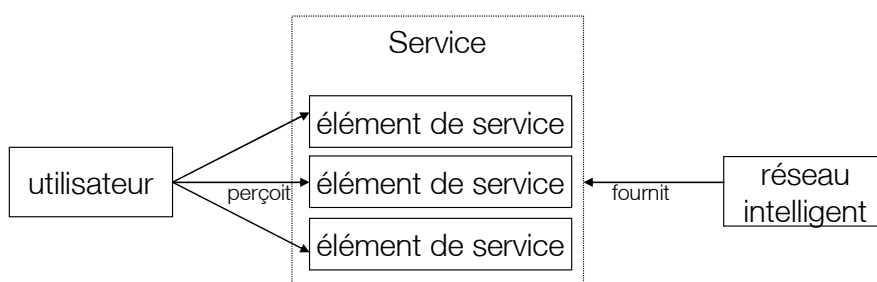


Figure 2 : Structure en éléments de service

A la suite de [Magedanz, 2007], on peut considérer ce modèle comme précurseur des approches de composition de service, actuellement en vogue à travers par exemple les services dits web 2.0.

Toutefois, cette décomposition de services en éléments de services n'a jamais été pleinement exploitée dans la communauté RI. Premièrement, les spécifications Q.12x2 restent muettes sur la façon d'identifier des éléments de services, au-delà du critère du bon sens (c'est-à-dire en essayant de dégager des points communs à partir de services connus). Deuxièmement, ces mêmes normes ne détaillent pas la façon d'utiliser les éléments de service sur les plans inférieurs pour aboutir à une architecture logique, et notamment la façon de répercuter une composition de service sur les plans inférieurs. Troisièmement, les industriels n'ont pas pris appui sur ces éléments de service pour spécifier leurs services du point de vue des utilisateurs. Comme le souligne N. Simoni dans [Simoni, 2007],

"La norme cite [les éléments de service] presque à titre d'exemple, sans définir les règles de structure, ni celles de composition de service."

Les travaux sur le plan service se sont in fine concentrés principalement sur la problématique des interactions de service, comme exposé par exemple dans [Keck, 1998], une interaction de service désignant un comportement non souhaité se produisant lorsque plusieurs services sont

déclenchés durant la même communication. A partir de la définition de téléseices (comme l'appel de base), et des services supplémentaires compatibles avec les téléseices, mais dont la compatibilité entre eux est à déterminer au cas par cas, des mécanismes formels de description de services ont été élaborés, par exemple en SDL (*Specification and Description Language*) [ITU, 1992b]. Ces descriptions de service se sont fortement orientées vers la détection a priori des incompatibilités entre services, avec des résultats toutefois mitigés, comme nous l'avons par exemple détaillé dans [2006-3] ou [2006-4]. Ces descriptions se sont donc peu appesanties sur le plan service, où l'on ne peut détecter que des cas d'incompatibilité flagrante, mais se sont concentrées sur les plans suivants.

Passons au plan fonctionnel global. Sur ce plan, un service est décrit comme un enchaînement de composants, les SIB. Ce plan est défini de la façon suivante dans la norme Q.1203 [ITU, 1997b] :

"Le plan fonctionnel global modélise la fonctionnalité du réseau d'un point de vue global. (...) Dans ce plan, les services et les éléments du service sont redéfinis en termes de larges fonctions du réseau nécessaires pour leur support. Ces fonctions ne sont spécifiques ni aux services ni aux éléments du service (SF, service feature) et sont appelées modules indépendants du service (SIB, service independent building block)."

Les SIB ne sont pas un raffinement des éléments de service. Ils sont indépendants des services et modélisent, comme indiqué dans la première phrase de la définition ci-dessus, les "larges fonctions" réalisables par le réseau (c'est-à-dire les commutateurs, les plates-formes Réseau Intelligent et les ressources associées).

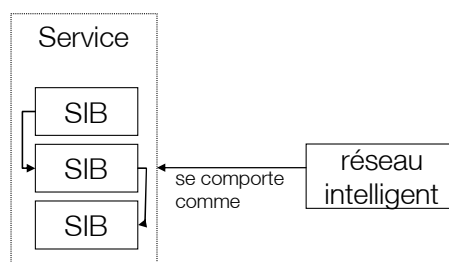


Figure 3 : structure en SIB

Un SIB est donc une fonction du système technique supportant les services, fonction que l'on cherche à isoler comme unitaire et composable avec d'autres fonctions pour répondre à un besoin de service (décrit lui dans la vue service). Treize SIB ont été définis dans la norme CS1 [ITU, 1993c], dont par exemple *charge* pour déterminer la taxation d'un appel, *screen*

pour comparer une valeur à celles d'une liste, ou *queue* pour mettre en file d'attente un appel. Ces treize SIB devaient suffire à décrire tous les services possibles à partir des éléments de service du plan service. Toutefois, les SIB étant définis comme l'abstraction des fonctionnalités du réseau, les spécificités des différents équipementiers ont donné lieu à un nombre excessif de SIB propriétaires en sus des SIB normalisés. Par ailleurs, les services rentables à mettre en œuvre, compte-tenu des coûts liés au RI, étaient vraisemblablement en nombre limité.

L'une des grandes questions laissées ouvertes par le modèle conceptuel du RI est sans doute la suivante : Comment passe-t-on du plan service au plan fonctionnel global, c'est-à-dire comment traduit-on une description du comportement du service tel que perçu par l'utilisateur en une description du comportement du réseau ? Dans le cadre du RI, ce passage, cette traduction étaient laissés à l'expertise du développeur de service. A travers les plans fonctionnel global, fonctionnel réparti et physique, c'est un passage de la spécification du service au code logiciel qui était visé. Ainsi, des travaux industriels et universitaires, décrits par exemple dans [Najm, 1999], ont porté sur des ateliers de création de service permettant de transcrire un service spécifié sous forme d'un enchaînement de SIB en code exécutable sur les plates-formes RI. Une telle perspective a bien sûr conduit à complexifier la vue fonctionnelle globale, les SIB devant non seulement décrire une architecture de service, mais spécifier avec exactitude son comportement pour pouvoir en déduire une implémentation. Idéalement, le travail de développement des services aurait ainsi été mené au niveau du plan fonctionnel global. La valeur ajoutée du développeur de service résidait alors justement dans le passage du plan service au plan fonctionnel global, les autres transitions pouvant être en théorie automatisées. Dans la pratique, cette approche n'a jamais pu être rendue opérationnelle. D'une part, comme nous l'avons dit, l'objectif d'automatisation entraîne une complexification de la description du service, cette description devenant de facto un codage du service et ainsi l'apanage d'experts en Réseau Intelligent ; d'autre part cette automatisation n'a jamais pu être entièrement réalisée, des reprises manuelles du code final étant toujours nécessaires.

Dans cette thèse, nous retiendrons du RI la distinction entre description des services dans le plan service et modélisation du comportement du réseau dans les plans fonctionnel global et fonctionnel réparti. Nous ne viserons pas ici la création de service, par exemple à travers une génération de code à partir d'une description architecturale d'un service, mais nous nous limiterons à la prise en compte de plusieurs points de vue pour décrire les architectures

de services. L'architecture des services constituera pour ainsi dire une référence vis-à-vis des équipes de développement.

I.1.2 De TINA au NGN

L'initiative TINA-C (*Telecommunications Information Networking Architecture Consortium*), décrite par exemple dans [Inoue, 1998], a tenté de dépasser les limites du RI mais n'a pas travaillé sur la question du passage du plan service au plan fonctionnel global. Un concept de session de service générique, indépendant des services rendus car applicable quel que soit le service¹, a au contraire été défini, une session étant une relation temporaire entre des objets pour accomplir une tâche durant un temps donné (ces objets seront aussi bien une abstraction d'éléments perçus par les utilisateurs qu'une abstraction d'éléments techniques). Le premier des *requirements* de TINA-C est ainsi le suivant :

"1: Support of a wide range of services. The TINA service architecture has to support telecommunication, management and information services and should be open to allow the introduction of new classes of services. The service architecture addresses the evolution of services, and should be able to support new requirements and business needs."

Pour remplir cette exigence, plusieurs rôles et plusieurs types de sessions ont été définis, comme illustré dans la figure ci-dessous.

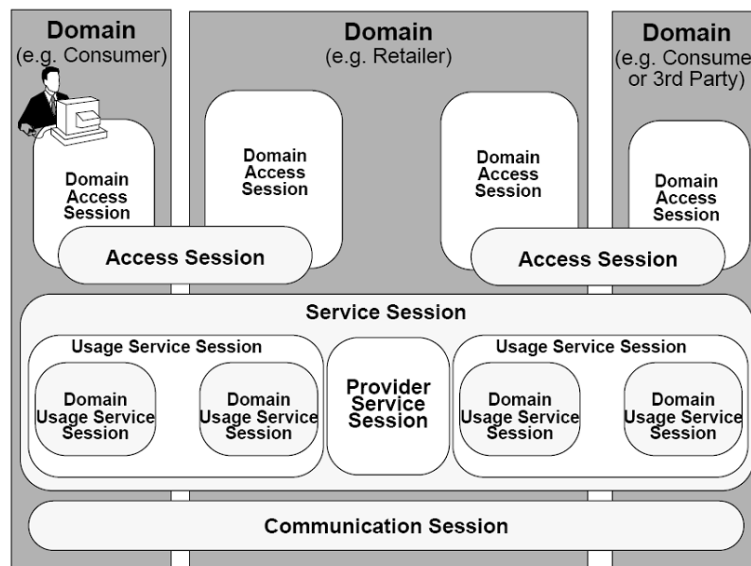


Figure 4 : Architecture de session TINA-C

Ces travaux sont sans doute plus proches de la définition d'une place de marché de services que de la description d'une architecture de service. Il est d'ailleurs étonnant de constater, à la suite de [Magedanz, 2007], que ces concepts n'ont pas débouché sur le déploiement d'applications dans le monde télécom, mais ont par contre trouvé une traduction concrète dans le monde IT avec les architectures SOA. TINA-C a également ouvert la voie à la prise en compte de méthodes du monde IT dans les services télécoms, notamment les méthodes objet et les composants logiciels.

Du côté de l'ITU-T, des travaux ont été menés sur le GII (pour *Global Information Infrastructure*), afin de rapprocher les industries de télécommunication, de l'informatique et du divertissement, travaux présentés par exemple dans [Asatani, 1996] ou [Morre, 1998]. En ce qui concerne les services, la norme Y.110 [ITU, 1998] définit une typologie en couches des services, classés selon les fonctions qui les supportent. On distingue ainsi dans cette norme :

- les fonctions basicielles (*baseware*), incluant notamment les fonctions de transport de l'information et les fonctions de traitement et de stockage ;
- les fonctions intergicielles (*middleware*), se basant sur les fonctions basicielles, et qui incluent notamment les fonctions de création d'application/service.

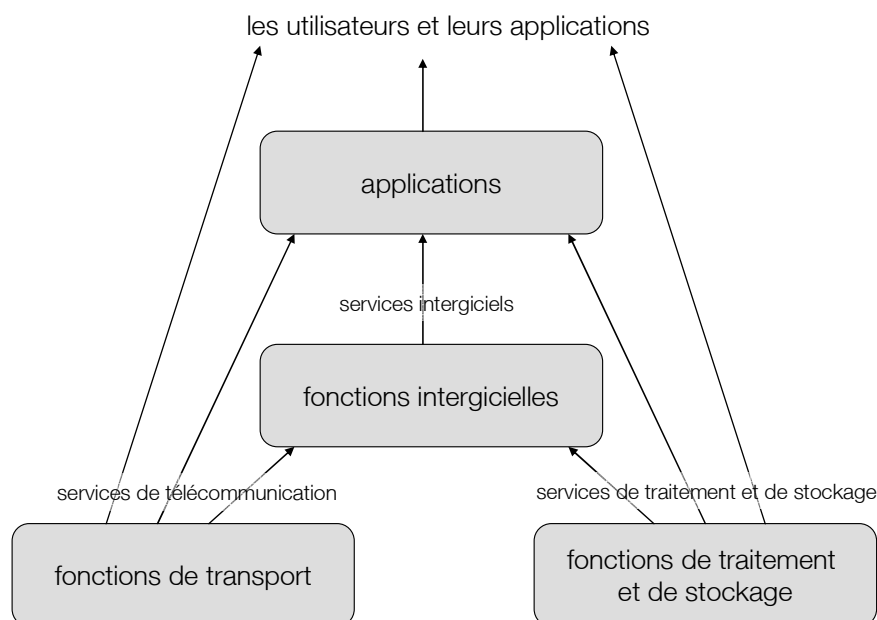


Figure 5 : Typologie des fonctions GII

¹ Ce faisant, elle a peut-être annoncé le NGN, qui sera spécifié sans vision service explicite comme nous le verrons dans le chapitre suivant.

Sur la figure ci-dessus, une flèche d'une fonction A vers une fonction B signifie que la fonction A fournit un service à la fonction B, et donc que la fonction B utilise la fonction A. Les fonctions de transport de l'information sont par exemple celles de RNIS, du RTC ou d'Internet. Les fonctions de traitement et de stockage sont par exemple celles de PC, de STB (*Set-Top Box*), de serveur de fichiers, de serveur vidéo. Les fonctions intergicielles sont par exemple la sécurité, la conversion de formats, l'authentification, la facturation, les ponts de conférence. Ce modèle est toutefois donné sans mode d'emploi sur ses conditions de validité et sur son cycle de vie, alors que les évolutions des services télécoms de ces dernières années conduiraient nécessairement à le modifier. Il présente probablement un manque d'homogénéité entre les fonctions basicielles (abstraction des ressources de transport ou de traitement/stockage de l'information) et les fonctions intergicielles (qui sont des fonctionnalités transverses aux services et aux ressources). Ce découpage pourrait donc être utilisé pour décrire un service, mais ne permet pas vraiment de caractériser une architecture de service.

L'ETSI a également distingué différents types de services. Dans la spécification technique TS 22.101, la dernière version étant [3GPP, 2009], les services de communication sont divisés en services conversationnels, services de messagerie, services de recherche d'information (*retrieval services*) et les services de diffusion (*broadcast*), que ce soit sous contrôle de l'utilisateur (par exemple la vidéo à la demande) ou non (par exemple la télévision). Cette distinction reste toutefois de haut-niveau, et à nouveau sans explications sur son origine et ses conditions de validité.

Plus récemment, dans le cadre du NGN, qui sera décrit au début du chapitre II, plusieurs initiatives ont vu le jour autour des services. Des débats ont eu lieu [Cochennec, 2002] pour déterminer si le NGN devait standardiser des services, ou si les services devaient exclusivement rester du ressort des fournisseurs de service utilisant le NGN. Le modèle économique des opérateurs européens comme fournisseurs de service (et non uniquement comme fournisseur de capacités réseau) et leur volonté de se différencier par les services, associé au souhait de "rentabiliser" le déploiement d'un NGN par des services, a conduit plusieurs organismes à s'intéresser plus avant aux services.

I.1.3 OMA et les enablers

L'OMA (*Open Mobile Alliance*)¹ est un consortium né en 2002 regroupant des industriels télécoms et IT dans le but suivant.

"The mission of the Open Mobile Alliance is to facilitate global user adoption of mobile data services by specifying market driven mobile service enablers that ensure service interoperability across devices, geographies, service providers, operators, and networks while allowing businesses to compete through innovation and differentiation."[OMA, web]

Cet objectif résume bien la position des organismes de standardisation vis-à-vis des services. Pour éviter de standardiser des services et faciliter une possibilité de différenciation autre que tarifaire entre opérateurs, l'OMA, mais aussi l'ETSI et l'ITU-T, ont choisi de standardiser des capacités de service, appelées *service capabilities* au 3GPP, *service support capabilities* à l'ITU-T et *service enablers* à l'OMA. Les capacités de support de service de l'ITU-T incluent typiquement [Carugi, 2005] la présence, la localisation, la gestion de groupes, la gestion de messages, le *broadcast/multicast* ou la gestion des équipements. Les capacités de service de l'ETSI sont par exemple la présence [3GPP, 2007a], la messagerie [3GPP, 2007b] ou la conférence [3GPP, 2007c]. Les enablers de service de l'OMA regroupent [OMA, 2007b] la synchronisation de données, la gestion d'équipements, la gestion de droits électroniques, le téléchargement, la notification d'emails, la messagerie instantanée, la présence ou la localisation mobile. Ces éléments, que nous appellerons tous enablers pour simplifier, sont des modules applicatifs réutilisables, dont le comportement vis-à-vis de leur environnement est entièrement spécifié. Un enabler d'un équipementier peut ainsi théoriquement être utilisé par une plate-forme de service d'un autre équipementier. Cette approche permet une meilleure interopérabilité entre les équipements, les opérateurs et les fournisseurs de services. Elle permet également d'améliorer l'expérience utilisateur (*user experience*). En effet, chaque enabler a une responsabilité clairement définie, par exemple sur des données telles que la présence des utilisateurs ou sur leurs préférences, ce qui permet à l'utilisateur de fournir une seule fois ces données pour tous les services qui les utilisent. Ainsi, si chaque service nécessitant comme information la présence des utilisateurs implémente une fonctionnalité de capture et de remontée de cette présence, l'utilisateur devra remonter sa présence (ou installer des clients logiciels de présence sur ses équipements), autant de fois que de services. Avec un

¹ <http://www.openmobilealliance.org/>

enabler de présence, l'enabler centralise cette information et la distribue à tous les services concernés. Dans les cas plus techniques, comme par exemple la synchronisation de données entre terminal mobile et plate-forme de service, le service ne va pas accéder à une information, mais réutiliser un mécanisme de synchronisation entre serveur et terminal.

Les enablers peuvent être vus comme des briques de base (building blocks) pour construire des services, tel que présenté dans l'article [2004-2]. Alors que les SIBs doivent être considérés comme des opérateurs logiques (par exemple le SIB Compare, le SIB Screen, le SIB Translate), entièrement indépendants des services spécifiés par leur enchaînement¹, les enablers ne sont au contraire pas indépendants du service offert, mais en constituent un élément, réutilisable dans d'autres services. Un enabler peut être rapproché en ce sens d'un élément de service du plan service du RI. La différence avec un élément de service réside dans le fait qu'un enabler prescrit non seulement le service rendu, mais également sa réalisation technique et l'interface applicative pour y accéder, comme le soulignent les définitions de l'OMA ci-dessous.

"(an enabler is) A technology intended for use in the development, deployment or operation of a Service; defined in a specification, or group of specifications, published as a package by OMA." [OMA 2007a]

"An enabler should specify one or more public interfaces." [OMA 2007b]

Un enabler peut être vu comme une déclinaison dans le monde télécom du principe de composant logiciel réutilisable de SOA (*Service Oriented Architecture*), que nous introduirons dans la section suivante. Les travaux sur les enablers sont dans la continuité du mouvement de spécification d'API pour les réseaux de télécommunications, notamment à l'initiative du groupe Parlay [Parlay, web], fondé en 1998. Parlay avait initialement comme objectif de spécifier des interfaces applicatives au-dessus des équipements télécoms de contrôle d'appel afin de permettre la réalisation par des fournisseurs de services tiers de services simples, par exemple un service de lancement d'appel (*clic-to-dial*). Ce groupe s'est ensuite rapproché de l'initiative OSA (*Open Service Access*) de l'ETSI. Des spécifications communes ont alors été publiées, notamment pour des APIs de contrôle d'appel, de messagerie ou de présence [Moerdijk, 2003].

¹ Cet aspect est d'ailleurs souligné dans leur désignation comme indépendant des services (*service independent*)

Un enabler diffère néanmoins d'une API comme d'un service SOA, car il n'est pas seulement spécifié par son interface, mais par sa place dans un environnement, constitué d'autres enablers, de ressources réseau, de services, le tout relié par des points de référence. A la différence des architectures SOA, il n'y a pas de protocole unique d'accès à un enabler comme peut l'être http, mais un enabler va s'intégrer dans un environnement protocolaire complexe. De plus, alors que la sémantique des opérations réalisables par un service SOA est propre à ce service, cette sémantique va être ici induite par les protocoles. Le comportement d'un enabler est également spécifié plus complètement que celui d'une API. Enfin, un enabler repose sur une technologie précise (par exemple syncML pour la synchronisation de données), à la différence d'une API qui se veut en général indépendante de la technologie sous-jacente, mais si cette dernière transparaît souvent à travers les paramètres d'entrée de l'API.

Par ailleurs, comme indiqué dans la définition de l'OMA ci-dessus, un enabler présente un caractère normatif (*defined in a specification*). Au final, un composant ou une technologie sont constitués en enabler de part la standardisation. L'ITU-T, l'ETSI, ou l'OMA ne proposent donc pas de critères pour déterminer le contour d'un enabler, mais une méthode : la standardisation. Et une standardisation concertée entre enablers. En effet, s'ils étaient spécifiés indépendamment les uns des autres, les enablers risqueraient de se recouvrir partiellement, c'est-à-dire d'offrir des fonctions redondantes. Les fournisseurs de services ne pourraient alors que difficilement combiner différents enablers pour réaliser un service. Comme indiqué par l'OMA, nous serions dans la situation suivante :

"Integration and deployment of services is complicated and expensive; high implementation efforts for applications wanting to use several capabilities; there is no common integration of the different services from the point of view of the end-user (e.g. no common group management or user profile across multiple services)." [OMA 2007b]

Pour éviter cela, l'OMA a recours à la notion de fonction intrinsèque, *intrinsic function*. Une fonction intrinsèque est définie comme

"those functions that are essential in fulfilling the intended task of the specified enabler. For example, the Position Calculation function is Intrinsic to Secure User Plane Location; Authentication is intrinsic to Single Sign On; Encryption is an intrinsic function of Digital Rights Management." [OMA 2007b]

L'OMA spécifie alors qu'un enabler ne doit contenir que des fonctions intrinsèques.

"any requirements or features that are not intrinsic to an enabler should not be specified within the enabler's specification."

Cette exigence permet d'assurer que différents enablers ne vont pas offrir la même fonction. Ceci est garanti par exemple qu'une fonction d'authentification ne sera pas présente dans chaque enabler, alors qu'elle est requise par la plupart d'entre eux. Cette notion de fonction intrinsèque n'épuise pourtant pas la question du contour fonctionnel des enablers. Tout d'abord, une fonction intrinsèque peut soit correspondre à une fonctionnalité de service, perçue par l'utilisateur – comme par exemple une fonction de gestion des groupes de contacts (*group management*) ou de profil utilisateur (*user profile*) –, soit correspondre à l'abstraction d'une technologie – comme une fonction de *push* ou de *watermarking*. Ensuite, la séparation entre fonctions intrinsèques et extrinsèques n'est pas évidente. Elle reste in fine subjective, comme reconnu par l'OMA.

"The classification of intrinsic and non-intrinsic is subjective and needs to be done on a per enabler basis"

Les enablers de l'OMA, les capacités de service de l'ETSI ou les capacités de support de service de l'ITU-T sont des outils importants pour la convergence de service que nous présenterons au chapitre suivant. Ils permettent en effet à différents services télécoms de partager et de réutiliser des données et des fonctions. Mais leur mode de construction est très lié à la standardisation. C'est par la vertu des débats contradictoires en standardisation que leur contour fonctionnel (leurs fonctions intrinsèques) est déterminé et légitimé.

Comment étendre à l'intérieur des fournisseurs de service ou des opérateurs ce mode de construction de services convergents ? Un opérateur peut-il structurer ses services en construisant ses propres enablers ? Comment ? Une solution est de mettre en place des instances de "standardisation" interne d'enablers, jouant le même rôle que des organisations comme l'OMA dans la construction d'enablers non redondants et dans leur légitimation. Mais de part la nature hiérarchique des entreprises, le débat y sera sans doute moins contradictoire que dans une instance de standardisation rôtée à la prise en compte de points de vue et d'intérêts divergents. Une autre solution, qui peut venir en appui de la première, est de mettre en place des règles et des procédures – conformément aux modes de fonctionnement des grandes entreprises –, pour d'une part identifier les fonctions des services susceptibles de faire l'objet d'un enabler et pour d'autre part aider les équipes de développement de services à identifier les enablers pertinents pour leurs projets et leur permettre de valider a priori la

cohérence du service ainsi construit. Nous allons dans les chapitres III et IV présenter une démarche qui pourra servir de fondement pour de telles règles et procédures.

Pour conclure ce parcours sur la notion de service dans le monde télécom, nous pouvons insister sur deux points. Tout d'abord, le plan service du RI et la notion d'élément de service afférente. Ces éléments de service sont un premier pas vers des composants de service réutilisables pour réaliser différents services. Ensuite, la notion d'enabler, qui a poussé plus loin cette notion de composant de service en couplant le service rendu à une implémentation effectivement réutilisable. Ce couplage entre service rendu et implémentation uniformise toutefois les architectures de service en imposant une seule solution technique pour répondre à un besoin de service. Ceci conduit de fait à une standardisation des enablers, ce qui rigidifie leur cycle de vie (un enabler mettra du temps à être spécifié puis à être mis en œuvre). Et en conséquence le cycle de vie des services, si l'on souhaite que tout nouveau service repose au maximum sur des enablers existants.

Nous allons maintenant examiner la description des architectures de service dans la communauté web.

I.2 Dans la communauté Web

I.2.1 Les services web comme unité architecturale de base

Du point de vue architectural, le web peut être vu comme un ensemble de ressources accessibles par des liens hypertextes. Ces ressources sont les unités architecturales de base du web. Ce terme de ressource est à comprendre de la façon suivante, d'après le RFC 3986 "Uniform Resource Identifier (URI) : Generic Syntax".

"The term "resource" is used in a general sense for whatever might be identified by a URI. Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., "today's weather report for Los Angeles"), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity)."

Dans la communauté web, les services sont ainsi vus comme une ressource parmi d'autres, au même titre qu'un document électronique ou une image, ce service étant, comme toute

ressource, accessible par un URI. Une fois les mécanismes de base du web mis en place, la communauté web s'est plus spécifiquement intéressée aux services au début des années 2000 en spécifiant des mécanismes dédiés aux services. Le consortium W3C (pour *World Wide Web Consortium*) définit dans son glossaire un service web (en anglais *web service*) de la façon suivante.

"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL)." [W3C, web]

Plus précisément, un service web est une application ou un composant logiciel qui vérifie les propriétés suivantes :

- Il est identifié par un URI ;
- Ses interfaces et ses liens (binding) peuvent être définis en XML, notamment dans le langage WSDL ;
- Ces définition (interfaces et liens d'un service) peuvent être découvertes par d'autres Web services, par exemple via un annuaire UDDI ;
- Il peut interagir directement avec d'autres Web services à travers le langage XML et en utilisant des protocoles Internet.

Une architecture technique de référence pour les services web a rapidement émergé [Channabasavaiah, 2003]. Cette architecture est composée du consommateur de web services, du fournisseur de web services et de l'annuaire de web services. Le fournisseur de web services publie les services qu'il expose dans un annuaire de web services. Le consommateur de web services consulte cet annuaire pour trouver le service adéquat, puis invoque le fournisseur correspondant.

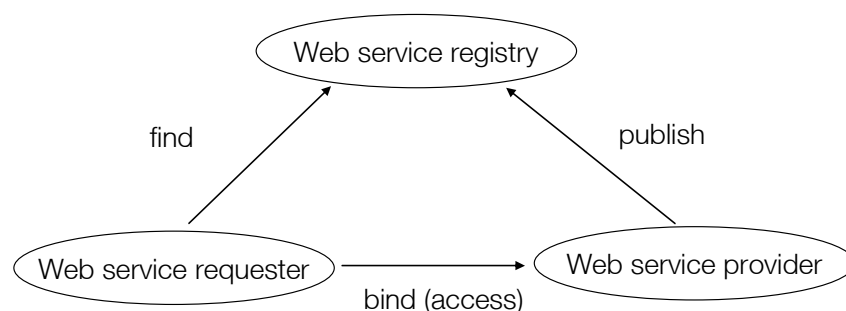


Figure 6 : Architecture de référence des services web

Comme présenté par exemple dans [Karunamurthy, 2007], ces notions peuvent être rapprochées de celles de *broker* et *retailer* de TINA-C ou du framework de Parlay [Moerdijk, 2003].

Les web services sont décrits par leur interface, habituellement suivant le langage WSDL (pour *Web Services Description Language*) qui spécifie la signature du service, c'est-à-dire ses paramètres d'entrée attendus et ses paramètres de sortie. Dans la communauté web, une architecture de service consiste donc en un ensemble de services web interagissant à travers le motif présenté ci-dessus (figure 6). Une réflexion plus large sur la description des liens entre services a été menée, notamment dans le cadre de la communauté du web sémantique.

I.2.2 La description sémantique de ressources

L'expression Web sémantique, attribuée à Tim Berners-Lee [Berners-Lee, 2001], désigne plus une vision que des outils précis. Cette vision peut être résumée par l'objectif suivant : permettre aux machines de raisonner sur les sources d'information du Web pour en déduire des informations de plus-haut niveau afin de répondre à des requêtes. Par exemple, construire automatiquement une liste des peintres néo-classiques français, à partir de l'information brute disponible sur le Web. Ou encore une liste des œuvres d'un écrivain. Des méthodes basées uniquement sur la recherche de mots-clés et sur la proximité textuelle de ces mots-clés ne sont pas suffisantes. Elles ne permettront pas par exemple de distinguer nettement les livres d'Albert Camus et les livres sur Albert Camus. Ou de distinguer entre le rattachement d'un peintre à un courant et son rejet de ce courant, une proximité textuelle pouvant être trouvée dans les deux cas. De telles manipulations de données par des machines nécessitent une structuration de l'information accessible par le web dans un formalisme permettant des traitements automatisés sur des sources multiples et diverses. Pour cela, la communauté Web sémantique utilise la notion de métadonnée, une métadonnée étant une donnée servant à décrire une autre donnée [Berners-Lee, 2001]. Toutes les ressources informationnelles peuvent être décrites par des métadonnées. A la différence des données de base, ces métadonnées suivront une sémantique compréhensible par des machines. Ces métadonnées ne sont pas de simples mots-clés, mais sont structurées. Ainsi, un peintre et son rattachement à un courant artistique pourront être indiqués dans les métadonnées décrivant un tableau numérisé. Pour être efficaces, ces métadonnées devront bien sûr utiliser les mêmes mots-

clés, par exemple le terme "néoclassique". Publié en 1999, le langage RDF pour *Ressource Description Framework* permet de décrire des ressources à l'aide de métadonnées, ainsi que les liens entre ces métadonnées¹. Des logiciels d'intelligence artificielle appelés moteurs d'inférence peuvent alors conduire des raisonnements logiques à partir de ces métadonnées.

Un accord sur une liste restreinte et cohérente de métadonnées à utiliser pour décrire les ressources est toutefois illusoire dans un environnement ouvert de la taille du web. La communauté Web sémantique a donc également travaillé sur des modèles de représentation de la connaissance. Ces travaux ont notamment produit des langages de définition d'ontologies, comme OWL pour *Ontology Web Language* ; une ontologie étant entendue, selon la célèbre définition de T.R. Gruber dans [Gruber, 1993], comme

"la spécification d'une conceptualisation d'un domaine de connaissance"

Une ontologie définira ainsi un ensemble de métadonnées pour décrire un domaine, et les liens entre ces métadonnées à travers des prédicats. Une ontologie pourra ainsi prendre en compte les relations entre métadonnées. Elle a usuellement vocation à être construite en commun et partagée par plusieurs acteurs œuvrant dans un même domaine de connaissance. Qui souhaite décrire une ressource peut alors rattacher les métadonnées qu'il utilise à cette fin à une ontologie existante, en indiquant les liens entre ses métadonnées et les concepts de l'ontologie à laquelle elles se réfèrent. Une ontologie peut être construite, mais aussi générée automatiquement à partir d'un ensemble représentatifs de documents structurés, avec toutefois des résultats mitigés, comme développé par exemple dans [Bedini, 2008].

I.2.3 La description sémantique de services web

La communauté Web sémantique s'est attachée à appliquer ces principes aux services web. Elle s'est ainsi intéressée à la description des services en lançant des initiatives pour ajouter une description sémantique aux services web, comme indiqué par exemple dans [Zeng, 2001], [Zhixiong, 2007] ou [Arroyo, 2006]. Les services web sont alors considérés comme des ressources et décrits par des métadonnées, ces métadonnées se référant à une ontologie de services. Deux directions complémentaires ont été investiguées. D'une part, des spécifications ont été proposées pour ajouter des annotations sémantiques aux fichiers WSDL, comme par

¹ Une métadonnée ou un lien pouvant eux-mêmes être considérés comme une ressource et décrits à l'aide d'autres métadonnées.

exemple SAWSDL pour *Semantic Annotations for WSDL* [Mentzas, 2007]. D'autre part, d'autres spécifications ont visé, de façon plus ambitieuse mais moins incrémentale, à décrire un service dans ses différentes dimensions, et non uniquement ses paramètres d'entrée et de sortie. Deux services web présentant la même signature peuvent en effet réaliser des services tout à fait différents. Un service ayant pour paramètre d'entrée deux numéros de téléphone peut par exemple être aussi bien un service de *clic-to-dial* qu'un service de renvoi du premier numéro vers le second. Des spécifications comme OWL-S (*OWL for Services*) ont pour objectif de supporter la construction d'ontologies de description de services web. Un service web est décomposé en trois dimensions. Le profil de service, le modèle de processus et l'ancrage (*grounding*). Le profil de service permet de décrire ce que le service réalise ; le modèle de processus permet de décrire comment le service est utilisé et l'ancrage permet de décrire comment interagir applicativement avec le service. Le profil et le modèle de processus sont des caractérisations abstraites du service, alors que l'ancrage fait le lien entre ces descriptions abstraites et les messages concrets échangés par le producteur et le consommateur du service web, via un protocole précis.

D'autres travaux se sont concentrés sur la construction de places de marché (*marketplace*) de services web. Ces travaux visent à mettre en place un environnement de découverte et d'accès à des services, ces services étant décrits sémantiquement selon des ontologies cohérentes. Ceci permettrait alors de découvrir des services à partir d'un besoin exprimé par l'utilisateur, grâce à ces mécanismes d'inférence. De plus, les annotations sémantiques faciliteraient la création automatique d'un service composé pour répondre à un besoin. Des travaux comme [Shiaa, 2008] ou [Lécué 2009] proposent ainsi un algorithme de correspondance entre les paramètres d'entrée et de sortie de services web, ces paramètres étant décrits sémantiquement afin de détecter les relations entre eux (équivalence, appartenance...). Un enchaînement de web services peut alors être conçu automatiquement en réponse à un besoin exprimé sous forme de paramètres d'entrée et de sortie. La communauté web sémantique propose ainsi d'automatiser les tâches de découverte et de composition de services en modélisant formellement la connaissance des humains susceptibles de réaliser ces tâches. Dans une vision cible, les utilisateurs pourraient formuler (par exemple en langage naturel) leur besoin afin qu'un service soit conçu automatiquement pour y répondre, la place de marché se basant pour cela sur les services enregistrés chez elle. Cette dernière agrègerait des liens vers une multitude de service sans cohérence a priori, la sélection des services et leur fonctionnement cohérent étant assurés par des technologies sémantiques (tels que les

annotations sémantiques, les ontologies, les moteurs d'inférence). Les concepts décrivant chaque service et décrivant les utilisateurs (préférences, profil, contexte) seraient définis sous forme d'ontologies. Des raisonnements sur ces ontologies assureraient la cohérence entre ces concepts. Ces approches visent ainsi à résoudre les difficultés que nous évoquerons dans la section suivante pour les services SOA, comme l'identification des services, la détermination de leur granularité et la gestion de leur cycle de vie.

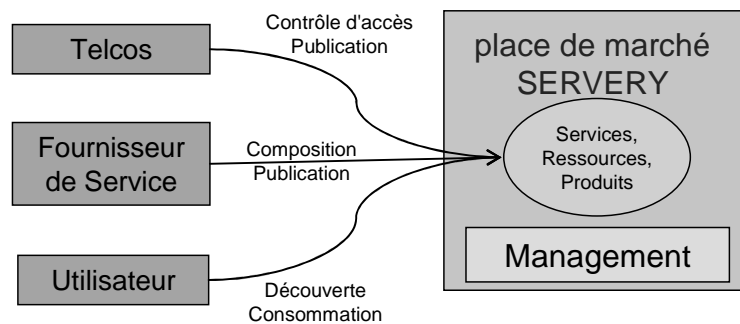


Figure 7 : Principe d'une place de marché dans le projet CELTIC Servery

Concernant les architectures télécoms, des travaux ont alors porté sur la construction d'ontologies pour les services de communication, comme par exemple dans le projet IST SPICE [SPICE, web]. Des places de marché de service sont également à l'étude pour les services télécoms, comme dans le projet CELTIC Servery [Servery, web]. Toutefois, ces approches restent prospectives, car elles se heurtent à plusieurs points bloquants.

Tout d'abord, un problème de masse critique et de maturité industrielle, comme nous l'avons indiqué dans [2007-6]. Les outils sémantiques sont encore en phase de développement académique, les outils de construction d'ontologie ou d'inférence restent peu utilisés en dehors de projets de recherche, leur pertinence, leur fiabilité et leur facilité d'utilisation n'étant pas pour l'instant au rendez-vous. Un problème de passage à l'échelle se pose également, notamment au niveau des performances et de la gestion de la complexité.

Ensuite et plus fondamentalement, un problème de cycle de vie des ontologies. Dans le paradigme sémantique, la pertinence de la réponse à un besoin donné repose sur la qualité de l'ontologie qui va permettre de relier les métadonnées décrivant les services et les besoins des utilisateurs. Comment construire ces ontologies et les faire évoluer ? Cela peut être fait de façon centralisée, une équipe assurant la construction cohérente puis la maintenance d'une ontologie. Mais cette équipe se reposera alors tous les problèmes d'exhaustivité, de cycle de vie, d'identification des services et de granularité que ces mécanismes étaient censés résoudre.

L'outil ontologique ne présente alors qu'un faible avantage par rapport à des procédés plus frustrés comme la spécification d'une liste de métadonnées. Cela peut également être fait de façon automatisée, en détectant les mots-clés les plus utilisés dans les interfaces des services, et en les reliant entre eux. Mais comme indiqué dans [Bedini, 2008], de tels mécanismes automatiques ne sont pertinents que s'ils se basent sur une classification cohérente, et induisent sinon des ontologies incohérentes. L'annotation sémantique des services pose également problème. Si cette annotation est faite de façon centralisée, par une équipe en charge de l'annotation sémantique des nouveaux services, le bénéfice est alors minime par rapport à un fonctionnement plus traditionnel, par exemple celui préconisé pour la gestion du cycle de vie des services SOA. Et dans le cadre d'une place de marché fournissant une multitude de service, cette équipe serait rapidement débordée. Mais obliger les concepteurs de service à annoter eux-mêmes leurs services en se référant à une ontologie existante serait extrêmement contraignant. Ils devraient en effet connaître non seulement les langages et les outils sémantiques, mais également les ontologies existantes et être capables de faire le lien entre les concepts qu'ils manipulent et ceux de l'ontologie, cela équivalant quasiment à un travail de spécification de leur service sous forme sémantique. On observe en effet fréquemment, comme indiqué par exemple dans [Hepp, 2007] ou [Roche, 2007], un décalage entre le vocabulaire technique des développeurs de services et le vocabulaire des utilisateurs de services. Une correspondance entre les deux n'est pas immédiatement accessible, car il ne s'agit pas d'une traduction, de synonymes¹, mais de deux points de vue différents sur le service, sans relation univoque entre les concepts manipulés par chacun.

De plus, on peut relever une certaine contradiction entre l'approche des architectures orientées service et l'approche sémantique. En effet, le web sémantique, comme indiqué par exemple dans [Patel-Schneider, 2007], suit un paradigme de boîte blanche, ou de monde ouvert (*open world*), c'est-à-dire que l'univers du discours tend à englober l'ensemble des services fournis, y compris dans le détail de leurs propriétés [Rolland, 2000]. Alors que, les plates-formes de service SOA, comme nous le verrons dans la section suivante, ont une approche de type boîte noire ou de monde clos (*closed world*), exposant le minimum vers l'extérieur de leur fonctionnement interne afin de garantir un couplage lâche (*loose coupling*) entre les différents services.

¹ Ce qu'une ontologie peut tout à fait gérer.

Enfin, le paradigme sémantique implique que les données soient compréhensibles par des machines, mais sans doute au détriment des utilisateurs humains. A l'instar de certaines descriptions formelles SDL dans le RI, une ontologie n'est pas immédiatement lisible ; les données sémantiques sont d'abord conçues pour être interprétées par des machines. Cela rend plus délicate la généralisation de ces méthodes pour la conception des architectures de service.

Que retenir de ce parcours ? A la différence de la communauté télécom, qui cherchait à identifier des composants de services indépendants les uns des autres pour ensuite les mettre en relation afin de réaliser un service, la communauté du web et du web sémantique met l'accent sur l'importance des liens, vecteur essentiel de la connaissance. Les services, et même les éléments de service ou les services dits unitaires ne sont pas des briques totalement indépendantes. Il y a des liens, des prérequis, des dépendances entre les services et avec les infrastructures, que les méthodes sémantiques cherchent précisément à capturer dans des ontologies. Dans cette thèse, nous ne nous placerons pas explicitement dans le paradigme sémantique, mais nous tenterons de répondre aux questions ci-dessus, laissées en suspens par ce paradigme, à travers la conception d'une classification des services télécoms et des liens entre les différentes vues d'un service.

I.3 Dans la communauté IT

Le terme IT (pour *Information Technology*) désigne l'ensemble de l'informatique d'entreprise. L'association américaine des technologies de l'information [ITAA, web] définit ainsi le terme IT comme

"the study, design, development, implementation, support or management of computer-based information systems, particularly software applications and computer hardware."

I.3.1 Les architectures orientées service

Dans le monde informatique, les travaux sur les services sont pilotés par les besoins des systèmes d'information (SI). Un SI peut être défini de la façon suivante [O'Brien, 1997].

"Un système d'information est un ensemble de personnes, de procédures et de ressources qui recueillent de l'information, la transforment et la distribuent au sein d'une organisation."

Les systèmes d'information sont un levier essentiel de la stratégie d'une entreprise ; ils ont la charge de mettre en œuvre les processus métiers des entreprises en servant de support à la gestion de l'information. La réalisation informatisée d'un système d'information est nommée système informatique. L'alignement du SI avec le métier des entreprises est un enjeu majeur, comme détaillé par exemple dans [Henderson, 1993] ou plus récemment dans [Simonin, 2007].

Or, le métier des entreprises de services évolue vers un décloisonnement des entités organisationnelles pour répondre aux besoins des clients. Cette évolution est particulièrement notable dans le secteur économique des services, comme nous le développerons dans le chapitre II. Les SI doivent évoluer en conséquence en passant d'une organisation "verticale" très cloisonnée, où chaque entité organisationnelle gère son SI de façon autonome car ces SI n'ont pas besoin de communiquer ni d'échanger (les processus métier qu'ils supportent étant propres à une seule entité organisationnelle), à une organisation plus "horizontale", où les SI des différentes entités organisationnelles échangent continuellement pour supporter des processus transverses entre les différentes fonctions de l'entreprise, afin de répondre aux besoins clients. Considérons par exemple le domaine de la bancassurance. Si, il y a 15 ans, un réseau bancaire pouvait obliger ses clients à changer de numéro de compte lorsqu'ils changeaient d'agence de rattachement, aujourd'hui, un même client peut non seulement le conserver, mais aussi accéder avec ce même numéro de compte à des produits financiers ou assurantiels variés via de multiples canaux de distribution (agence, centre d'appel, web). Ce décloisonnement est aussi la règle dans le secteur du e-commerce, comme illustré par exemple dans [Morley 2003].

Le monde IT a forgé le concept de SOA (pour *Service Oriented Architecture*) afin de faciliter ce décloisonnement. Dans le paradigme SOA, les applications du SI sont décomposées en services logiciels, c'est-à-dire des unités applicatives indépendantes réalisant une fonction métier déterminée et limitée, accessible par l'intermédiaire d'une interface ouverte et définie ; ce concept de service logiciel a été essentiellement implémenté à l'aide des services web, vus précédemment. Une architecture SOA a été définie par ses promoteurs initiaux d'IBM dans [Channabasavaiah, 2003] comme

"an application architecture within which all functions are defined as independent services with well defined invocable interfaces which can be called in defined sequences to form business processes"

Dans les méthodes d'analyse des processus métier, comme [Simonin, 2004], à la suite de [Booch, 1999], un processus métier est entendu comme

"une séquence d'actes de gestion, réalisée par l'entreprise, qui produit un résultat dont la valeur est perceptible et mesurable pour un acteur individuel du domaine modélisé."

Dans le paradigme SOA, cette séquence d'actes de gestion peut être réalisée par un enchaînement d'appels à des services logiciels, du moins lorsque ces actes de gestion sont informatisés. Les processus métier peuvent également être spécifiés dans un langage approprié, comme BPML pour *Business Process Modeling Language*, et une traduction peut être faite de cette description métier vers un script des services logiciels à enchaîner, exprimé par exemple en BPEL pour *Business Process Execution Language*.

Outre cette notion de services logiciels, l'approche SOA précise aussi des principes d'architecture. Les services logiciels doivent être :

- fortement cohérents, c'est-à-dire que l'ensemble des fonctionnalités offertes par un même service logiciel doit être cohérent ;
- faiblement couplés (*loose coupling*), c'est-à-dire qu'un service logiciel ne doit manipuler des données placées hors de sa responsabilité que via d'autres services logiciels et non directement.

Différents services logiciels peuvent alors être composés, c'est-à-dire invoqués les uns à la suite des autres pour réaliser un service plus complexe. Par exemple, un service logiciel de traduction automatique suivi d'un service logiciel d'envoi de SMS permettront d'envoyer un texte traduit par SMS. Un enchaînement d'un service A puis d'un service B sera possible lorsqu'un paramètre de sortie du service A est du même type qu'un paramètre d'entrée du service B. Des langages comme BPEL permettent de spécifier de tels enchaînements, en incluant des conditions logiques. Ces scripts ou services logiciels composés sont un moyen d'automatiser des processus métier.

Mais en tentant de réaliser les processus métier des entreprises à l'aide de services logiciels, la communauté IT a constaté que le principal enjeu n'était pas technique. Techniquement, les protocoles d'invocation ou les langages de spécification d'enchaînement de services logiciels sont opérationnels. Mais le principal enjeu réside dans l'identification des services logiciels à construire. Comment délimiter le périmètre de ces unités applicatives

indépendantes ? Un service logiciel doit-il être de bas niveau, avec un service par fonction identifiable ou de plus haut niveau, avec un ensemble de fonctions cohérentes réalisant un objectif métier ? Des services de bas niveau, c'est prendre le risque du foisonnement ; impossible de gérer le cycle de vie de tous ces services. Et difficile pour un concepteur de déterminer les services à utiliser. Des services de haut niveau, c'est prendre le risque d'un décalage entre les objectifs métiers ayant présidé à l'instauration d'un service et les objectifs métiers du moment. Ces objectifs varient en effet en fonction de la stratégie de l'entreprise et de l'environnement externe (régulation, législation, concurrence, marché...)¹. Et comment maximiser l'indépendance entre services lorsque les services sont de haut niveau ? Ces problématiques sont finalement assez proches de celles que nous avons rencontrées ci-dessus pour les enablers.

La transposition du paradigme SOA aux services de communication est efficace dans une certaine mesure. Elle permet en effet d'offrir un accès réutilisable à quelques fonctions des plates-formes et réseaux de télécommunication ; ces fonctions étant souvent celles déjà identifiées par les travaux du groupe Parlay, comme par exemple l'initiation d'appel (*click-to-call*), l'envoi de SMS ou la géo-localisation. Cette transposition rencontre toutefois un écueil majeur. Les services logiciels du monde IT sont conçus initialement pour accéder à des fonctions du système d'information, c'est-à-dire classiquement des fonctions de traitement de données. Invoquer un service signifie soit demander des données, soit demander un traitement sur des données. La valeur d'un service de communication quant-à elle ne réside pas tant dans sa capacité à traiter des données, mais dans sa capacité à mettre en relation. Le cœur d'un service de communication n'est ainsi pas réductible à une série de requêtes/réponses entre machines, car différentes parties-prenantes vont être impliquées dans le service, comme l'appelant et l'appelé dans un service de téléphonie. Les protocoles, comme SIP ou SMTP, qui structurent les échanges entre parties sont essentiels pour les services de communication, mais ne sont pas pris en compte dans le paradigme SOA, où les protocoles ne sont vus que comme un support pour les invocations de service logiciels. En outre, le paradigme SOA ne permet pas de prendre en compte des aspects essentiels des services de communication, comme la qualité de l'interface homme-machine, ou l'effet réseau que permet l'utilisation d'un même protocole, portant une même sémantique.

¹ Un pan de la communauté IT a ainsi étudié l'alignement entre la stratégie d'une entreprise, ses processus métier et son système d'information, comme par exemple [Henderson, 1993], [Bleistein, 2006] ou [Simonin,

Dans cette thèse, nous tenterons de proposer un mode de description des services de communication qui réponde aux préoccupations évoquées. Identification du périmètre des services sous un angle de vue non logiciel et en lien avec le métier de fournisseur de services. Puis construction d'une architecture applicative qui puisse prendre en compte aussi bien les services logiciels que les protocoles usuels du monde télécom. Nous allons maintenant voir comment la communauté IT a formalisé cette notion d'angle de vue pour décrire l'architecture d'un système.

I.3.2 La notion de vue

Face à la complexité des systèmes informatiques, la communauté du génie logiciel a petit à petit élaboré des notions permettant de maîtriser cette complexité. Dans les années 90, sous l'impulsion notamment de l'OMG, elle a adopté la notion de vue. Le standard IEEE 1471 [IEEE, 2000] définit par exemple de la façon suivante les notions de vue et de point de vue :

"A view is a representation of a whole system from the perspective of a related set of concerns."

Un point de vue est la spécification du mode de construction d'une vue :

"A viewpoint is a specification of the conventions for constructing and using a view, a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis."

Le standard introduit également les attributs d'une vue :

"Each viewpoint shall be specified by a viewpoint name, the stakeholders to be addressed by the viewpoint, the concerns to be addressed by the viewpoint, the language, modeling techniques, or analytical methods to be used in constructing a view based upon the viewpoint."

Chaque vue d'un système doit ainsi répondre à des préoccupations de parties-prenantes (*stakeholders*) du système. Pour un système d'information, ces parties-prenantes seront par exemple l'utilisateur final du système, l'architecte, le développeur ou l'exploitant.

Cette notion de point de vue a été employée dans de nombreux cadres (*frameworks*). Un des plus connus est sans doute le cadre 4+1 proposé par Kruchten en 1995 [Kruchten,

1995]. Comme son nom l'indique, il propose 4 points de vue sur le système, réunis par des scénarios qui illustrent les fonctionnalités essentielles du système.

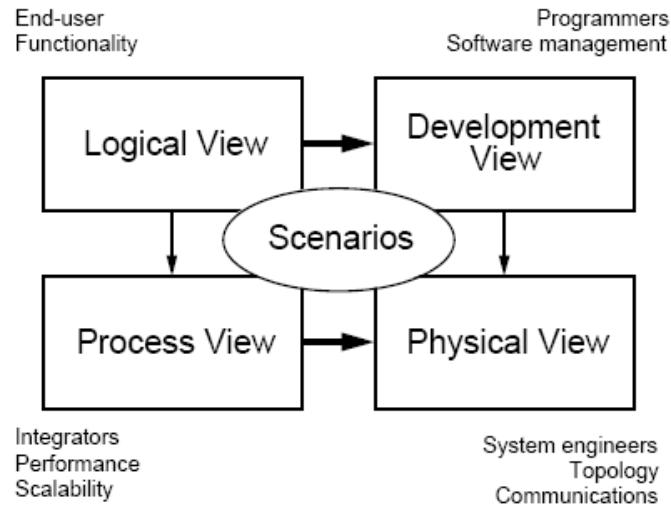


Figure 8 : Points de vue du cadre 4+1

Les attributs de ces vues (tels que présentés ci-dessus), sont décrits dans le tableau suivant.

Tableau 1 : Points de vue du cadre 4+1

Point de vue	Logique	Processus	Développement	Physique	Scénarios
Objectif	Indiquer le service que le système doit fournir	Indiquer les aspects concurrence et synchronisation de la conception	Décrire l'organisation statique du logiciel	Décrire l'implémentation du logiciel sur le matériel	Découvrir les éléments clés de la conception et de la validation
Parties-prenantes	Architecte, utilisateur	Architecte, concepteur, intégrateur	Architecte, développeur, manager	Architecte, concepteur	Architecte, utilisateur, développeur
Préoccupations	Fonctionnalités	Performance, disponibilité...	Organisation, réutilisation, portabilité....	Scalabilité, performance	Compréhension
Concepts de modélisation	Classes, associations, héritage	Evénements, messages	Sous-systèmes, composants	Machine, bande-passante	Classes, événements, étapes

Dans la suite de notre travail, cette notion de vue ou de point de vue jouera un rôle essentiel. Elle peut tout à fait être rapprochée de la notion de plan du Réseau Intelligent, mais elle la clarifie. Alors que les plans du RI pouvaient être vus comme une structure en couches,

le concepteur d'un plan ne prenant en compte que le plan au dessus du sien, la notion de vue est plus souple et précise les critères de constitution d'une vue, comme outil pour répondre aux préoccupations de parties prenantes du système. Nous allons maintenant étudier comment cette notion de vue a été appliquée aux systèmes d'information.

I.3.3 L'architecture d'entreprise et l'urbanisation

Les méthodes d'architecture d'entreprise ou d'urbanisation ont émergé avant celles des architectures orientées service, comme une déclinaison de la notion de vue dans le cadre des systèmes d'information d'entreprise. Mais ces méthodes restent tout à fait pertinentes pour mettre en place des architectures SOA, car elles visent à prendre en charge la difficulté majeure évoquée ci-dessus, c'est-à-dire la mise en place de services logiciels en cohérence avec la stratégie de l'entreprise.

Les méthodes d'architecture d'entreprise, ou EA pour *Enterprise Architecture*, ont pour objectif de faciliter le pilotage de l'évolution des systèmes d'information d'entreprise. Elles proposent de représenter le SI à travers diverses vues, afin de le rendre compréhensible à des décideurs tout en assurant une certaine cohérence entre la vue du SI qu'auront ces décideurs et celle qu'auront les équipes de développement puis les équipes d'hébergement. Elles s'appuient généralement sur quatre démarches clés, comme indiqué dans [Simonin, 2009] et [Urba-EA, 2006] :

- la cartographie de l'existant, c'est-à-dire le recensement et la description des ressources (humaines, logicielles) de l'entreprise et leur positionnement dans le système d'information, ainsi que leur responsabilité sur les informations manipulées par le SI ;
- la définition d'une cible idéale, pleinement conforme à la stratégie de l'entreprise ;
- la migration vers la cible, c'est-à-dire la définition de priorités à atteindre dans un temps déterminé pour rapprocher le SI existant de la cible ;
- la déclinaison de ces priorités dans des projets de développement. L'architecture mise en place dans ces projets doit alors s'inscrire dans la cible.

Dans le monde anglo-saxon, de nombreuses méthodes d'EA ont été proposées, comme par exemple le cadre (*framework*) de Zachman [Zachman, 1987], le DoDAF pour *Department of Defense Architecture Framework* [DoDAF, web], le TOGAF pour *The Open Group Architecture Framework* [TOGAF, web], ou l'ITIL pour *Information Technology*

Infrastructure Library [ITIL, web]. Il y a ainsi un certain foisonnement des méthodes d'EA, au point que certains [Schekkerman, 2003] puissent le dépeindre comme une jungle. Ces différentes méthodes se situent généralement dans une optique plutôt anglo-saxonne. Elles sont relativement peu contraignantes (par exemple pour la modélisation des différents points de vue) et ne cherchent pas à hiérarchiser et à systématiser cette modélisation, le plus important étant finalement la dynamique induite dans l'entreprise [Salinesi, 2008].

En France, pays de Descartes, un cadre plus rationnel a vu le jour avec l'urbanisme des systèmes d'information. L'urbanisme a émergé au cours des années 90 dans de grandes entreprises, en particulier dans le domaine bancaire, confrontées à la maîtrise de leur système d'information. Jacques Sassoon, premier maître d'œuvre de l'urbanisme au Crédit Agricole, puis à la Société Générale, a popularisé l'application de la métaphore de la cité aux systèmes d'information [Sassoon, 1998]. Dans cette optique, un système d'information peut être structuré, à la façon d'une ville haussmannienne, en zones, en quartiers et en îlots, reliés entre eux par des voies de communication. Ces découpes en blocs ainsi que les règles de construction applicables à chacun de ces blocs peuvent être documentées rationnellement, à l'image d'un Plan Local d'Urbanisme (anciennement Plan d'Occupation des Sols). Un comité de validation pourra alors autoriser ou non chaque projet de modification du système d'information en fonction du respect ou non des règles applicables au bloc dans lequel il se situe. Ces principes ont été complétés, notamment par Christophe Longépé, responsable de l'urbanisme chez Sema puis à la Société Générale, en formalisant des vues, avec une structure et des règles associées à chaque vues. Il distingue ainsi dans [Longépé, 2006] quatre points de vue sur le système d'information, la découpe en blocs étant essentiellement l'apanage des vues fonctionnelles et applicatives.

"

- *La vision des processus métier qui décrit les processus métier mais aussi leurs relations ;*
- *la vision fonctionnelle qui décrit les fonctions que le système d'information doit supporter ;*
- *la vision applicative qui décrit l'ensemble des éléments logiciels du système informatique automatisant le système d'information;*
- *la vision technique, qui décrit l'architecture technique globale."*

Autrement dit, la vue métier répond à la question du pourquoi, la vue fonctionnelle à la question du quoi, la vue technique à la question du avec quoi et la vue applicative à la

question du comment [Simonin, 2009]. Ces quatre vues sont reprises dans la figure suivante, issue de [Longépé, 2006]

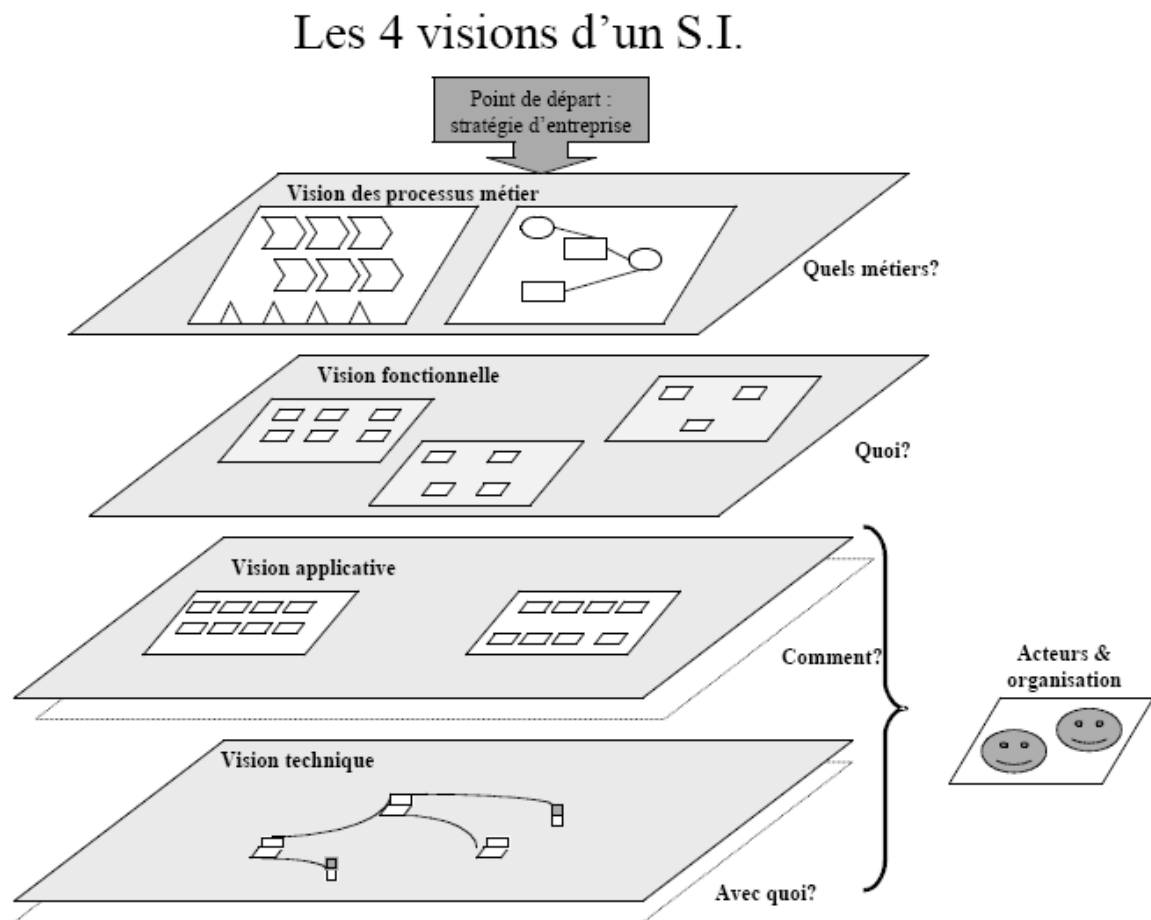


Figure 9 : Les quatre vues d'un système d'information

L'urbanisation est alors définie comme la démarche d'application des principes de l'urbanisme des systèmes d'information dans une entreprise particulière.

"Urbaniser, c'est organiser la transformation progressive et continue du système d'information visant à le simplifier, à optimiser sa valeur ajoutée et à le rendre plus réactif et flexible vis à vis des évolutions stratégiques de l'entreprise, tout en s'appuyant sur les opportunités technologiques du marché. L'urbanisme définit des règles ainsi qu'un cadre cohérent, stable et modulaire, auquel les différentes parties prenantes se réfèrent pour toute décision d'investissement dans le système d'information." [Urba-si, web]

Comme souligné par [Contini, 2002], l'urbanisation repose, à la différence des méthodes d'EA, sur le présupposé que l'on peut identifier des invariants suffisamment stables pour hiérarchiser des blocs et pour édicter des règles à validité sinon perpétuelle, du moins de

longue durée. Ainsi, dans son premier plan d'urbanisme à la Société Générale, J. Sassoon avait considéré comme pérennes les produits bancaires, unités de base de l'offre d'une banque. C. Longépé considère lui plusieurs types d'invariants. Les invariants issus de la vue métier, appelés classes concepts. Les invariants liés aux objectifs d'un système d'information d'entreprise (une zone échange, une zone gisement de données, une zone référentiel de données et de règles, une zone pilotage, une zone opération, une zone ressource). Des invariants issus de la vue technique, comme le modèle d'architecture technique *3-tiers*. Cette notion d'invariant n'est pas immédiatement compatible avec le modèle de l'entreprise agile, s'adaptant sans cesse aux évolutions externes pour maximiser ses profits. Une difficulté, mais aussi sans doute un intérêt, de l'urbanisme réside dans l'identification de ces invariants dans le métier de l'entreprise, au-delà des évolutions tactiques. En appliquant dans le chapitre III cette démarche aux services télécoms, nous nous confronterons d'ailleurs à cette question des invariants pour les services de communication.

Les principes de l'EA et de l'urbanisme ont été déjà appliqués par les opérateurs télécoms. Ils l'ont été non aux services télécoms eux-mêmes, mais aux activités internes des opérateurs ou des fournisseurs de service télécoms. Autrement dit, le travail de définition d'un cadre et d'organisation du système d'information a été mené sur les activités de prise de commande des services, de facturation des services et de supervision des services, mais non sur l'utilisation de ces services. De façon significative, le terme "système d'information" désigne chez les opérateurs télécoms les systèmes réalisant ces activités de prise de commande, facturation et supervision, mais non les plates-formes de service qui vont rendre le service. Ceci s'explique sans doute pour des raisons historiques. Les opérateurs ont longtemps rendu uniquement des services réseaux, autrement dit l'établissement d'une connectivité. L'enjeu essentiel était alors la mise en place de cette connectivité (son *provisioning*), sa facturation et sa supervision¹. Dans leurs démarches d'urbanisation, les opérateurs se sont ainsi focalisés sur leurs processus internes, et non sur les processus qu'ils partageaient avec leurs clients.

Ces travaux ont été menés notamment via un organisme de standardisation, le TMF pour *TeleManagement Forum*. Les domaines clés adressés par le TMF dans ce domaine sont ainsi la définition d'un cadre de référence (*framework*) des processus métier, la définition d'un

¹ Des travaux récents ont toutefois appliqué les principes de l'urbanisme aux services réseau, notamment F. Menai dans [Menai, 2005]

ensemble de processus métiers standards, la définition de systèmes permettant la réalisation des processus métiers et l'implémentation des solutions. Le TMF a ainsi formalisé le modèle eTOM pour *enhanced Telecom Operation Map*, résumé dans la figure ci-dessous. Pour plus de détails, on pourra par exemple se référer à [2006-7] ou directement à [eTOM, web].

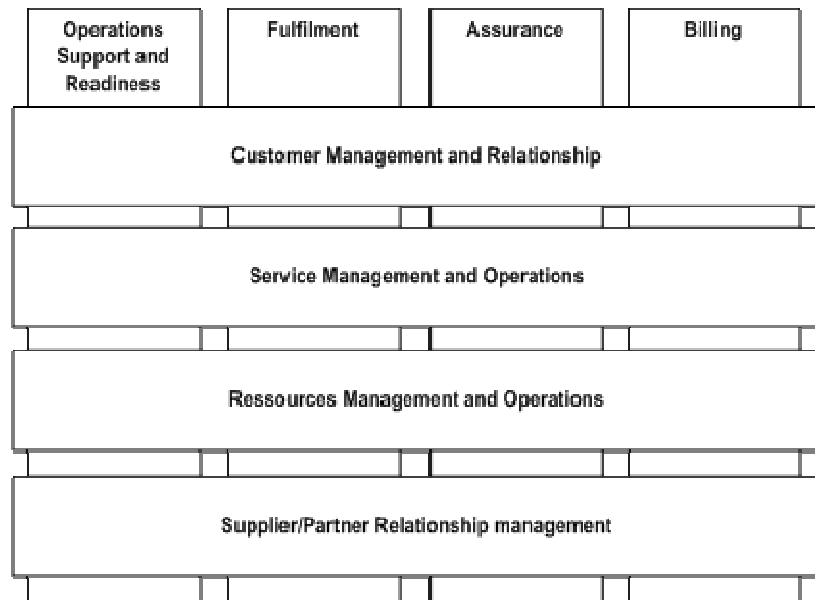


Figure 10 : Cadre de référence eTOM

Les lignes et les colonnes de cette figure constituent pour la gestion des services télécoms les invariants que nous venons d'évoquer. Les lignes représentent les grands domaines du système d'information, que l'on peut rapprocher des zones de l'urbanisation. Ce sont la relation client, la gestion des services, la gestion des ressources et la gestion des fournisseurs. Les colonnes représentent les grandes familles de processus métier interne des opérateurs télécoms :

- la prise de commande (*fulfillment*) pour vendre et mettre en place un service ;
- l'assurance pour garantir la qualité de service souhaitée ;
- la facturation (*billing*) pour facturer le service rendu ;
- les fonctions support (*operations support and readiness*) pour assurer la bonne marche des trois précédents.

Comme nous l'avons indiqué, ces travaux n'ont pas abordé ce qui fait la valeur d'un service pour ses utilisateurs, c'est-à-dire l'usage de ce service. Les plates-formes de service

sont ainsi vues comme des ressources, au même titre qu'un routeur ou commutateur. Mais les plates-formes de service de communication sont elles aussi partie-prenantes dans un système. Elles supportent des interactions avec les utilisateurs. Et elles interagissent avec d'autres plates-formes de service et avec des équipements réseau, pour qu'un service soit assuré de bout en bout. Les questions usuelles de l'urbanisme des systèmes d'information, comme la structuration en blocs, la répartition des responsabilités fonctionnelles entre ces blocs et les moyens de communication entre eux, s'appliquent tout à fait aux plates-formes de service actuelles, dont l'architecture demande à être structurée et rationalisée. Nous verrons dans le chapitre suivant comment ce besoin d'architecture découle de l'évolution des services de communication vers des services *user-centric*, centrés sur l'utilisateur. Contrairement à l'approche que nous avons proposée dans [2007-7], également présente dans par exemple [Vu Duong, 2003], une extension de eTOM pour prendre en compte l'usage des services ne semble pas pertinente. Les invariants des services ne sont en effet pas les mêmes que les invariants des processus internes de gestion de services, que nous avons résumés à travers les colonnes et les lignes de la figure ci-dessus. Il nous faudra dans le chapitre III déterminer ce qui est stable dans des services en rapide évolution.

Synthèse

Au début ce chapitre, nous avons défini une architecture comme la façon dont un système est constitué par des éléments de base, en vue de son fonctionnement. Quels sont ces éléments de base dans les communautés télécom, web et IT ? Ce sont sans doute les fonctionnalités et le comportement du réseau pour le RI, les plates-formes de service dans les services NGN, les ressources pour le web, les services logiciels dans les architectures orientées service.

Si ces éléments sont définis au cas par cas et sans méthode, nous tombons dans le foisonnement, la confusion, l'absence de maîtrise avec ses dangers. Avec les méthodes d'urbanisation, nous avons vu que pour orienter la conception de ces éléments vers la stratégie d'une entreprise, il fallait identifier une cible, des invariants sur lesquels s'appuyer pour identifier ces éléments, pour décrire leur architecture. La recherche de ces invariants n'est pas l'apanage de l'urbanisation, mais traverse les communautés télécom, web et IT. Ce sont les éléments de services et les SIBs pour les services RI, les enablers pour les plates-formes de service NGN, les concepts d'ontologies pour les ressources web, les activités de processus métier pour les services SOA. Pour les services de communication, cette recherche d'invariants manque souvent de méthode, d'une part en n'indiquant pas la façon de gérer leur cycle de vie (comment et pourquoi les introduire ou les supprimer ?) et d'autre part en ne distinguant pas toujours clairement entre les différents aspects d'un service (fonctionnel, technique, applicatif). Nous verrons dans le chapitre II quels sont les éléments de base constituant les services télécoms. Puis nous proposerons dans le chapitre III une méthode pour construire des invariants permettant de décrire rigoureusement ces éléments de base.

Chapitre II La convergence de service

Avant de chercher à décrire les architectures de service télécom, nous devons caractériser précisément ces services et leurs évolutions. La section 1 dresse ainsi un panorama des services de communication actuels et conclut à un manque d'outil de compréhension et de règles de conception pour ces nouveaux services. Comme les services de communication s'inscrivent dans le secteur plus large des services en général, nous resituerons dans la section 2 les nouveaux services de communication dans des tendances communes au secteur économique des services et chercherons comment l'évolution du secteur des services peut nous aider à mieux les appréhender. Cela nous amènera dans la section 3 à reformuler notre problématique en insistant sur le rôle central des utilisateurs dans les nouvelles générations de services.

II.1 Panorama des services de communication

Soulignons tout d'abord que le terme service n'est pas utilisé ici dans le sens logiciel vu au chapitre précédent pour les services web. Un service est ici défini comme une activité métier immatérielle mise à disposition par un fournisseur de service et apportant une valeur ajoutée à un utilisateur [2006-2]. Les activités métier immatérielles considérées pour les services de télécommunications sont notamment les activités de communication à distance. Comme nous le verrons, ces activités nécessitent de nombreuses interactions entre l'utilisateur et le fournisseur de service.

II.1.1 Les services télécoms et le NGN

Contrairement aux services d'Internet, les services télécoms n'ont historiquement pas été tirés par les usages, mais par la technique et notamment par la standardisation. Ceci a par exemple été le cas pour la téléphonie à commutation de circuit RTC, pour les services RI (Réseau Intelligent) ou pour la téléphonie mobile GSM (Global System for Mobile Communications, anciennement Groupe Spécial Mobile). La conduite des activités de standardisation implique évidemment une vision sur les services à offrir et les évolutions du métier des opérateurs, mais cette vision service demeure le plus souvent implicite. Par contre, les acteurs de la

standardisation s'accordent au préalable sur une vision technique, sur les objectifs techniques à réaliser. La dernière grande vague de standardisation dans les services de communication est le NGN (pour Next Generation Network), qui a été décliné par l'ETSI à travers l'IMS (pour IP Multimedia Subsystem). Les principes du NGN ont été définis à la fin des années 90 par les équipementiers et opérateurs télécoms. Les services qui pouvaient être offerts par les opérateurs étaient alors conditionnés au réseau d'accès utilisé, par exemple le GSM pour les services CAMEL (pour Customized Applications for Mobile Network Enhanced Logic), le RTC pour les services RI. La vision technique du NGN a donc été de réunifier les différents réseaux des opérateurs (le RTC, le réseau GSM, les réseaux de données), en un seul réseau supportant plusieurs services. Ceci a été réalisé en suivant un motif en couches (*layers*) classique dans les télécoms : une couche N fournit des interfaces à la couche N+1 et utilise les interfaces fournies par la couche N-1. Le NGN généralise, par ailleurs, le principe de la séparation entre la signalisation et les flux, principe introduit historiquement dans le RTC avec le réseau sémaphore, réseau logique dédié au transport de la signalisation, et repris par les protocoles de voix sur IP définis à la fin des années 90 à l'IETF avec RTP puis à l'ITU avec H.323 et H.248.

L'architecture d'un réseau NGN selon l'ITU [Knightson, 2005] distingue ainsi une strate service (*service stratum*) et une strate transport (*transport stratum*). La strate transport comprend :

- les fonctions de transfert de données du réseau cœur et des réseaux d'accès, comme par exemple UTRAN (UMTS Terrestrial Radio Access), WLAN, xDSL),
- les fonctions de contrôle de ces fonctions de transfert (par exemple le contrôle de l'attachement d'un terminal au réseau, ou le contrôle d'admission),
- le profil utilisateur de niveau transport (par exemple les données liées à l'attachement au réseau),
- ainsi que les fonctions de manipulation de flux media (par exemple le jeu d'annonces ou le transcodage).

La strate service inclut :

- les fonctions de contrôle des services (par exemple établissement de sessions d'appels, déclenchement de services),

- le profil utilisateur de niveau service (par exemple la liste des services souscrits à déclencher)
- et les fonctions applicatives (*application functions*), qui sont indépendantes des fonctions de contrôle des services et doivent offrir la flexibilité nécessaire pour réaliser des services variés. Ces fonctions applicatives peuvent être ajoutées successivement par le contrôle de service durant le processus d'établissement de session et peuvent également initier ou terminer des sessions.

Toutes ces fonctions et leur architecture sont représentées sur la figure suivante.

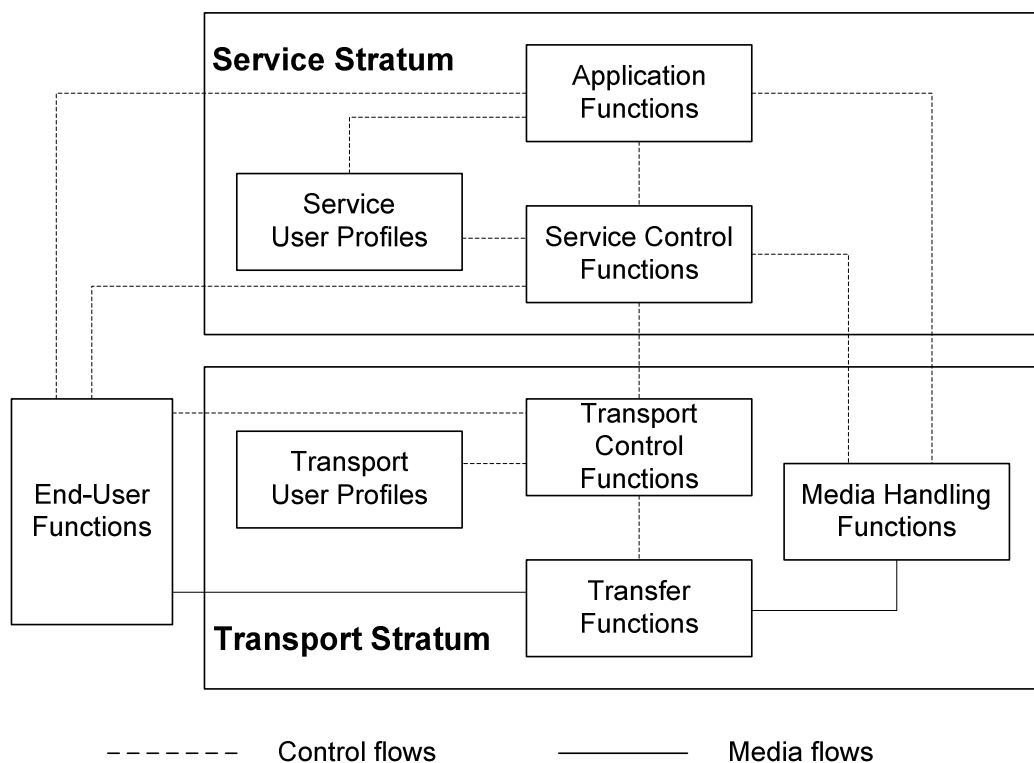


Figure 11 : Architecture NGN

La vision technique ayant piloté la standardisation du NGN a été claire et étayée, comme nous venons de le voir succinctement (pour de plus amples développements sur ce sujet, le lecteur pourra se référer aux articles [Knightson, 2005] et [Carugi, 2005] parus dans IEEE Communications Magazine). Cette vision peut être considérée pertinente, dans la mesure où d'une part elle se situe en continuité avec l'évolution historique des télécoms vers des réseaux multiservice et où d'autre part elle repose sur des technologies et protocoles relativement banalisés (par exemple SIP), même si cela se traduit souvent par l'ajout de quelques spécificités. Par contre, la vision des services réalisables avec le NGN n'a jamais été

formalisée. En effet, la communauté télécom qui a œuvré à la standardisation du NGN a majoritairement considéré que la définition des services n'était pas de son ressort. On lit ainsi la phrase suivante en 2002 sous la plume de Jean-Yves Cochenec, vice-chair de la commission SG13 à l'ITU-T, dans un article visant à populariser les travaux en cours autour du NGN [Cochennec, 2002] :

"To face the explosion of user demand for new services, NGN should support the provision of all kinds of services".

Cette citation illustre l'absence d'expression de besoin claire sur les services qui seront offerts par le NGN. Autrement dit, la spécification du NGN n'a pas été faite pour répondre à des besoins de services précis, mais s'est voulue générique, indépendante des services offerts. Dans le Réseau Intelligent, une démarche allant de la spécification à l'implémentation des services avait été définie avec les quatre plans du modèle INCM (pour Intelligent Network Conceptual Model), incluant la définition d'éléments de services (Service Features) sur le plan service puis de SIB (Service Independent Building blocks) sur le plan fonctionnel global. Aucun cadre conceptuel équivalent n'a été défini pour les services NGN. Même si des tentatives ont eu lieu, notamment avec Parlay/OSA, celles-ci n'ont jamais débouché sur un cadre reconnu et normatif pour les services NGN. On peut expliquer cette approche par la volonté d'offrir un réseau multiservice, donc en ne préjugant pas des services offerts. Mais ce qui est valable pour les couches basses d'un réseau ne l'est plus pour les fonctions applicatives. Les choix de protocoles et de mécanismes au niveau de la couche contrôle induisent de fait des limites et des contraintes aux services réalisables. Ces limites n'ont été ni explicitées ni débattues. La communauté NGN aurait par exemple pu définir un domaine de validité pour le NGN : pour quels types de services le NGN est-il utile et dans quels cas ne l'est-il pas ? Ces réflexions ont ainsi lieu a posteriori avec par exemple des controverses sur l'utilité de l'IMS. Il est vrai que répondre à ces questions est particulièrement délicat, en particulier à cause du lien fort entre service et technologie dans la culture télécom. De même, lors de la définition des services Réseau Intelligent mobile, aucune réflexion de grande ampleur n'avait été menée pour en décrire les buts et les domaines d'application, le choix des services déployés s'étant finalement fait de façon empirique.

II.1.2 Vers la convergence

En l'absence d'une analyse des besoins et d'une méthode pour concevoir des services NGN, on peut estimer que la communauté NGN a, logiquement, réutilisé implicitement les services qu'elle connaissait, c'est-à-dire les services de type RI. Les mécanismes de déclenchement de service du NGN sont ainsi parfaitement adaptés aux services des années 90, des services téléphoniques qui consistent principalement en une modification du processus d'appel. Malgré une volonté de généricité (déclenchement possible à partir de n'importe quel élément protocolaire d'un message SIP), ces mécanismes de déclenchement sont en effet d'une part liés au protocole SIP, c'est-à-dire à un protocole d'initiation de sessions d'appels et d'autre part provisionnés uniquement par l'opérateur (n'offrant donc pas d'ouverture à la composition de services par des utilisateurs). Significativement, un élément comme le SCIM (Service Capability Interaction Management, voir par exemple [2006-3]) prévu pour offrir plus de souplesse aux invocations de services n'a jamais été spécifié en détail. La prise en compte des besoins de nouveaux services au niveau de ce SCIM conduirait en effet à une grande complexité de mise en œuvre, comme illustré par exemple dans l'article [2007-2]. Le NGN est donc en résumé une infrastructure sécurisée permettant l'établissement de sessions de communication temps-réel entre usagers, cet établissement de session pouvant être piloté par des applications.

Les acteurs du monde des télécoms ont toutefois du traduire cette vision technique en une vision service, afin de construire et commercialiser leurs offres. Cette vision service est souvent résumée par le mot "convergence". Ce terme est à la fois employé par des universitaires, comme [Arbanowski, 2004], et par des opérationnels, comme [Lombard, 2008], et même dans la presse spécialisée sous des appellations comme "convergence fixe-mobile" ou "convergence voix/données", comme synthétisé dans [Broadbent, 2008]. Mais le mot convergence est trompeur car il réunit en fait deux convergences : celle des réseaux et celle des services.

La **convergence de réseau** désigne la possibilité d'offrir les mêmes services à partir de différents réseaux d'accès. C'est la traduction immédiate en termes de service offerts des possibilités techniques du NGN. Par exemple, le même service de téléphonie peut être utilisé à travers un accès ADSL ou un accès mobile 3G. Cette convergence des réseaux a été décrite métaphoriquement comme le passage d'une architecture en "silo" (où les couches de transfert, de contrôle et d'applications sont fortement couplées) à une architecture en couches (où ces

couches sont clairement distinctes et reliées via des interfaces standardisées, généralement ouvertes), comme illustré sur la figure ci-dessous.

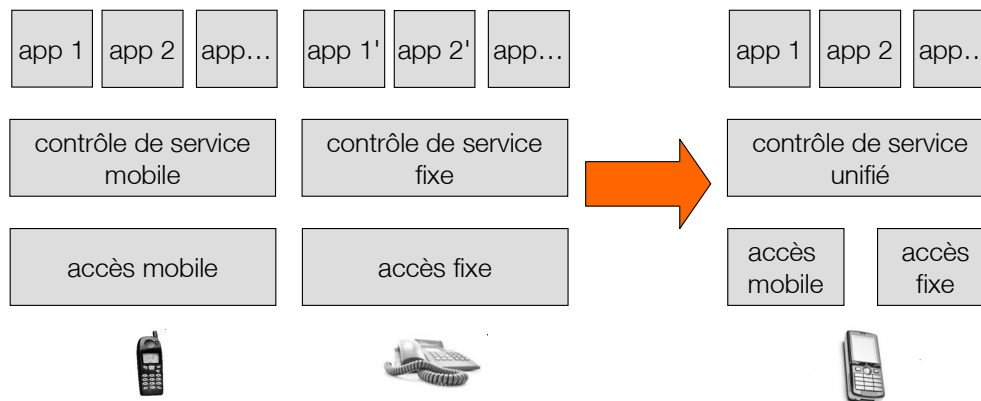


Figure 12 : La convergence de réseau

Le terminal est logiquement interfacé avec les couches de contrôle et avec les applications, et non principalement avec la couche de transfert.

La **convergence de service** quant-à elle n'est pas la simple traduction commerciale des possibilités techniques du NGN, mais elle constitue justement selon nous l'expression de besoin dont a manqué le NGN dans sa phase de définition. Cette convergence de services désigne l'intégration de services différents (différentes fonctionnalités, voire différents fournisseurs) au sein d'un même environnement de service, à l'initiative de l'utilisateur, ces services pouvant alors partager certains attributs et coopérer entre eux. Par exemple, dans un tel environnement, l'utilisateur peut avoir souscrit un service de téléphonie et un service de webmail qui partageront tous deux le même carnet d'adresses contenant à la fois les numéros de téléphone et les adresses email. Ou alors, l'état de présence d'une personne, captée par un service de présence, peut être un critère pour le déclenchement d'un renvoi d'appel. Lorsqu'un nouveau service est ajouté à l'environnement d'un utilisateur, ce service peut coopérer avec les services existants, par exemple un nouveau service de messagerie instantanée utilisera le même carnet d'adresses pour la construction de la liste d'amis. Le terme d'environnement de service désigne l'expérience ressentie par l'utilisateur que ses différents services font partie d'un même environnement, et non un environnement au sens logiciel. L'utilisateur est ici métaphoriquement au centre d'un écosystème de services. Cette vision est donc également nommée "user-centric", notamment par des acteurs de l'industrie des télécoms comme France Telecom [Lombard, 2008], mais aussi Telefonica, par exemple dans [Lizcano, 2008].

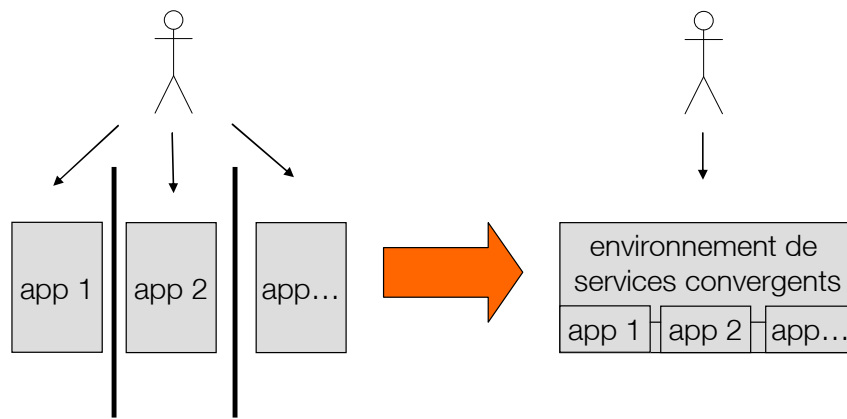


Figure 13 : La convergence de service

Des services de communication innovants cherchent par exemple à mêler la communication vocale, la messagerie instantanée, la présence ou le carnet d'adresses, tel que décrits en 2004 dans [2004-2] et lancés en 2007 à travers l'initiative RCS pour Rich Communication Suite. Cette convergence peut être considérée comme la déclinaison pour les services télécoms des évolutions fondamentales à l'œuvre dans le secteur socioéconomique, comme nous le verrons dans la section suivante. Nous pouvons avec [Griffin, 2007] résumer cette vision comme le passage du paradigme de la "*killer application*" à celui du "*killer environment*". Nous verrons dans la section 3 de ce chapitre comment un tel environnement de service peut être qualifié de système de service. Cette convergence de service peut être réalisée indépendamment de la convergence de réseau, comme illustré par des services Internet. Des fournisseurs de services comme Google ou Yahoo commencent à offrir des services user-centric. Ainsi pour Google, le mail (gmail), l'agenda (google agenda) et la communication instantanée (google talk) sont capables de coopérer afin de fournir un environnement de service unifié à l'utilisateur.

La convergence de service peut également être combinée avec la convergence de réseau. L'environnement de service décrit ci-dessus peut alors être accédé directement depuis de multiples réseaux d'accès. Dans ce cas, la problématique principale réside d'avantage dans l'adaptation des interfaces des services aux terminaux de ces réseaux d'accès (PC, téléphone mobile), plutôt que dans l'accès réseau aux plates-formes de service, qui peut de toute façon être réalisé indirectement via le réseau fédérateur Internet. Des projets européens comme SPICE [SPICE, web] ou Servery [Servery, web] ont étudié l'apport de l'IMS (et donc de la convergence de réseau) pour la convergence de service. Les résultats quoique intéressants ne montrent toutefois pas une contribution décisive de l'IMS à la convergence de service, à notre sens pour les raisons liées à la spécification du NGN discutées ci-dessus.

II.1.3 Implémentation de services convergents

Pour réaliser de tels services convergents, les acteurs du monde des télécoms ont travaillé en continuité avec les travaux de standardisation antérieurs. Comme nous l'avons vu dans le chapitre précédent, ils ont développé la notion de capacité de service (*service capability*) ou d'*enabler*, comme étant un élément standardisé, sur lesquels les industriels vont s'appuyer pour concevoir des services. Une capacité de service peut être par exemple la messagerie, ou la présence. Ces *enablers* permettent de coupler fortement un service rendu avec une technologie et une implémentation technique. Par exemple, dans un *enabler* de messagerie, le service rendu d'envoi de messages est couplé avec le protocole MMS et l'architecture de MMS UA (User Agent) et MMS Center. Dans un *enabler* de présence, la souscription, la collecte et la notification de la présence des utilisateurs sont couplées avec le protocole SIMPLE et une architecture d'Application Server de présence. Ceci permet de mettre en relation différents services reposant sur ces capacités, même lorsqu'ils sont proposés par des fournisseurs différents, puisqu'ils reposent in fine sur la même technologie. Des travaux ont même été menés pour qu'un fournisseur de service puisse proposer ces capacités à des fournisseurs de services tiers (notamment avec Parlay/OSA). Comme nous l'avons évoqué dans le chapitre précédent, ces travaux de standardisation laissent toutefois sans réponse deux questions majeures : comment identifier les capacités de service au-delà d'une simple "homologation" de technologies existantes (et donc plus pilotée par la technique que par le service rendu), et comment combiner effectivement des capacités de services différentes pour fournir un service complet.

La composition de services peut être vue comme une généralisation et une extension de ce principe de capacités de service. En effet, le principe de la composition de services est d'assembler différents services d'un ou plusieurs fournisseurs de service pour en former un nouveau. Selon les cas, cette composition peut être réalisée soit par un développeur professionnel, soit par un développeur amateur, soit même par un utilisateur final, comme développé dans [2009-3] ou [2008-9]. Dans tous les cas, le service élémentaire inclus dans une composition joue le rôle de capacité de service ou d'*enabler* vis-à-vis du service composé. De plus, cette approche peut être vue comme une réponse au besoin d'un environnement de service personnalisé. En effet, c'est soit l'utilisateur final, soit des développeurs plus à même de répondre à des besoins particuliers des utilisateurs qu'un opérateur qui vont réaliser ces compositions. L'apport de cette approche de composition de services n'est pas tant dans le

principe de composer des services, déjà présent dans les objectifs du réseau intelligent, comme souligné par [Magedanz, 2007] ou par [Simoni, 2007], que de proposer des technologies opérantes pour réaliser cet objectif. Cette approche reste toutefois confinée pour l'instant à des services limités, comme Yahoo Pipes¹ et peine à être généralisée. L'industrialisation d'un tel environnement unifié de services se heurte en effet à de nombreux écueils, comme par exemple :

- l'identification des services composables ou des enablers : tout service peut-il être composé avec d'autres ou non ? Comment déterminer le périmètre fonctionnel d'un service de base ou d'un enabler ?
- la cohérence fonctionnelle de la composition : le service composé ou l'assemblage d'enablers fournit-il un résultat cohérent pour l'utilisateur ? Comment identifier en amont les risques d'incohérence ? Comment réutiliser les mêmes données (par exemple un profil utilisateur) dans plusieurs services ?
- l'hétérogénéité des services à composer ou des enablers : les enablers et les services sont hétérogènes, à la fois en terme d'interface (protocole, API...) et en terme de fonctionnement (paramètres, modèles de données), comment mettre en correspondance ces différents éléments ?

Ces sujets sont peu traités dans les travaux existants. Il n'y a pas de mode de conception clair et global pour ces services à composer. Il n'y a pas non plus de claire dissociation entre service rendu et solution techniques. Pourtant, puisque la valeur ajoutée d'un service composé par rapport à un service standard est de répondre à un besoin spécifique, la problématique la plus importante est celle de la pertinence du service rendu, même si une solution technique pour composer les services est évidemment indispensable. Il reste à industrialiser la problématique du service rendu. Pour cela, nous allons nous aventurer au-delà des services de communication, dans le secteur des services en général pour y chercher une juste formulation de cette problématique.

II.2 Evolutions communes du secteur des services

A côté des services de communication, le secteur des services est composé de plusieurs autres domaines, détaillés dans les classifications de l'activité économique, telle que celle publiée par

¹ <http://pipes.yahoo.com/pipes/>

[Naics, 2007] : services de santé, services sociaux, services d'éducation, services financiers ou d'assurance, services de l'administration publique. Le secteur des services est le principal moteur de l'activité et de la croissance dans les pays développés, comme illustré dans le tableau suivant, publié par le Bureau International du Travail dans [ILO, 2008]. On peut par ailleurs relever que l'économie américaine est devenue presque exclusivement une économie de services (80% du PIB).

Tableau 2 : Part sectorielle dans l'emploi global dans les économies développées

Année	Proportion du secteur dans l'emploi global	
	1997	2007
Secteur agricole	6.1%	3.9%
Secteur industriel	28.3%	24.5%
Secteur des services	65.6%	71.5%

Ces taux illustrent la transformation de nos sociétés occidentales de sociétés industrielles, pilotées par l'offre, en sociétés de services, pilotées par la demande, comme souligné par exemple par [Child, 2001], [Chesbrough, 2006] ou [Spohrer, 2008]. Cette transformation a été relevée par de nombreux auteurs, à compter des années 80. Jean-Claude Delaunay constate ainsi dans [Delaunay, 1987] que :

"La société industrielle issue des deux derniers siècles est en voie de disparition. Si les principaux aspects de la révolution des services actuellement en cours sont observables par tous, ils ne sont pas pour autant aisément interprétables".

Quelques années plus tard, l'entrepreneur Gérard Mulliez écrit l'ouvrage "La dynamique du client - Une révolution des services" [Mulliez, 1994], avec le consultant américain Richard Whiteley, auteur de [Whiteley, 1991]. L'auteur conclut sur le passage "d'une société de consommation à une société de services". Cette transformation est maintenant effective. La plupart des entreprises du secteur des services ont intégré dans leur stratégie le passage d'une "orientation produit" (*product centricity*) à une "orientation client" (*customer centricity*), comme indiqué par exemple dans [Shah, 2006], [Day, 2006] ou [Kellogg, 2006]. Cette orientation client peut être vue comme un aboutissement logique de la notion même de service. Au fur et à mesure des études sur cette notion de service, la relation avec le client est en effet apparue comme un facteur différenciant les services des produits. Ainsi, d'après André Barcet, dans sa thèse de référence sur les services [Barcet, 1987] :

"Les services ne se produisent pas immédiatement comme les biens. La raison tient d'abord au fait que le service est un acte qui met en relation. De ce point de vue il est radicalement différent du bien. Si celui-ci suppose qu'il y ait un acte, un processus de production, ce processus a un objectif qui est le résultat de la production, le bien lui-même. La finalité du processus de production peut se concevoir comme étant dans le bien lui-même. Le service n'a pas de ce point de vue une finalité en lui-même, mais dans les effets qu'il a. Le service est dominé dans sa logique par ce que nous avons appelé le service rendu."

Comme relevé depuis longtemps [Zeithaml, 1996], un service est réalisé lors de son usage. Alors que la phase de réalisation d'un produit, par exemple une voiture, est indépendante de la phase d'usage du produit, par exemple la conduite de cette voiture, un service ne peut être réalisé sans utilisateur (ce qui préexiste à l'usage est la définition (spécification) du service et la capacité à le rendre). Ce point est habituellement nommé inséparabilité de la production et de la consommation (*inseparability of production and consumption* [Gronroos, 2007]). Pour aller plus loin dans notre étude, nous devons considérer un service comme à la fois et inséparablement activité du fournisseur et service rendu au consommateur. Citons de nouveau [Barcet, 1987] :

"On peut donc dire que la notion de service a dans une très large mesure un caractère très ambigu à partir du moment où elle peut être saisie ou appliquée à des niveaux très différents (...). Cette caractéristique n'est sans doute pas fortuite. Elle est bien d'une certaine manière la concrétisation d'une difficulté propre au service de ne pas pouvoir permettre la distinction entre la production du service et son utilisation ; les deux actes intellectuellement séparables sont au moins en partie concomitants et liés."

Tout en gardant à l'esprit leur concomitance, nous allons caractériser les services et leurs évolutions d'abord du point de vue de la production, puis de point de vue de l'utilisation. Nous verrons ensuite comment les travaux récents tentent d'inclure ces deux points de vue dans un système de service.

II.2.1 Le point de vue de la production du service : du besoin au service

La différence entre produits et services n'est pas simplement une distinction entre matériel et immatériel. L'essentiel dans une approche service est de rendre service au client, c'est-à-dire de mettre à sa disposition des compétences et de l'effort pour réaliser une prestation (définie conjointement) qu'il ne peut ou ne veut réaliser lui-même. La mise en place de services découle des besoins des utilisateurs. Ces besoins ne sont pas "standards", mais individuels et spécifiques. Frédéric Bastiat, économiste et homme politique français du XIXe siècle,

précurseur dans sa façon de fonder l'économie sur les services et cité par de nombreux auteurs anglo-saxons comme [Lusch, 2008], notait déjà dans ses Harmonies Economiques en 1851 :

"Besoin, effort, satisfaction: voilà l'homme, au point de vue économique. Nous avons vu que les deux termes extrêmes (besoin et satisfaction) étaient essentiellement intransmissibles, car ils s'accomplissent dans la sensation, ils sont la sensation même, qui est tout ce qu'il y a de plus personnel au monde, aussi bien celle qui précède l'effort et le détermine, que celle qui le suit et en est la récompense. C'est donc l'Effort qui s'échange, et cela ne peut être autrement, puisque échange implique activité, et que l'Effort seul manifeste notre principe actif."

Puis :

"Je viens de définir le service. C'est l'effort dans un homme, tandis que le besoin et la satisfaction sont dans un autre."

Selon cette définition, un service est donc fondamentalement l'effort (ou plus précisément la chaîne des activités) du ou des fournisseurs du service que le consommateur du service sollicite pour obtenir la satisfaction d'un besoin. Ce type de définition a été repris par la communauté du marketing, comme par exemple

"Services are deeds, processes and performances" [Zeithaml, 1996]

Ou bien encore

"[Services are] any act or performance that one party can offer to another that is essentially intangible" [Kotler, 1988]

Concrètement, comment peut-on construire un service à partir de besoins individuels des consommateurs de service ? Nous pouvons distinguer deux types d'approche.

Dans une **approche standardisée**, les organisations (entreprises, administrations) considèrent les services comme des produits de masse, les concevant, les produisant et les commercialisant comme des produits. La production de services est ainsi objectivement décorrélée des besoins effectifs des clients de ces services et se base sur une définition des services par le marketing ou la R&D de l'entreprise à partir des besoins estimés des clients en général. Ces besoins estimés peuvent être par exemple issus d'analyses du marché et des offres concurrentes ou d'une représentation implicite des besoins des clients par les concepteurs de service [Flichy, 2008]. L'organisation des entreprises industrielles reflète en général ce mode de fonctionnement, avec des "silos" spécialisés chacun sur une fonction

particulière, par exemple un silo conception, un silo production, un silo informatique... [Kellogg, 2006]

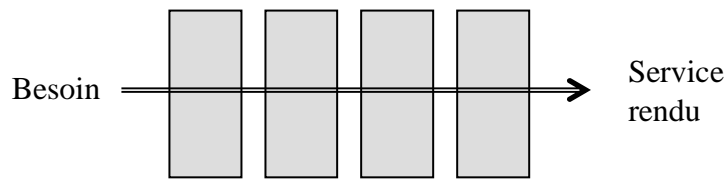


Figure 14 : Approche standardisée de la production de service

Un service "standard" (c'est-à-dire ni personnalisé, ni personnalisable) est proposé en bout de chaîne. Le client a le choix entre plusieurs offres qui packagent ces services, de la même façon qu'il a le choix entre plusieurs modèles de voitures chez un constructeur. La production précède le besoin, l'offre précède la demande, c'est un mode d'organisation typique d'une société industrielle, société de l'offre. Dans le domaine des télécommunications, on peut penser au minitel ou au GSM, services "standardisés" (de part la position monopolistique dans le cas du minitel et effectivement standardisé pour le GSM), qui ont précédé les besoins des utilisateurs. Ce modèle, qualifié de modèle cathédrale dans [Lombard, 2008], est sans doute devenu inadapté pour le secteur des télécommunications, en constante et rapide évolution.

Une **approche individualisée** est également possible. Certains services sont effectivement personnalisés et adaptés à la demande des clients, mais sont souvent le fait d'individus ou de structures légères. On notera ainsi la différence entre le médecin généraliste dont le service s'adapte au patient et l'hôpital où le patient doit, dans une certaine mesure, s'adapter au service. Lorsque le fournisseur du service est un individu, le client peut négocier avec le fournisseur pour que son besoin soit pris en compte de façon personnalisée. Par exemple, un plombier pourra adapter sa prestation au besoin spécifique de son client, à condition que la satisfaction de ce besoin fasse appel à des prestations qu'il propose. Certaines organisations s'inspirent de ce modèle en formalisant les prestations qu'elles peuvent proposer et les besoins auxquels elles peuvent répondre, par exemple à travers la rédaction de "projets d'établissement" dans le domaine médico-social ou éducatif. Le service rendu à chaque usager peut alors être formalisé dans ce cadre. On lit ainsi à l'article 8 de la loi n° 2002-2 du 2 janvier 2002 rénovant l'action sociale et médico-sociale :

"Un contrat de séjour (...) est élaboré avec la participation de la personne accueillie ou de son représentant légal. Ce contrat (..) définit les objectifs et la nature de la prise en charge ou de l'accompagnement dans le respect des

principes déontologiques et éthiques, des recommandations de bonnes pratiques professionnelles et du projet d'établissement. Il détaille la liste et la nature des prestations offertes ainsi que leur coût prévisionnel."

En termes d'organisation, cela se traduit par une structure horizontale, chaque couche correspondant à des métiers distincts et des compétences spécifiques, proposant donc des prestations spécifiques. Un besoin client est alors pris en charge par une équipe pluridisciplinaire qui le traite de bout en bout. Comme illustré dans [Kellogg, 2006], les membres de l'organisation porteurs de compétences distinctes, de métiers différents, échangent et se coordonnent alors par un mécanisme informel de traduction, chaque métier ayant non seulement un regard spécifique sur le besoin à satisfaire et sur les solutions à apporter, mais également un vocabulaire spécifique pour l'exprimer. Ces mécanismes de coordination sont représentés dans la figure par des flèches pointillées entre les couches.

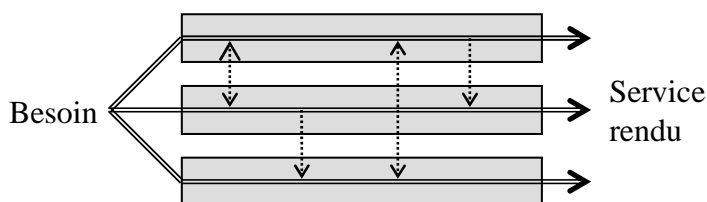


Figure 15 : Approche individualisée de la production de service

Le service rendu n'est plus standardisé, mais personnalisé pour la personne qui en bénéficie. Cette approche présente toutefois des inconvénients importants pour les organisations industrielles. Elle nécessite en effet une grande coordination entre les différentes entités organisationnelles, ce qui se traduit par exemple par de nombreuses réunions [Kellogg, 2006]. Chaque cas étant traité de façon unique, il est difficile de capitaliser sur des processus métier stables. Dans le cadre des télécommunications, une telle approche peut être envisagée pour des offres sur mesures à des grands-comptes, mais est hors de propos pour des services de masse. Par ailleurs, comme souligné dans [Rust, 2006], cette approche est difficilement réalisable pour des services automatisés basés sur des machines, comme les services de communication.

Une **approche front-office back-office** permet de répondre à ces inconvénients. Il s'agit de mettre en place un frontal pour traiter les besoins des clients, ce frontal étant chargé de faire appel aux différentes entités organisationnelles et de les coordonner pour répondre à ce besoin. Cela permet à ces entités de garder des activités relativement standardisées, tout en pouvant répondre à un besoin spécifique.

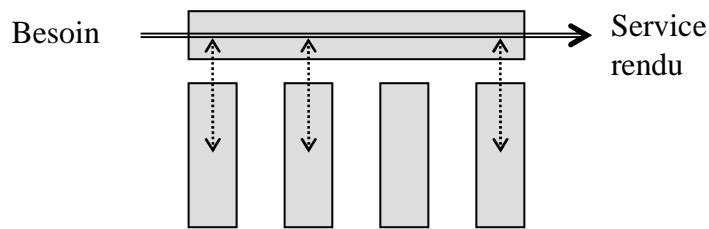


Figure 16 : Approche front-office / back-office de la production de service

Ce mode de fonctionnement est notamment identifié par [Kwan, 2008] ou [Kellogg, 2006] comme un compromis pour fournir des services adaptés au client tout en conservant une organisation efficace. Nous retrouvons ce motif dans plusieurs domaines, par exemple dans la découpe des entreprises de service entre leurs lieux d'interaction avec les clients et leurs lieux de traitement de leurs processus internes. On peut également lire ce motif dans la mise en place dans les services publics de guichets uniques, par exemple avec la création en France de "maisons de l'emploi" ou de "maisons des personnes handicapées" pour prendre en charge la globalité des besoins des personnes en recherche d'emploi ou en situation de handicap, tout en conservant des organismes distincts traitant différents aspects. Cette zone *front-office* est alors chargée de décomposer le besoin du client en services standards que peuvent traiter les différents silos, et si nécessaire de mettre en œuvre ces services standards pour le compte du client afin de lui fournir un résultat synthétique. Comme présenté dans [Zeithaml, 2006], ceci permet d'améliorer "l'expérience utilisateur" en faisant coopérer des entités différentes afin de répondre à des besoins clients spécifiques. Certains auteurs comme [Rust, 2006] soulignent que ce motif d'organisation permet de passer d'une production de masse (*mass production*) à des services de masse (*mass services*), et qu'il est ainsi typique d'une société de services, société de la demande.

II.2.2 Le point de vue de l'utilisation du service : client, consommateur, utilisateur

Si nous avons jusqu'ici fait indistinctement usage des termes de client, de consommateur ou d'utilisateur, ils doivent néanmoins être distingués. Un client est défini dans Le Robert comme une "*personne qui requiert des services moyennant rétribution*". Alors qu'un utilisateur est une personne qui use de, c'est-à-dire une personne qui "*fait en sorte qu'une chose produise un effet souhaitable en la faisant fonctionner, agir*". L'utilisateur est ainsi par sa définition même dans une posture active vis-à-vis du service, ce qui n'est pas nécessairement le cas du

consommateur. Ces définitions rejoignent celles du TeleManagement Forum [eTOM, web]. Nous pouvons donc distinguer trois rôles en vis-à-vis du producteur de service, en adoptant la terminologie suivante :

- le client, avec qui le producteur de service a une relation commerciale
- le consommateur, bénéficiaire du service qui en reçoit la valeur par la satisfaction de son besoin
- l'utilisateur, qui agit pour faire fonctionner le service

Dans les services télécoms, ces trois rôles sont souvent confondus, excepté dans le cas des services aux entreprises, où le client sera le service informatique ou télécom, le consommateur l'entreprise comme organisation et l'utilisateur les employés de cette entreprise.

Nous ne nous intéresserons pas ici à la notion de client, car nous ne n'étudierons pas la relation commerciale entre client et fournisseur de service, décrite en détail par le TMF à travers les processus de commande/livraison, service après-vente et facturation (*fulfillment, assurance, billing*). Les études sur les SLA (Service Level Agreement), pour mesurer l'adéquation entre un service fourni et le contrat le spécifiant, entrent également dans cette catégorie.

La notion de consommateur nous amène à la question de la valeur du service consommé. Etudier la valeur d'un produit ou d'un service pour son consommateur est l'objet d'une discipline, l'analyse de la valeur, "*Méthode de compétitivité, organisée et créative, visant la satisfaction du besoin de l'utilisateur par une démarche spécifique de conception à la fois fonctionnelle, économique et pluridisciplinaire*" [AFNOR, 1991]. Cette discipline s'intéresse aux produits dans un sens large, définis comme "*Ce qui est (ou sera) fourni à un utilisateur pour répondre à son besoin*", par exemple un matériel, un service, un système, un ouvrage, un processus industriel... Il est à noter que les normes de l'AFNOR citées ici emploient le mot utilisateur et non consommateur, mais il s'agit bien du consommateur selon le cadre terminologique introduit ci-dessus. Pour la suite, nous nous intéresserons uniquement à l'analyse fonctionnelle, c'est-à-dire la partie de cette discipline qui regarde comment les besoins sont satisfaits par le service rendu, et non les aspects liés aux coûts. Le terme analyse ne présente pas ici la même signification que dans le domaine du génie logiciel, l'analyse fonctionnelle ne se limitant pas à une analyse du besoin, mais incluant également les choix structurants de conception du produit à travers la notion de fonction technique que nous allons

introduire ci-dessous. Le concept fondamental de l'analyse fonctionnelle est celui de fonction :

"Parmi les fonctions du produit, apparaissent les fonctions de service qui expriment la participation du produit à la satisfaction des consommateurs. Ce sont, soit des fonctions d'usage, soit des fonctions d'estime, suivant respectivement leur nature objective et subjective. Elles sont à distinguer des fonctions techniques, qui ne sont pas directement perceptibles par le consommateur, mais auxquelles le concepteur fait appel, dans une solution donnée du produit, pour que celui-ci remplisse les fonctions de service."[AFNOR, 1991]

Une fonction de service est ainsi définie comme

"Action attendue d'un produit (ou réalisée par lui) pour répondre à un élément du besoin d'un consommateur donné". [AFNOR, 1991]

Alors qu'une fonction technique est définie comme

"Action interne au produit (entre ses constituants) choisie par le concepteur-réalisateur, dans le cadre d'une solution, pour assurer des fonctions de service". [AFNOR, 1991]

L'objectif d'une analyse fonctionnelle est de documenter le passage des besoins aux solutions techniques, par la médiation des fonctions de service et des fonctions techniques.

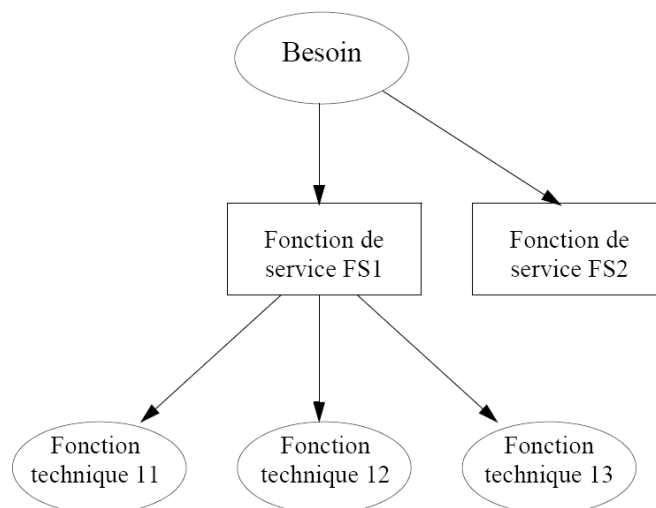


Figure 17 : Démarche d'analyse fonctionnelle

Des méthodes de type FAST (Functional Analysis System Technique) ou SADT (Structured Analysis and Design Technique) permettent notamment de formaliser ce passage sous forme

de diagrammes. La représentation des SIBs du Réseau Intelligent sur le plan fonctionnel global s'inspire d'ailleurs du formalisme SADT.

Regardons maintenant plus en détail la notion d'utilisateur. [Spohrer, 2008] définit un service comme un processus d'échange au cours duquel la valeur économique est coproduite entre l'utilisateur et le fournisseur. L'utilisateur est en effet impliqué dans la création de la valeur du service. Ainsi, dans le domaine médical, un médecin généraliste aura besoin de la participation de l'utilisateur (i.e. du patient) pour le soigner, par exemple en exposant correctement ses symptômes, en agissant pour être examiné (dire ah...), en suivant le traitement. Une partie de la valeur de la prestation médicale sera ainsi déterminée par la bonne volonté du patient, le médecin et le patient coproduisent la valeur du service. Ces illustrations semblent loin des services de communication, mais nous indiquent ce qui constitue la "quintessence" d'une relation de service. Revenons aux services de communication avec un exemple de service aux entreprises. Dans un service de téléphonie sur IP hébergé par un opérateur, également appelé Centrex IP, le bénéficiaire du service – le consommateur –, est l'entreprise qui en reçoit la valeur (gain sur le budget télécom, gain de productivité grâce à une meilleure collaboration entre ses employés, notamment en situation de nomadisme). Cette valeur n'est pas uniquement produite par le fournisseur du service, mais également par les employés, de par l'utilisation pertinente qu'ils font de ce service.

Un modèle économique d'audience rend cette distinction encore plus intéressante. Dans un tel modèle, les services sont gratuits et le fournisseur du service est rémunéré par des annonceurs publicitaires. La valeur du service pour son fournisseur dépend donc de sa capacité à générer des revenus publicitaires, c'est-à-dire de sa capacité à ce que des publicités soient vues. Dans un modèle d'audience, l'acteur essentiel devient donc l'utilisateur, celui qui regarde les publicités lorsqu'il utilise le service. Ce sont les goûts et les préférences de cet utilisateur qu'il s'agit de cibler afin de maximiser les revenus publicitaires.

Dans le cadre des services de communication, la notion d'utilisateur permet de désigner toutes les personnes impliquées dans le rendu du service, quoique externes au fournisseur de service. On retrouve cette question dans tous les services de réseaux, comme la téléphonie, mais aussi les réseaux sociaux. Ainsi, dans un service de blog, quel est le consommateur de la fonctionnalité de commentaire ? La réponse n'est pas évidente et la notion d'utilisateur permet de travailler sur le rendu du service tout en éludant cette question.

D'une façon générale, dans les services de télécoms, le fournisseur du service télécom fournit à l'utilisateur la capacité de réaliser lui-même des actions.

II.2.3 Du service au système de service

Cette importance de la notion d'utilisateur et de co-création de valeur est soulignée ces dernières années par des chercheurs universitaires et industriels tentant d'aller au-delà des travaux existants de la communauté marketing pour conceptualiser les services. Ce travail est par construction pluridisciplinaire car les différents aspects des services sont étudiés dans des disciplines variées, telles que l'économie, le marketing, la sociologie, la gestion (*management*) ou l'informatique. Cette tentative de conceptualisation est néanmoins menée dans une perspective IT, notamment sous la houlette d'IBM, avec en ligne de mire une meilleure adaptation entre l'offre IT et les besoins du secteur des services, identifié comme relais de croissance pour l'informatisation des processus métier. Comme exposé dans le texte fondateur [Chesbrough, 2006], cette thématique s'est nommée science des services (*service science*) ou SSME, pour Service Science, Management and Engineering. Un service y est d'abord défini comme un processus d'échange entre des parties-prenantes.

"A service includes an exchange between two or more parties" [Metcalfé, 2001]

"This exchange is co-generated by both parties" [Chesbrough, 2006]

L'étude de ce processus d'échange conduit ensuite ce chercheurs à envisager un service comme un système complexe, composé d'acteurs (et notamment celui qui fournit le service et celui qui en profite), de produits (machines, logiciels...) et d'information partagée, de processus liant l'ensemble¹, comme illustré par les citations suivantes.

"Service systems comprise service providers and service clients working together"
[Tien 03]

"The actors of a service system (client, provider) "are connected by value propositions, and shared information" [Spohrer, 2007]

"Service systems are value-creation networks composed of people, technology, and organizations" [Maglio, 2006]

¹ Cette notion de processus rejoint d'ailleurs la notion de logique de service, classique dans les services de communication.

"In general, a service system can be considered to be a combination or recombination of three essential components—people, processes and products."
[Tien, 2008]

"We define a service system as a valuecoproduction configuration of people, technology, other internal and external service systems, and shared information"
[Spohrer, 2007]

Un système de service peut être composé d'autres systèmes de service. Ces notions permettent de prendre en compte plus simplement la diversité des services, et notamment le *self-service*. Dans ce cas, l'utilisateur est fortement impliqué dans la production de valeur. Ainsi, dans une cafétéria, le fait que l'utilisateur porte son plateau de stand en stand peut produire de la valeur à la fois pour le consommateur (qui choisit plus facilement les plats qu'il souhaite et bénéficie d'un prix attractif), et pour le fournisseur (qui fait l'économie de coûts de personnel et logistiques). Consommateur, utilisateur et fournisseur sont ainsi parties-prenantes d'un système de service, où l'utilisateur et le fournisseur collaborent pour satisfaire les besoins du consommateur.

Avec cette communauté de recherche SSME, on passe ainsi d'une vision usuelle d'un service comme la fourniture d'un bien immatériel en réponse à un besoin...

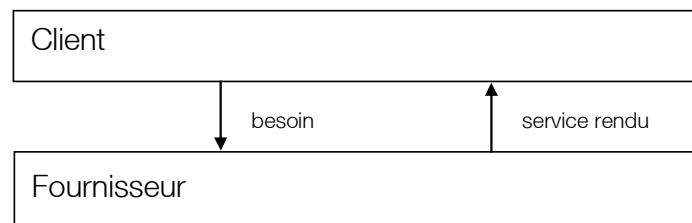


Figure 18 : Le service comme fourniture d'un bien immatériel

... à une vision plus large où plusieurs acteurs collaborent pour co-créeer de la valeur à travers la réalisation d'un processus de service.

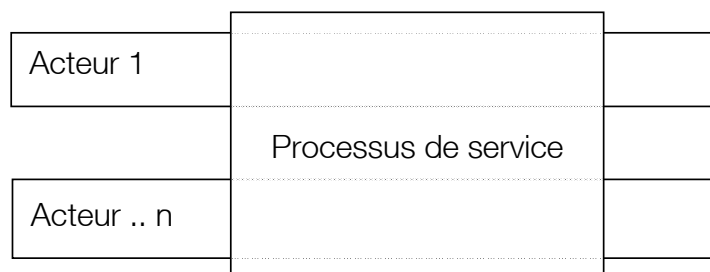


Figure 19 : Le service comme système

II.3 Application aux services télécoms

Nous pouvons maintenant relier les évolutions des services télécoms à la lumière de la section précédente.

II.3.1 Les services télécoms du point de vue de la production ?

Du point de vue de la production, les services télécoms présentent les motifs que nous avons vus précédemment. A travers ces motifs, nous retrouvons l'évolution des architectures de service télécom introduites dans la première section de ce chapitre. La convergence de service consiste en effet à passer d'un motif en "silo", où le consommateur doit lui-même répondre à son besoin en invoquant des applications hétérogènes à un motif front-office / back-office où une entité de coordination, un environnement de service unifié, se charge d'invoquer les différentes applications. Ce motif de front-office / back-office se retrouve ainsi dans l'architecture IMS, où le S-CSCF (Serving - Call Session Control Function), éventuellement complété par le SCIM (Service Capability Interaction Management), assurent l'invocation de diverses applications suite à une requête de l'utilisateur du service.

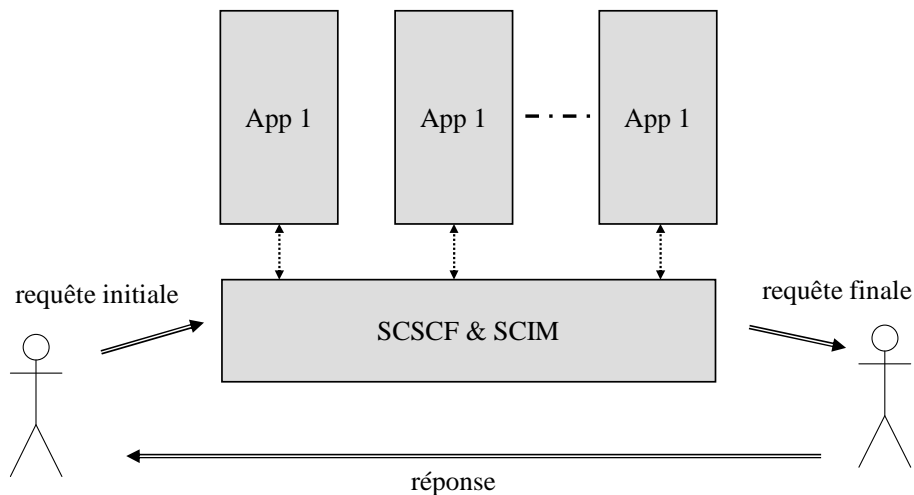


Figure 20 : Environnement de service IMS

Ce même motif front-office / back-office se retrouve également dans le monde web, par exemple avec les travaux sur un service broker SOA.

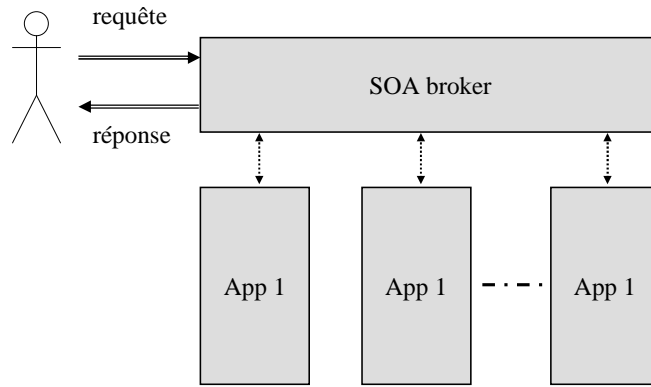


Figure 21 : Environnement de service SOA

Ces technologies sont vues comme un moyen de réaliser un environnement de services convergents, au sens de la section 1.3 de ce chapitre. Toutefois, les problématiques de divergence entre protocoles et solutions techniques rendent délicate la constitution d'un front-office unique. Ceci conduit à réaliser la convergence de service au sein d'environnements techniques compliqués. Certains travaux, comme [Blum, 2008a] distinguent ainsi un accès aux applications "par le bas" en SIP ou "par le haut" en http.

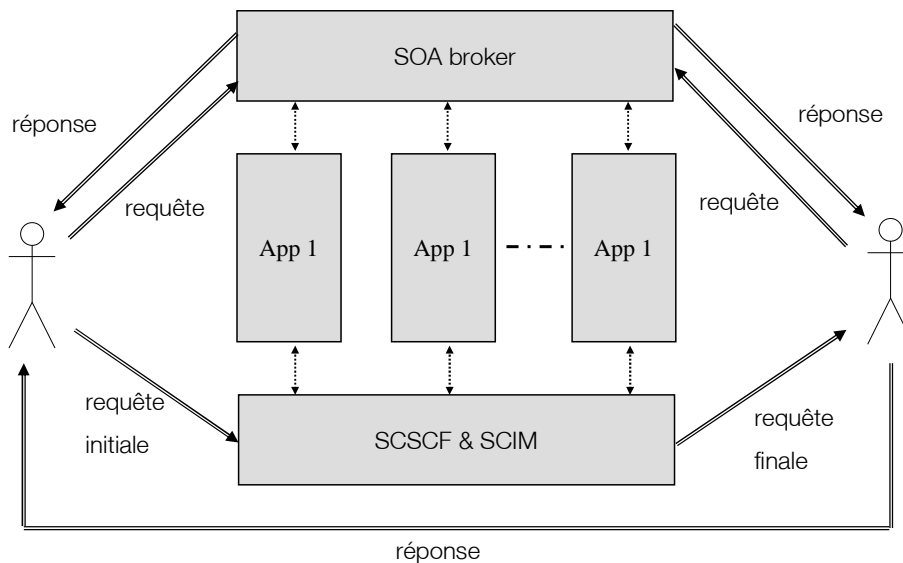


Figure 22 : Environnement de services convergents SOA + IMS

Cette séparation en multiples front-office rend le système difficilement gérable et évolutif. En effet, les mêmes informations devront être dupliquées plusieurs fois, induisant des risques de divergence. Réaliser un front-office commun aux différents services nécessite deux pré-requis. Tout d'abord, il faut être capable de mettre en place un environnement technique supportant les différents protocoles et solutions considérées, comme défendu dans [2008-1]

ou [2009-2]. De notre point de vue, il ne peut y avoir qu'un seul front-office, interface d'accès aux services aux services, même si des mécanismes techniques de modification de l'établissement d'une session existent par ailleurs.

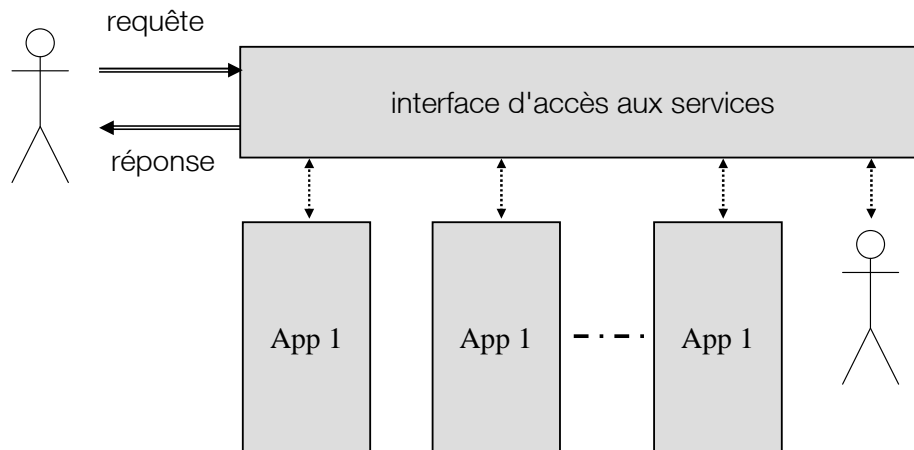


Figure 23 : Environnement de services convergents sans couture

Ensuite, il faut être capable de segmenter clairement les fonctions des différentes applications, de même que chaque entité organisationnelle définissait dans la section précédente les prestations qu'elle offrait. En effet, ceci est indispensable pour coordonner ces applications en déterminant quelle application invoquer en réponse à quel besoin. Ceci permet également d'identifier et de réutiliser des informations communes aux différentes applications, comme des profils utilisateurs, des préférences. Nous devons donc définir une sémantique qui puisse être transverse aux différents services télécoms et qui permette de décrire leurs fonctions et leurs données. En d'autres termes, nous devons déterminer des invariants sur lesquels nous appuyer pour décrire une architecture de service télécom. Cela sera l'objet du chapitre III. Pour cela, nous devons tout d'abord considérer les services télécoms du point de vue de l'utilisation.

II.3.2 Les services télécoms du point de vue de l'utilisation ?

Les méthodes d'analyse fonctionnelle appliquées aux services télécoms ont principalement donné lieu à une méthodologie, qui consiste d'abord en la rédaction d'un dossier d'expression de besoins de service contenant les besoins auxquels doit répondre le service, puis en l'élaboration en réponse d'un cahier des charges proposant des solutions. Dans les faits, ces dossiers d'expression de besoins s'appuient le plus souvent déjà sur des solutions techniques pressenties. Les besoins listés sont ainsi fréquemment non pas le résultat d'une analyse des

besoins des consommateurs, mais basés soit sur les fonctionnalités que la solution technique pressentie est capable de rendre, soit sur les fonctionnalités des services concurrents. Les besoins sont généralement exprimés sans distinguer les aspects fonctionnels des aspects non-fonctionnels et sans séparer entre les besoins des différents acteurs du service. Ceci s'explique par l'absence de formalisme commun qui pourrait être utilisé pour réaliser cette expression de besoins : il n'existe ni de liste normée des besoins possibles, ni de liste des fonctions qui pourraient être proposées pour répondre à ces besoins.

La construction d'une telle liste commune des besoins auxquels peuvent répondre les services télécom se révèle en effet malaisée. Quels sont les besoins à considérer ? Certains travaux comme [Ghadiri, 2007] tentent de classer les services télécoms au sein d'une catégorisation théorique des besoins humains, telle la pyramide de Maslow. Outre le fait que ce type de classification des besoins humains soit sujet à caution, ces approches sont largement inopérantes, car elles ne nous fournissent aucun outil et sont trop générales pour les services télécoms.

Pour avancer, la prise en compte des points de vue et des objectifs des différentes parties-prenantes (*stakeholders*) participant au rendu service nous semble essentielle et trop peu étudiée. Si le point de vue de ces parties-prenantes peut sembler subjectif, il est en fait tout à fait classique pour décrire les exigences (*requirements*) sur un système [Rolland, 2000]. Il est ainsi par exemple préconisé pour rédiger des cas d'utilisation UML, par exemple dans [Cockburn, 1997]. Cela procède d'une approche boîte noire : le service est décrit tel que vu par ceux qui le manipulent de l'extérieur.

"Linking use cases to actors' goals is so very significant because it shifts the writer's attention away from the function lists that most programmers worry about and puts it back on the users: What the users really want to accomplish with the software are their goals when using the software. If the software supports those goals, the software will yield the greatest business value." [Cockburn, 1997]

Les différents acteurs du service sont ainsi des collaborateurs, des parties prenantes dans la réalisation du service. Ces acteurs sont notamment l'utilisateur et le fournisseur du service.

"The primary actor is only one of the stakeholders. The organization that owns the system is another, a government regulatory agency might be another, and there might be others as well." [Cockburn, 1997]

Nous pourrions appliquer ces principes en tentant de nous abstraire des solutions techniques (c'est-à-dire de *"the function lists that most programmers worry about"*). On évoquera ainsi

par exemple le besoin de communiquer par écrit plutôt que le besoin d'envoyer un email. Même si historiquement, l'apparition d'un besoin est souvent induite par l'existence d'une solution technique, une telle décorrélation facilite la prise en compte de plusieurs solutions techniques répondant au même besoin. Nous pouvons ainsi identifier certains besoins généraux des consommateurs auxquels répondent les services télécoms :

- communiquer instantanément avec autrui, par la voix, la vidéo ou l'écrit ;
- communiquer par messagerie ;
- consommer des contenus multimédia.

Ces besoins se traduisent en fonctions de service au sens de l'analyse fonctionnelle. Une fonction de communication instantanée, une fonction de communication par messagerie, une fonction de consommation de contenus. A ces besoins des consommateurs s'ajoutent les besoins du fournisseur de service, par exemple la gestion des droits des utilisateurs à accéder aux services, en fonction de leurs souscriptions.

Nous pouvons en outre identifier une hiérarchie dans ces besoins. La gestion des droits est ainsi un besoin sur la satisfaction duquel vont s'appuyer les autres. Mais cette liste de besoins ou de fonctions de service reste toutefois de haut-niveau et ne permet pas de déduire une sémantique précise pour décrire le service rendu par les différents services télécoms. Nous pourrions préciser ces besoins en y rattachant des solutions techniques, comme l'email ou la messagerie instantanée. Cela pourrait nous aider à constituer une segmentation des offres commerciales en fonction des grandes catégories de besoins auxquelles elles répondent mais cela ne nous aiderait pas pour autant à élaborer une sémantique commune permettant de répondre aux questions laissées en suspens dans la section 1, comme le périmètre fonctionnel des enablers, ou la cohérence fonctionnelle d'une composition de services télécoms. Nous verrons dans le chapitre suivant que ces besoins de haut-niveau constituent un point de départ pour identifier des processus de services dans une vue métier des services télécoms.

II.3.3 Comment architecturer les services télécoms ?

Nous avons introduit dans la section 3.1 de ce chapitre le besoin d'une sémantique commune pour décrire les services télécoms. Dans la section 3.2, nous avons vu qu'une approche par les besoins des consommateurs ne permettait pas de construire avec précision une telle sémantique. Pour avancer, nous allons maintenant envisager les services télécoms comme systèmes de services. Nous avons présenté à la section 2.3 un service comme un processus

d'échange, de co-création de valeur. Plutôt que sur le couple classique consommateur/producteur de service, nous allons insister dans la suite de cette étude sur le lien utilisateur-système de service, ce qui nous conduira naturellement à étudier la façon dont l'utilisateur participe au service, cette perspective nous semblant la plus pertinente à explorer. Utilisateur et système de service ne sont pas à considérer comme deux entités distinctes (comme peuvent l'être les rôles consommateur/fournisseur), mais dans une perspective systémique : l'utilisateur est l'un des composants du système de service. Cette perspective diffère d'approches visant à composer automatiquement des services à partir de besoins de consommateurs comme [Shiaa, 2008] ou [Margaria, 2008]. Ces approches, axées sur le couple consommateur/producteur butent d'une part sur la formalisation de son besoin par le consommateur et d'autre part sur l'industrialisation (passage à l'échelle) de la conciliation entre les mots clés du besoin exprimé par le consommateur et les mots clés des interfaces logicielles des services proposés par le développeur/producteur (les uns étant dans une logique d'usage et les autres dans une logique technique), comme nous l'avons vu au chapitre précédent à propos des mécanismes Web Sémantique. Une perspective plus pertinente à moyen-terme est pour nous de laisser l'utilisateur traduire les besoins du consommateur, suivant le principe du self-service. En reprenant les termes de l'analyse fonctionnelle, cela revient à exposer à l'utilisateur un premier niveau de fonctions techniques des services et le laisser construire lui-même le service qui répondra à ses besoins en tant que consommateur. Cette vision est généralement désignée sous le nom d'UGS pour User Generated Services.

Dans cette perspective, l'utilisateur du service et le fournisseur du service sont deux parties-prenantes, deux acteurs du service qui collaborent pour réaliser un processus de service générant de la valeur pour le consommateur du service. Ceci s'applique bien aux services télécoms, qui sont des services où l'utilisateur participe activement. Dans un service sans participation de l'utilisateur, Alice dicterait un message pour Bob à un service de secrétariat qui trouverait les coordonnées de Bob et enverrait le message. Dans un service télécom, Alice cherche l'adresse de Bob dans son carnet d'adresses, écrit un email et l'envoie. L'utilisateur et le fournisseur de service collaborent au sein du système "service" pour fournir une valeur au consommateur. L'utilisateur et le consommateur sont généralement la même personne. Néanmoins ce n'est pas nécessairement le cas. Nous avons ainsi vu que le consommateur du service télécom pouvait être une entreprise et l'utilisateur un salarié de cette entreprise, par exemple dans un service de centre d'appels. Dans tous les cas, c'est la

compétence de l'utilisateur qui lui permet de traduire les besoins du consommateur dans les services auxquels il a accès. Dans cette perspective, la notion de base pour aboutir à une sémantique commune décrivant les services télécoms n'est plus celle de besoin, mais celle d'action. L'utilisateur et le fournisseur participent conjointement au rendu du service au consommateur en réalisant des actions métier, ces actions étant supportées par des solutions techniques. La figure suivante illustre ces relations.

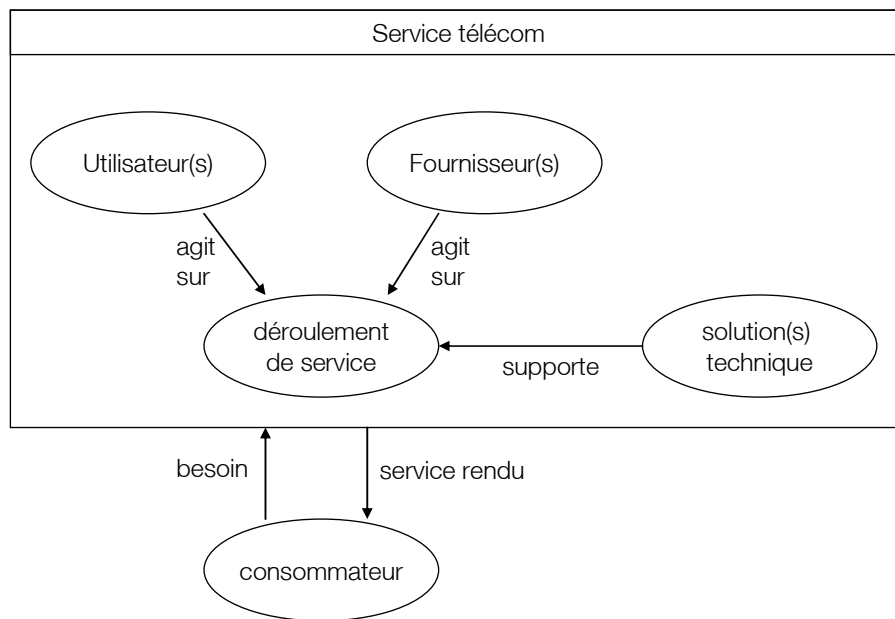


Figure 24 : Déroulement d'un service télécom

On peut être amené à considérer plusieurs utilisateurs ou plusieurs fournisseurs pour le même service. Ainsi, pour un service de téléphonie, on considérera l'utilisateur appelant, le fournisseur du service de téléphonie et l'utilisateur appelé. Les actions du fournisseur de service et celles de l'utilisateur ne sont pas par nature différentes. Par exemple, dans le cas des services de contenus autoproduits comme YouTube, l'utilisateur réalise certaines actions (par exemple la mise à disposition de vidéo) qui sont dévolues au fournisseur de service dans un service classique de vidéo à la demande (VOD). Dans les services dits web 2.0, il n'est d'ailleurs pas rare que la plupart des actions soient effectuées par l'utilisateur.

Le service rendu au consommateur peut être décrit par des fonctions de service, en référence à l'analyse fonctionnelle. Ces fonctions de services correspondent en effet à une action du produit pour répondre à un besoin du consommateur, le produit étant entendu ici comme le système de service dans son ensemble, en incluant l'utilisateur. Mais ce qui nous intéresse ici, pour construire notre sémantique de description des services, n'est pas le service

rendu au consommateur mais le déroulement du service à travers les actions de l'utilisateur et du fournisseur de service. Ces actions se rapprochent des fonctions techniques dans le vocabulaire de l'analyse fonctionnelle, car elles décrivent le fonctionnement interne du produit, toujours entendu comme le système de service incluant l'utilisateur.

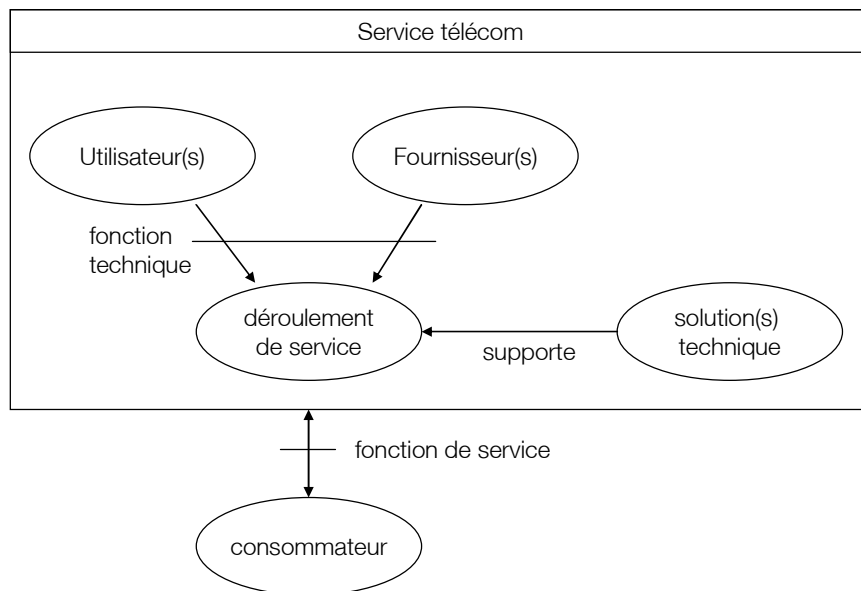


Figure 25 : Fonctions techniques et fonctions de service pour les services télécoms

Par exemple, pour une fonction de service de communication par messagerie, des fonctions techniques susceptibles d'assurer cette fonction de service seraient la recherche et la sélection d'un contact dans une liste de contacts, l'envoi d'un message, sa remise dans la boîte aux lettres du destinataire.

Dans le chapitre suivant, une part importante de notre travail consistera à établir ces fonctions techniques comme référentiel sémantique pour décrire les services télécoms. Nous définirons une fonction technique comme une capacité d'action, susceptible d'être mise en œuvre par les utilisateurs ou les fournisseurs. Le support du déroulement du service par une solution technique devra également être décrit afin de faciliter l'intégration dans un unique front-office. Nous verrons dans le chapitre suivant comment décrire ces solutions techniques à l'aide de motifs techniques communs aux différentes solutions.

Afin de construire une référence pour décrire les services télécoms, nous allons donc étudier d'une part le déroulement des services pour en déduire des fonctions techniques, et d'autre part étudier les motifs communs dans les solutions techniques. Nous nous appuyons pour cela sur les concepts de l'architecture d'entreprise, introduits au chapitre précédent. Nous

avons vu que la communauté de l'urbanisation des systèmes d'information considérait classiquement les systèmes d'information selon quatre points de vue: la vue métier, la vue fonctionnelle, la vue technique et la vue applicative. Dans notre cas, il ne s'agit pas d'étudier le système d'information d'une entreprise en partant du déroulement de ses processus métier internes, mais d'étudier les systèmes de service télécoms, en partant du déroulement des services. Néanmoins, les concepts de l'architecture d'entreprise nous fournissent une base solide pour construire notre sémantique de description des services selon différents points de vue. Nous allons ainsi construire un référentiel d'architecture pour les services télécoms, selon des principes proches des référentiels d'urbanisation d'entreprise ou des référentiels d'architecture d'entreprise (*enterprise reference architecture*). Nous nous proposons ainsi de décrire les systèmes de service télécom selon quatre vues :

- Une vue métier décrivant le déroulement d'un service tel que perçu par les parties-prenantes collaborant au rendu de ce service.
- Une vue fonctionnelle décrivant les fonctions techniques (au sens de l'analyse fonctionnelle) utilisées pour réaliser ce service. Nous verrons dans la section 2 du prochain chapitre comment les fonctions technique présentées ci-dessus nous aideront à identifier les invariants nécessaires à cette vue.
- Une vue technique décrivant les principaux rôles techniques et protocoles du service.
- Une vue applicative décrivant les applications réalisant ce service.

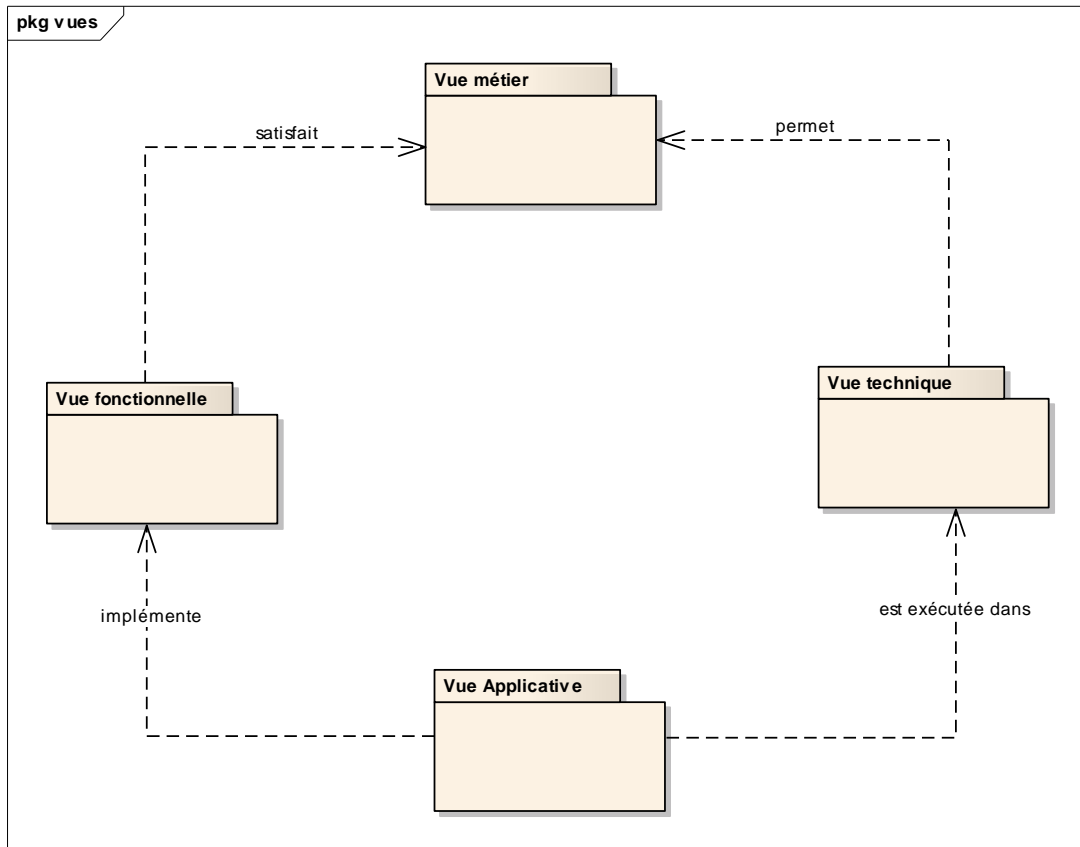


Figure 26 : Les quatre dimensions d'un service télécom

Nous verrons dans les chapitres suivants les concepts clés de ces vues et leurs relations, qui seront formalisés dans un méta-modèle.

Si ces quatre vues nous permettent de décrire un service télécom, la question de la sémantique de référence reste entière. Chaque service pourrait en effet être décrit sur ces vues avec un vocabulaire propre pour désigner ses fonctions ou ses rôles techniques. Il nous faut donc construire des vues de référence qui vont porter notre sémantique commune. Les vues fonctionnelles et techniques d'un service¹ devront ensuite décrire ce service avec les composants de ces vues de référence. Le prochain chapitre est consacré à la construction de ces vues de référence. La vue métier de référence contiendra le déroulement des principaux services de l'opérateur tels que perçus par les parties-prenantes de ces services. La vue fonctionnelle de référence listera les capacités d'action offertes par les services télécoms. La vue technique de référence décrira les principaux rôles techniques et protocolaires disponibles

pour implémenter ces services. Nous pourrions ensuite décrire l'architecture d'un service à travers une vue fonctionnelle de ce service (se basant sur la vue fonctionnelle de référence), une vue technique (se basant sur la vue technique de référence), et une vue applicative indiquant comment les fonctions de la vue fonctionnelle sont exécutées dans l'environnement d'exécution de la vue technique.

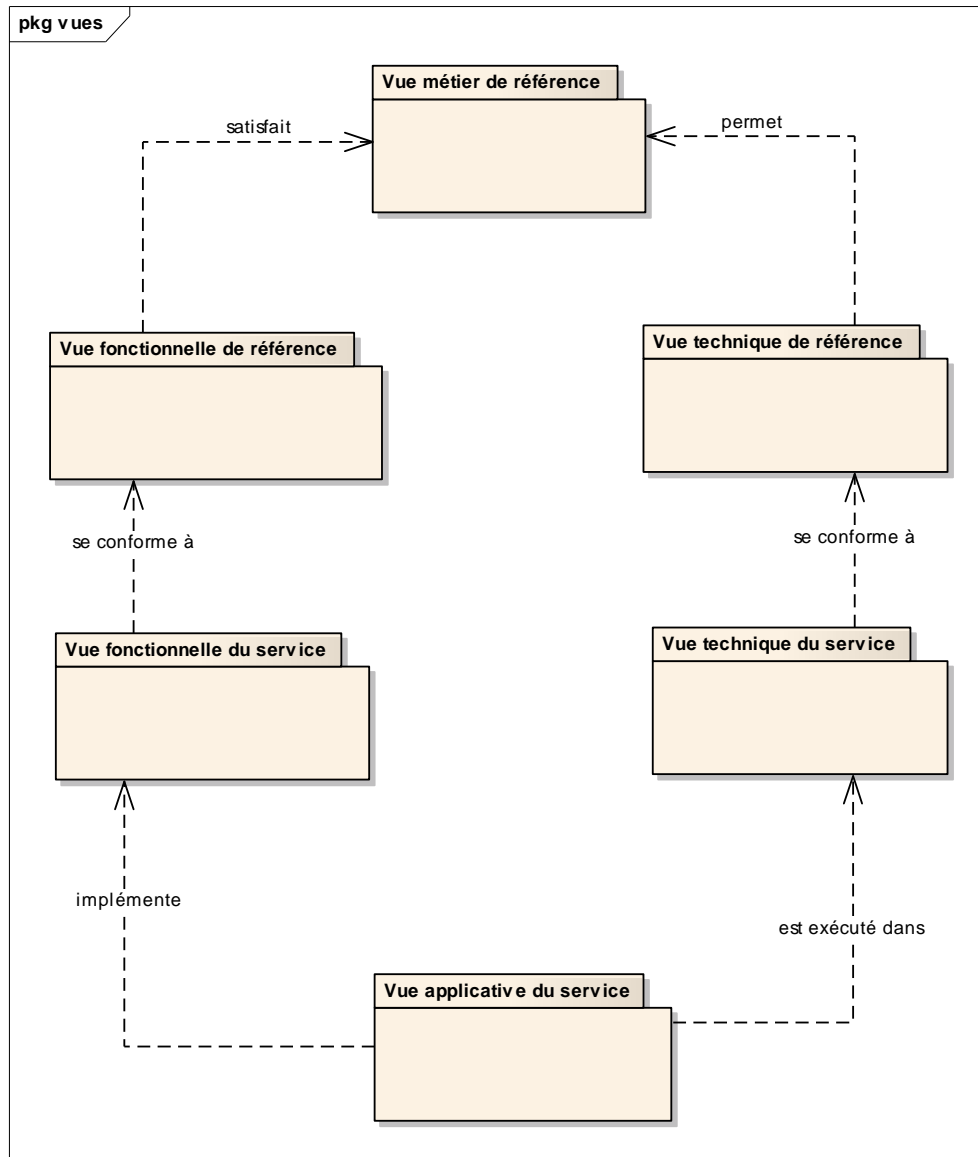


Figure 27 : Méthode de description de l'architecture d'un service télécom

¹ Nous verrons qu'une vue métier n'est pas nécessaire pour décrire chaque service. Et nous ne proposerons pas de sémantique de référence pour la vue applicative, chaque service y étant décrit avec son vocabulaire propre, MAIS en se référant aux descriptions normées des vues fonctionnelles et techniques.

Dans les deux chapitres suivants, nous verrons donc d'abord dans le chapitre III comment construire ces vues de référence, puis dans le chapitre IV comment les utiliser pour décrire les services télécoms, que ce soient des services simples, des enablers ou des services composés. A chaque étape, nous élaborerons un méta-modèle afin de formaliser les relations entre concepts mis en œuvre. Au chapitre III, nous proposerons un méta-modèle d'urbanisme de service pour construire le référentiel d'architecture des services télécoms. Au chapitre IV, nous introduirons un méta-modèle d'architecture de service pour décrire un service télécom à partir de ce référentiel. Le chapitre V sera consacré aux applications des résultats obtenus.

Notre travail ne se basera pas sur les outils issus du web sémantique tels les ontologies. Notre élaboration d'une sémantique de référence peut en effet être assimilée à la construction d'une ontologie, c'est à dire "*la spécification d'une conceptualisation d'un domaine de connaissance*", selon la définition précédemment citée de [Gruber, 1993]. Nous préférons néanmoins nous baser sur les outils conceptuels de l'architecture d'entreprise, beaucoup plus ciblés et structurés, présentant des contraintes syntaxiques assez fortes, à travers la méta-modélisation. Nous nous appuyerons ainsi sur une fondation éprouvée du domaine plutôt que de spécifier un dictionnaire ex nihilo. Nous verrons au chapitre V, que nos résultats pourront tout à fait servir de corpus de base à la constitution d'une ontologie de description des services télécoms. Comme évoqué au chapitre précédent, une ontologie fournit en effet des résultats d'autant plus pertinents que son corpus de base est cohérent. Les concepts que nous introduirons dans les deux méta-modèles mentionnés ci-dessus peuvent être vus comme des traits de construction pour aboutir à une sémantique de référence, traits pouvant être gommés dans des applications ultérieures.

Synthèse

Nous avons cherché au long de ce chapitre à identifier les éléments constituant un service de communication. Nous avons distingué entre le point de vue de la production et le point de vue de l'utilisation. Du point de vue de l'utilisation, un service est caractérisé par des fonctions de service, représentant la valeur du service pour son consommateur, et par des fonctions techniques, représentant les actions conjointes de l'utilisateur et du fournisseur pour réaliser ces fonctions de service. Les services de communication sont en effet de plus en plus des *self-services* dont l'utilisateur est impliqué dans le rendu, notamment dans la vision de la convergence de service où l'utilisateur navigue au sein d'un environnement unifié de services. Du point de vue de la production, un service est caractérisé par une solution technique supportant ces fonctions, ainsi que par une découpe entre ce qui est perçu par l'utilisateur (*front-office*) et ce qui ne l'est pas (*back-office*), une découpe bien pensée étant essentielle pour le *self-service* et donc pour réaliser la vision de convergence de service. Les fonctionnalités présentées dans un *front-office* correspondront logiquement à des fonctions techniques du point de vue de l'utilisation, la réalisation de ces fonctions étant ensuite portée par le *back-office*.

Nous proposons de rapprocher ces éléments constitutifs d'un service des quatre points de vue de l'urbanisation des systèmes d'information : la vue métier pour les fonctions de service, la vue fonctionnelle pour les fonctions techniques, la vue technique pour les solutions techniques, la vue applicative pour le lien entre solution technique et fonctions techniques et en particulier pour la répartition des fonctions entre *front-office* et *back-office*.

Nous devons maintenant identifier des invariants, une sémantique de référence pour construire une vue métier de référence, une vue fonctionnelle de référence et une vue technique de référence. Cela sera l'objet du chapitre suivant, où nous allons proposer une méthode de construction de ces invariants que nous illustrerons avec quelques exemples. Nous pourrons alors, dans le chapitre IV, décrire de façon rigoureuse l'architecture d'un service de communication à travers une vue fonctionnelle du service, une vue technique du service et une vue applicative du service reliant les deux précédentes. Cela nous permettra notamment de comparer facilement un service avec un autre et d'identifier des composants réutilisables, comme nous le verrons dans le chapitre V.

Chapitre III Construire un référentiel d'architecture pour les services télécoms

Au cours du chapitre précédent, nous avons insisté sur les multiples dimensions d'un service télécom. Service rendu par des parties-prenantes suivant un déroulement défini, mais aussi solution technique supportant les actions de ces parties-prenantes. Nous avons vu également qu'il nous manquait une sémantique de référence pour décrire formellement le déroulement des services télécoms, et que ce manque rendait confuses des problématiques architecturales comme la construction d'enablers ou la composition de services. A partir des bonnes pratiques de l'urbanisation des systèmes d'information, nous avons proposé de décrire les services télécoms, à la fois dans leur dimension fonctionnelle et dans leur dimension technique, selon quatre vues. Nous allons maintenant nous confronter à cette question : Comment construire cette sémantique de référence ? Autrement dit, à partir de quels invariants peut on décrire une architecture de service télécom ? La réponse se fera en trois temps. Dans une première section, il s'agira d'étudier les actions réalisées au cours d'un service (existant ou cible) par les acteurs collaborant au rendu de ce service et d'en déduire une vue métier de référence. Dans une deuxième section, nous caractériserons le fonctionnel des services par les capacités d'actions fournies par les services télécoms pour construire une vue fonctionnelle de référence. Nous verrons dans une troisième section comment modéliser les solutions techniques permettant l'implémentation de ces services afin d'identifier une vue technique de référence. Chacune de ces trois sections suivra un déroulement similaire : introduction des notions utilisées, présentation d'une méthode de travail, puis évaluation et application de cette méthode aux services télécoms. Au cours de chaque section, nous formaliserons les concepts introduits et les liens qui existent entre eux par un méta-modèle d'urbanisme, support pour la construction de notre sémantique de référence. La vue applicative ne sera pas abordée dans ce chapitre, car elle ne possède pas à proprement parler de sémantique de référence, étant décrite à partir des vues fonctionnelles et techniques. Nous verrons dans le chapitre suivant comment cette sémantique de référence nous aidera à clarifier nos problématiques architecturales en introduisant alors la vue applicative.

III.1 Une vue métier de référence pour les services télécoms

Qu'est ce qu'un service télécom vu des acteurs qui participent à son rendu ? D'une façon générale, un service télécom donne à ses utilisateurs la capacité de réaliser des actions de communication. Les services d'un opérateur télécom fournissent ainsi la capacité de téléphoner, d'échanger des messages, de regarder une chaîne de télévision, etc. Du point de vue d'un utilisateur donné, le service est déclenché par des actions de sa part, comme par exemple composer le numéro de téléphone ou sélectionner une chaîne de télévision. Les interactions avec l'utilisateur sont une part essentielle des services de communication, le service ne pouvant pas être rendu sans cette interaction. Selon la classification développée dans [Pinhanez, 2008], les services télécoms peuvent donc être considérés comme *customer-intensive*, c'est-à-dire nécessitant pour leur rendu des entrées régulières de leurs utilisateurs, en suivant la terminologie développée dans le chapitre précédent. Notre proposition clé dans la section suivante sera de décrire le fonctionnel d'un service télécom à travers les actions qu'il permet à ses utilisateurs de réaliser. Dans cette section, nous allons nous attacher à justifier l'existence d'une description stable de ces actions à travers une vue métier pour les services télécoms. Nous rechercherons une médiation entre une description haut-niveau de la finalité d'un service et une description fine des besoins d'un utilisateur précis. Cette approche est compatible avec la description d'un service par les intentions de ses utilisateurs, dans la mesure où il y a un lien fort entre la description d'une intention d'agir et la description d'une action. La notion d'action nous permettra néanmoins plus de précision qu'une notion générique d'intention, notamment pour l'enchaînement des actions entre plusieurs acteurs.

III.1.1 Comment définir les activités de la vue métier ?

Pour les systèmes d'information d'entreprise, la vue métier est usuellement constituée des processus métier de l'entreprise. Par exemple, le processus d'achat ou le processus de vente. Comme indiqué dans [Simoni, 2007], un processus peut être défini comme une chaîne d'activités reliées pour réaliser un objectif métier. Chaque activité est un ensemble de tâches, manuelles ou automatiques, dont la réalisation nécessite des ressources et produit une valeur ajoutée. Quelles sont donc les activités dans une vue métier d'un service télécom ? Dans la logique du chapitre précédent, ce sont les actions, réalisées par les parties-prenantes du service. Pour développer cette proposition, nous sommes immédiatement confrontés à une

difficulté : quelles sont précisément les actions à considérer ? Avec comme corollaire : comment définir l'utilisateur qui agit ?

On peut en effet considérer des actions spécifiques à une personne et correspondant chez elles à un besoin précis, par exemple téléphoner à sa grand-mère pour lui souhaiter son anniversaire. Téléphoner à son voisin sera alors une action distincte. Dans ce cas, une action est indissociable de la personne réelle qui la réalise : c'est monsieur untel qui téléphone à sa grand-mère. L'utilisateur est alors une personne donnée. A ce niveau, les études concernant les services télécoms seront des études sociologiques. Ces recherches s'intéressent par des enquêtes à des actions de personnes réelles (par exemple une femme, de 36 ans, habitant en région parisienne, ayant suivi des études supérieures, qui téléphone depuis son domicile, le matin avant de partir au travail, pour appeler un plombier parce que son robinet fuit), puis regroupent ces personnes en catégories pour en tirer statistiquement des tendances plus générales.

On peut également considérer les actions physiques réalisées dans le cadre de l'usage d'un service - par exemple décrocher le combiné, appuyer sur des touches pour composer un numéro, taper du texte, cliquer sur un bouton, ainsi que les représentations mentales permettant de comprendre le sens de ces actions. L'utilisateur est alors défini uniquement par ses capacités physiques et mentales. A ce niveau, les études concernant les services télécoms seront des études de conception et d'ergonomie des interfaces de services. Elles conduisent à segmenter les utilisateurs en fonction de leurs capacités motrices, psychomotrices, cognitives. L'étude de modèles mentaux visera par exemple à catégoriser les représentations que se font diverses personnes d'un service.

Mais notre travail sur les architectures de service télécoms, nous conduit vers une autre vue : des actions décontextualisées qui correspondent à la finalité technique objective du service. Ni celle de souhaiter son anniversaire à sa grand-mère, ou d'appuyer sur des touches, mais celle de téléphoner, c'est-à-dire de demander une communication, d'accepter une demande de communication, de pouvoir se parler. L'utilisateur qui réalise ces actions peut alors être vu comme un rôle, défini uniquement par sa capacité à manipuler (créer, émettre, recevoir...) des informations conformément aux possibilités offertes par le service. L'utilisateur dont nous parlons ici n'est donc pas une personne réelle, mais le rôle joué par une personne lorsqu'elle utilise un service (par exemple le rôle d'appelant ou le rôle d'appelé dans

un service téléphonique), comme défendu par exemple dans [Lamb, 2003]. Notons que nous utiliserons par la suite le plus souvent le terme UML d'acteur plutôt que celui de rôle.

Cette conception de l'utilisateur prête toutefois le flanc à la critique formulée par la sociologie des usages, par exemple dans [Nygaard, 1986], de ne voir l'utilisateur que comme un sous-système technique, comme une représentation générique que l'ingénieur concepteur du service se fait des utilisateurs de son service [Flichy, 2008]. Il semble néanmoins, comme nous l'esquisserons ci-dessous, que le langage fournit un cadre qui va bien au-delà des représentations du concepteur du service, à condition de rester dans le langage commun aux concepteurs et aux utilisateurs.

Nous venons de cerner le niveau d'action que nous souhaitons considérer pour la description des services télécoms d'une façon intuitive. Un cadre théorique précis nous emmènerait sur des terrains éloignés des télécommunications en mobilisant des travaux sur les théories de l'action et du langage. Afin de ne point diverger excessivement, nous allons tout d'abord éclairer notre discussion précédente en comparant un service télécom avec un outil, puis nous esquisserons quelques considérations théoriques dont le but sera davantage de montrer la possibilité d'une justification rigoureuse que de la réaliser.

On parle communément d'outils de communication pour désigner des services télécoms usuels, comme le courrier électronique, la messagerie instantanée, mais aussi le téléphone ou la télévision. Un outil fournit la capacité de réaliser une action simple. Le Robert définit ainsi un outil comme un "*objet fabriqué qui sert à agir sur la matière, à faire un travail*". Considérons par exemple un outil communément utilisé, un outil *mass market* : le tournevis. Cet outil permet de réaliser l'action élémentaire de visser une vis. Cette action peut être incluse dans une action plus vaste, par exemple monter un meuble. Elle peut également être elle-même décomposée en actions physiques plus élémentaires : appuyer le tournevis contre la vis, tourner le poignet dans le sens des aiguilles d'une montre... Toutefois, ces actions plus élémentaires peuvent être vues comme contingentes. On peut ainsi visser également avec une visseuse électrique, les actions physiques étant alors différentes. L'action de visser peut pourtant être considérée en dehors de tout contexte de réalisation, de façon décontextualisée. De plus, l'action de visser nous désigne implicitement un acteur : celui qui réalise l'action de visser. Cet acteur n'est pas une personne réelle, mais une abstraction, un rôle, qui nous permet de parler de l'action de visser en général, et non de l'action d'une

personne précise. Les actions que nous souhaitons considérer pour la description des services télécoms sont intuitivement du même type que les actions réalisables avec un outil.

Les travaux théoriques sur la sémantique de l'action insistent sur la validité de diverses descriptions d'une unique action, comme nous l'avons intuitivement envisagé précédemment. Le philosophe français Paul Ricœur discute ainsi dans [Ricœur, 1990] les travaux d'Elisabeth Anscombe, célèbre pour ses analyses des notions d'action et d'intention :

"Je rappelle l'exemple qui a rendu fameuse l'analyse d'E. Anscombe : Un homme pompe de l'eau dans la citerne qui alimente une maison en eau potable. Quelqu'un a trouvé le moyen de contaminer systématiquement la source au moyen d'un poison lent dont les effets se font sentir quand il est trop tard pour les soigner. La maison est régulièrement habitée par un petit groupe d'agitateurs qui agissent pour le compte de meneurs politiques qui sont à la tête d'un vaste Etat. (...) L'homme qui a contaminé la source (...) a mis au courant de son calcul, en même temps que de la nature du poison l'homme qui fait marcher la pompe. La mort des habitants de la maison aura bien entendu toutes sortes d'autres effets, par exemple : un certain nombre de personnes inconnues à ces hommes recevront des legs dont ils ne connaissent pas l'origine(...) La question posée par cet exemple est la suivante : qu'est-ce que l'homme est en train de faire ? Quelle est la description de son action ? Réponse : la question admet autant de réponses que permet l'échelonnement des « en vue de » ; toutes les descriptions sont également valables. En particulier, on peut aussi bien dénommer l'action en fonction de la première chose qu'on fait ou en fonction du dernier résultat visé. "

Nous passons ainsi du besoin d'identifier les actions des parties prenantes d'un service au besoin de décrire leurs actions. Entre téléphoner, téléphoner à sa grand-mère, ou décrocher le combiné en appuyant sur des touches, l'action peut être considérée comme identique, mais la description de cette action diffère.

Et la description d'une action nécessite de la nommer. On retrouve cet impératif de nommage dans les travaux théoriques sur le langage, comme par exemple chez le philosophe du langage John Searle (connu dans la communauté de l'intelligence artificielle pour son paradoxe de la chambre chinoise) dans [Searle, 2005] :

"My dog has very good vision, indeed much better than mine. But I can still see things he cannot see. We can both see, for example, a man crossing a line carrying a ball. But I can see the man score a touchdown and the dog cannot (...). To see a touchdown scored he would have to be able to represent what is happening as the scoring of a touchdown, and without language he cannot do that."

C'est le langage qui nous permet de décrire les actions, mais aussi de les penser - au-delà des actions physiques dont la perception est sensorielle. Dans les langues européennes, ce

nommage est dévolu aux verbes dits d'action. Le sujet du verbe d'action est l'acteur (ou l'agent) et le complément est l'objet sur lequel l'action est exercée. Dans l'action de téléphoner à sa grand-mère pour lui souhaiter son anniversaire et dans l'action de téléphoner à son plombier parce que son robinet fuit, le verbe d'action est commun : il s'agit dans les deux cas de téléphoner, même si le but et le contexte diffèrent.

Ces verbes d'actions ont un aspect définitionnel. Le langage est en effet une structure abstraite, imposant des contraintes interprétatives et impliquant des présupposés logiques [Kaufmann, 2005]. Un verbe d'action comme celui de marquer un essai (*to score a touchdown*), pour reprendre l'exemple de Searle, fait ainsi référence à un ensemble de règles constitutives : poser les deux pieds dans la zone d'en-but adverse avec le ballon en main lors d'une séquence de jeu valide. On parle parfois de jeu de langage pour rendre compte de cet aspect définitionnel, indépendamment des actualisations empiriques particulières par des acteurs particuliers. Ce passage par le langage nous permet ainsi d'explicitier une description d'actions décontextualisées, où l'on ne s'intéresse pas à la réalisation d'une action par une personne particulière, mais à une description impersonnelle de ce qu'implique sémantiquement l'usage d'un verbe d'action et d'un complément d'objet direct. Nous serons donc par la suite attentifs aux mots utilisés pour décrire nos actions dans le cadre des services télécoms, et l'on observera que ces mots sont communs à toutes les entreprises du secteur des télécoms, comme par exemple envoyer un email, regarder une chaîne de télévision ou envoyer un SMS. En effet, en reprenant la terminologie de [Rust, 2006], les services télécoms sont largement standardisés (non pas dans le sens de la validation par un organisme de standardisation, mais dans le sens où ces services sont fournis à l'identique à des dizaines de millions de personnes).

Ceci nous donne quelques éléments pour préciser le type d'actions à considérer dans notre cas. Ce sont des actions indépendantes de tout contexte de réalisation (comme la personne réalisant l'action, son intérêt à agir, le fournisseur du service, le moyen physique d'accès au service...). Elles sont décrites par des verbes d'action qui sont communs à tous les contextes de réalisation. Elles impliquent des présupposés logiques que nous développerons par la suite sous forme de processus.

Une autre approche, proposée dans certains travaux pour la description de services, par exemple [Rolland, 2009], serait de se fonder sur la notion d'intention. Cette notion peut être rapprochée de celle d'action, comme pointé par E. Anscombe qui affirme dans son ouvrage [Anscombe, 1957] :

"« Roughly speaking, a man intends to do what he does. » But of course that is very roughly speaking. It is right to formulate it, however, as an antidote against the absurd thesis which is sometimes maintained: that a man's intended action is only described by describing his objective" (§25)

En analysant les travaux menés sur ces notions d'intention et d'action, P. Ricœur en vient même à affirmer dans [Ricœur, 1990] :

"L'implication mutuelle entre question quoi ? et pourquoi ? joue en plein et dans les deux sens : de la description (de l'action) vers l'explication (de l'action), mais aussi à rebours, de l'explication vers la description". (p.90)

Et l'on retrouve ce lien fort entre intention et description chez beaucoup de philosophes de l'action, notamment Donald Davidson dans son essai de référence "Actions, Reasons and Causes" [Davidson, 1963].

Nous privilégierons ici toutefois la notion d'action à celle d'intention. Elle est de notre point de vue un point d'entrée plus efficient dans le cadre de ce travail, car plus proche des concepts de la vue métier de l'urbanisation des systèmes d'information, comme le concept d'activité métier. Elle permet d'une part de se focaliser sur les interactions entre service et utilisateur, et d'autre part de prendre en compte les enchaînements d'actions sous forme de processus, en explicitant les différents rôles joués par les utilisateurs. Il est ainsi plus simple de considérer que l'action d'un acteur d'accepter un appel téléphonique est précédée par celle d'un autre acteur de passer un appel plutôt que de considérer que l'intention de l'appelant de passer un appel doit précéder l'intention de l'appelé de l'accepter.

Après cette digression visant à mieux appréhender la nature des concepts manipulés, nous allons maintenant étudier comment identifier ces actions dans les services télécoms.

III.1.2 Comment construire une vue métier de référence ?

Nous avons insisté dans la section précédente sur les actions des utilisateurs d'un service télécom, décrites indépendamment de tout contexte particulier de réalisation à travers des verbes d'action et des compléments d'objets directs. Les actions du fournisseur du service peuvent être décrites semblablement. Par exemple, le fournisseur du service email va délivrer les emails à leurs destinataires, le fournisseur de service télévisuel va distribuer des chaînes de télévision. Notre proposition est que, selon le point de vue de l'utilisation, le déroulement d'un service peut être représenté comme un enchaînement d'actions réalisées par les acteurs du système de service, ces acteurs incluant les rôles du ou des utilisateurs et ceux du ou des fournisseurs de service. Illustrons cette proposition avec un service en dehors du monde

télécom. Dans un service postal, l'expéditeur poste une lettre et le fournisseur de service postal l'achemine vers le destinataire. Nous avons ainsi décomposé un déroulement typique du service postal en deux actions. Cette décomposition est intuitivement évidente, mais elle n'est pas immédiatement justifiable. La difficulté réside en effet dans le critère de choix des acteurs et des actions. On aurait pu ajouter par exemple un acteur qui écrit la lettre, un acteur qui colle le timbre, un acteur qui trie le courrier, ou un acteur qui le transporte en train. Nous pouvons toutefois identifier des critères objectifs conduisant à ce choix. Tout d'abord, les actions sont celles perçues lors de l'utilisation du service. Les actions du fournisseur du service déclenchées par une action de l'utilisateur seront donc perçues comme une seule action qui correspond à ce que l'utilisateur demande au fournisseur comme service. Des actions comme trier le courrier ou le transporter ne sont donc pas prises en compte, mais uniquement l'action d'acheminer le courrier, qui inclut implicitement le triage du courrier et son transport. L'enjeu est en effet de dissocier une vue "externe" du service qui manifeste sa valeur ajoutée fonctionnelle d'une vue interne, habituellement privilégiée pour l'architecture des services, qui correspond aux processus internes du fournisseur du service et est donc contingente à une réalisation technique particulière. Cette vue externe sera commune même si les réalisations techniques diffèrent, par exemple entre un service de téléphonie IMS et un service de téléphonie peer-to-peer. Ensuite, les actions prises en compte sont les actions nécessaires, quel que soit le contexte particulier de réalisation d'un service. Des actions comme coller un timbre ou écrire une lettre n'apparaissent donc pas car elles sont contingentes à un certain mode de réalisation. Ces critères sont tout à fait cohérents avec les critères de définition d'une activité métier dans un processus métier.

L'enchaînement des actions est un enchaînement typique, commun à tous les contextes particuliers de réalisation. Réciproquement, on peut considérer cet enchaînement d'actions comme constitutif du service. Ainsi, lorsqu'un acteur poste une lettre, puis qu'un acteur l'achemine, cela peut être vu comme sémantiquement constitutif d'un service postal. Quel que soit le contexte, quel que soit le vocabulaire employé par le fournisseur du service, un tel service sera perçu comme un service postal. Il s'agit bien sûr d'une vision a posteriori, qui ne reflète absolument pas la façon dont les éléments caractérisant un service évoluent au cours du temps, mais vise uniquement à indiquer le sens actuel du service. On peut considérer cet enchaînement comme la définition formelle de l'utilisation d'un service postal (cet exemple est bien sûr présenté à titre illustratif et nous n'avons pas recherché sa complétude, des aspects essentiels comme la validité de l'affranchissement ou la validité de l'adresse du destinataire

n'ont pas été pris en compte ; nous verrons par la suite comment de telles contraintes peuvent être modélisées dans le cas des services télécoms). Cette définition formelle peut être vue comme une connaissance tacite minimale partagée entre tout utilisateur du service et son fournisseur. Cette connaissance n'est usuellement pas codifiée. Il n'y a pas de mode d'emploi explicite d'un service postal, pas plus qu'il n'y a de mode d'emploi explicite d'un service de coiffure – exemple que nous avons discuté dans [2009-1] – ou d'une consultation médicale. Comme souligné dans [Chesbrough, 2006], l'explicitation d'une telle connaissance tacite est un aspect important de l'informatisation du secteur des services.

Utilisateurs ou fournisseurs de service peuvent donc être représentés de façon générique par des acteurs, chaque acteur étant responsable de la réalisation d'une action ou d'un ensemble d'actions lorsque celles-ci sont nécessairement réalisées par le même acteur. Le déroulement d'un service peut alors être vu comme un processus, où le domaine modélisé n'est plus les activités de l'entreprise, mais le système de services télécoms. La définition d'un processus métier vue au premier chapitre peut être modifiée de la façon suivante (les modifications ne sont pas en italique) pour définir un processus de service :

"Un processus de service est une séquence d'actions, réalisée par les parties-prenantes du service, qui produit un résultat dont la valeur est perceptible et mesurable pour chacune de ces parties-prenantes."(p.11)

Comme nous l'avons évoqué, ces parties-prenantes correspondent aux collaborateurs métier des méthodes d'analyse des processus métier [Simonin, 2009] :

"Un Collaborateur métier représente un rôle ou un ensemble de rôles joué dans la réalisation des Processus métier par un individu ou une unité organisationnelle, interne au domaine modélisé." (p.9)

Selon cette terminologie, l'utilisateur sera un collaborateur métier, tout comme le fournisseur de service.

Nous utiliserons les notations UML préconisées dans le cadre de l'analyse de processus métier et représenterons le déroulement d'un service par un diagramme d'activités, les acteurs par des couloirs (*swimlane*) dans ces diagrammes et les actions par des activités UML. Concernant la terminologie, nous préférons parler de "service" plutôt que de "métier" (par exemple de processus de service plutôt que de processus métier), afin de souligner que nous ne prenons pas le point de vue de la production des services (qui est le métier des entreprises de service), mais bien celui de leur utilisation.

Revenons à notre exemple de service postal. Les actions de service s'enchaînent les unes après les autres, dans un ordre défini. Ainsi, l'action d'acheminer la lettre est nécessairement précédée de l'action de poster la lettre : l'action d'acheminer une lettre nécessite une lettre postée. Ceci se traduit logiquement par une dépendance de l'action "acheminer une lettre" envers l'action "poster une lettre". On remarquera d'ores et déjà que la dépendance est bien depuis l'action d'acheminer la lettre vers l'action de poster la lettre et non dans le sens inverse : une lettre acheminée a toujours été d'abord postée. Cet enchaînement d'actions peut assez naturellement être représenté par un diagramme d'activité UML, comme illustré par le schéma ci-dessous.

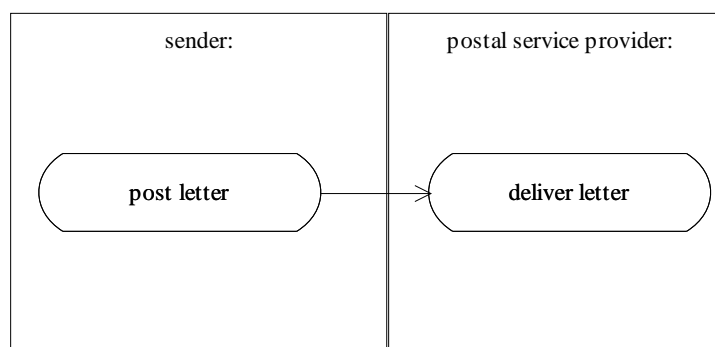


Figure 28 : Processus d'un service postal

La délimitation des responsabilités entre les acteurs apparaît immédiatement. L'acteur "sender" est responsable de l'action "post letter" alors que l'acteur "postal service provider" est responsable de l'action "deliver letter". A la différence des modélisations usuelles de processus métier de la vue métier des systèmes d'information, l'utilisateur du service (c'est-à-dire l'expéditeur *sender*) apparaît dans le processus non comme déclencheur du processus, mais comme collaborateur, responsable de l'accomplissement de certaines actions. Alors qu'une modélisation orientée vers les processus métier internes d'un fournisseur de service postal ferait apparaître le postage d'une lettre comme un événement déclenchant le processus, ce postage par l'expéditeur devient, dans notre perspective service, une action "post letter".

Les actions constitutives des services télécoms sont des actions de manipulation d'information (sélection, transmission, affichage...). Ceci découle naturellement de la définition des télécommunications comme "ensemble des procédés de transmission d'informations à distance par tout moyen électrique, radioélectrique, optique ou

électromagnétique" (Le Robert). Nous pouvons déduire la nature de ces informations à partir des objets manipulés dans les actions considérées. Si nous continuons notre exemple du service postal (qui est un service de transmission d'information à distance, par des moyens manuels et mécaniques), dans l'action de poster une lettre, l'objet manipulé est "une lettre". Grammaticalement, l'information manipulée dans une action sera le complément d'objet direct du groupe verbal décrivant l'action. Chaque action crée ou modifie des objets nécessaires au service. L'action de poster une lettre produit ainsi une lettre postée. Une action peut également nécessiter un objet produit par une autre action. Par exemple, l'acheminement d'une lettre nécessite une lettre postée. Il est intéressant de constater que la connaissance de ces objets est tacitement partagée entre les différents acteurs (utilisateur, fournisseur) d'un service ; et leur explicitation est tout l'enjeu des processus de service. De plus, ces objets peuvent être décrits par un état particulier. Par exemple, une lettre peut être dans l'état écrite, dans l'état postée, dans l'état acheminée ou dans l'état lue.

Cet objet peut être rapproché de la notion d'entité métier définie dans les méthodes d'analyse des processus métier :

"Une Entité métier est la représentation (image dans le système d'information) d'un concept manipulé par l'entreprise dans son activité. C'est un objet produit ou utilisé par un ou plusieurs Processus métier, et donc manipulé par les Collaborateurs métier. C'est une notion significative du métier, directement compréhensible et partageable par les différents acteurs individuels du domaine modélisé." (p.8)

"Une Activité métier est (...) définie par les informations (Entités métier) qu'elle utilise en entrée et qu'elle produit en sortie" (p.8)

Suivant notre terminologie, nous les nommerons des entités de service. En adaptant la définition ci-dessus à notre contexte, une entité de service peut être définie de la façon suivante (les modifications ne sont pas en italique).

"Une Entité de service est la représentation (image dans le système d'information) d'un concept manipulé par les parties-prenantes d'un service dans leurs actions. C'est un objet produit ou utilisé par un ou plusieurs Processus de service, et donc manipulé par les parties-prenantes du service. C'est une notion significative du service, directement compréhensible et partageable par les différents acteurs individuels du domaine modélisé."

Ces entités de service peuvent être ajoutées dans le diagramme d'activité UML vu précédemment. Une flèche d'une action vers une entité de service signifie que l'action produit

cette entité de service, et une flèche d'une entité de service vers une action signifie que l'action utilise cette entité de service. L'état d'une entité de service une fois l'action terminée est représenté entre crochets.

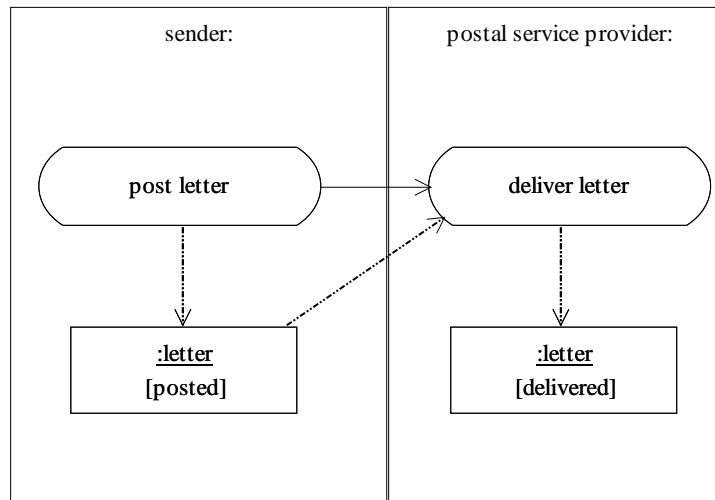


Figure 29 : Processus et entités de service

Les liens entre entités de service peuvent quant-à eux être représentés dans un diagramme de classe UML, appelé usuellement modèle d'information dans les méthodes d'architecture d'entreprise. Les liens entre entités de service sont des liens structurels, indépendants de l'état de l'entité. La figure suivante illustre ainsi le lien entre lettre et timbre.



Figure 30 : Relations entre entités de service

Nous avons ainsi formalisé sommairement une vue externe du déroulement d'un service postal. Cette formalisation en enchaînement d'actions présente l'avantage d'être facilement compréhensible par tous les intervenants impliqués dans le service. Elle peut notamment être validée par une division marketing comme cible en termes de comportement d'un service vu de l'utilisateur. Ceci rejoint en effet des pratiques marketing comme la

définition de parcours utilisateur (*user journey*), illustrée dans [Dubberly, 2008]. Cela rejoint également les méthodes de conception de services (au sens ergonomie/usage et non au sens applicatif), comme par exemple la définition d'un service comme enchaînement de tâches [Zeithaml, 1985] ou la représentation en cercle autour de l'utilisateur des séquences d'actions nécessaires [Cai, 2007]. De plus, l'identification des acteurs du service et des liens entre ces acteurs peut être corrélée avec la chaîne de valeur du service et son modèle de rôle (*business model*). Même si les relations considérées entre les acteurs ne sont pas les liens contractuels ou les flux financiers, il y a souvent recoupement entre relation contractuelle et relation d'utilisation.

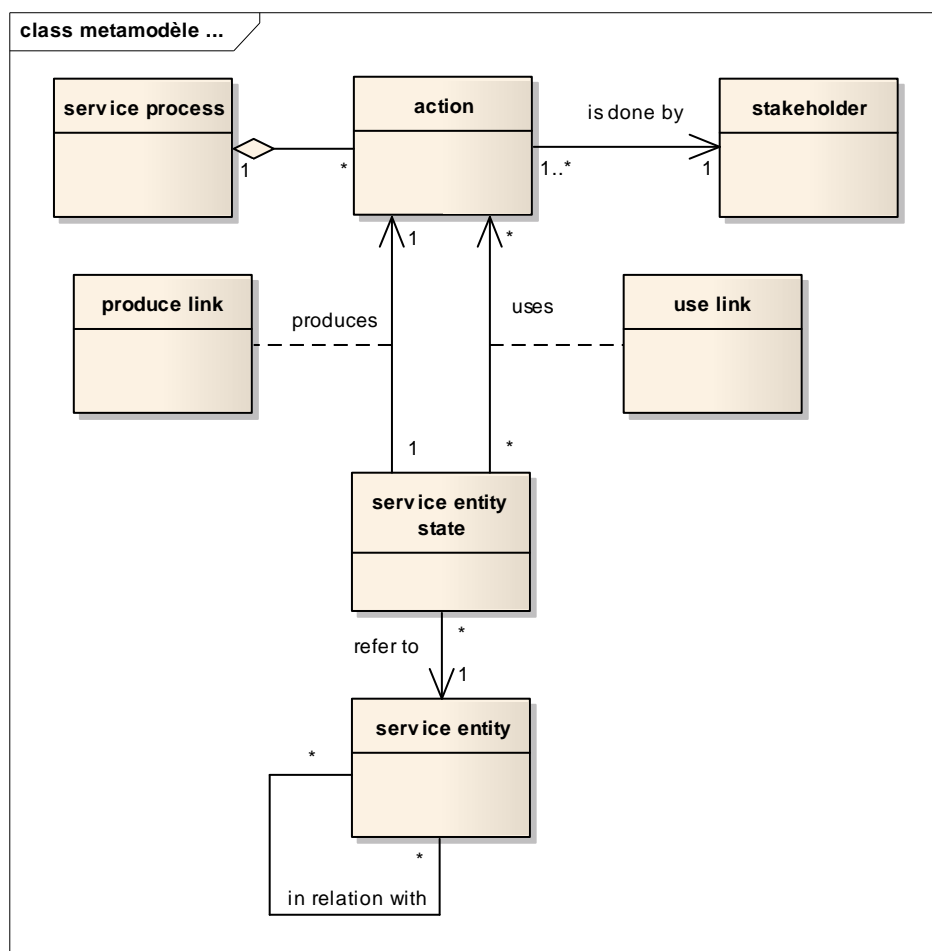


Figure 31 : Méta-modèle de la vue métier de référence

Nous pouvons maintenant synthétiser les notions que nous avons utilisées et les relations entre elles dans un méta-modèle. Une action est réalisée dans le cadre d'un processus de service (*service process*). Elle est effectuée par une partie-prenante du service (*stakeholder*). Elle produit et utilise des entités de service (*service entity*) dans un état

particulier (*service entity state*), à travers les classes d'association *produce link* et *use link* (l'intérêt d'utiliser des classes d'association apparaîtra dans la suite du méta-modèle). Les entités de service peuvent être en relation les unes avec les autres.

III.1.3 Evaluation et résultats

Transposons maintenant aux services télécoms. Arrive immédiatement une difficulté : les services télécoms sont habituellement considérés par les ingénieurs directement sous un angle technique ou applicatif. Les processus de service ou les modèles d'entités ne sont pas construits formellement, même s'ils sont contenus plus ou moins explicitement dans des expressions de besoin, émises par exemple par une division marketing. Nous allons donc les formaliser, en s'appuyant pour cela sur le déroulement des services. Pour les services spécifiés dans des protocoles standardisés, comme SIP pour les services conversationnels, nous pouvons toutefois débusquer des processus de service cachés dans les standards protocolaires. Les acteurs du service et leurs actions peuvent en effet y apparaître en filigrane, par exemple sous les traits d'Alice, de Bob, de Carol, prénoms communément utilisés dans les standards IETF pour désigner les utilisateurs. Le RFC 3261 spécifiant le protocole SIP contient ainsi 152 occurrences du mot Alice, 198 occurrences du mot Bob et 41 occurrences du mot Carol. On peut ainsi y lire :

"Alice calls Bob using his SIP identity, a type of Uniform Resource Identifier (URI) called a SIP URI" (p.11)

Ou encore :

"The 200 (OK) contains a message body with the SDP media description of the type of session that Bob is willing to establish with Alice" (p.15)

Nous voyons ainsi dans le texte même du RFC les acteurs du service (Alice l'appelante et Bob l'appelé), leurs actions (Alice appelle Bob) et les informations qu'ils manipulent (une SIP URI qui correspond au numéro de téléphone, un type de session média SDP à établir entre eux).

Ce travail a été réalisé ci-après pour les services télécoms les plus emblématiques. Une telle modélisation en processus de service permet d'explicitier et de remonter au niveau du processus des concepts qui peuvent sembler techniques (comme SIP URI ou session média), mais qui sont en fait des entités essentielles pour l'utilisation du service, dont la connaissance est partagée entre utilisateurs et fournisseurs du service. Des groupes non techniques comme

une division marketing sont alors mieux à même de comprendre et de valider le déroulement d'un service. A titre illustratif, nous considérerons donc ici les processus de service télécoms suivants, correspondant chacun à une fonction de service (au sens de l'analyse fonctionnelle), c'est-à-dire la réponse à un type de besoin des consommateurs de services télécoms :

- Envoyer et consulter des messages
- Effectuer un appel téléphonique
- Choisir et jouer un contenu multimédia (une vidéo ou une musique)
- Accéder à un service souscrit (se connecter, *to sign-in*)

Les événements déclencheurs des activités du processus sont représentés sous forme de signal (⊗). D'autres processus pourraient également être décrits comme la publication de son contexte (localisation, présence) et la consultation de celui d'un tiers, ou bien la consultation d'un calendrier et la prise de rendez-vous, ou encore le stockage de fichiers en ligne. Pour ne pas alourdir notre propos, nous nous limiterons aux quatre processus indiqués.

L'envoi et la consultation de messages sont décrits par le processus de service suivant.

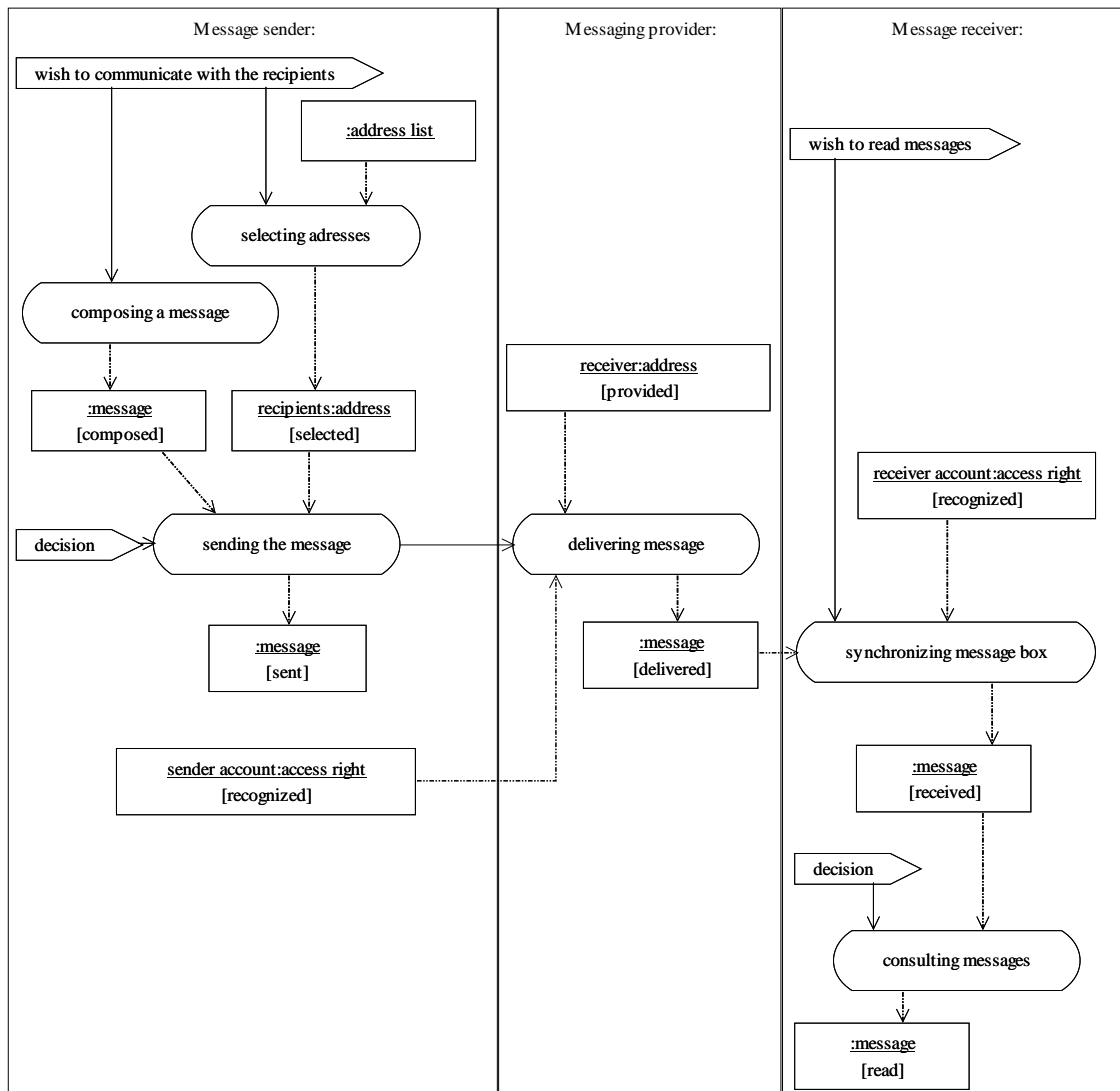


Figure 32 : Processus d'envoi et de consultation de messages

Les parties-prenantes (les acteurs) sont l'émetteur, le destinataire et le fournisseur de service de messagerie. Le processus commence avec le souhait d'un consommateur du service de communiquer par messagerie avec des destinataires. L'utilisateur *sender* sélectionne leurs adresses dans une liste, compose le message et l'envoie. Le fournisseur du service délivre le message dans les boîtes aux lettres des destinataires, après avoir vérifié les droits d'accès du sender et la validité des adresses des destinataires. Lorsqu'un consommateur du service souhaite lire les messages, l'utilisateur *receiver* accède à sa boîte aux lettres, après vérification de ses droits d'accès, et peut alors consulter ses messages.

Les entités de service du processus sont les suivantes :

- *address* qui représente une adresse de joignabilité comme un numéro de téléphone ou une adresse email
- *address list* qui représente une liste d'adresses, par exemple un carnet d'adresses ou un annuaire
- *message* qui représente un message texte
- *access rights* qui représente le droit pour le destinataire d'accéder à sa boîte aux lettres

Les relations entre ces entités sont indiquées dans la figure suivante. Une liste d'adresses contient des adresses. Un message possède une adresse d'émetteur et n adresses de destination.

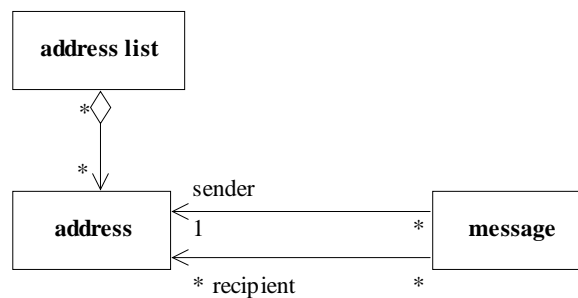


Figure 33 : Relations entre entités du service de messagerie

L'émission et la réception d'appels téléphoniques sont décrites par le processus de service suivant.

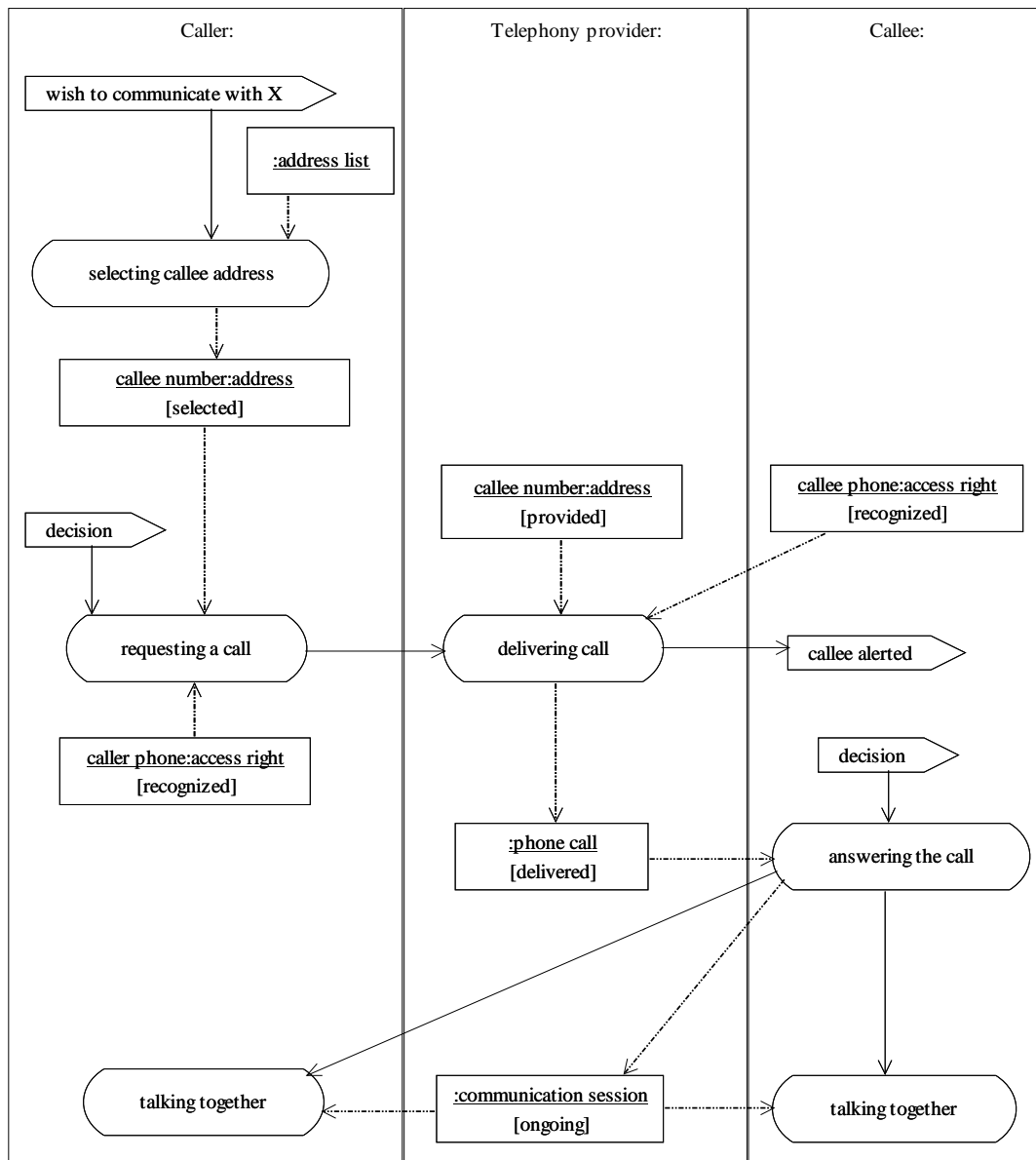


Figure 34 : Processus d'émission et de réception d'appels

Les parties-prenantes (les acteurs) sont l'appelant, l'appelé et le fournisseur de service de téléphonie. Suite à un souhait d'un consommateur du service de communiquer instantanément vocalement avec un interlocuteur, l'utilisateur *caller* sélectionne le numéro de téléphone, puis émet l'appel, à condition de disposer des droits d'accès au service. Le fournisseur du service délivre l'appel, après avoir vérifié la validité du numéro de téléphone de l'appelé, ce qui se traduit typiquement chez l'appelé par une sonnerie. Si l'appelé *callee* répond, cela produit une

session de communication qui permet à l'appelant et l'appelé de communiquer. Nous pouvons remarquer des similitudes, mais aussi des différences avec le processus précédent. Ainsi, la vérification des droits est effectuée au moment de la remise du message dans un cas et au moment de l'émission de l'appel dans l'autre. Ces différences sont inévitables, dues par exemple à des différences de perception entre les personnes décrivant ces processus. Nous verrons dans la section suivante de ce chapitre comment les motifs fonctionnels permettent, via une harmonisation des actions, de faire ressortir les similitudes et de distinguer entre différences superficielles et différences structurelles.

Les entités de service du processus sont les suivantes :

- *address* qui représente une adresse de joignabilité comme un numéro de téléphone ou une adresse email
- *address list* qui représente une liste d'adresses, par exemple un carnet d'adresses ou un annuaire
- *phone call* qui représente une demande d'établissement d'un appel téléphonique
- *access rights* qui représente le droit pour l'utilisateur d'accéder au service
- *communication session* qui représente un appel téléphonique établi

Les relations entre ces entités sont indiquées dans la figure suivante. Une demande d'établissement d'appel téléphonique possède un appelant et un appelé. Une session de communication se réfère à une demande d'établissement d'appel téléphonique.

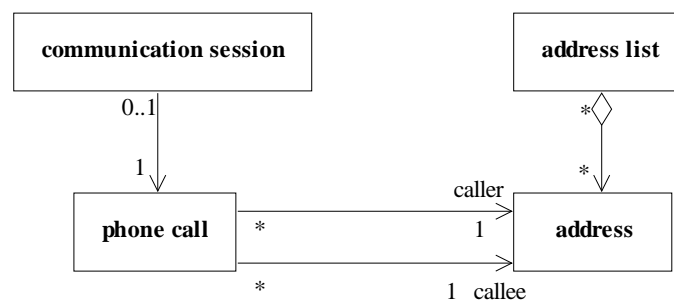


Figure 35 : Relations entre entités du service de téléphonie

La sélection et le jeu d'un media audiovisuel sont décrits par le processus de service suivant.

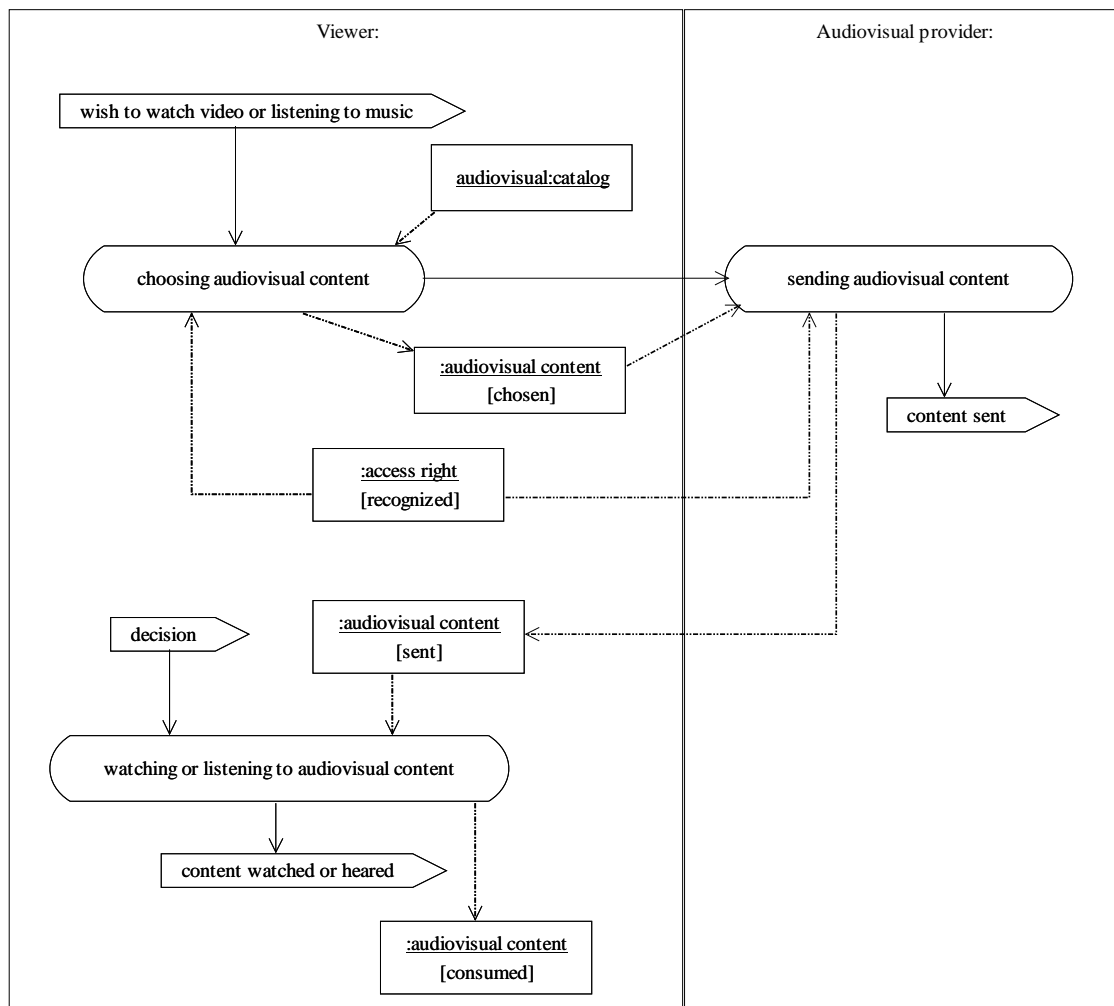


Figure 36 : Processus de sélection et de jeu d'un média

Les parties-prenantes (les acteurs) sont le spectateur (*viewer*) et le fournisseur de service audiovisuel. Un consommateur du service souhaite consommer un contenu audiovisuel. L'utilisateur *viewer* sélectionne ce contenu dans un catalogue, en fonction de ses droits d'accès au contenu du catalogue. Le fournisseur du service vérifie ces droits et diffuse le contenu. L'utilisateur peut alors consommer le contenu audiovisuel.

Les entités de service du processus sont les suivantes :

- *catalog* qui représente la liste des contenus audiovisuels disponibles pour un consommateur donné
- *access rights* qui représente le droit pour l'utilisateur d'accéder au service
- *audiovisual content* qui représente un contenu media (télévisuel ou musical)

Les relations entre ces entités sont indiquées dans la figure suivante. Un catalogue de contenus audiovisuels contient des contenus audiovisuels.

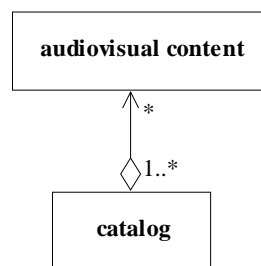


Figure 37 : Relations entre entités du service audiovisuel

L'accès à un service est décrit par le processus de service suivant.

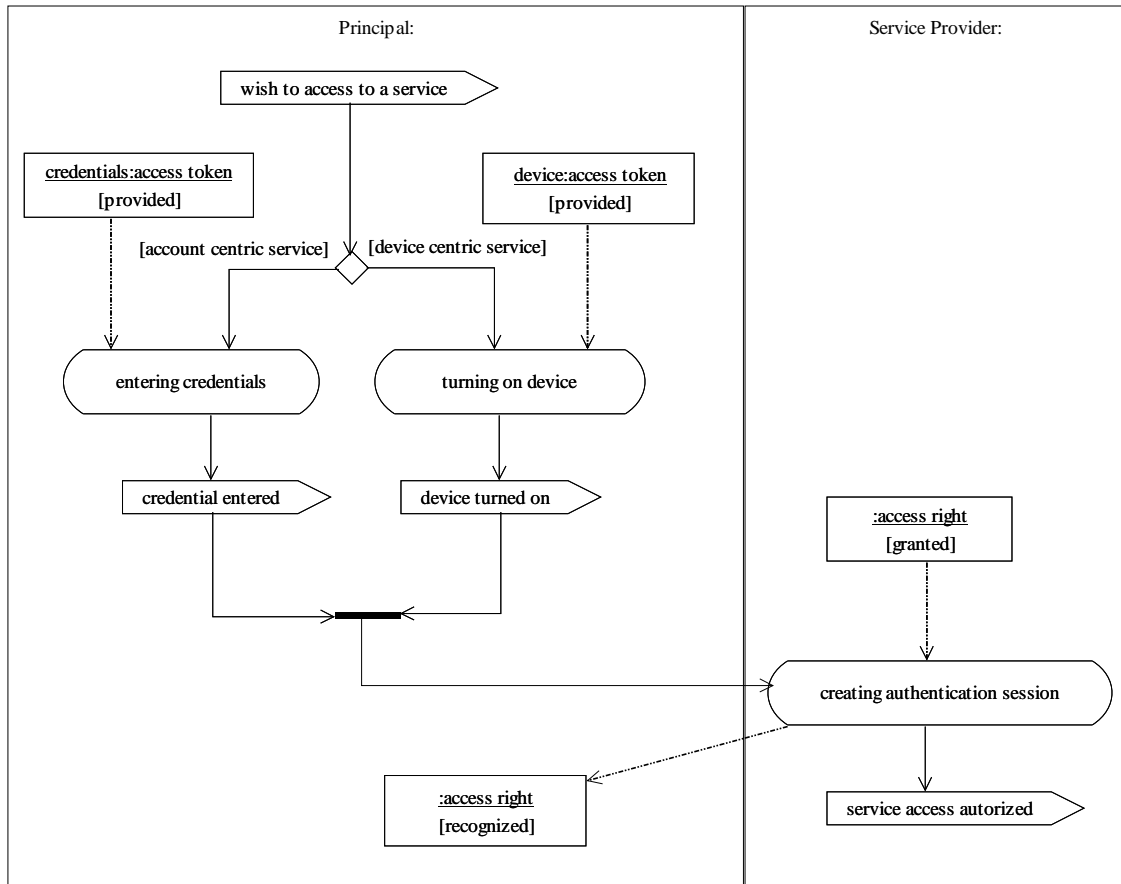


Figure 38 : Processus d'accès à un service

Les parties-prenantes (les acteurs) sont la personne à identifier (*principal*, en reprenant la terminologie du consortium Liberty Alliance [Liberty, web], centré sur la gestion de l'identité en ligne) et le fournisseur du service auquel elle cherche à accéder. Un consommateur souhaite accéder à un service. L'utilisateur principal entre ses *credentials* (typiquement un identifiant et un mot de passe) s'il est identifié auprès du fournisseur de service par un compte utilisateur, ou alors il met en route son équipement (matériel ou logiciel) si c'est cet équipement (par exemple une carte SIM) qui est identifié auprès du fournisseur de service. Le fournisseur de service détermine les droits affectés à ces credentials ou à ces équipements et crée une session d'authentification grâce à laquelle le principal bénéficie d'un accès aux services auxquels il a droit.

Les entités de service du processus sont les suivantes :

- *access token* qui représente le moyen d'accès au service. Ce peut être par exemple un identifiant et un mot de passe ou bien un appareil configuré avec un certificat (comme un téléphone mobile avec une carte SIM)
- *access rights* qui représente les droits pour l'utilisateur d'accéder aux services

Les relations entre ces entités sont indiquées dans la figure suivante. Un droit d'accès à un service est affecté à un moyen d'accès.

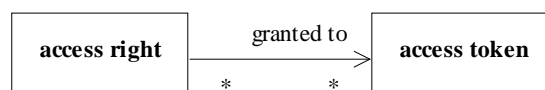


Figure 39 : Relations entre entités d'accès à un service

III.2 Une vue fonctionnelle de référence pour les services télécoms

Nous avons vu au cours de la section précédente qu'un service télécom, tel un outil, offre à ses utilisateurs la capacité de réaliser des actions, ces actions pouvant être décrites de façon impersonnelle et décontextualisée. Nous allons maintenant étudier comment formaliser ces capacités d'actions à travers la vue fonctionnelle de référence des services télécoms.

III.2.1 Comment définir les composants de la vue fonctionnelle ?

Nous avons vu que la vue fonctionnelle des systèmes d'information d'entreprise est définie par C. Longépé [Longépé, 2006] comme décrivant *"les fonctions du système d'information permettant de supporter les processus métier"*. L'architecture fonctionnelle est alors *"la structuration du système d'information en blocs fonctionnels communicants"*. La définition de ces blocs présente une certaine pérennité et ils peuvent être considérés comme invariants sur le temps d'un projet. Dans la démarche d'urbanisation, cette structuration suit celle d'un Plan Local d'Urbanisme, comme nous l'avons vu au cours du premier chapitre. Les fonctions du système d'information sont découpées en zones fonctionnelles, quartiers fonctionnels et îlots fonctionnels. Un îlot fonctionnel est décrit par les fonctions élémentaires qu'il réalise. Les fonctions élémentaires correspondent ainsi au grain le plus fin à partir duquel s'opèrent les regroupements en îlots, quartiers et zones. Nous allons maintenant nous concentrer sur l'identification de ces fonctions élémentaires dans le cas des services télécoms. Comme évoqué dans la section 3.3 du chapitre II, ces fonctions élémentaires correspondent aux fonctions techniques, au sens de l'analyse fonctionnelle présentée dans la section 2.2 du même chapitre II. Rappelons leur définition. Une fonction technique est une :

"Action interne au produit (entre ses constituants) choisie par le concepteur-réalisateur, dans le cadre d'une solution, pour assurer des fonctions de service."

Dans notre cas des services télécoms, le concepteur-réalisateur est celui qui conçoit le déroulement d'un service, celui qui spécifie le service vu de l'utilisation et non celui qui développe la solution technique. Les fonctions techniques ne décrivent alors pas une solution technique, mais la façon dont se déroule le service. Elles décrivent ce que devra rendre la solution technique. Considérons par exemple l'analyse fonctionnelle d'un vélo. Les fonctions de service avancer, tourner et ralentir sont assurées respectivement par les fonctions techniques de transmission de mouvement, de guidage et de freinage. Ces fonctions

techniques sont ensuite réalisées par une solution technique : la transmission de mouvement sera par exemple réalisée par le pédalier, la chaîne, les pignons et la roue arrière, mais pourraient également être réalisée par un moteur, des ponts, des arbres et des roues dans un autre contexte d'implémentation. Ces fonctions techniques sont relativement standardisées au sein d'un même domaine technique. Les mots transmission et distribution désignent ainsi en mécanique des fonctions techniques précises et différentes, quoique leur sens en français puisse être parfois proche. Une liste de fonctions techniques communes nous permettra donc à la fois de décrire un service de façon standardisée et à la fois de mieux identifier les solutions techniques susceptibles de le réaliser. Comme nous nous plaçons dans le cadre terminologique de l'urbanisation des systèmes d'informations, nous parlerons plutôt de fonctions élémentaires que de fonctions techniques, d'autant que le mot "technique" prête à une confusion avec la vue technique d'un service. Nous allons maintenant chercher à déduire ces fonctions élémentaires à partir des processus de service vus dans la section précédente.

Notre proposition est de définir pour les services télécoms une fonction élémentaire comme une capacité d'action fournie par un service à ses parties-prenantes. Par capacité d'action, on entend une possibilité d'agir offerte à un acteur d'un service. Ainsi, un service de téléphonie offre à l'appelant la capacité de passer un appel, au fournisseur la capacité de l'acheminer et à l'appelé la capacité d'y répondre. Un service de vidéo à la demande offre au spectateur la capacité de sélectionner une vidéo dans un catalogue et de la regarder, au fournisseur la capacité de la diffuser. Ces capacités d'actions ne sont pas propres à un service donné ou à une utilisation donnée, mais sont décontextualisées. Une fonction élémentaire peut ainsi être vue comme un support pour qu'un acteur réalise une action. Nous avons vu dans la section précédente que les actions constitutives des services télécoms pouvaient être décrites par un verbe d'action et par un complément d'objet direct désignant l'information manipulée. Par exemple, dans l'action d'envoyer un email, le verbe d'action est "envoyer" et l'information manipulée, en position de complément d'objet direct, est "un email". La capacité d'action support sera alors la "capacité d'envoyer un email". Cette capacité d'action permet de caractériser le fonctionnel d'un service. Par exemple, un service de messagerie fournira la capacité d'envoyer un email.

Mais une liste de fonctions élémentaires n'est pas suffisante pour décrire précisément le fonctionnement d'un service. Si nous nous contentions de décrire un service en listant les capacités d'action qu'il fournit, nous serions moins précis que dans la vue métier, où les

processus de service n'étaient pas uniquement constitués d'une liste d'actions, mais spécifiaient le séquençement de ces actions et les informations échangées (à travers les entités de service). Pour pallier à cette limite, nous pouvons formaliser des opérations fonctionnelles, chaque opération fonctionnelle correspondant à l'interface de réalisation d'une et une seule fonction élémentaire. Si la fonction élémentaire est la "capacité à envoyer un email", l'opération correspondante sera "envoyer un email". Nous voyons immédiatement un lien fort avec les actions de la vue métier, lien que nous allons étudier dans la section suivante. Nous pourrions alors, comme nous le verrons au chapitre suivant avec la vue fonctionnelle d'un service, décrire un service par un enchaînement d'opérations fonctionnelles. De plus, une opération recevant des données d'entrée et produisant des données de sortie, nous allons faire correspondre ces données avec des entités de service.

Il est en effet intéressant de considérer dans notre modélisation non seulement les capacités d'actions fournies par les services, mais également les informations manipulées, comme nous l'avons vu au chapitre précédent avec les entités de service. Toutefois, les entités de service ne sont pas suffisantes pour décrire avec exactitude les informations manipulées. Ainsi, si l'on considère uniquement les entités d'email ou de message, nous ne pouvons pas distinguer entre un message brouillon, un message envoyé, un message reçu, alors que ce sont bien des données différentes. De plus, nous ne pouvons pas considérer un message reçu comme une instance d'un type "message", car "message reçu" n'est pas une instance, mais bien également un type. Les états des entités sont un moyen de distinguer, mais il semble plus pertinent de disposer d'une donnée spécifique à chaque état afin de les manipuler plus aisément tout en se conformant aux bonnes pratiques de l'architecture d'entreprise. Nous appellerons données fonctionnelles de service de telles données. Ces données fonctionnelles sont reliées à une opération, comme "message envoyé" est lié à "envoyer un message". Plus précisément, une donnée fonctionnelle peut être définie comme le type de données qui est produit par une opération lorsqu'elle est utilisée : l'action d'envoyer un message produit un message envoyé. Chaque donnée fonctionnelle est issue d'une entité de service dans un état particulier, par exemple la donnée fonctionnelle "message envoyé" est issue de l'entité de service "message" dans l'état envoyé. Un service offrant une fonction élémentaire doit être capable à la fois de proposer la capacité d'action et de gérer les données produites par cette action. Par exemple, un service offrant la capacité de composer un email doit offrir à la fois des outils d'édition du message et gérer les messages composés.

Comment formaliser les fonctions élémentaires, les opérations et les données fonctionnelles pour la description d'un service ? Suivant les bonnes pratiques de l'architecture d'entreprise et du génie logiciel (UP et UML), telles que décrites nous proposons de les formaliser par des classes UML, stéréotypées

- **SEF** comme *Service Elementary Function* pour fonction élémentaire de service¹,
- **SFO** comme *Service Functional Operation* pour opération fonctionnelle de service
- **SFD** comme *Service Functional Data* pour les données fonctionnelles de service.

L'accès à une donnée fonctionnelle SFD se fait à travers une opération SFO. Nous avons choisi que chaque opération SFO ne produise qu'une seule donnée fonctionnelle SFD. Ceci permet de réaliser un lien fort entre les données manipulées et les traitements manipulant ces données. Cela signifie que chaque type d'action produit un et un seul type de données, et chaque type de données est produit par un et un seul type d'action. Une opération fonctionnelle produit une et une seule donnée fonctionnelle en paramètre de sortie et en utilise n (avec $n \geq 0$) en paramètre d'entrée. Ces notions sont synthétisées dans le méta-modèle UML suivant.

¹ Au chapitre précédent, nous avons déjà vu le terme de "fonction de service" dans le cadre de l'analyse fonctionnelle. Le terme de fonction élémentaire de service est ici employé dans un sens totalement autre ; une fonction élémentaire de service dans le cadre de notre modélisation correspond à une fonction technique au sens de l'analyse fonctionnelle.

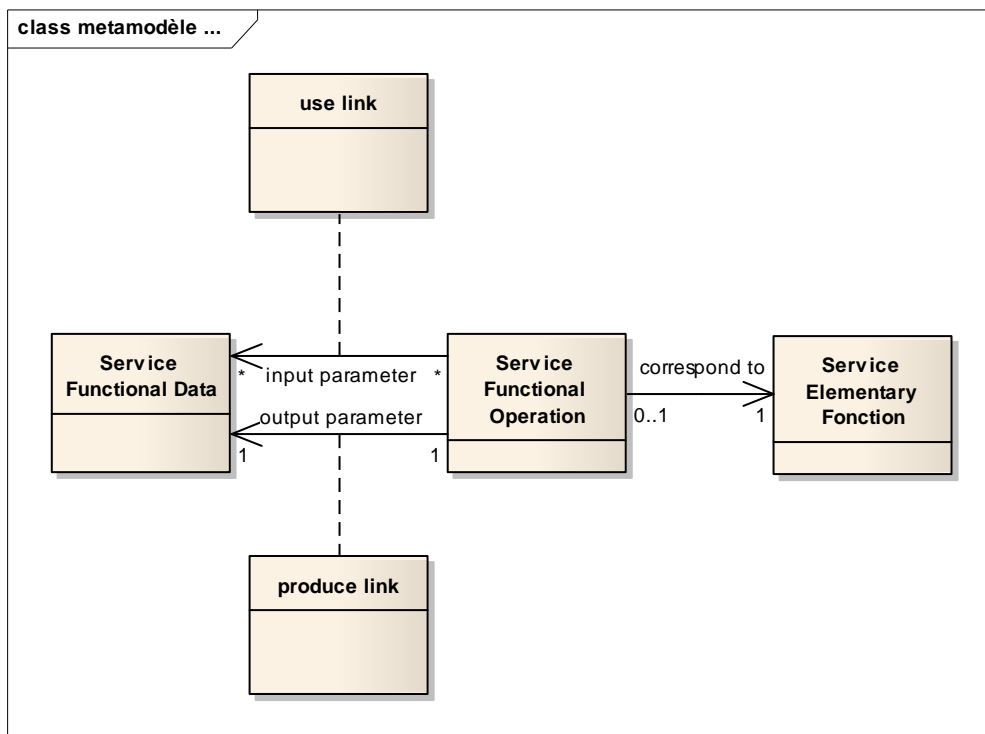


Figure 40 : Notions de SEF, SFO et SFD

Les opérations peuvent être vues comme des opérateurs de manipulation des données fonctionnelles. Par exemple, l'action "envoyer un mail" va utiliser la donnée fonctionnelle "message composé" pour produire la donnée fonctionnelle "message envoyé". Ces relations d'utilisation sont la traduction au niveau fonctionnel du fait qu'une activité métier nécessite pour sa réalisation une entité de service (dans un état particulier) produite par une autre activité métier. L'intérêt de ces opérations sera explicité dans le chapitre suivant avec l'utilisation de diagrammes de séquence pour décrire un service.

Par ailleurs, les relations entre opérations SFO peuvent se traduire par des dépendances entre fonctions élémentaires SEF. Lorsqu'une opération SFO-1, SFO-1 correspondant à une fonction élémentaire SEF-1, utilise une donnée fonctionnelle SFD-2 produite par une autre opération SFO-2, SFO-2 correspondant à une fonction élémentaire SEF-2, ceci équivaut à une dépendance de SEF-1 vers SEF-2. Une telle dépendance signifie que SEF-1 a besoin de SEF-2 pour réaliser sa fonction propre. Il y a ainsi un lien fort entre une donnée fonctionnelle, une opération fonctionnelle et une fonction élémentaire.

Dans le cadre de l'urbanisme des systèmes d'information, les fonctions élémentaires sont regroupées en îlots fonctionnels. Ces îlots produisent et utilisent des données

fonctionnelles. Ils produisent et utilisent des opérations fonctionnelles, ces liens de production et d'utilisation (ou voies fonctionnelles) constituant la vue externe de leurs relations. Ces îlots fonctionnels sont regroupés ensuite en quartiers fonctionnels, et ces quartiers sont eux-mêmes regroupés en zones fonctionnelles. Dans le cadre des services télécoms, il nous semble intéressant d'adopter le même découpage, non seulement pour se conformer aux bonnes pratiques de l'urbanisation des systèmes d'information mais également pour disposer de concepts de plus haut-niveau, qui permettent de masquer les détails d'un domaine à travers des regroupements tels que les quartiers ou les zones.

Nous formaliserons ces îlots fonctionnels par des classes UML stéréotypées **SFC** pour *Service Functional Component*, composant fonctionnel de service, le terme composant rappelant qu'un îlot fonctionnel possède une vue externe. L'utilisation par un composant SFC A d'une opération SFO B (lien *uses*) signifie que l'opération SFO A fournie par ce SFC A (lien *provides*) a comme paramètre d'entrée (*input parameter*) une donnée SFD B produite (*output parameter*) par l'opération SFO B.

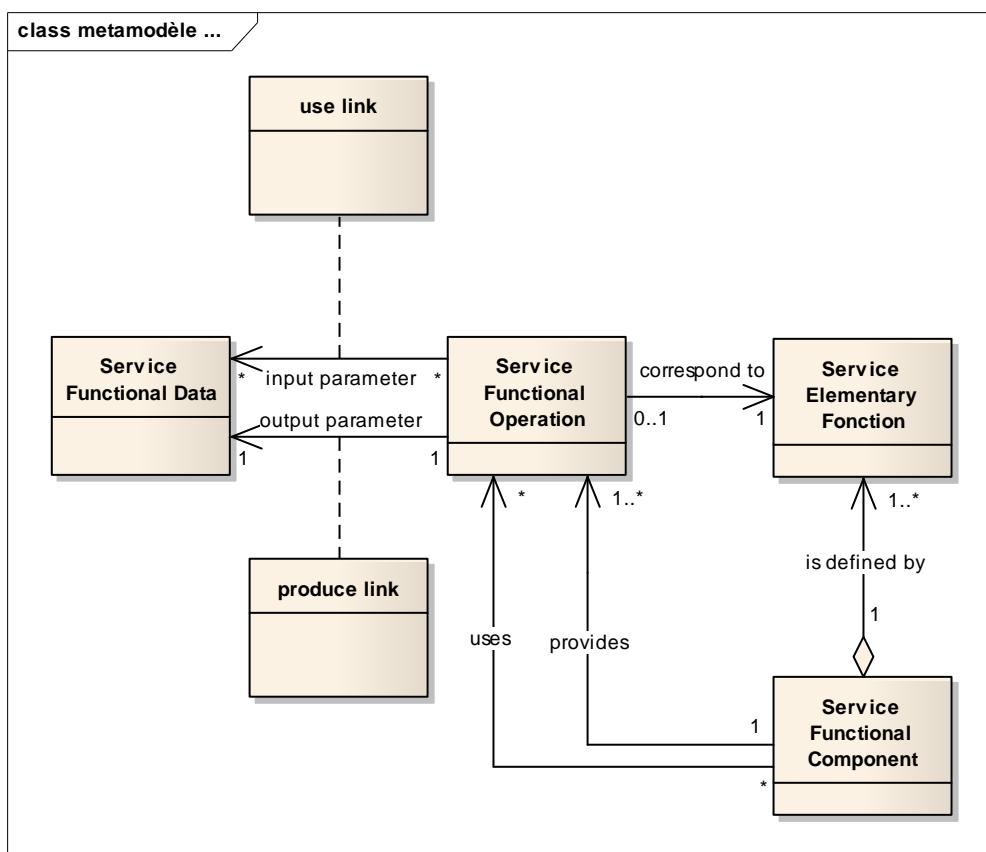


Figure 41 : Méta-modèle de la vue fonctionnelle de référence

Après avoir défini la signification des concepts de l'urbanisme des systèmes d'information dans le cadre des services télécoms, nous allons maintenant étudier comment construire des SEF, SFO et SFD de référence et comment en déduire des SFC de référence.

III.2.2 Comment construire une vue fonctionnelle de référence ?

Notre travail précédent pour construire des processus métier de service nous donne un outil pour identifier les fonctions élémentaires SEF à partir des actions réalisées dans le cadre d'un service. Toutefois, nous cherchons à identifier des capacités d'actions génériques et non liées à un service particulier. Nous avons vu l'importance de la médiation du langage pour arriver à ce résultat.

Nous devons donc tout d'abord élaborer une liste permettant de décrire de façon non ambiguë le fonctionnement des services télécoms pour leurs parties-prenantes. Cela passe par une nomenclature commune pour nommer toutes les fonctions élémentaires. Nous proposons tout d'abord de nommer une fonction élémentaire par "capacité à", suivi d'un verbe d'action, puis de l'entité de service manipulée dans l'action en position de complément. Par exemple "capacité à envoyer un message". Concernant les entités de service, un travail d'harmonisation lexicale peut être mené, par exemple entre des mots équivalents comme texto et SMS. Concernant les verbes d'actions, nous pouvons les harmoniser a priori en définissant une liste très réduite de verbes d'actions à utiliser pour nommer les capacités d'actions.

Pour cela, une première possibilité serait de lister tous les verbes d'actions des processus de service vus dans la section précédente, puis choisir arbitrairement un verbe d'action à privilégier parmi une liste de verbes d'actions synonymes ou au sens proche – cette démarche est usuelle dans les bonnes pratiques SOA [Nuffel, 2007]. On pourrait alors obtenir un résultat tel que celui présenté dans le tableau suivant.

Tableau 3 : Exemples de verbes à privilégier

Verbe d'action	Nom choisi	Définition
To fetch, to collect, to capture, to get, to pull To send, to publish, to issue	Capability to issue	Capacité à publier, à émettre une information vers des destinataires spécifiés ou non
To save, to safeguard, to store	Capability to store	Capacité à stocker une information

		existante de façon pérenne dans un emplacement
To sort, to filter, to match, to select To choose, to search, to browse	Capability to browse	Capacité à naviguer dans une liste d'informations pour en sélectionner une ou plusieurs.
To apply, to enforce, to follow, to check	Capability to enforce	Capacité à vérifier la validité d'une information par rapport à une référence
To display, to show, to play, to present, to render To read, to watch, to hold, to pause, to stop, to retrieve, to resume	Capability to render	Capacité à rendre perceptible (visible, audible, tangible) une information et à contrôler ce rendu (pause, reprise, arrêt)
To create, to set, to compose, to modify, to delete To start, to end	Capability to create	Capacité à créer une information, à la modifier et à la supprimer
To push, to distribute, to notify, to provide, to deliver to transfer, to forward, to route To receive, to answer	Capability to deliver	Capacité à remettre une information à un ou plusieurs destinataires précis, incluant la notification du destinataire
To download, to copy	Capability to copy	Capacité à dupliquer une information

Une telle harmonisation est certes efficace pour élaborer les fonctions élémentaires à partir des processus de service, mais présente des inconvénients. Tout d'abord, la granularité des fonctions reste dépendante de celle définie dans les processus de service. Nous n'obtenons donc pas l'indépendance souhaitée vis à vis des spécificités des processus de service. Ensuite, cette harmonisation uniquement lexicale ne fournit pas un outillage suffisant pour aider à l'identification des fonctions. En appliquant cette méthode, on obtiendrait par exemple comme fonctions "*message issuing*" et "*content issuing*" ou "*message rendering*" et "*content rendering*", sans pouvoir déterminer ce qui est exactement commun dans ces deux fonctions mis à part le verbe les décrivant. Enfin, cela ne permet pas de prendre en compte la stratégie fonctionnelle de l'opérateur télécom, c'est-à-dire quelles familles de fonctions élémentaires sont dans le périmètre de l'opérateur, quelles sont les capacités d'action génériques que fournit l'opérateur quelles que soient les entités de service manipulées.

Pour répondre à ces limites, nous allons introduire des motifs fonctionnels, conçus par généralité et abstraction à partir des services fournis par un opérateur télécom et

correspondant à sa stratégie fonctionnelle. Par stratégie fonctionnelle, nous entendons ici les types de fonctions qui sont de la compétence d'un opérateur télécom. Cette stratégie fonctionnelle correspond à un invariant stratégique, au-delà de la variété tactique des services mis en œuvre. En effet, certaines capacités d'action proches des services télécoms ne sont en fait pas du ressort du service lui-même mais de services connexes. Ainsi, la création d'un document ou d'une feuille de calcul est du ressort d'applications comme Word ou Excel, mais pas des services télécoms. La création de fichier ne fait ainsi pas partie de nos motifs fonctionnels. La fourniture de capacités de création de fichiers, par exemple avec des outils bureautiques tels que traitements de texte ou tableurs n'entre pas aujourd'hui dans la stratégie fonctionnelle des fournisseurs de service télécoms. On peut rapprocher cette notion de stratégie fonctionnelle de celle de barrière de métier évoquée notamment par D. Lombard dans [Lombard, 2008].

Avec ces motifs fonctionnels, nous nous intéressons aux actions génériques que permet un service télécom, générique signifiant ici indépendamment de l'entité manipulée dans cette action. Un motif fonctionnel est composé d'une opération fonctionnelle générique, produisant et utilisant des données fonctionnelles génériques. Une opération fonctionnelle générique est une opération qui peut potentiellement manipuler tout type d'entité de service. Par exemple, "émettre" est une opération fonctionnelle générique qui pourra tout aussi bien s'appliquer à une entité "message" dans une opération "émettre un message", ou à une entité "appel" dans une opération "émettre un appel", voire à une entité "contenu multimédia" dans une opération "émettre un contenu multimédia". Une donnée fonctionnelle générique est le type de donnée fonctionnelle manipulée par une opération fonctionnelle générique. Une opération fonctionnelle générique "émettre" nécessitera par exemple une donnée fonctionnelle générique "adresse de destination". La nécessité d'une adresse de destination est structurellement incluse dans la signification de "émettre". Ces données fonctionnelles génériques peuvent être vues comme un rôle joué par des données vis-à-vis d'une opération fonctionnelle générique. Avec ces motifs fonctionnels, nous allons ainsi obtenir non pas juste une liste de verbes pour nommer nos opérations, mais une définition des paramètres d'entrée et de sortie que doit contenir une opération lorsqu'elle est nommée par un verbe donné. Ces concepts sont repris dans le méta-modèle ci-dessous.

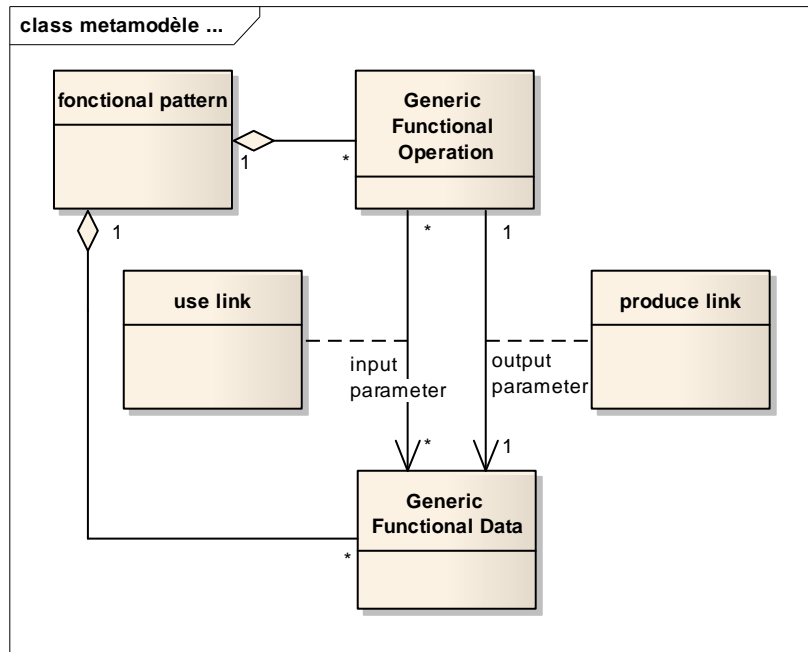


Figure 42 : Notion de motif fonctionnel

Ces concepts seront formalisés par des classes UML stéréotypées

- **GFO** pour *Generic Functional Operation*, opération fonctionnelle générique,
- **GFD** pour *Generic Functional Data*, donnée fonctionnelle générique.

Nous avons identifié trois motifs typiques des services télécoms et qui se combinent dans les services existants.

Le premier est le motif historique des télécommunications : **le motif de communication**. Une fonction d'envoi permet de transformer une information initiale en message, avec une adresse de destination et un émetteur. Une fonction de remise permet de transformer ce message en information reçue, selon des règles de distribution et des enregistrements associant adresses de destination et destinataires.

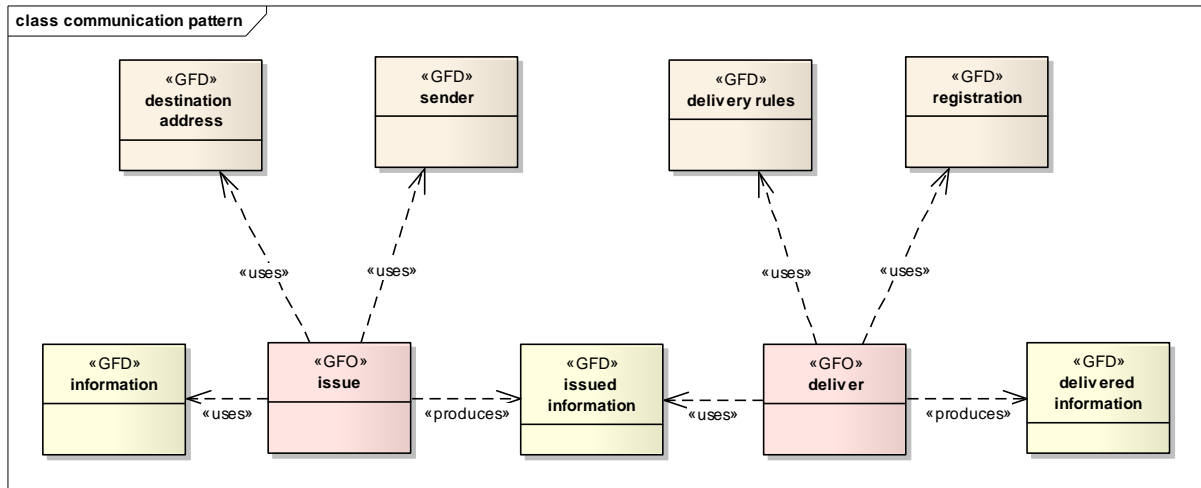


Figure 43 : Motif fonctionnel de communication

Le second motif est le motif du web et plus généralement des média : **le motif de diffusion**. Une fonction de publication permet de transformer une information initiale (par exemple un post sur un blog, une vidéo) en information publiée avec un propriétaire et au sein d'un container (par exemple un blog, un catalogue). Une fonction de sélection permet de naviguer dans le container pour choisir une information, conformément à des autorisations.

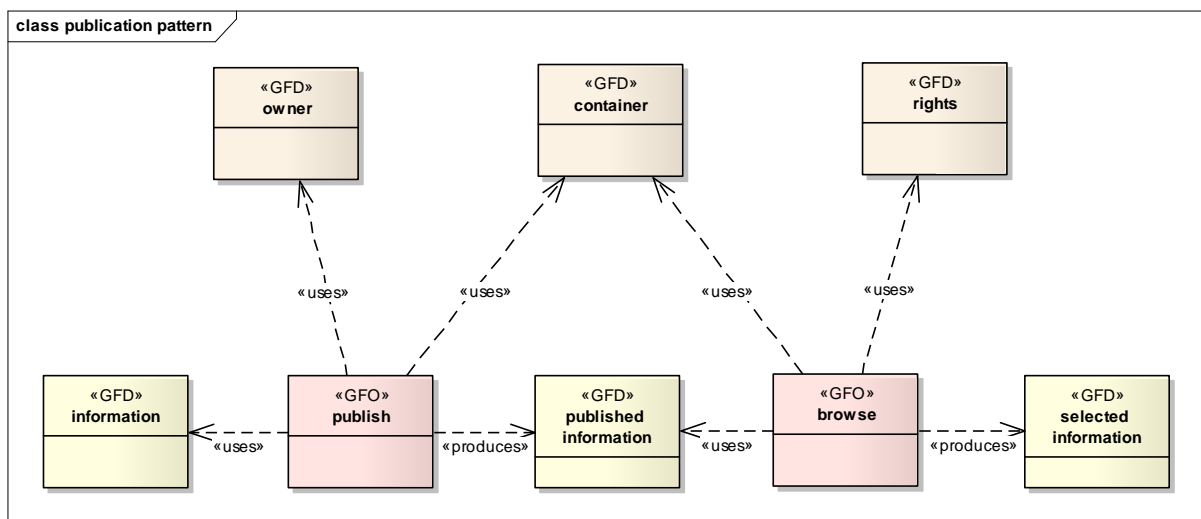


Figure 44 : Motif fonctionnel de diffusion

Enfin, un troisième motif est un motif de paramétrage d'informations : **le motif de configuration**. Il permet de définir les informations utilisées dans les motifs précédents, comme les règles de distribution, les autorisations ou les containers. Une fonction de définition permet de définir une référence, éventuellement à partir d'autres informations. Une

fonction d'application permet de déterminer une information pertinente à partir d'informations initiales et de références (par exemple l'authentification d'un utilisateur à partir d'un identifiant et d'un mot de passe saisi en fonction d'un identifiant mot de passe référence produit un *token* d'authentification)

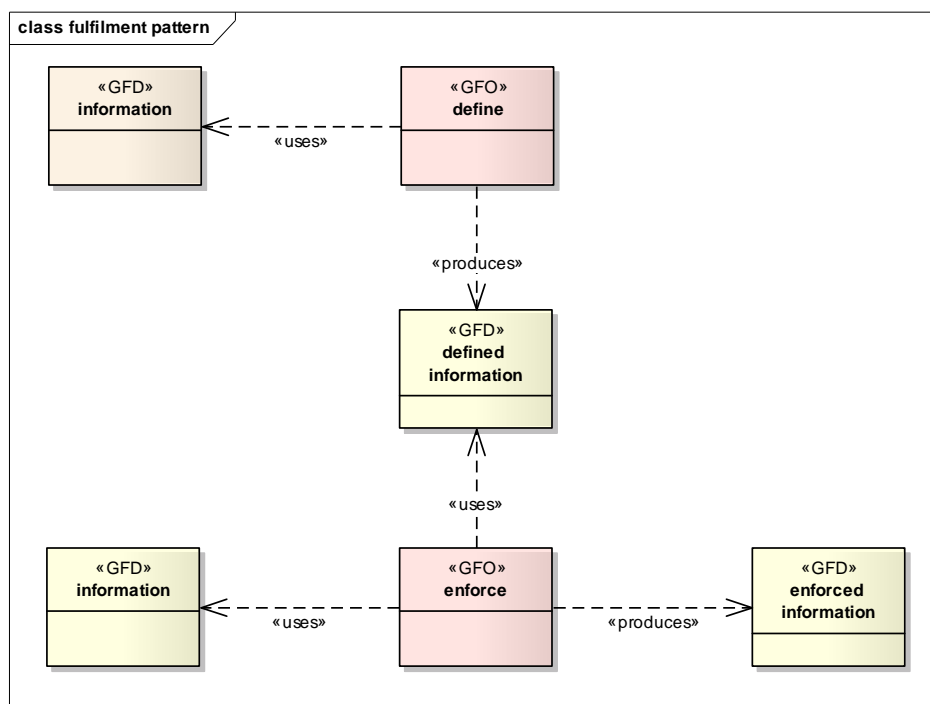


Figure 45 : Motif fonctionnel de configuration

Les opérations fonctionnelles génériques de ces motifs sont récapitulées dans le tableau suivant. Dans les sens proche, on a regroupé les verbes désignant une action vue du fournisseur de service (par exemple *to distribute*) et ceux désignant la même action vue de l'utilisateur (par exemple *to receive*).

Tableau 4 : Liste des opérations fonctionnelles génériques (GFO)

GFO	Définition	Sens proches
Issue	Emettre une information vers des destinataires, spécifiés par une adresse de joignabilité	To capture, to pull To send, to issue, to compose
Deliver	Remettre un message à un ou plusieurs destinataires, spécifiés par leur adresse de joignabilité, éventuellement en fonction de règles de distribution	To push, to distribute, to notify, to provide, to deliver to transfer, to forward, to route To receive, to answer, to download

Publish	Publier une information au sein d'un contenant	To store, to fetch, to collect, To publish, to upload
Browse	Naviguer dans un ensemble d'éléments d'un contenant, pour en sélectionner un ou plusieurs	To sort, to filter, to match, to select, to get To choose, to search
Define	Définir, configurer une référence	To define, to create, to configure, to setup, to set
Enforce	Vérifier la validité d'une information par rapport à une référence	To apply, to enforce, to follow, to check

Pour identifier les opérations fonctionnelles SFO des services télécoms, nous allons maintenant combiner ces motifs fonctionnels avec les processus d'usage de la vue métier. Ceci permet d'ancrer les SFO, les SEF et les SFD à la fois dans le déroulement des services et à la fois dans les grandes familles de capacités d'action fournies par les services télécoms. A partir de ces motifs, nous allons pouvoir définir tout d'abord des règles de nommage. Nous privilégierons la langue anglaise sur ce point, afin que cette nomenclature puisse être internationalisée.

Une opération SFO sera nommée par un verbe d'action tel que décrit dans un motif fonctionnel, suivi de l'entité de service manipulée lors de l'action. Par exemple "*issue message*".

Une fonction élémentaire SEF sera nommée par le gérondif¹ du verbe de l'opération correspondante, précédé de l'entité de service manipulée lors de l'action et suivie du mot "*capability*" pour indiquer qu'il s'agit d'une capacité d'action. Par exemple "*message issuing capability*". En français par l'infinitif du même verbe d'action suivi de l'entité de service manipulée et précédé du mot "capacité de", comme dans "capacité de créer un message".

La donnée fonctionnelle de service SFD paramètre de sortie d'une SFO est nommée en anglais par le verbe d'action du SFO à la forme passive suivi de l'entité de service manipulée, par exemple "*issued message*" pour *SFO issue message*. En français, la donnée fonctionnelle serait nommée par le verbe d'action à la forme passive précédé du concept manipulé, par exemple "message émis". En effet, comme une donnée fonctionnelle est fortement liée à l'opération qui la produit, nous proposons de refléter ce lien fort dans leur nommage.

¹ Le temps anglais du gérondif étant particulièrement approprié pour désigner une action.

Une opération SFO nommée "B A", où B est un verbe d'action et A une entité de service, consiste en la réalisation par un système de service de l'action B sur l'entité A. Par exemple, l'opération fonctionnelle SFO "émettre un appel" désigne la réalisation de l'action d'émettre un appel par le système de service composé notamment de l'appelant, du terminal téléphonique, du réseau, de l'opérateur. Une SFD nommée "B-ed A" est la donnée produite lors de cette action, par exemple "appel émis". Une SEF nommée "A B-ing capability" est la capacité offerte par un système de service d'appliquer l'action B à l'entité de service A. Ce nommage formel reflète la façon dont sont construites les opérations et les données fonctionnelles. On peut y ajouter un nommage plus convivial, exprimant de façon plus aisément compréhensible le sens du composant ou de la donnée fonctionnelle. Par exemple, on pourra associer le nom convivial "*send message function*" à la *SEF message issuing capability* ou le nom convivial "*sent message data*" à la *SFD issued message*. Ceci permet à la fois de conserver une harmonisation du nom des SEF, SFO et SFD d'après les motifs fonctionnels vus ci-dessus et en même temps d'adapter leurs noms aux vocabulaires des différents domaines.

Voyons maintenant comment déduire les opérations SFO à partir de la vue métier. Nous avons vu dans la section précédente qu'une action de la vue métier produisait et utilisait des entités de service dans un état particulier. Nous avons vu que l'association entre action et entités dans un état particulier pouvait être modélisée par deux classes d'association, une pour la production de l'entité et l'autre pour son utilisation. Dans cette section, nous avons vu qu'une classe d'association liait une opération SFO avec la donnée SFD rendue par cette opération et une autre classe d'association liait cette opération SFO aux données SFD nécessaires en entrée de cette opération. Pour déduire une vue fonctionnelle de référence de notre vue métier de référence, nous allons faire correspondre sémantiquement ces classes d'association, comme indiqué dans la figure ci-dessous.

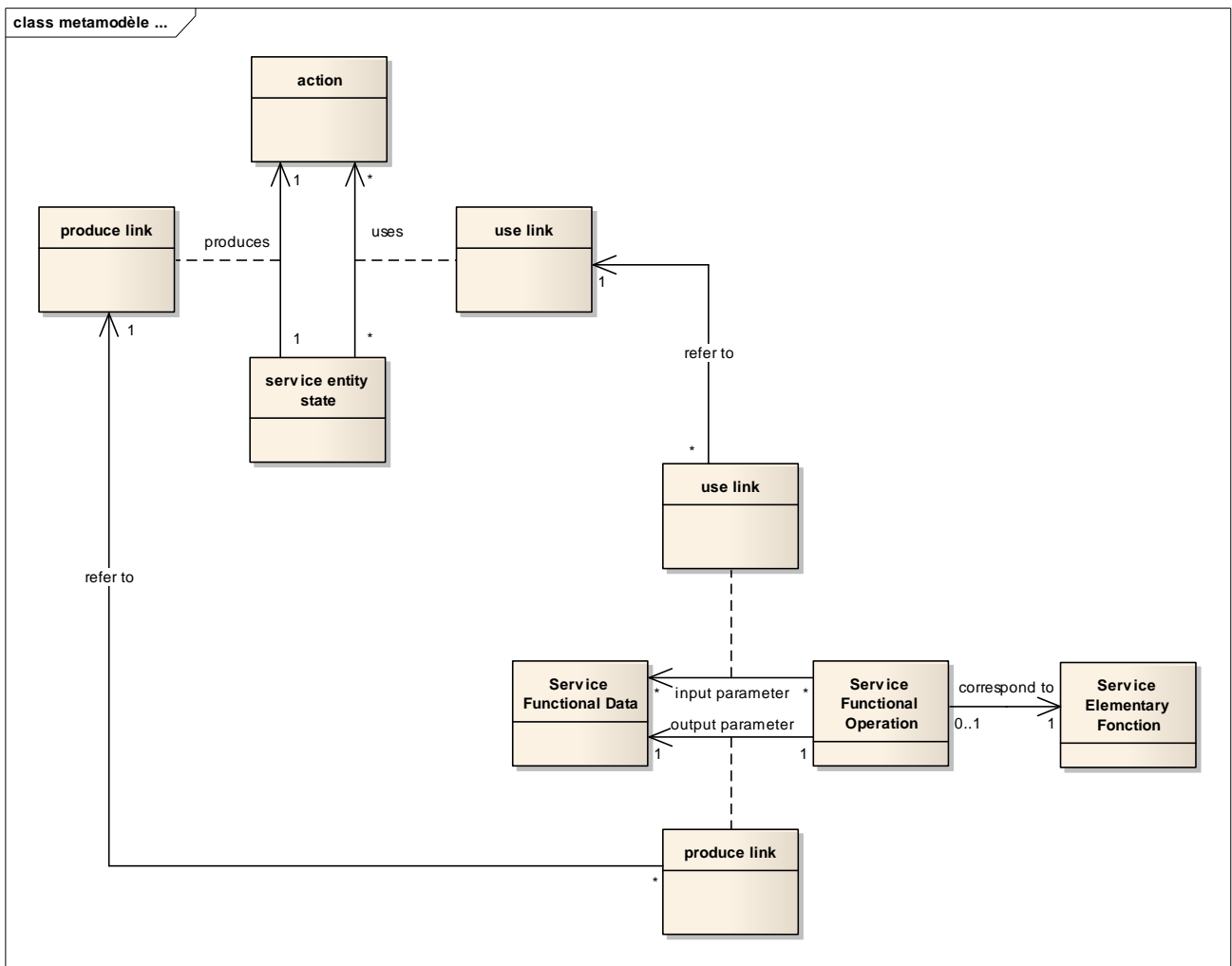


Figure 46 : Lien entre vue fonctionnelle de référence et vue métier de référence

Ceci nous permet de relier par exemple une action d'envoi de message produisant un message envoyé avec une opération SFO "émettre message" produisant une donnée SFD "message émis".

Il nous reste à expliciter la prise en compte des motifs fonctionnels dans l'élaboration des SEF, SFO et SFD. Les opérations GFO et les données GFD sont également associées par deux classes d'association, l'une concernant la production de la donnée et l'autre son utilisation. Une correspondance sémantique entre ces classes d'association et celles reliant SFO et SFD sera également réalisée, comme indiqué dans la figure ci-dessous.

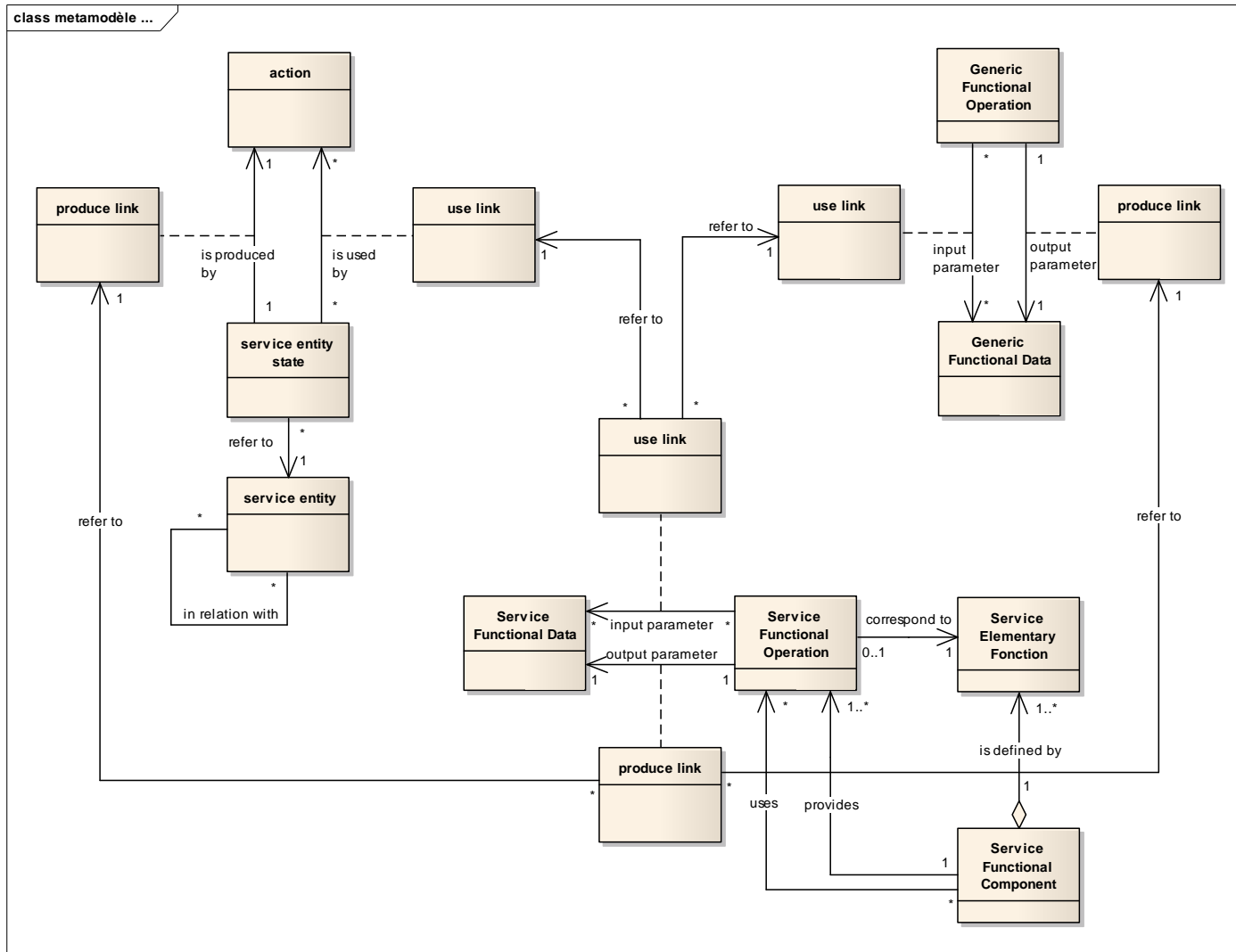


Figure 47 : Lien avec les motifs fonctionnels

Ceci nous permet enfin de construire notre vue fonctionnelle de référence en rapprochant sémantiquement une action d'envoi de message produisant un message dans l'état envoyé avec une opération GFD "émettre" produisant une donnée GFD "information émise" pour aboutir à une opération SFO "émettre un message" produisant une donnée SFD "message émis".

Récapitulons. Chaque activité UML d'un processus d'usage peut donner lieu à une SFO et une SEF à condition que cette activité puisse être corrélée avec l'un des motifs fonctionnels vu précédemment. On applique donc la méthode suivante sur nos processus :

1. Si une activité produit une et une seule entité de service, alors on passe directement à l'étape 2. Si une activité produit plusieurs entités de service, alors elle est scindée en activités plus élémentaires produisant une et une seule entité de service dans un état particulier. Si une activité ne produit pas d'entité de service, alors on détermine une entité de service manifestant la valeur ajoutée de cette activité. Si ce n'est pas possible, on la fusionne avec une autre activité.
2. Si une activité peut être corrélée avec une opération générique GFO de l'un des motifs fonctionnels, et que l'entité produite par cette activité peut être corrélée avec une donnée générique GFD, alors on déduit une opération SFO de cette activité et une donnée fonctionnelle SFD de l'entité de service dans un état particulier produite par cette activité, cette SFD constituant le paramètre de sortie de la SFO. Sinon, on retravaille le processus pour que cela soit le cas¹. On itère cette étape 2 pour que toutes les activités soient étudiées.
3. Pour chaque opération SFO, on considère chaque donnée fonctionnelle générique GFD utilisée par la GFO avec laquelle la SFO est corrélée. Si une donnée générique peut se rapporter à une entité de service dans un état particulier utilisée par l'activité dont est déduite la SFO, alors la SFO utilise en paramètre d'entrée la SFD issue de cette entité de service dans un état particulier. Des entités de services omises dans les processus de service peuvent être ajoutées si nécessaire.
4. De chaque SFO est déduite une fonction élémentaire SEF correspondant à la capacité pour un système de service d'offrir cette SFO.

Grâce à ces règles, nous pouvons obtenir un modèle unique de SEF, SFO et SFD à partir de processus d'usage réalisés par différentes équipes, dans la mesure où les règles de construction d'un processus d'usage ont été respectées. Des paramètres d'entrée SFD pourront éventuellement être ajoutés au niveau des SFO s'ils ont été oubliés ou omis dans les processus d'usage. Une réflexion sur la valeur ajoutée de chaque SFO, vu comme un opérateur de production d'une donnée fonctionnelle à partir de n autres, peut en effet conduire à identifier

¹ Ce travail d'adaptation de la vue métier pour pouvoir déboucher sur des fonctions élémentaires et des opérations fonctionnelles peut notamment être mené à partir d'une étude d'alignement telle que définie dans [Simonin, 2009]

de nouveaux liens d'utilisation, qui devront rétroactivement pouvoir être justifiés dans les processus d'usage.

Appliquons, par exemple, cette méthode avec le processus de service d'appel téléphonique. L'activité *requesting call* produit l'entité *phone call* dans l'état *requested* et utilise l'entité *reachable address* dans l'état *selected*. Reprenons les quatre étapes précédentes :

1. L'activité *requesting call* ne produit pas d'entité. La valeur ajoutée de cette activité est l'émission d'une demande d'appel vers un destinataire. Cette valeur ajoutée peut être manifestée par la production d'une entité de service "*phone call*" dans l'état "*requested*".

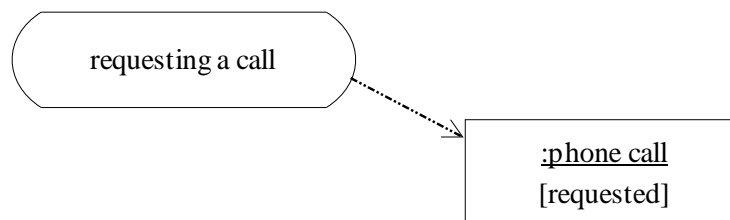


Figure 48 : Modification d'un processus de service

2. L'activité *requesting call* peut être corrélée avec la fonction générique *GFO issue* du motif fonctionnel de communication. *GFO issue* correspond en effet à l'émission d'une information vers des destinataires. L'entité *phone call* dans l'état *requested* peut être corrélée avec la donnée générique *GFD issued information*, qui correspond à une information émise. On déduit donc de l'activité *requesting call* une opération *SFO issue phone call* et une donnée fonctionnelle *SFD issued phone call* est déduite de l'entité *phone call* dans l'état *requested*.

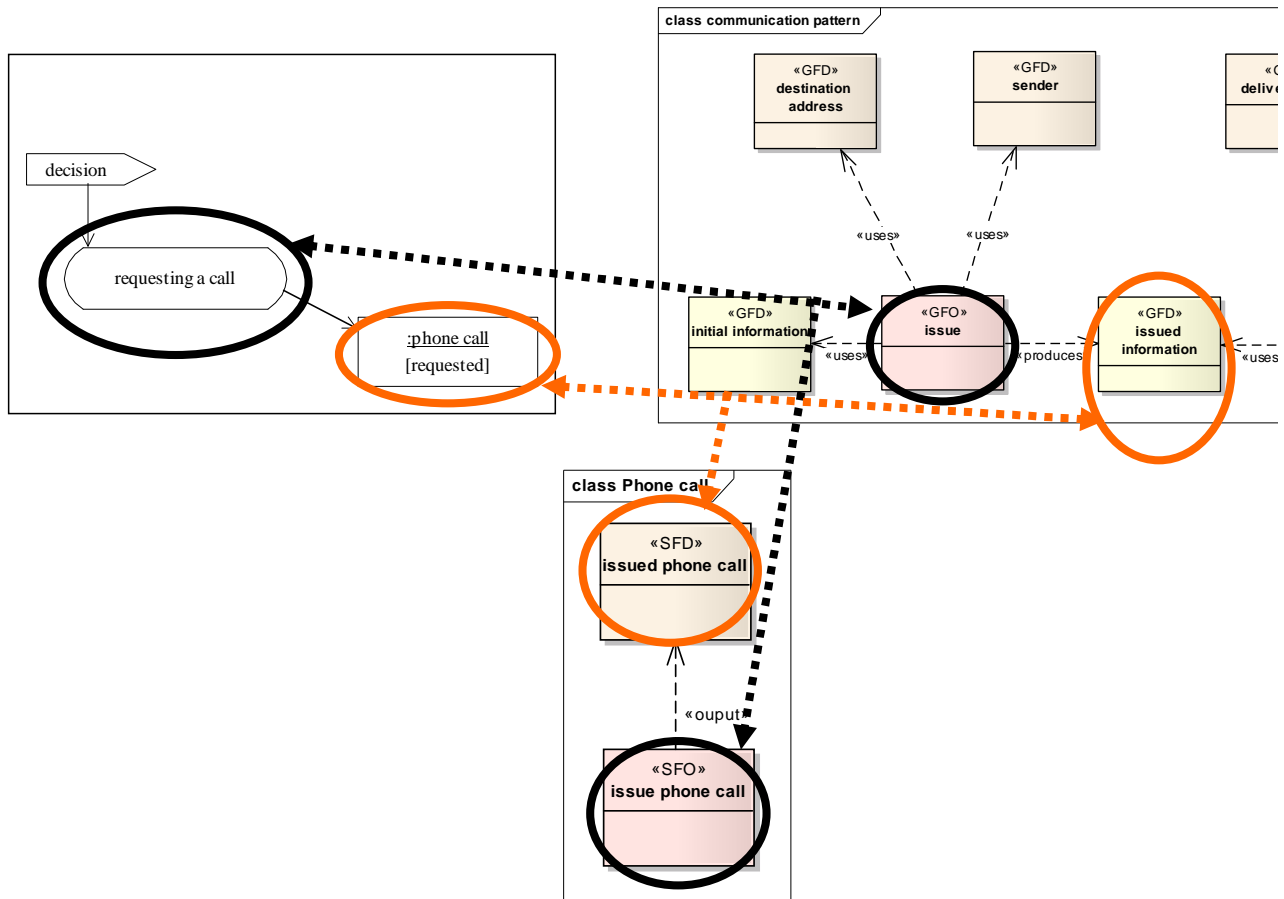


Figure 49 : Construction d'une opération fonctionnelle et de la donnée fonctionnelle produite

- L'opération générique *GFO issue* utilise une donnée générique *GFD destination address*, qui peut se rapporter à l'entité *address* dans l'état *selected*, puisque l'entité *address* représente l'adresse de joignabilité du destinataire de la demande d'appel.

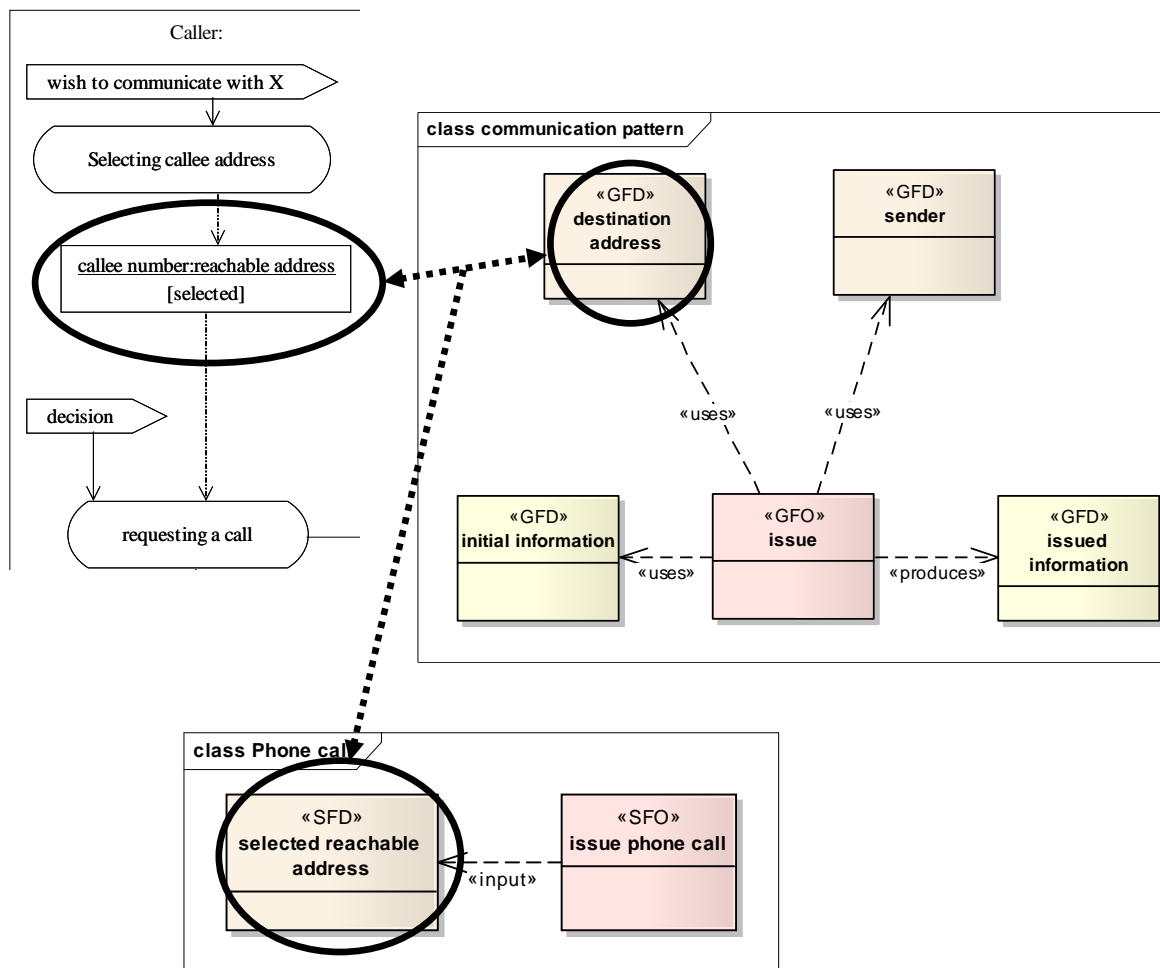


Figure 50 : Construction d'une donnée fonctionnelle utilisée

- Une fonction élémentaire *SEF phone call issuing capability* est déduite de l'opération *SFO issue phone call*.

Par ailleurs, nous avons vu précédemment l'intérêt de regrouper des fonctions élémentaires SEF en composants fonctionnels SFC. Ce regroupement peut se faire suivant plusieurs logiques.

Tout d'abord, ce regroupement pourrait faire correspondre les SFC avec les applications existantes implémentant les fonctions élémentaires SEF. L'inconvénient majeur de cette méthode est de confondre la logique de la vue fonctionnelle d'un service avec la logique de la vue applicative, vue que nous expliciterons au cours du chapitre à venir. Ensuite, ce regroupement peut être laissé à l'expertise des architectes et urbanistes de service, selon des critères de proximité métier entre SEF. Cela constitue une possibilité tout à fait valable, et la pertinence de tels regroupements peut être améliorée et justifiée par des procédés de conception automatisée des îlots, quartiers et zones fonctionnelles en fonction de l'intensité des liens d'utilisation par une opération SFO de données générées par d'autres opérations SFO, chaque SFO correspondant à une fonction élémentaire SEF. Un tel procédé est défini dans [2007-10] ou [Simonin, 2009], avec notamment une axiomatisation, et une automatisation par un algorithme bactériologique. Cette approche nécessite toutefois un nombre conséquent de scénarios caractéristiques de l'ensemble des services télécoms d'un opérateur.

Nous allons donc effectuer un regroupement très simple, peut-être simpliste. Les fonctions élémentaires SEF, les opérations SFO et les données SFD seront regroupées dans un SFC selon l'entité dont elles découlent, entité rappelée dans leur nommage. Un SFC permettra ainsi de manipuler une entité de service de la vue métier à travers différentes opérations. Ainsi, la fonction élémentaire *SEF message issuing capability* sera-t-elle contenue dans le composant *SFC message*. Cet îlot "*message*" offrira alors une opération *SFO issue message* et sera responsable de la donnée produite par cette opération, c'est-à-dire *SFD issued message*. Avec ce mode de regroupement, les îlots fonctionnels combinent deux sources de légitimité. Ils sont à la fois issus de la vue métier de part l'entité qu'ils gèrent et issus des motifs fonctionnels de part les opérations qu'ils offrent sur cette entité.

III.2.3 Evaluation et résultats

Tout d'abord, il importe de présenter une alternative possible pour la construction des fonctions élémentaires SEF, alternative que nous n'avons pas retenue ici. Elle consisterait à déduire les SEF des fonctions techniques réalisées par les plates-formes de service et les applications informatiques d'un opérateur télécom, en les généralisant et en les abstrayant. Une fonction comme "acheminer un message SIP", rendue par le cœur IMS et ses Application Servers, pourrait être abstraite en une fonction élémentaire *SEF phone call delivering*

capability. Cette alternative n'a pas été retenue. Même si cela peut être une façon relativement intuitive de procéder pour des ingénieurs, elle présente l'inconvénient de fonder les SEF non sur les capacités d'action offertes aux acteurs des services, mais sur les capacités techniques offertes par les plates-formes de service. Les SEF tendent alors à refléter le fonctionnement de ces plates-formes et non le fonctionnement du service ; et nous quittons subrepticement le point de vue de l'utilisation pour prendre le point de vue de la production des services. Un ingénieur aura ainsi tendance à considérer qu'un service comme Skype ou un service conversationnel basé sur l'IMS ne réalisent pas les mêmes SEF car leur réalisation technique est très différente – peer-to-peer dans un cas et centralisée dans l'autre –, alors que le fonctionnement du service en lui-même, en dehors de tout aspect technique et non-fonctionnel comme la QoS ou la sécurité, est relativement similaire. De plus, l'expérience montre que la tâche d'abstraction du fonctionnement des plates-formes de service est délicate en termes organisationnels. Chaque équipe projet tend naturellement à mettre en avant les spécificités de sa plate-forme plutôt que les fonctions communes, l'existence organisationnelle de l'équipe étant justifiée par ces spécificités. Il est donc difficile pour les équipes projet de construire elles-mêmes une vue fonctionnelle de référence qui sera commune à tous les projets. D'un autre côté, la construction par une équipe tierce d'urbanistes, non impliqués dans les différents projets, nécessite de la part de ces urbanistes une expertise pointue sur l'ensemble des plates-formes de service existantes, afin de pouvoir en abstraire et généraliser le fonctionnement. Mais cette expertise pointue est généralement l'apanage des personnes en contact avec les technologies, et donc des équipes projets, non d'une équipe transverse d'urbanistes. Les urbanistes devraient alors réunir des experts des différentes équipes projets pour les écouter et définir avec eux les fonctions communes, l'apport des urbanistes étant principalement méthodologique. Cette démarche peut donner des résultats, mais ne fonctionne que dans une ambiance collaborative, lorsque les pesanteurs organisationnelles ne sont pas trop fortes. De plus, les fonctions définies risquent d'être de granularité différente selon les domaines.

Déduire les SEF des activités de processus de service corrélés avec des motifs fonctionnels présente l'avantage d'une démarche objective, en partant des processus de service. La constitution de la vue fonctionnelle de référence n'est pas ainsi contaminée par les enjeux techniques des différentes équipes projets, en restant uniquement au niveau des capacités d'actions offertes. Cela permet de partir du service rendu, qui est naturellement exprimé dans un langage non technique (pas un jargon technique), ce qui facilite la tâche de

généralisation. Cette approche peut d'ailleurs servir de base à une équipe d'urbanistes qui collectera auprès des équipes projets non plus le fonctionnement de leurs plates-formes, mais les processus de service qu'elles réalisent, ces processus étant beaucoup plus simples à établir objectivement. Cela permet également de travailler a priori, sur des services non encore déployés par un opérateur télécom et non uniquement sur un nombre de souches de plates-formes ou d'applications suffisamment large pour pouvoir généraliser leur fonctionnement. La seule condition est que les actions de base permises par ces services puissent être identifiées, c'est-à-dire qu'elles soient suffisamment répandues (par exemple chez d'autres acteurs que les opérateurs télécoms, ou alors non informatisées) pour qu'elles puissent être nommées.

Par contre, cela pourrait présenter l'inconvénient de l'exhaustivité. Faudra-t-il analyser tous les services offerts par un opérateur télécom sous forme de processus d'usage pour en déduire les SEF ? Il n'en est rien, car seuls certains services, les plus représentatifs, doivent être analysés, les autres services pouvant se résumer à une composition d'éléments existants dans ces services significatifs (avec éventuellement des aspects techniques différents, comme nous le verrons dans le chapitre suivant). En effet, les différents services télécoms sont le plus souvent une recombinaison d'éléments existants dans d'autres services, comme cela avait été déjà noté dans le Réseau Intelligent avec les éléments de service évoqués dans le premier chapitre, un service étant une composition d'éléments de services et ces éléments de services étant en nombre fini (22 pour les spécifications RI CS1). De même, un nombre fini de SEF permet de décrire l'ensemble des services offerts par un opérateur tel que France Telecom. Nous les appellerons les SEF de référence. Deux cas se présentent donc pour décrire un nouveau service. Soit ce nouveau service peut être décrit avec les SEF de référence (ce qui est la très grande majorité des cas), soit ce nouveau service contient réellement une rupture en terme de capacité d'action fournie aux acteurs du service et doit être décrit sous forme de processus de service afin d'en déduire les concepts manipulés et la ou les nouvelles SEF qui permettront de décrire ce nouveau service, généralement en combinaison avec des SEF de référence. Cette ou ces nouvelles SEF ne seront pas incorporées immédiatement au corpus des SEF de référence. Elles ne seront considérées comme de référence que lorsque les services les nécessitant pèseront suffisamment dans la stratégie de l'entreprise.

Cette discussion nous amène à la question du cycle de vie des SEF. La proximité des SEF avec les processus de service permet de relier les SEF avec une stratégie d'entreprise, plus facilement que dans une approche qui partirait des applications existantes, lesquelles ne

se relie pas directement à la stratégie de l'entreprise. Il est en effet difficile de déterminer le poids d'une application dans la stratégie de l'entreprise, alors qu'il est beaucoup plus aisé de déterminer le poids d'un service offert. Une validation des processus de service par l'organisation en charge du marketing d'une entreprise peut garantir l'alignement des activités métiers de ces processus avec la stratégie de l'entreprise. Le cycle de vie d'une SEF est donc le suivant. Une SEF apparaît pour formaliser une nouvelle capacité d'action introduite par un nouveau service. Cette SEF devient de référence lorsqu'un ou des processus de service contenant l'activité dont elle est issue sont considérés (par exemple, par une entité de marketing stratégique) comme pesant suffisamment dans la stratégie de l'entreprise. Cette SEF est retirée des SEF de référence lorsque le ou les processus de service contenant l'activité dont elle est issue ont perdu tout poids dans la stratégie de l'entreprise. La stratégie de l'entreprise évoluant indépendamment des SEF, la cohérence entre processus d'usage et SEF peut être vérifiée à chaque évolution à l'aide d'outils automatiques, comme décrit dans [Simonin, 2009] avec la définition d'une métrique mesurant l'alignement entre processus d'usage et SEF.

Par ailleurs, nous verrons au chapitre V que des outils informatiques peuvent être mis en place pour que chaque projet décrive son architecture fonctionnelle à partir des fonctions de la vue fonctionnelle de référence.

Examinons maintenant les SFO et les SFD correspondant aux processus de service décrits dans la section précédente : envoyer et consulter des messages, effectuer un appel téléphonique, choisir et jouer un contenu multimédia, accéder à un service souscrit. En appliquant notre méthode détaillée ci-dessus, nous obtenons les SFO.

Tableau 5 : Liste de SFO correspondant aux processus de services précédents

SFO	Issue du GFO	Issue de l'activité
SFO define access right	GFO define	Grant access right (fulfillment process)
SFO enforce access right	GFO enforce	Creating authentication session
SFO define access token	GFO define	Provide access (fulfillment process)
SFO define catalog	GFO define	-
SFO define communication session	GFO define	Answering the call
SFO browse content	GFO browse	Choosing audiovisual content
SFO deliver content	GFO deliver	Watching or listening audiovisual content
SFO issue content	GFO issue	Sending audiovisual content

SFO publish content	GFO publish	-
SFO define delivery device	GFO define	-
SFO browse message	GFO browse	Consulting messages
SFO deliver message	GFO deliver	Delivering message
SFO define message folder	GFO define	-
SFO issue message	GFO issue	Sending the message
SFO publish message	GFO publish	Synchronizing mailbox
SFO define messaging rule	GFO define	-
SFO deliver phone call	GFO deliver	Delivering call
SFO issue phone call	GFO issue	Requesting a call
SFO browse address	GFO browse	Selecting callee address (phone call process) / selecting addresses (messaging process)
SFO define address list	GFO define	-
SFO publish address	GFO publish	-
SFO define registration	GFO define	-
SFO enforce registration	GFO enforce	-
SFO define routing rule	GFO define	-
SFO enforce routing rule	GFO enforce	Delivering call

Nous constatons que certains SFO ne sont issus d'aucune activité. En effet, les motifs fonctionnels nous incitent à prendre en compte des SFO pertinents pour lesquels nous n'avons pas identifié d'activités. Ceci nous amène en retour à retravailler la vue métier de référence pour y proposer ces nouvelles activités. A titre illustratif, pour le processus de sélection et de jeu d'un média audiovisuel, le motif de diffusion nous pousse à ajouter la publication du contenu et la constitution d'un catalogue, ce qui est tout à fait pertinent. De plus, le motif de communication nous permet de remarquer l'oubli d'une activité de sélection de l'équipement (player vidéo ou audio) à utiliser pour jouer le contenu. Le processus pourrait ainsi devenir le suivant (les modifications sont colorées). Nous voyions ainsi comment les motifs fonctionnels nous permettent rétroactivement d'harmoniser et de fiabiliser la vue métier de référence.

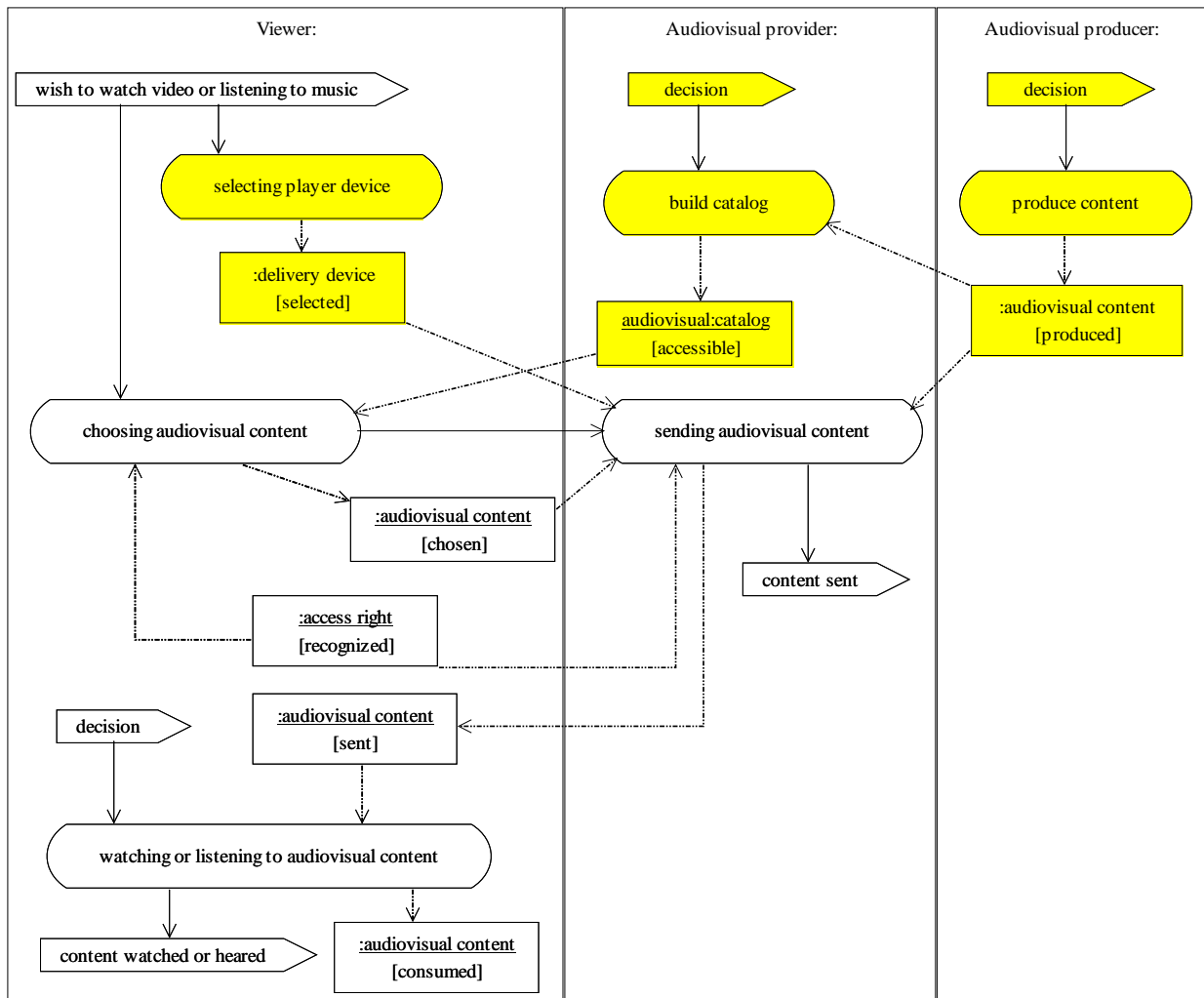


Figure 51 : Amélioration du processus de sélection et de jeu d'un média

Les activités ajoutées ont bien une pertinence pour l'utilisation du service. La vue métier de référence peut donc être consolidée par le travail sur les motifs fonctionnels.

Par ailleurs, les SFO obtenus peuvent être définis en détail et reliés aux SFD qu'ils produisent, comme dans le tableau suivant.

Tableau 6 : Description des SFO et des SFD obtenus à partir des processus de service précédents

Nom formel du SFO	Nom convivial du SFO	Description	Nom formel du SFD produit	Nom convivial du SFD généré	Description
SFO define access right	manage right	création ou modification d'un droit d'accès à un service ou à des données pour un utilisateur, cette création pouvant être faite par le service, le SI ou par un autre utilisateur (autorisation d'accès à des données personnelles)	SFD defined access right	access right data	droit d'accès à une cible (par exemple un service, un contenu, un contexte)
SFO enforce access right	authenticate	authentification d'un utilisateur et contrôle de ses droits à accéder à une cible (service, contenu...)	SFD enforced access right	service access token data	droit d'accès à un service dont la validité a été vérifiée dans un contexte particulier d'accès à une cible (service, contenu...) par un demandeur authentifié par son access token
SFO define access token	manage identity	création ou modification d'un compte ou d'une identité chez un fournisseur de service, en incluant la création des éventuels credentials associés	SFD defined access token	identity data	identité utilisateur et credentials associés
SFO define catalog	manage catalog	création ou modification et mise à disposition d'un catalogue de contenu	SFD defined catalog	content catalog data	catalogue de contenu créé ou modifié par un agrégateur de contenu
SFO define communication session	setup communication session	établissement d'une session de communication suite à un appel téléphonique accepté, avec la possibilité de mise en garde, reprise, fin	SFD defined communication session	communication session data	communication téléphonique établie suite à un appel

SFO browse content	access catalog	navigation dans un catalogue de contenu (y compris la recherche) pour sélectionner un élément	SFD selected content	targeted content	élément d'un catalogue de contenu choisi par un utilisateur, par exemple suite à une navigation dans le catalogue
SFO deliver content	render content	remise par un diffuseur d'un contenu media à un utilisateur, individuellement ou en broadcast, en jouant ce contenu, avec la possibilité de pause, reprise, arrêt	SFD delivered content	media render data	contenu media reçu par le destinataire de ce contenu, avec la possibilité de pause, reprise, arrêt
SFO issue content	serve content	émission d'un contenu media à partir d'un point de stockage de ce contenu (par ex organiquement une tête de réseau)	SFD issued content	media stream data	contenu media émis vers un destinataire
SFO publish content		publication d'un contenu media par un ayant droit de ce contenu	SFD published content	content master data	contenu audiovisuel ou audio publié par l'ayant droit d'un contenu
SFO define delivery device		choix du media player auquel diffuser un contenu	SFD defined delivery device	media player data	adresse de la cible d'une émission de contenu
SFO browse message	display message	navigation dans une liste de messages reçus (y compris la recherche), éventuellement pour sélectionner un élément	SFD selected message	read message data	message affiché ou ayant été affiché
SFO deliver message	message delivery	remise d'un message dans la boîte aux lettres de son ou ses destinataires	SFD delivered message	incoming message data	message reçu par le destinataire de ce message
SFO define message folder	manage message box	création et modification d'un espace de stockage de message (boîte aux lettres) et de l'arborescence de dossiers associée	SFD defined message folder	message folder data	espace de stockage de message (boîte aux lettres) et arborescence de dossiers associée

SFO issue message	send message	composition et envoi d'un message texte vers un ou plusieurs destinataires, en incluant l'ajout d'attachements	SFD issued message	sent message data	message composé et émis par un utilisateur vers un ou plusieurs destinataires
SFO publish message	store message	stockage, classement ou suppression d'un message reçu dans une boîte aux lettres, soit par une règle, soit manuellement	SFD published message	stored message data	message classé automatiquement ou manuellement par un utilisateur
SFO define messaging rule	edit messaging rule	définition de règle de redirection, filtrage ou classement des messages reçus	SFD defined messaging rule	messaging rule data	règle de redirection, de filtrage ou de classement des messages reçus
SFO deliver phone call	route incoming call	remise d'un appel téléphonique entrant	SFD delivered phone call	incoming call request data	appel téléphonique reçu par un utilisateur
SFO issue phone call	route outgoing call	émission d'un appel téléphonique sortant vers un destinataire	SFD issued phone call	outgoing call request data	appel téléphonique émis par un utilisateur
SFO browse address	display directory	navigation dans une liste d'adresses de joignabilité (y compris la recherche), éventuellement pour en sélectionner un élément en vue de communiquer	SFD selected reachable address	communication target data	adresse de joignabilité choisie par un utilisateur en vue d'une action (par exemple l'envoi d'un message)
SFO define address list	set contact list	Mise en place d'une liste d'adresses (carnet d'adresses, liste de contacts, annuaire...)	SFD defined address list	address book data directory	Liste d'adresses de joignabilité comme un carnet d'adresses ou un annuaire

SFO publish address		création ou modification d'une adresse de joignabilité, par exemple dans un annuaire ou dans un carnet d'adresses	SFD published address	address record data	adresse créée ou modifiée, soit par un utilisateur, soit par un administrateur (par exemple dans un annuaire)
SFO define registration	register terminal	établissement d'une association temporaire entre un équipement connecté au réseau et une adresse de joignabilité (par exemple un n° de téléphone)	SFD defined registration	terminal registration data	association temporaire entre une adresse de joignabilité et un équipement connecté au réseau
SFO enforce registration		détermination de l'enregistrement à utiliser dans un cas donné	SFD enforced registration		
SFO define routing rule	edit communication rule	définition de règle de renvoi, filtrage des appels reçus	SFD defined routing rule	reachability rule data	règle de renvoi, filtrage des appels reçus
SFO enforce routing rule		détermination de la règle de routage à utiliser dans un cas donné	SFD enforced routing rule		

Les SFD utilisées en paramètre d'entrée par chaque SFO sont indiquées dans le tableau suivant.

Tableau 7 : Données fonctionnelles utilisées par chaque SFO

SFO	SFD utilisées en input par la SFO
SFO define access right	SFD defined access token
SFO enforce access right	SFD defined access right, SFD defined access token
SFO define access token	-
SFO define catalog	-
SFO define communication session	SFD delivered phone call
SFO browse content	SFD defined catalog, SFD published content, SFD enforced access right
SFO deliver content	SFD issued content
SFO issue content	SFD selected content, SFD enforced access right, SFD delivery device
SFO publish content	SFD defined catalog
SFO define delivery device	-
SFO browse message	SFD published message, SFD defined message folder, SFD enforced access right
SFO deliver message	SFD issued message
SFO define message folder	SFD enforced access right
SFO issue message	SFD selected address, SFD enforced access right
SFO publish message	SFD delivered message, SFD defined routing rule, SFD defined message folder
SFO define messaging rule	SFD enforced access right, SFD defined message folder
SFO deliver phone call	SFD issued phone call, SFD enforced access right, SFD enforced routing rule, SFD enforced registration
SFO issue phone call	SFD selected address, SFD enforced access right
SFO browse address	SFD enforced access right, SFD defined address list, SFD published reachable address
SFO define address list	-
SFO publish address	SFD defined address list, SFD enforced access right
SFO define registration	-
SFO enforce registration	SFD defined registration, SFD enforced access right
SFO define routing rule	-
SFO enforce routing rule	SFD defined routing rule

Les SEF correspondant à chaque SFO sont indiquées dans le tableau suivant.

Tableau 8 : Fonctions élémentaires correspondant à chaque SFO

SFO	SEF correspondante
SFO define access right	SEF access right defining capability
SFO enforce access right	SEF access right enforcing capability
SFO define access token	SEF access token defining capability
SFO define catalog	SEF catalog defining capability
SFO define communication session	SEF communication session defining capability
SFO browse content	SEF content browsing capability
SFO deliver content	SEF content delivering capability
SFO issue content	SEF content issuing capability
SFO publish content	SEF content publishing capability
SFO define delivery device	SEF delivery device defining capability
SFO browse message	SEF message browsing capability
SFO deliver message	SEF message delivering capability
SFO define message folder	SEF message folder defining capability
SFO issue message	SEF message issuing capability
SFO publish message	SEF message publishing capability
SFO define messaging rule	SEF messaging rule defining capability
SFO deliver phone call	SEF phone call delivering capability
SFO issue phone call	SEF phone call issuing capability
SFO browse address	SEF address browsing capability
SFO define address list	SEF address list defining capability
SFO publish address	SEF address publishing capability
SFO define registration	SEF registration defining capability
SFO enforce registration	SEF registration enforcing capability
SFO define routing rule	SEF routing rule defining capability
SFO enforce routing rule	SEF routing rule enforcing capability

Le contenu des deux tableaux précédents peut également être représenté sous la forme graphique de diagrammes de classes avec des liens de dépendance entre SFO et SFD. Un diagramme de classe peut par exemple représenter les opérations SFO d'un composant SFC et leurs dépendances. A titre illustratif, voici les diagrammes de classe de quelques SFC.

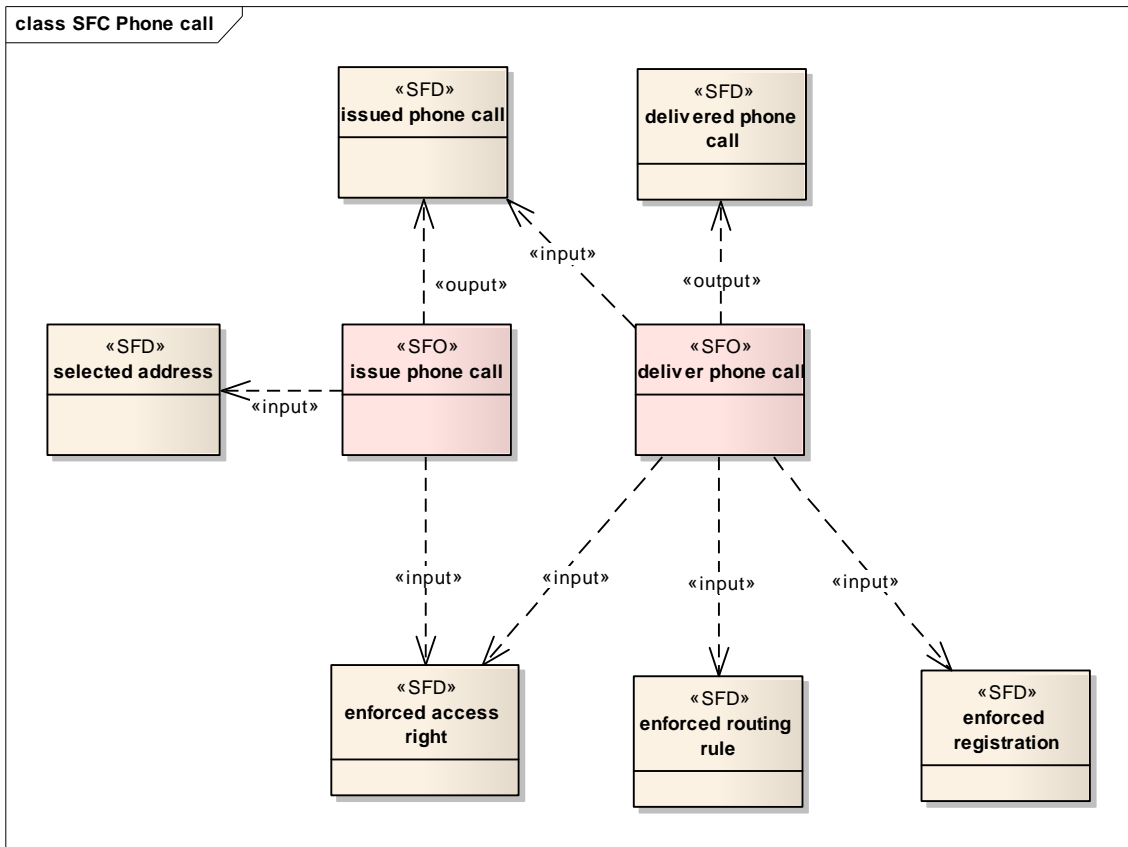


Figure 52 : Opérations offertes par *SFC Phone call*

On reconnaît aisément le motif fonctionnel de communication, uniquement complété par les droits d'accès.

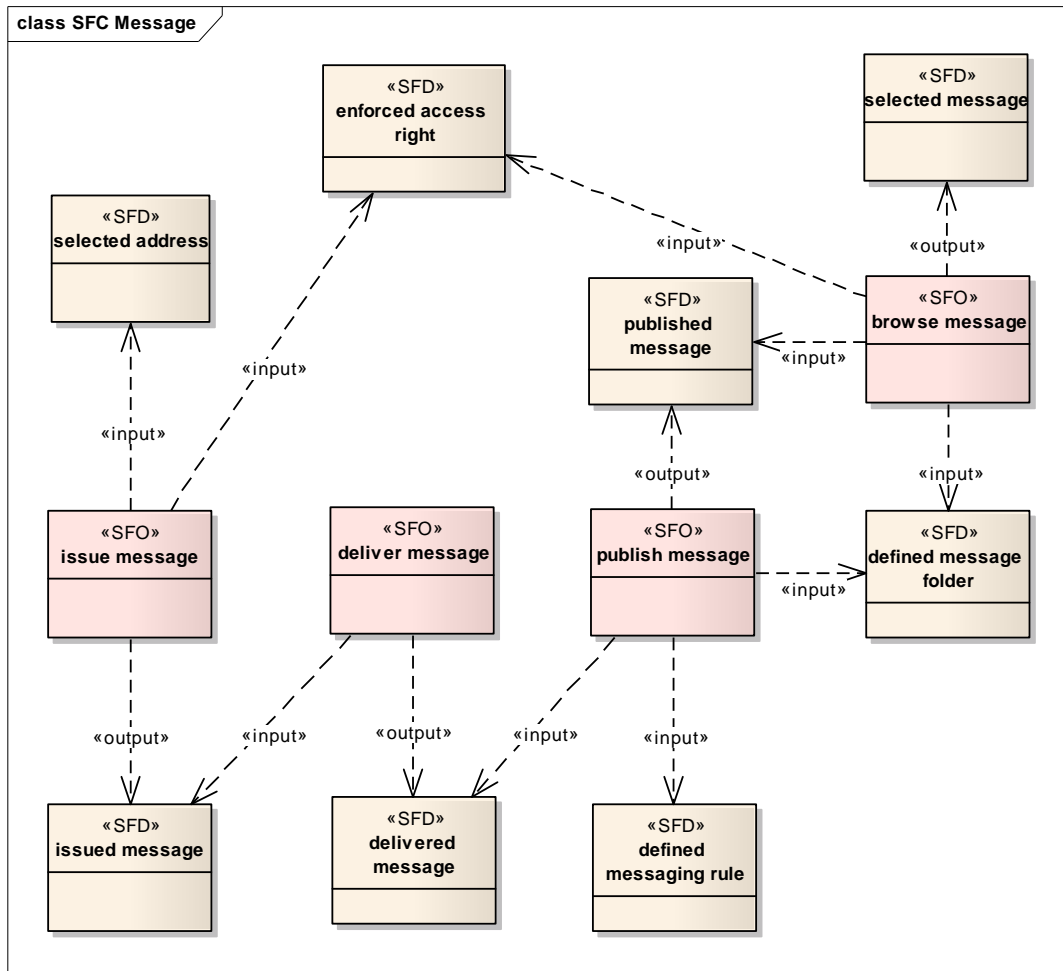


Figure 53 : Opérations offertes par *SFC Message*

Ici, l'on peut identifier un enchaînement du motif fonctionnel de communication (pour l'envoi/remise du message) puis du motif de diffusion (pour la publication/consultation du message). L'opération de publication du message correspond à l'activité de synchronisation de la boîte de message. Les règles de classement des messages dans la messagerie ne sont pas des règles de routage du motif de communication, mais des règles liées à la publication du message.

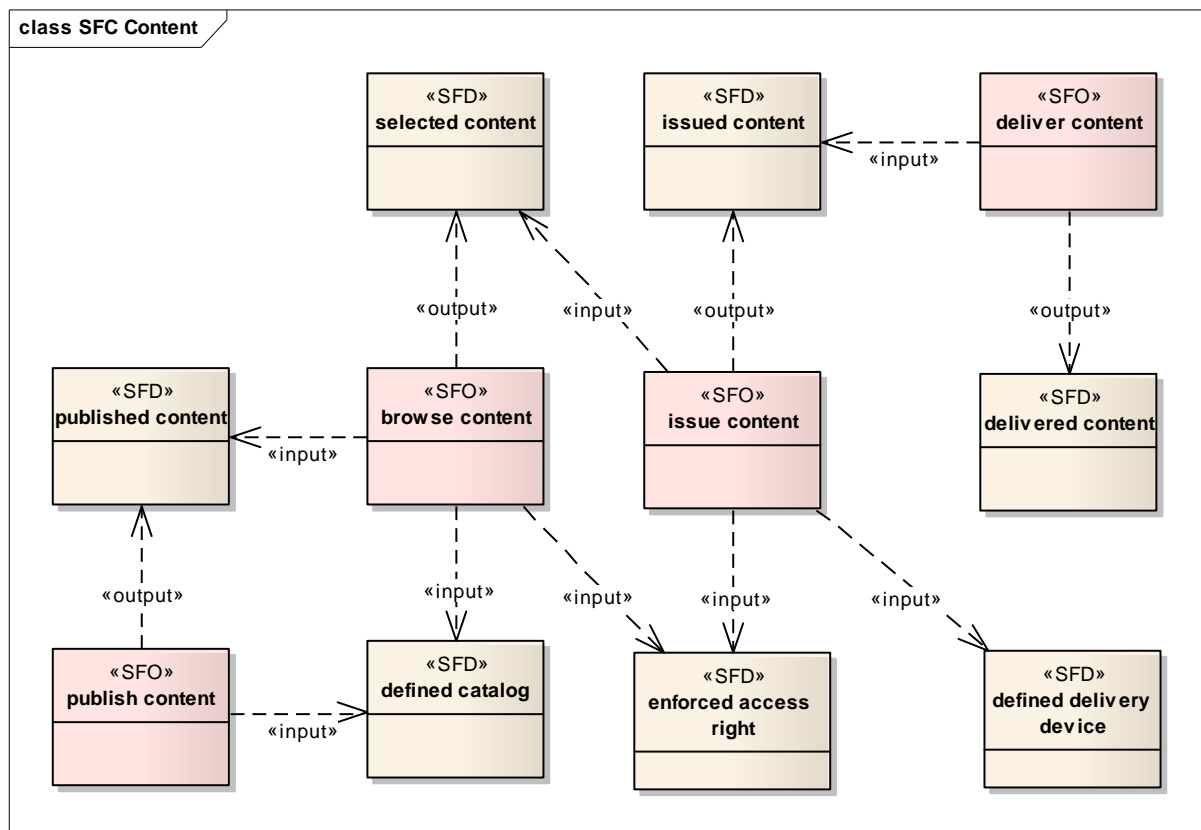


Figure 54 : Opérations offertes par *SFC Content*

Ici, l'on reconnaît un enchaînement du motif de diffusion (pour la publication puis la sélection du contenu), puis du motif de communication (pour l'envoi du contenu et la remise à son destinataire). *SFD defined delivery address* correspond à l'adresse de destination du motif de communication. Cet élément, non identifié au niveau de la vue métier, est néanmoins un élément essentiel, correspondant au choix du player media de réception.

Les regroupements des fonctions élémentaires SEF en composants fonctionnels SFC, puis en quartiers fonctionnels et en zones fonctionnelles sont détaillés dans le tableau ci-dessous. Le regroupement en SFC se fait comme expliqué à la fin de la section III.2.2. Le regroupement en quartiers et zones est ici proposé de façon arbitraire, suivant la proximité des fonctions manipulées. L'enjeu des quartiers et des zones est principalement un enjeu de classement et de compréhension plus rapide de la vue fonctionnelle de référence. Comme pour les îlots, une conception automatisée peut être réalisée, comme proposé dans [2007-10] ou [Simonin, 2009].

Tableau 9 : Regroupement en quartiers et zones fonctionnelles

Fonction élémentaire SEF	Composant fonctionnel SFC	Quartier fonctionnel	Zone fonctionnelle
SEF access right defining capability	SFC access right	identity and rights	user characteristics
SEF access right enforcing capability			
SEF access token defining capability	SFC access token	identity and rights	user characteristics
SEF catalog defining capability	SFC catalog	content publication and access	Multimedia
SEF communication session defining capability	SFC communication session	communication setup	live communication
SEF address browsing capability	SFC address	address book	user characteristics
SEF address publishing capability			
SEF address list defining capability	SFC address list	address book	user characteristics
SEF content browsing capability	SFC content	content publication and access	Multimedia
SEF content delivering capability			
SEF content issuing capability			
SEF content publishing capability			
SEF delivery device defining capability	SFC delivery device	content control	Multimedia
SEF message browsing capability	SFC message	message publication and access	messaging
SEF message delivering capability			
SEF message issuing capability			
SEF message publishing capability			
SEF message folder defining capability	SFC message folder	messaging control	messaging
SEF messaging rule defining capability	SFC messaging rule	messaging control	messaging
SEF phone call delivering capability	SFC phone call	communication setup	live communication
SEF phone call issuing capability			
SEF registration defining capability	SFC registration	communication control	live communication
SEF registration enforcing capability			
SEF routing rule defining capability	SFC routing rule	communication control	live communication
SEF routing rule enforcing capability			

Nous obtenons ainsi 4 zones fonctionnelles :

- Messaging, composé des quartiers fonctionnels
 - Messaging control,
 - Message publication and access ;
- Live communication, composé des quartiers fonctionnels
 - Communication control,
 - Communication setup ;
- Multimedia, composé des quartiers fonctionnels
 - Content publication and access,
 - Content control ;
- User characteristics, composé des quartiers fonctionnels
 - Address Book,
 - Identity and rights.

Ces zones pourraient être complétées par l'étude d'autres types de services télécoms dans la vue métier. Ainsi, la prise en compte des services de présence conduirait sans doute à ajouter le contexte utilisateur à la zone User characteristics.

Nous pouvons représenter le contenu des zones et des quartiers par des diagrammes de classe. Ainsi, le diagramme suivant présente les SFC et leurs liens d'utilisation et de production de SFO pour les quartiers Communication setup et Address book.

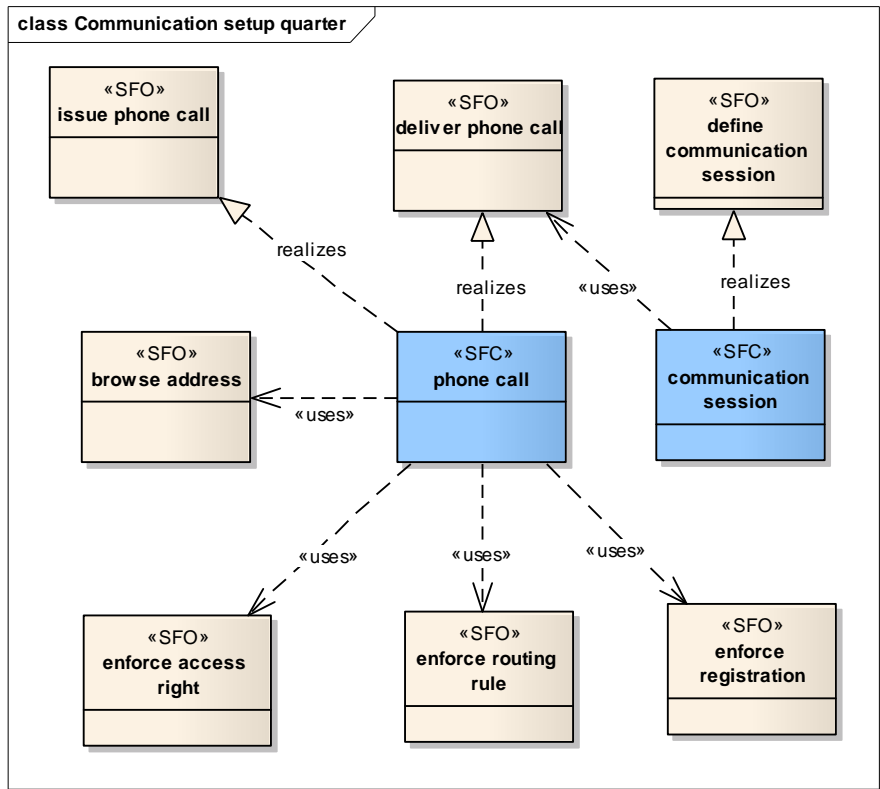


Figure 55 : Quartier Communication setup

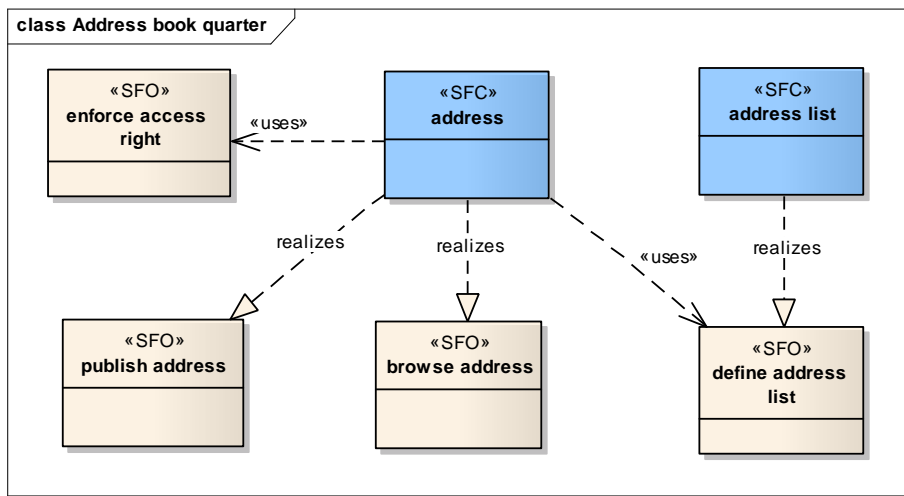


Figure 56 : Quartier Address book

Après cette étude d'une vue fonctionnelle de référence pour les services télécoms, nous allons maintenant aborder la construction d'une vue technique de référence, vue technique pleinement décorrélée de la vue fonctionnelle.

III.3 Une vue technique de référence pour les services télécoms

Deux services rendant le même service et dont le déroulement est proche sinon identique peuvent néanmoins être réalisés techniquement de façon très différente. Par exemple, Skype ou MSN messenger rendent un service relativement similaire (du moins en ce qui concerne leurs nombreuses fonctionnalités communes) alors que l'un est réalisé de façon distribuée à travers un réseau de pairs et l'autre est réalisé de façon centralisé en mode client/serveur. Il nous faut donc pour décrire complètement un système de service être capable de distinguer ces deux cas. Cela passe par une étude des solutions techniques que nous avons volontairement laissées de côté dans notre parcours des sections précédentes afin de dissocier les considérations techniques des considérations sur le service rendu et sur son déroulement, considération trop souvent imbriquées. Nous allons donc maintenant étudier comment constituer une vue technique de référence des services télécoms, en complément des vues métier et fonctionnelle de référence. Nous verrons dans le chapitre suivant comment relier ces différentes vues pour décrire l'architecture d'un service dans sa globalité.

III.3.1 Comment définir les éléments de la vue technique ?

La vue technique des systèmes d'information d'entreprise est définie dans [Longépé, 2006] comme décrivant *"l'ensemble des matériels, logiciels de base et technologies utilisées"*, l'architecture technique étant *"la structuration des moyens d'infrastructure technique à mettre en œuvre pour informatiser l'activité de l'entreprise"*. De façon similaire, la vue technique de référence pour les services télécoms contiendra les solutions techniques qui peuvent être mises en œuvre pour informatiser le déroulement des services.

Pour passer des SEF aux solutions techniques, nous pourrions procéder, comme dans l'analyse fonctionnelle vue au chapitre II, par raffinements successifs des composants fonctionnels. Considérons la *SEF call issuing capability*, capacité à émettre un appel. Une fonction technique subséquente serait par exemple "émettre une requête SIP" ou "émettre une requête H.323". "Emettre une requête SIP" pourrait également être un raffinement d'autres fonctions, par exemple *"personal context publishing capability"*, pour publier un contexte de présence. Nous obtiendrions ainsi un arbre de fonctions techniques, chaque niveau raffinant le précédent.

Toutefois, un tel inventaire de toutes les fonctions techniques de bas niveau et de leurs liens possibles semble difficile à mener sur tout le périmètre d'un opérateur télécom. D'une part, une multitude de fonctions techniques différentes existent, notamment dans le domaine de services web et media. Autant les fonctionnalités des services télécom classiques se rattachent plus ou moins directement à des protocoles comme SIP, H.323, SMTP... Autant les fonctionnalités des services web et media ne sont pas rattachées à un protocole particulier, mais à un développement logiciel spécifique. Un tel travail est plus du ressort de chaque projet de développement que d'une démarche d'architecture globale.

Le fait que les fonctions de services télécoms classique se rattachent à des standards explique d'ailleurs que le domaine de l'architecture fonctionnelle (à travers une vue fonctionnelle telle que présentée dans la section précédente) soit peu développé dans la communauté des télécoms. Une solution technique – par exemple l'IMS – supporte de façon privilégiée un service spécifique – par exemple la téléphonie – car cette solution technique a été spécifiée dans cet objectif. Dans le monde du logiciel, les solutions techniques (telles que base de données, serveur PHP, moteur de servlet...) sont génériques et ne sont intrinsèquement porteuses d'aucun service particulier, car elles sont programmables, aptes à recevoir un code qui en pilotera le comportement. C'est une différence importante entre services télécoms et services informatiques. Construire une architecture de service télécom consistera ainsi à identifier des fonctions techniques de bas niveau (par exemple "émettre une requête SIP") répondant au besoin du service, en se basant sur les fonctions des standards, puis à en déduire les composants techniques de la solution (terminal SIP, proxy SIP...) et leurs échanges de données via des protocoles. Ceci se rapproche tout à fait de la conception d'un objet technique. Ainsi, dans l'exemple de l'analyse fonctionnelle du vélo vu au chapitre précédent, la fonction technique de transmission de mouvement est réalisée par des échanges d'énergie entre pédalier, chaîne, pignons et roue arrière. La problématique des tests d'intégration et des tests de bout en bout est ainsi particulièrement importante pour les services télécoms. Il s'agit de s'assurer que les divers composants techniques de la solution, censés pouvoir fonctionner ensemble du fait de leur conformité à des standards, sont effectivement compatibles et assurent effectivement ensemble les fonctions techniques de bas niveau retenues par l'architecte du service, une fois assemblés dans leur environnement.

Par contre, concevoir une architecture logicielle consistera à identifier des modules logiciels cohérents pour répondre à un besoin fonctionnel, et à maîtriser le couplage entre ces

composants logiciels. Ces modules logiciels pourront alors être développés par des équipes différentes, évoluer indépendamment durant le cycle de vie du logiciel, être mis à jour séparément, dans le respect de leurs contrats. Le problème de la cohérence et du couplage se pose moins pour les services télécoms classiques, car il a été traité en amont dans la standardisation¹. Le périmètre d'une solution technique est celui défini dans le standard, et son couplage avec les autres composants est spécifié par le protocole. A contrario, l'architecte logiciel doit déterminer lui-même le périmètre des modules logiciels et leur couplage, en fonction de l'environnement utilisé, de critères fonctionnels et non fonctionnels. La cohérence peut notamment être syntaxique (un module gestion des erreurs, un module gestion des interfaces homme-machine...), temporelle (un module initialisation, un module arrêt...), ou fonctionnelle lorsqu'un module regroupe tous les éléments nécessaires à la réalisation d'une fonctionnalité, à la fois pour l'interface, les traitements et les données. Le couplage définit le niveau de dépendance entre modules. Un faible couplage favorise les évolutions en permettant de concentrer les modifications sur des modules identifiés et en minimisant les effets de bord. Une bonne architecture logicielle doit au minimum identifier le niveau de couplage entre les différents modules, afin de faciliter le cycle de vie du logiciel. Dans la mesure du possible, l'architecte logiciel applique le principe de cohérence forte / couplage faible : un module logiciel est fortement cohérent et faiblement couplé avec les autres composants. La cohérence fonctionnelle est habituellement considérée comme la plus forte (on pourra par exemple se référer sur ces niveaux de cohésion à [Pressman, 1992]).

La recherche d'une cohérence fonctionnelle se traduit par un lien de traçabilité entre SEF et modules logiciels, sans passage par un raffinement des fonctions techniques (du moins au niveau de l'architecture). Prenons l'exemple du développement logiciel d'un réseau social incluant une fonctionnalité de messagerie textuelle. Idéalement, l'architecte logiciel cherchera à regrouper toutes les fonctionnalités liées à cette messagerie au sein d'un module indépendant, faiblement couplé avec le reste du logiciel de réseau social. Ce module pourra ainsi évoluer indépendamment du reste du service, par exemple en permettant dans une version ultérieure d'envoyer et d'afficher des images en plus du texte. Les fonctions d'émission de message, de réception de message et de consultation des messages seront donc regroupées dans le même module logiciel. La topologie des éléments techniques

¹ Ce problème se retrouve toutefois au niveau de l'intégration et des tests, les standards ou leurs implémentations pouvant manquer de cohérence ou masquer des couplages forts.

(client/serveur, distribution des modules...) n'apparaît pas car elle est gérée par les couches basses banalisées du logiciel (par exemple, le mécanisme de consultation d'une page web par un navigateur, ou un bus de communication par événements entre modules...). Les mécanismes qui seront utilisés à l'intérieur de ce module logiciel (par exemple un mécanisme de souscription/notification aux messages reçus, ou une consultation régulière d'un entrepôt de messages, ou encore la consultation d'un entrepôt de messages suite à une notification d'arrivée de message) ne sont pas déterminés par l'architecte logiciel du service de réseau social, mais par le développeur de la fonctionnalité de messagerie. En effet, le couplage entre les modules logiciels aura été déterminé de telle façon (par exemple via la définition d'API entre le module de messagerie et les autres modules) que le choix d'un mécanisme ou d'un autre au niveau du module de messagerie n'impactera pas les autres modules. De même, dans une optique web, une fonctionnalité d'émission d'appel téléphonique ne donnerait pas nécessairement lieu à un composant terminal et un composant serveur d'appel, mais plutôt à un module logiciel unique, responsable à la fois de l'interface de saisie et des mécanismes serveur.

Comme nous visons ici à décrire les services de communication, que leur implémentation soit orientée télécom ou logicielle, un raffinement des fonctions techniques des services ne semble donc pas pertinent. Par contre, les solutions techniques existantes sont une contrainte, pour l'architecte logiciel comme pour l'architecte télécom. Dans notre exemple du réseau social, nous nous sommes implicitement situés dans un environnement technique homogène, un environnement web (navigateur, serveur web et base de donnée) au sein duquel est exécuté le service. Or l'architecte doit souvent composer avec des environnements techniques existants et variés. Ainsi, la conception de l'architecture de notre module de messagerie différencierait fortement si la fonctionnalité de messagerie devait être couplée avec un environnement technique IMS. Notre besoin est donc double. D'une part, définir une liste d'environnements techniques et surtout les liens de communication qui permettent le transfert d'information. D'autre part, définir les composants applicatifs réalisant les SEF au sein d'un ou plusieurs environnements techniques, ces composants applicatifs pouvant être aussi bien des modules logiciels, des applications, des plates-formes de service télécom telles que les enablers OMA ou des serveurs de contrôle d'appel.

Dans la suite de cette section, nous allons étudier comment spécifier cette liste d'environnements techniques et de liens de communication, en prenant à la fois en compte les

solutions techniques télécoms et logicielles. Nous pourrions ensuite dans le chapitre IV proposer avec la vue applicative d'un service une description des composants applicatifs comme implémentant une SEF et supportés par un environnement technique.

III.3.2 Comment construire une vue technique de référence ?

Lorsque l'on veut décrire la solution technique supportant un système de service, on décrit habituellement les composants techniques de ce système, son architecture technique. Les composants techniques d'un service de téléphonie seront par exemple un terminal, un serveur d'appel, etc. On observe que les composants techniques et leur topologie sont souvent communs entre différents systèmes de service. Par exemple, l'utilisation d'un style d'architecture en couche de type *3-tier* se retrouvera dans des développements réalisant des services divers.

D'une façon générale, on peut décrire un environnement technique à l'aide de deux concepts : les éléments techniques constitutifs de l'environnement et les liens de communication entre ces éléments techniques. Un lien de communication entre deux éléments techniques permet à ces éléments d'échanger des flux d'information à travers un mécanisme particulier, comme par exemple un protocole ou un appel de procédure distante. Dans notre exemple du vélo, le lien de communication correspondrait au lien entre deux éléments de la solution technique, comme les dents réalisent le lien entre le pédalier et la chaîne. Dans le monde télécom, ce lien est habituellement spécifié en définissant un point de référence couplant ces deux éléments. A la suite de [Menai, 2005], nous définissons ici le point de référence comme une abstraction du couplage entre deux éléments via un protocole spécifique. Cette abstraction est parfois nommée interface dans certains standards. On parle ainsi de l'interface Gm entre UE et P-CSCF. Autrement dit, Gm est une abstraction du couplage entre UE et P-CSCF via le protocole SIP. Nous préférons le terme de point de référence à celui d'interface pour ne pas induire de confusion avec la vue externe d'un composant, nommée également interface. Par souci de simplicité, nous nommerons le point de référence soit par son nom standardisé s'il existe, soit par un nom arbitraire en cas d'absence de standard.

En ce qui concerne les éléments techniques, ils prennent leur sens par leur insertion dans un environnement technique et par leurs couplages avec d'autres éléments à travers des

points de référence. Ainsi, un élément Web Service Consumer ne prend sens que dans un lien avec un Web Service Provider, via un point de référence Web Service. Un élément technique peut ainsi être vu comme un rôle vis-à-vis d'un ou plusieurs points de référence. Nous les appellerons en conséquence ces éléments techniques des rôles techniques. Le point de référence est orienté ; un rôle technique peut en effet soit être à l'origine des échanges avec un autre rôle technique, soit en être la destination, suivant en cela le motif d'échange de message (*message exchange pattern*), classique à la fois dans les architectures client/serveur et dans les architectures des réseaux de télécommunication [Menai, 2005]. Un rôle technique peut donc soit émettre des requêtes vers un autre rôle technique via un point de référence, soit recevoir des requêtes d'un autre rôle technique via un point de référence.

Suivant le modèle en couches classique dans les télécoms, formalisé par exemple dans les normes ITU-T G.805 et G.809, un point de référence ou un rôle technique peut abstraire des mécanismes de plus bas niveau. La figure suivante illustre les concepts de partition et de couches tels que définis à l'ITU

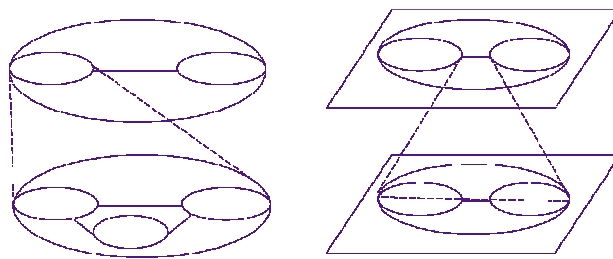


Figure 57 : Partition et décomposition en couches

Un point de référence peut ainsi se décomposer lui-même en deux rôles techniques et un point de référence de niveau inférieur. Un *call flow* de protocole implique typiquement plusieurs rôles techniques en ligne de vie. Et un rôle technique peut se décomposer en rôles techniques et en points de référence plus détaillés. Ainsi, une passerelle SIP/web service sera une composition d'un rôle *Web Service Provider* (pour les messages sortants côté web service), d'un rôle *Web Service Consumer* (pour les messages entrants côté web service) et d'un rôle UE (pour le côté SIP). Dans la définition des rôles techniques, nous n'entrerons pas dans le détail du fonctionnement d'un environnement technique, mais nous intéresserons plutôt aux frontières de cet environnement.

Ces notions peuvent être méta-modélisées comme indiqué dans la figure suivante.

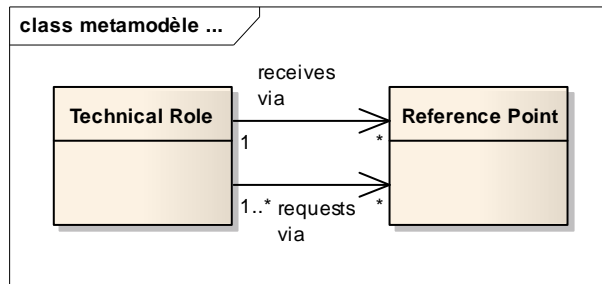


Figure 58 : Méta-modèle de la vue technique de référence

Ces notions seront représentées par des classes UML stéréotypées

- **TR** comme *Technical Role* pour les rôles techniques
- **RP** comme *Reference Point* pour les points de référence

Un point de référence est défini comme unidirectionnel. Lorsque les échanges peuvent être à l'initiative des deux rôles techniques, deux points de référence de même nom et de direction opposée seront définie. Les liens entre Technical Role et Reference Point sont nommés

- "*request via*" lorsque le rôle technique est à l'origine des échanges vers un autre rôle technique via le point de référence
- "*receives via*" lorsque le rôle technique est destinataire d'une requête issue d'un autre rôle technique via le point de référence. Ce rôle technique destinataire peut ensuite bien sûr répondre à cette requête via le point de référence.

III.3.3 Evaluation et résultats

Nous ne chercherons pas ici à identifier de façon exhaustive les environnements techniques, mais à en présenter quelques uns. Dans le cadre de ce travail, nous avons étudié les environnements techniques suivants, choisis de par leur importance actuelle dans le secteur des télécoms ou par leur caractère exemplaire, ainsi que par la diversité des applications couvertes qui laisse augurer de la généralisation et de l'adaptabilité de l'approche.

- L'architecture IMS,
- L'architecture SOA,
- L'architecture web 3-tier,
- L'architecture Internet Email,
- L'architecture SMS,
- L'architecture OMA XDM (XML Document Management),

- L'architecture IPTV,
- Les interfaces homme-machine (GUI, *Graphical User Interface*).

Concernant l'IMS, le motif technique est le suivant, tel que spécifié dans [ETSI, 2005].

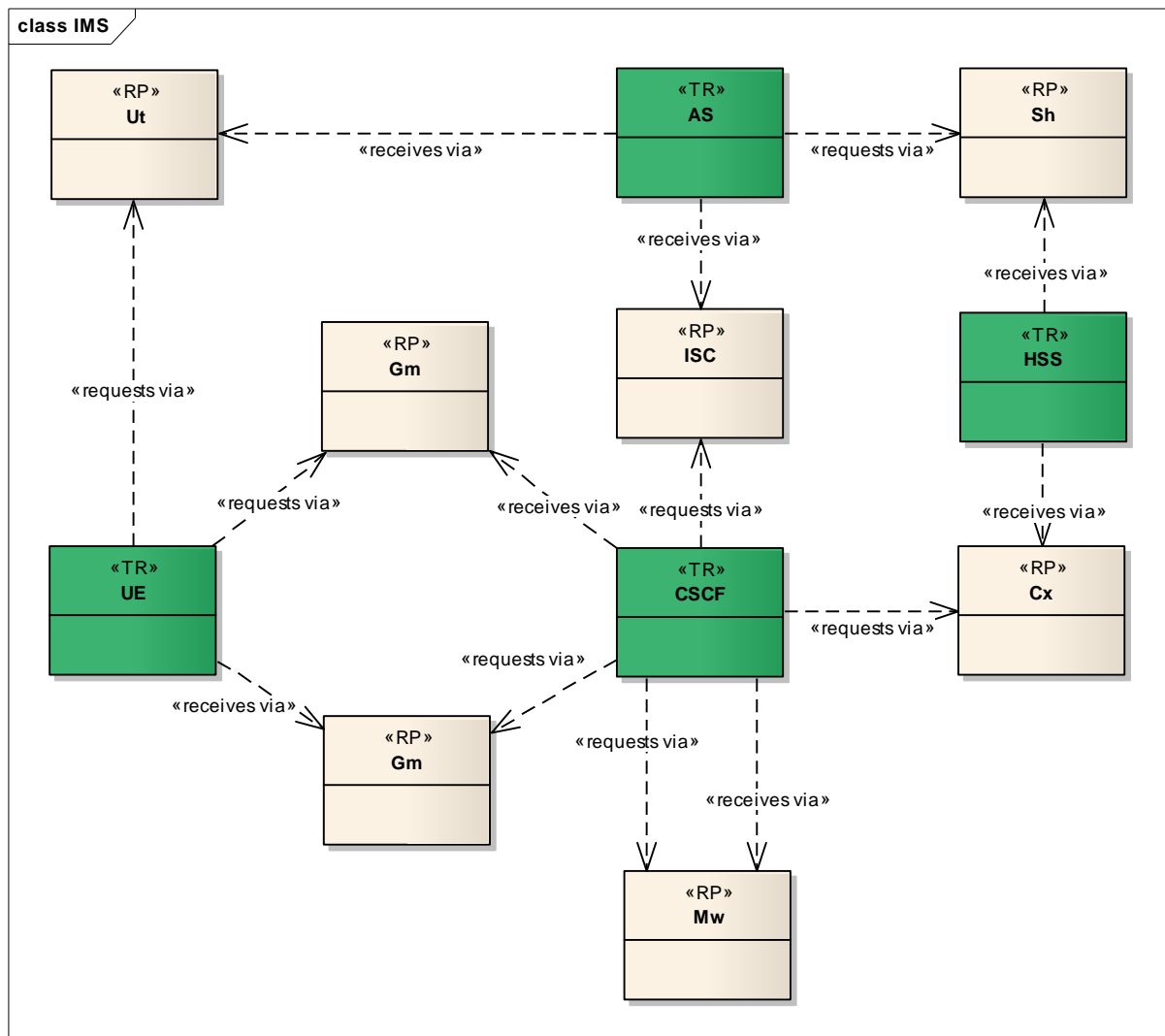


Figure 59 : Motif technique IMS

Les rôles techniques principaux de l'IMS sont les suivants.

Tableau 10 : Rôles techniques du motif technique IMS

Rôle technique	Description
UE pour <i>User Equipment</i>	<p>Le terminal de communication :</p> <ul style="list-style-type: none"> • Initie et reçoit des requêtes de communication via le point de référence Gm • Initie des requêtes vers des applications particulières, par exemple un service de présence, via le point de référence Ut

CSCF pour <i>Call Session Control Function</i>	<p>Le serveur d'appel :</p> <ul style="list-style-type: none"> • Reçoit et émet des requêtes de communication de l'UE via le point de référence Gm • Reçoit des requêtes de communication d'autres CSCF via le point de référence Mw • Emet des requêtes vers le HSS via le point de référence Cx • Emet des requêtes vers les AS via le point de référence ISC
HSS pour <i>Home Subscriber Server</i>	<p>Le serveur d'abonnés :</p> <ul style="list-style-type: none"> • Reçoit les requêtes du CSCF via le point de référence Cx • Reçoit les requêtes des AS via le point de référence Sh
AS pour <i>Application Server</i>	<p>Le serveur de services :</p> <ul style="list-style-type: none"> • Reçoit des requêtes du CSCF via le point de référence ISC • Reçoit des requêtes de l'UE via le point de référence Ut • Emet des requêtes vers le HSS via le point de référence Sh

Les protocoles usuellement choisis pour ces points de référence sont les suivants.

Tableau 11 : Protocoles usuels du motif technique IMS

Point de référence	Protocole
Gm	SIP
Mw	SIP
ISC	SIP
Cx	DIAMETER
Sh	DIAMETER
Ut	HTTP

Concernant le motif SOA (*Service Oriented Architecture*), le motif est le suivant, tel que décrit par exemple par [Channabasavaiah, 2003]. Il s'agit d'une formalisation de l'architecture technique présentée à la section I.2.1 (figure 6).

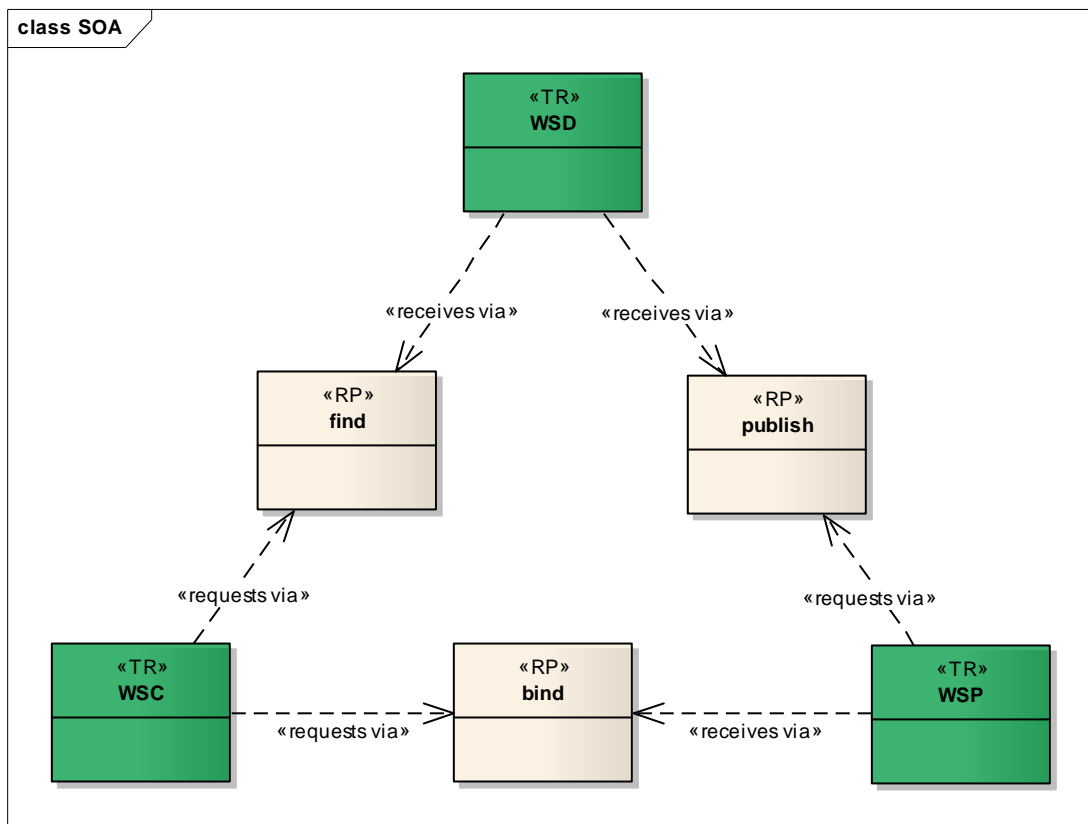


Figure 60 : Motif technique SOA

Les rôles techniques principaux de SOA sont les suivants.

Tableau 12 : Rôles techniques du motif technique SOA

Rôle technique	Description
WSC pour <i>Web Service Consumer</i>	Le client logiciel qui consomme un web service : <ul style="list-style-type: none"> • Recherche des web services via le point de référence find • Invoque des services web via le point de référence bind
WSP pour <i>Web Service Provider</i>	Le client logiciel qui fournit un web service : <ul style="list-style-type: none"> • Reçoit des appels via le point de référence bind • Publie les web services qu'il fournit via le point de référence publish

WSD pour <i>Web Service Directory</i>	Un annuaire de web services : <ul style="list-style-type: none"> • Reçoit les recherches des WSC via le point de référence find • Reçoit les publications des WSP via le point de référence publish
---------------------------------------	---

Les protocoles usuellement choisis pour ces points de référence sont les suivants.

Tableau 13 : Protocoles usuels du motif technique SOA

Point de référence	Protocole
bind	SOAP, REST
find	UDDI
publish	UDDI

Concernant les architectures 3-tiers, le motif classique, tel que décrit par exemple dans [Longépé, 2006], est le suivant.

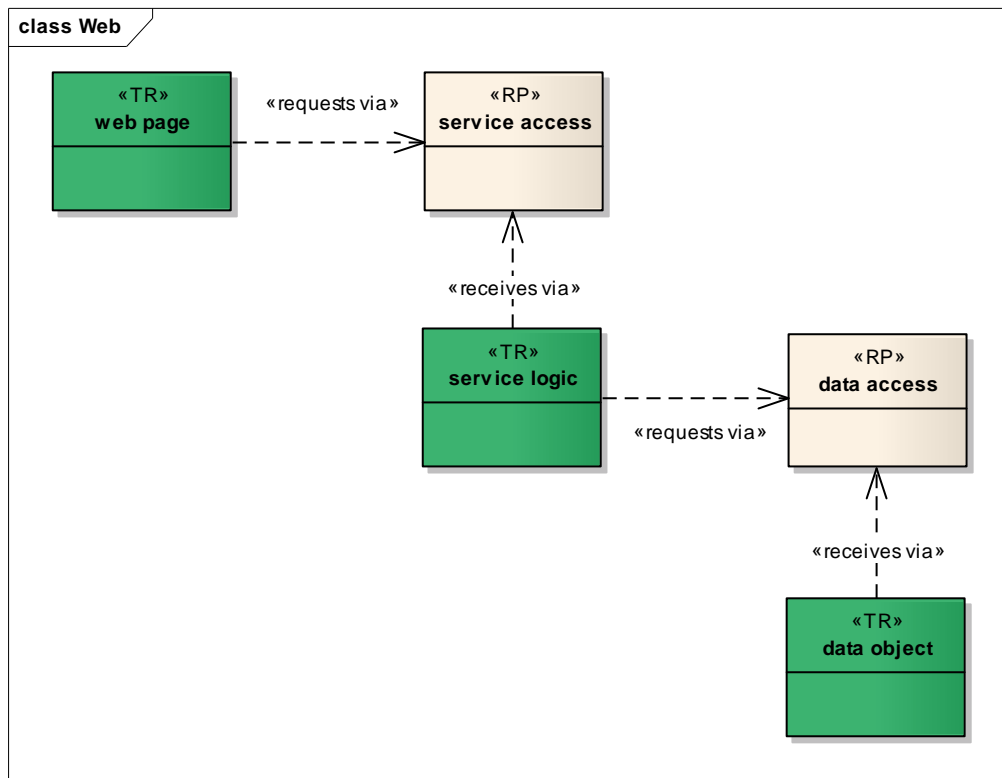


Figure 61 : Motif technique 3-tiers

Les rôles techniques principaux des architectures 3-tiers sont les suivants.

Tableau 14 : Rôles techniques du motif technique 3-tiers

Rôle technique	Description
Web page	La couche de présentation, assurant l'interaction avec l'utilisateur : <ul style="list-style-type: none"> • Emet des requêtes vers la logique de service via le point de référence service access et met en forme les réponses
Service logic	La couche métier, implémentant la logique du service : <ul style="list-style-type: none"> • Reçoit des requêtes des pages web via le point de référence service access • S'appuie sur les données persistantes à travers le point de référence data access
Data object	La couche des données, stockant les données pérennes : <ul style="list-style-type: none"> • Reçoit les requêtes de la logique de service via le point de référence data access

Les protocoles usuellement choisis pour ces points de référence sont les suivants.

Tableau 15 : Protocoles usuels du motif technique 3-tiers

Point de référence	Protocole
Service access	RPC protocols
Data access	Data request languages

Concernant l'email, le motif technique tel que décrit à l'IETF dans le RFC 5598 Internet Mail Architecture de l'IETF est le suivant.

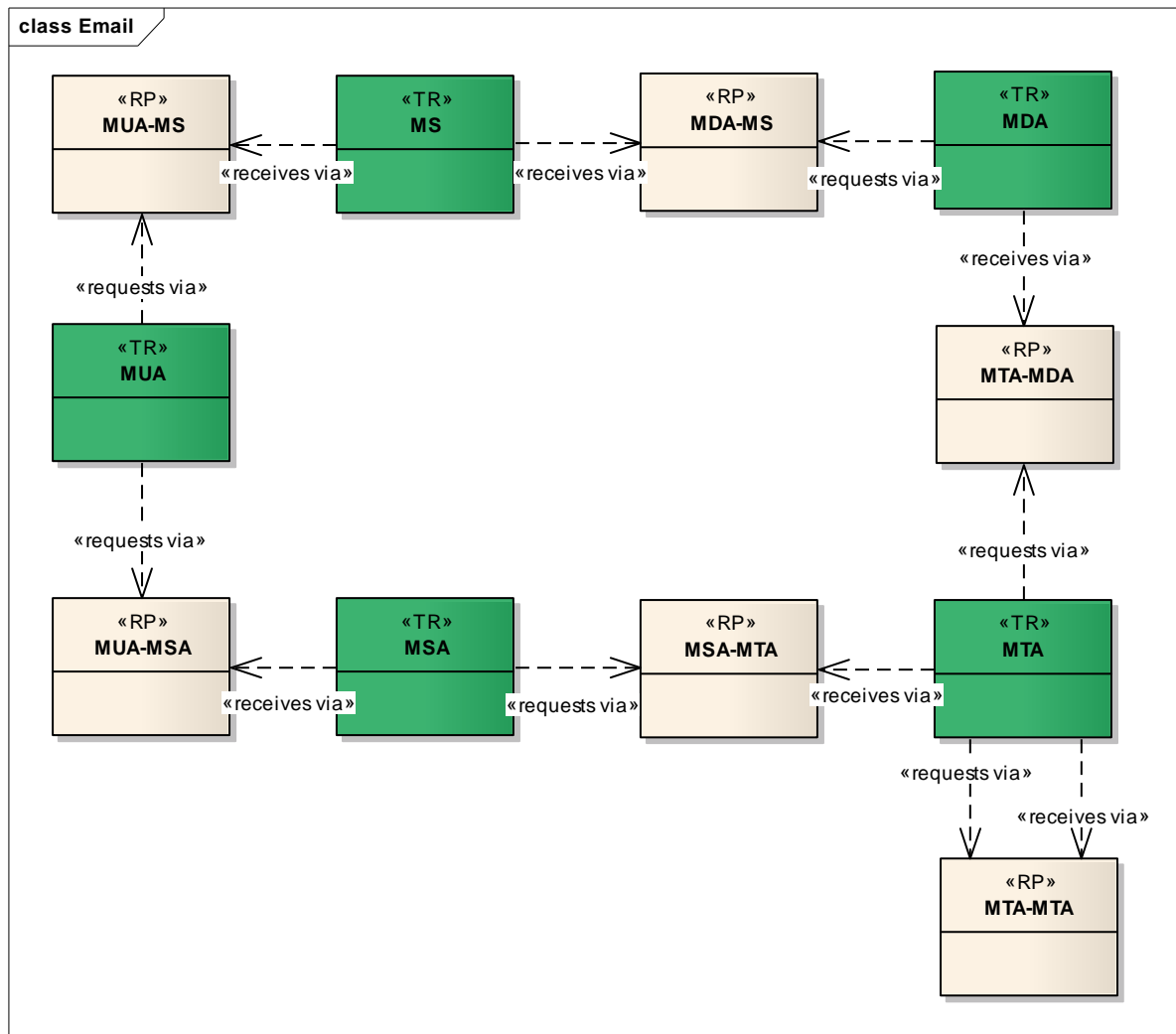


Figure 62 : Motif technique Email

A titre illustratif, le schéma produit par l'IETF est présenté à la figure 63. On y retrouve les éléments MUA, MSA, MTA, MDA et MS, ainsi que les liens entre ces éléments. La moindre formalisation de ce schéma rend toutefois son utilisation moins aisée comme base univoque pour une architecture technique.

Les rôles techniques principaux de la messagerie email sont les suivants.

Tableau 16 : Rôles techniques du motif technique Email

Rôle technique	Description
MUA pour <i>Mail User Agent</i>	L'agent de messagerie : <ul style="list-style-type: none"> • Envoie les emails au MSA via le point de référence MUA-MSA • Récupère les emails auprès du MS via le point de référence MUA-MS
MSA pour <i>Mail Submission Agent</i>	Agent gérant les emails émis par les utilisateurs locaux : <ul style="list-style-type: none"> • Reçoit des messages du MUA via le point de référence MUA-MSA • Relais les messages au MTA via le point de référence MSA-MTA
MTA pour <i>Mail Transfer Agent</i>	Agent gérant les emails issus d'autres domaines : <ul style="list-style-type: none"> • Emet vers ou reçoit des messages d'autres MTA via le point de référence MTA-MTA • Emet des messages vers le MDA
MDA pour <i>Mail Delivery Agent</i>	Agent gérant la remise des emails dans la boîte aux lettres : <ul style="list-style-type: none"> • Reçoit des messages du MTA via le point de référence MTA-MDA • Emet des messages vers le MS via le point de référence MDA-MS
MS pour <i>Message Store</i>	Espace de stockage des emails : <ul style="list-style-type: none"> • Reçoit des messages du MDA via le point de référence MDA-MS • Reçoit des requêtes du MUA via le point de référence MUA-MS

Les protocoles usuellement choisis pour ces points de référence sont les suivants.

Tableau 17 : Protocoles usuels du motif technique Email

Point de référence	Protocole
MUA-MSA	SMTP
MSA-MTA	SMTP
MTA-MTA	SMTP
MTA-MDA	SMTP
MDA-MS	SMTP
MUA-MS	POP, IMAP

Concernant le motif SMS, tel que décrit dans [3GPP, 2001]. On se situe ici au niveau protocolaire applicatif ; chaque point de référence pourrait être décomposé en procédures de niveau protocolaire inférieur.

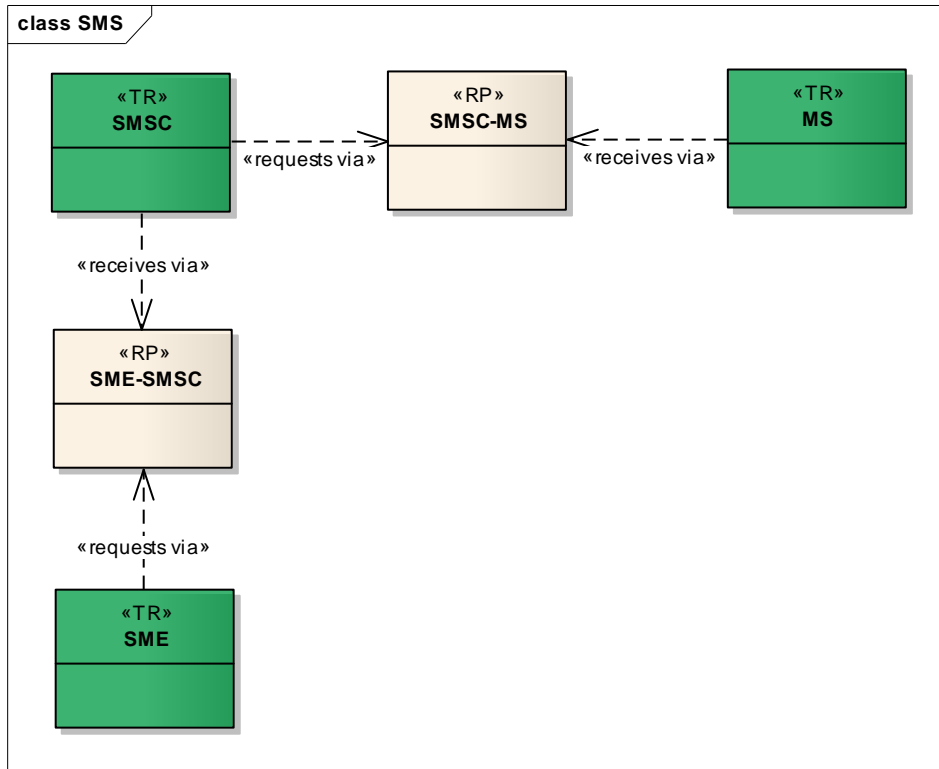


Figure 64 : Motif technique SMS

Les rôles techniques principaux du SMS sont les suivants.

Tableau 18 : Rôles techniques du motif technique SMS

Rôle technique	Description
SME pour <i>Short Message Entity</i>	Entité émettrice de SMS : <ul style="list-style-type: none"> • Envoie les SMS au SMSC via le point de référence SME-SMSC
SMSC pour <i>SMS Center</i>	Reçoit et distribue les SMS : <ul style="list-style-type: none"> • Reçoit des messages du SME via le point de référence SME-SMSC • Distribue les messages au MS via le point de référence SMSC-MS
MS pour <i>Mobile Station</i>	Terminal mobile recevant les SMS : <ul style="list-style-type: none"> • Reçoit les messages du SMSC via le point de référence SMSC-MS

Les protocoles usuellement choisis pour ces points de référence sont les suivants.

Tableau 19 : Protocoles usuels du motif technique SMS

Point de référence	Protocole
SME-SMSC	MAP
SMSC-MS	MAP

Concernant OMA XDM (pour *XML Document Management*), tel que défini dans [OMA, web], le motif est le suivant. OMA XDM est un enabler défini à l'OMA permettant la gestion de documents XML stockés dans un référentiel depuis un client. De même que pour le motif technique SMS, le point de référence peut être décomposé en procédures de niveau inférieur (par exemple un *aggregation proxy*).

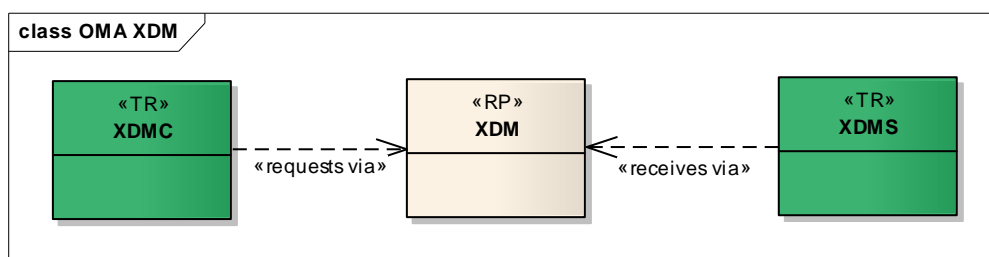


Figure 65 : Motif technique OMA XDM

Les rôles techniques principaux du XDM sont les suivants.

Tableau 20 : Rôles techniques du motif technique OMA XDM

Rôle technique	Description
XDMC pour <i>XDM Client</i>	Entité client pouvant manipuler un document XML : <ul style="list-style-type: none"> • Envoie des requêtes au XDMS via le point de référence XDM
XDMS pour <i>XDM Server</i>	Entité serveur stockant les documents XML : <ul style="list-style-type: none"> • Reçoit des messages du XDMC via le point de référence XDM

Les protocoles usuellement choisis pour ces points de référence sont les suivants.

Tableau 21 : Protocoles usuels du motif technique OMA XDM

Point de référence	Protocole
XDM	XCAP

Concernant la télévision sur Internet *IPTV*, nous pouvons considérer le motif suivant, repris et simplifié de [ITU, 2008].

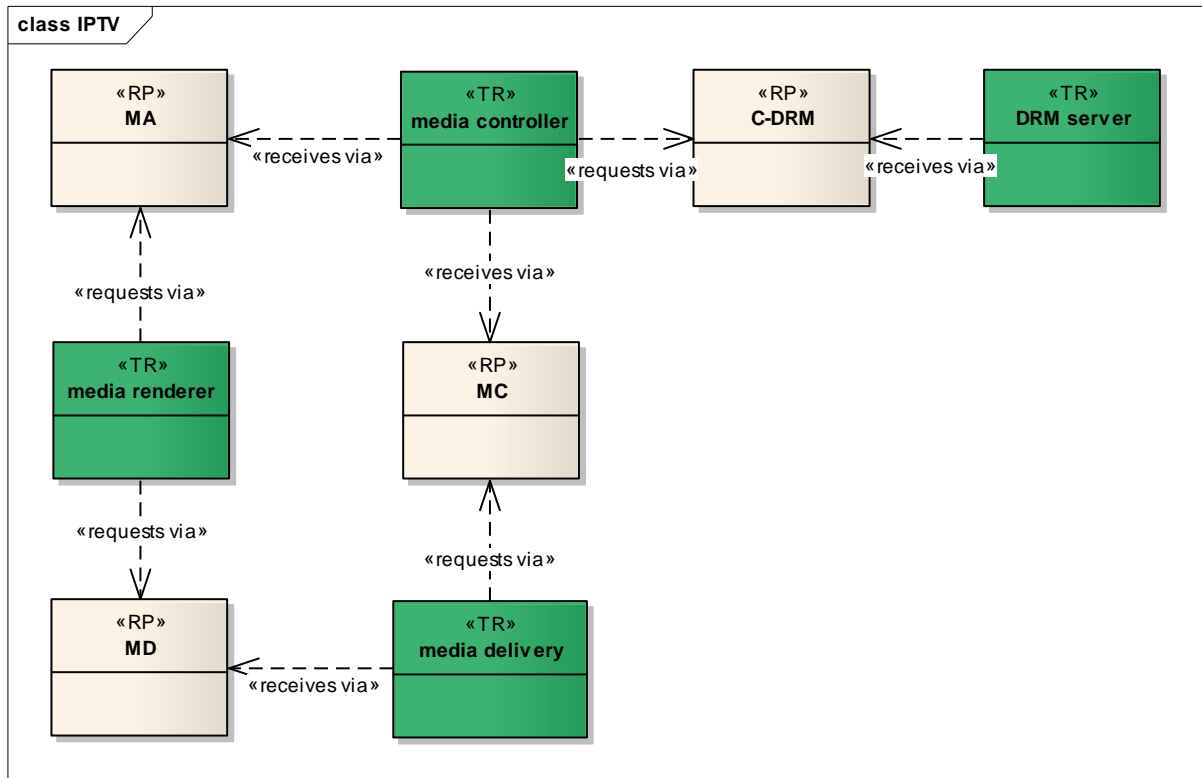


Figure 66 : Motif technique IPTV

Les rôles techniques principaux de l'IPTV sont les suivants.

Tableau 22 : Rôles techniques du motif technique IPTV

Rôle technique	Description
Media renderer	Entité jouant les media à l'utilisateur : <ul style="list-style-type: none"> • Envoie des requêtes pour demander un media au media controller via le point de référence MA • Envoie des requêtes pour jouer un media au media delivery via le point de référence MD
Media controller	Entité contrôlant les droits des utilisateurs et la disponibilité des media demandés : <ul style="list-style-type: none"> • Reçoit des messages du media renderer via le point de référence MA • Emet des requêtes au DRM server via le point de référence C-DRM

	<ul style="list-style-type: none"> • Reçoit des demandes de vérifications du media delivery via le point de référence MC
DRM server	Entité de gestion des droits : <ul style="list-style-type: none"> • Reçoit des messages du media controller via le point de référence C-DRM
Media delivery	Entité de diffusion des medias : <ul style="list-style-type: none"> • Reçoit des requêtes de diffusion de media du media renderer via le point de référence MD • Emet des requêtes de vérification au media controller via le point de référence MC

Les protocoles usuellement choisis pour ces points de référence sont les suivants.

Tableau 23 : Protocoles usuels du motif technique IPTV

Point de référence	Protocole
MA	RTSP, SIP
MC	COPS
MD	RTSP, RTP
C-DRM	license protocol

Concernant les interfaces homme-machine, le motif IHM est très simple.

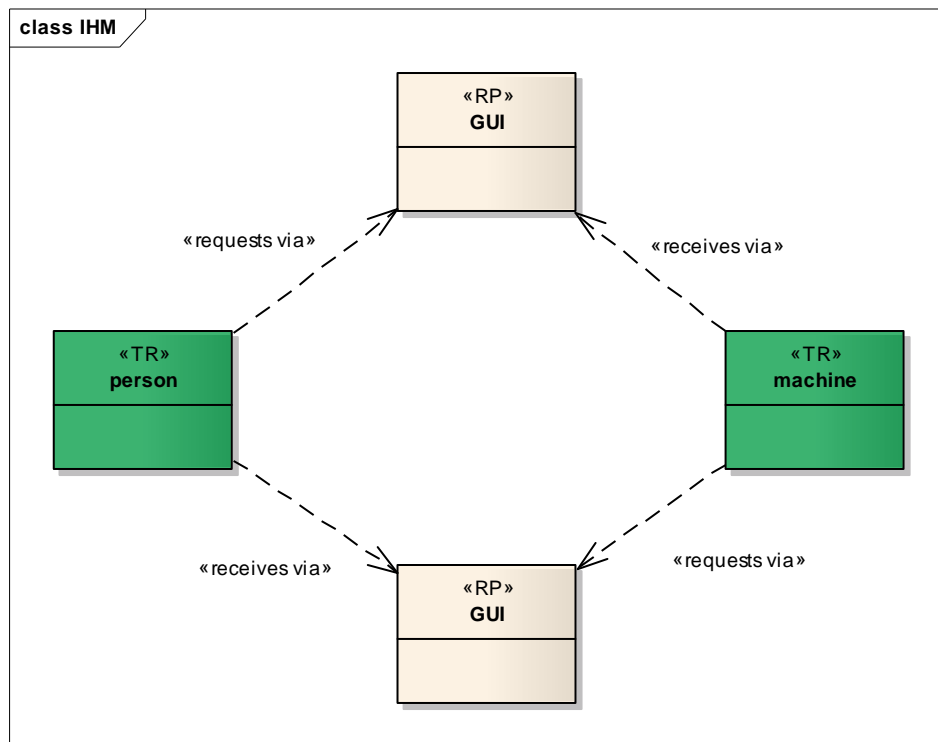


Figure 67 : Motif technique IHM

Un rôle technique *person* émet des requêtes vers une machine via une interface utilisateur GUI. Le point de référence GUI n'est pas réalisé par des protocoles, mais par des dispositifs rendant tangible les informations, comme un clavier, une souris, un écran, voire des interfaces de capture et d'interprétation de la voix ou des gestes dans le cas des IHM avancées. Le point de référence peut également être dans le sens inverse, lorsque c'est la machine qui sollicite en premier la personne, par exemple dans le cas d'une sonnerie de téléphone. Evidemment, une personne est ici uniquement considérée comme source et destination de flux, et non comme une partie-prenante du service comme dans la première section de ce chapitre.

Synthèse

Durant ce chapitre, nous nous sommes posé la question des invariants pour un service télécom. Alors que, dans une entreprise agile, les services évoluent sur un rythme rapide, tant pour les évolutions techniques que pour les fonctionnalités offertes, que reste-t-il de suffisamment stable pour pouvoir être considéré comme invariant à moyen-terme ? Nous avons identifié ici trois types d'invariants :

- les activités des processus de service : des invariants langagiers dans la description du service rendu par ses parties-prenante (utilisateur, fournisseur de service);
- les motifs fonctionnels : des invariants stratégiques dans les grandes fonctions rendues par un opérateur télécom (communication, diffusion ou configuration d'informations);
- les motifs techniques : des invariants techniques dans les protocoles ou dans les modèles d'architecture (par exemple l'IMS ou l'architecture dite *3-tiers*).

Ces différents invariants peuvent être décrits selon les vues de l'urbanisme des systèmes d'information pour constituer la sémantique de référence que nous attendions au chapitre II. La vue métier de référence contient ainsi les invariants langagiers avec les processus de service. La vue fonctionnelle de référence contient les invariants stratégiques corrélés avec les invariants langagiers pour donner des fonctions élémentaires SEF, des opérations fonctionnelles SFO et des données fonctionnelles SFD ; ces fonctions, opérations et données représentent ainsi la stratégie d'un opérateur appliquée à un domaine métier particulier. La vue technique de référence contient elle les invariants techniques, représentés par un ensemble de points de référence RP et de rôles techniques TR. Dans le chapitre suivant, nous allons étudier comment décrire des services de communication à l'aide de ces vues de référence.

Chapitre IV Utiliser le référentiel pour décrire l'architecture des services télécoms

Au cours du chapitre précédent, nous avons étudié la construction de vues de référence. Dans cette section, nous allons étudier comment les utiliser pour décrire un service télécom quelconque, un service existant comme un nouveau service. Pour cela, nous allons décrire un service ou une composition de services à travers trois vues : sa vue fonctionnelle, sa vue technique et sa vue applicative. Nous n'utiliserons pas de vue métier. Comme nous l'avons vu, la vue fonctionnelle de référence contient une sémantique de référence pour décrire le déroulement des services, qui permet notamment de comparer objectivement des vues fonctionnelles réalisées par des équipes différentes. La vue métier de référence nous est utile pour construire la vue fonctionnelle de référence et pour gérer son cycle de vie (ajout ou retrait d'un composant fonctionnel). Par contre, lors de la conception des services télécoms, ce sont les cas d'utilisation qui sont les plus utilisés. Nous verrons que ces cas d'utilisation peuvent être directement transcrits dans la vue fonctionnelle du service.

Nous nous servirons par contre de la vue applicative, que nous n'avons pas utilisée pour construire nos vues de référence. Cette vue applicative fait le lien entre les composants fonctionnels d'un service et les rôles techniques utilisés pour l'implémenter.

Reprenons les éléments d'un système de service introduits à la fin du deuxième chapitre (figure 24). La vue fonctionnelle d'un service décrira le déroulement du service, la vue technique d'un service décrira la solution technique retenue, et la vue applicative du service la façon dont le déroulement du service est réalisé par la solution technique.

Chaque section sera constituée de trois temps. Nous chercherons d'abord une méthode de description de l'architecture du service télécom selon la vue considérée, en prenant l'exemple d'un service de carnet d'adresses. Nous proposerons alors un méta-modèle d'architecture de service, distinct du méta-modèle d'urbanisme du chapitre III. Puis nous illustrerons cette méthode avec quelques services simples. Enfin, nous verrons comment traiter un service composé à travers un exemple.

IV.1 La vue fonctionnelle d'un service

Récapitulons le mouvement du chapitre précédent concernant les vues métier et fonctionnelles. Nous avons débuté avec les actions réalisées par les parties-prenantes d'un service, représentées sous forme de processus, et ce pour les services typiques d'un opérateur télécom. Nous en avons déduit des modèles d'entités de service qui couvrent les concepts majeurs de ces services typiques. Puis, à l'aide de motifs fonctionnels caractérisant les opérations rendues harmonisées suivant une nomenclature commune, nous avons construit des SFO, opérations fonctionnelles de service pour caractériser les actions des parties-prenantes d'un service et des SEF, fonctions élémentaires de service correspondant à ces opérations. Nous avons ensuite défini des SFD, données fonctionnelles de service, paramètres d'entrée et de sortie de chaque SFO. Puis nous avons regroupé ces SEF en SFC pour composants fonctionnels de service, un SFC offrant des SFO et pouvant ainsi caractériser les capacités d'actions offertes par un opérateur télécom. Dans cette section, nous allons revenir à la particularité des services en étudiant comment décrire un service télécom quelconque, et notamment un service composé, à partir de ses capacités d'actions, et donc formellement à partir des SFC, SFO et des SFD. Ces capacités d'actions sont un outil important pour construire des architectures de service user-centric. En effet, nous avons présenté la convergence de service comme la possibilité de mêler des services divers (*blended services*) pour répondre au besoin du consommateur du service. L'optique défendue ici est de considérer un service convergent comme un assemblage de capacités d'action offertes par des services différents. Nous poursuivons en cela la démarche initiée dans le réseau intelligent avec les éléments de service, où tout service était vu comme une composition d'éléments de services.

IV.1.1 Méthode de description

Il nous reste maintenant à étudier la meilleure façon de décrire un service ou une composition de services à l'aide du référentiel de la vue fonctionnelle, les SFC et les SFD. Selon les bonnes pratiques du génie logiciel, le point de départ pour décrire un système qui sera réalisé informatiquement est d'identifier des cas d'utilisation caractérisant ce système, ainsi que des

scénarios déclinant ces cas d'utilisation¹. Ces notions sont communes en génie logiciel et sont par exemple décrites en détail dans [Cockburn, 1997] ou [Alexander, 2004]. A l'aide de quelques scénarios bien choisis, on peut décrire de façon représentative un service. Plus le service possède de fonctionnalités plus il nécessite de scénarios pour être décrit. Les scénarios décrivant les cas d'erreur, essentiels pour le développement logiciel, ne seront pas mis en avant ici, car ils impacteront peu l'architecture fonctionnelle, c'est-à-dire les SFC décrivant le service et les dépendances entre eux.

Parcourons pas à pas l'exemple d'un service de carnet d'adresses. Dans un service de carnet d'adresses, le cas d'utilisation typique sera le suivant. Un acteur *owner*, propriétaire du carnet d'adresses, peut gérer ses contacts (ajouter, modifier, supprimer des contacts ou des adresses de ces contacts) ou consulter son carnet d'adresses. Nous voyons donc deux cas d'utilisation, "*manage contacts*" et "*consult address book*".

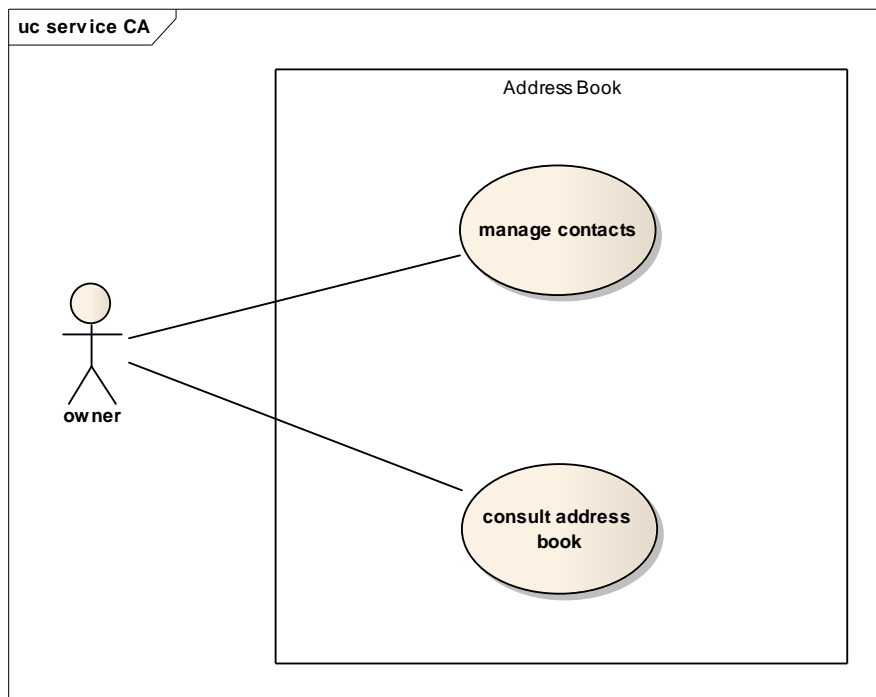


Figure 68 : Cas d'utilisation d'un service de carnet d'adresses

Pour le cas d'utilisation "*manage contacts*", nous considérerons le scénario 1 suivant : l'acteur *owner* se connecte à son carnet d'adresses et ajoute une entrée (nom, prénom et n° de

¹ Un scénario constitue un exemple de déroulement d'un service, où tous les embranchements logiques possibles dans le cas d'utilisation sont décidés. Pour un cas d'utilisation "téléphoner", un scénario sera par

téléphone) dans sa liste d'amis ; il valide et l'entrée est ajoutée. Le scénario 2 est le suivant, pour le cas d'utilisation "*consult address book*" : l'acteur *owner* se connecte à son carnet d'adresse, parcourt la liste de ses contacts et en sélectionne un. Il visualise alors les adresses saisies pour ce contact.

Dans la vue fonctionnelle d'un service, nous allons partir de ces scénarios représentatifs pour décrire le déroulement des services. Et ce en les traduisant dans la sémantique de référence de la vue fonctionnelle définie au chapitre précédent. Des scénarios établis pour des projets différents et par des équipes différentes seront ainsi exprimés selon une sémantique commune, et à ce titre comparables et positionnables les uns vis-à-vis des autres.

Voyons maintenant comment représenter la vue fonctionnelle d'un service. Nous proposons de formaliser cette vue fonctionnelle sous la forme de diagrammes de séquences composé de SFC, où chaque échange de message entre SFC manifeste une interaction entre ces SFC. Cette représentation sous forme de séquences permet de préciser la dynamique du service en montrant les enchaînements entre capacités d'actions. Les interactions entre SFC par des messages n'indiquent pas une succession temporelle, mais une relation de dépendance. Un message d'un SFC A vers un SFC B signifie que les capacités d'action modélisées par le SFC A nécessitent pour leur mise en œuvre le résultat de la mise en œuvre des capacités d'action modélisées par le SFC B, c'est-à-dire que le SFC A nécessite pour sa réalisation une donnée fonctionnelle produite par le SFC B et déclenche pour cela une opération fonctionnelle du SFC B. Répétons, ce déclenchement ne fait pas nécessairement suite temporellement à l'invocation du SFC A, mais il doit avoir eu lieu pour que le SFC A puisse réaliser son activité propre. Le message retour contient SFD B, la donnée fonctionnelle demandée. Cet ordre de dépendance fonctionnel est sans lien avec l'ordre temporel d'invocation de composants. Il pourra parfois être similaire à l'ordre temporel et parfois opposé. La ligne de vie du composant dans le diagramme de séquence ne manifeste pas la vie temporelle du composant, mais sa "vie fonctionnelle", c'est-à-dire l'activité de ce composant au cours de laquelle il obtient l'ensemble des données fonctionnelles qui sont nécessaires à sa réalisation. Les invocations par un SFC A d'autres composants fonctionnels sont le moyen pour récupérer ces données fonctionnelles. Elles nous permettent de montrer que le SFC A

exemple : "L'appelant décroche son téléphone et compose le n° de l'appelé ; il entend la tonalité de sonnerie puis l'appelé répond en décrochant son téléphone".

dépend de ces autres composants car il a besoin de données fonctionnelles dont ils sont maîtres, c'est-à-dire qu'ils produisent. Le diagramme de séquence peut être comparé à un processus de transformation de n données fonctionnelles primaires (rendues dans les messages retours des SFC ne faisant pas appel à d'autre SFC) en une donnée fonctionnelle finale indiquant le résultat du service (rendue dans le dernier message retour).

La démarche pour décrire un service est donc la suivante. Tout d'abord, comme nous l'avons vu, on identifie les cas d'utilisation représentatifs du service, puis les scénarios typiques de ces cas d'utilisation. A partir de ces scénarios typiques, on déduit les SFC impliqués dans le service, en établissant un diagramme de séquence par scénario représentatif du service. On détermine d'abord le premier composant fonctionnel du scénario, celui qui renverra la donnée fonctionnelle finale du scénario. Par exemple, si le scénario concerne un service de téléphonie et s'arrête à la sonnerie du téléphone de l'appelé, le premier composant fonctionnel du scénario sera le composant *SFC phone call*, dont l'opération *SFO deliver phone call* est en charge de la remise de l'appel à l'appelé, et qui renvoie une donnée fonctionnelle *SFD delivered phone call* correspondant à un appel remis à l'appelé. Ce premier composant fonctionnel est déclenché par une ligne de vie représentant le scénario lui-même, de façon usuelle en génie logiciel. Les autres composants fonctionnels de la séquence correspondent aux opérations requises par ce premier composant pour réaliser le scénario. Les messages échangés entre les composants permettent aux composants d'obtenir les données fonctionnelles dont ils ont besoin. Les dépendances existantes des composants fonctionnels de référence sont une aide pour l'architecte, afin de construire ses diagrammes de séquence. Lorsqu'un architecte ajoute dans une séquence fonctionnelle une opération SFO, il va en effet considérer toutes les données fonctionnelles dont a besoin en entrée cette SFO. Et pour chaque donnée fonctionnelle utilisée par cette SFO, il va étudier si cette donnée fonctionnelle présente une utilité dans le cadre du scénario considéré. Si oui, il ajoutera dans la séquence le composant produisant cette donnée fonctionnelle, qui sera invoqué par le SFC portant la SFO. De la même façon, des dépendances non prévues dans la vue fonctionnelle de référence peuvent être ajoutées, lorsque le scénario les nécessite. De plus, si un scénario de service nécessite une capacité d'action non définie dans les SFC de référence, de nouveaux composants fonctionnels peuvent être ajoutés pour décrire le service. Ces composants ne deviennent pas pour autant des SFC de référence, mais restent spécifique au service.

Continuons l'exemple du service de carnet d'adresses. La vue fonctionnelle du service est composée de séquences de SFC traduisant ces scénarios. Pour le scénario 1 d'entrée de contacts, les fonctions élémentaires nécessaires sont la capacité de publier une adresse de joignabilité (n° de téléphone, adresse email, adresse Skype...), la capacité de créer une liste d'adresses (par exemple une liste "amis", "collègues", mais aussi la liste racine), et la capacité d'authentifier et de valider l'accès. Ces fonctions élémentaires sont rattachées respectivement aux composants *SFC address*, *SFC address list* et *SFC access right*. Le *SFC address* est le premier SFC de la séquence, car il réalise l'objectif métier du scénario, c'est-à-dire l'ajout d'un contact avec une adresse de joignabilité. Le *SFC address* nécessite le *SFC access right* car l'accès au carnet d'adresses est dédié et celui qui y accède doit être authentifié. Le *SFC address* nécessite le *SFC address list* car l'ajout d'un contact se fait dans une liste de contacts, qui doit avoir été préalablement créée, que ce soit une nouvelle liste ou la liste par défaut.

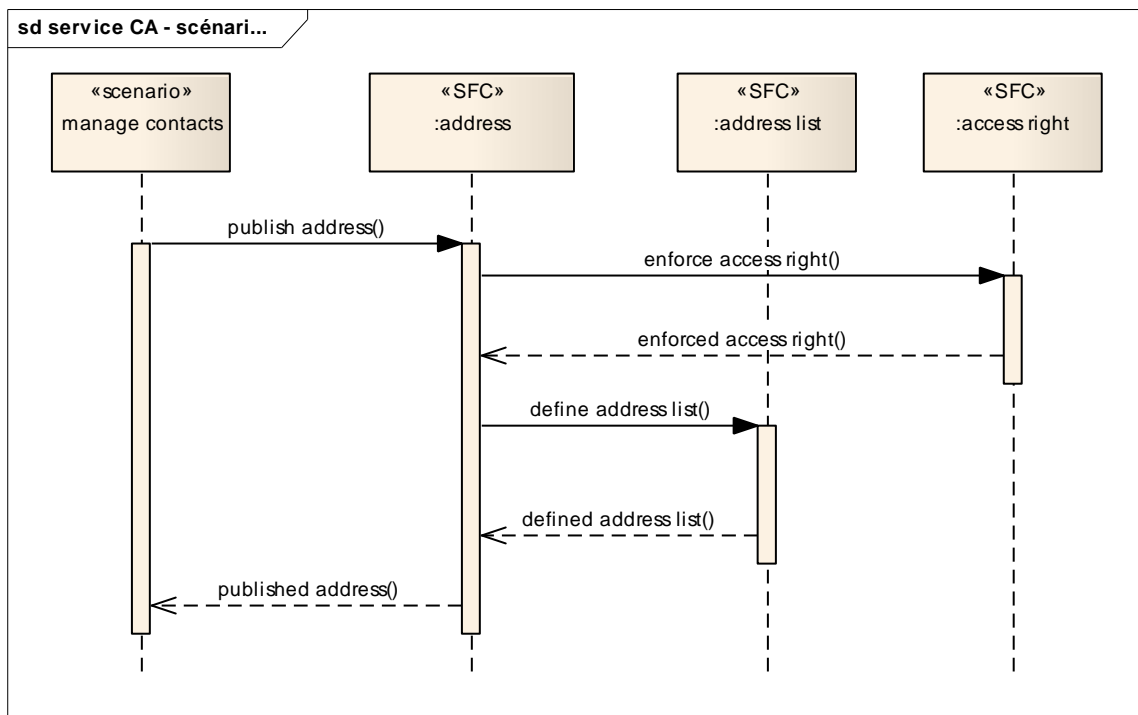


Figure 69 : Séquence fonctionnelle du scénario de gestion de contacts

Considérons maintenant la vue fonctionnelle du deuxième scénario. Les fonctions élémentaires mises en œuvre dans ce scénario sont la capacité de naviguer dans un carnet d'adresses, la capacité d'authentifier et de donner accès, la capacité de mettre en place une liste de contacts et la capacité de publier une adresse de joignabilité. Ces fonctions élémentaires sont respectivement portées par les *SFC address*, *SFC access right*, *SFC address*

list et *SFC address*. Le *SFC address* est le premier SFC de la séquence, car il réalise la finalité du scénario qui est de consulter des adresses de joignabilité. Le *SFC address* dépend du *SFC access right* car l'accès au carnet d'adresses est soumis à authentification et autorisation, ce qui ne serait par exemple pas le cas dans un service d'annuaire. Le *SFC address* dépend du *SFC address list*, car la consultation porte sur un carnet d'adresses particulier. Le *SFC address* invoque l'opération *SFO publish address* du *SFC address*, car la recherche se fait sur les adresses préalablement publiées.

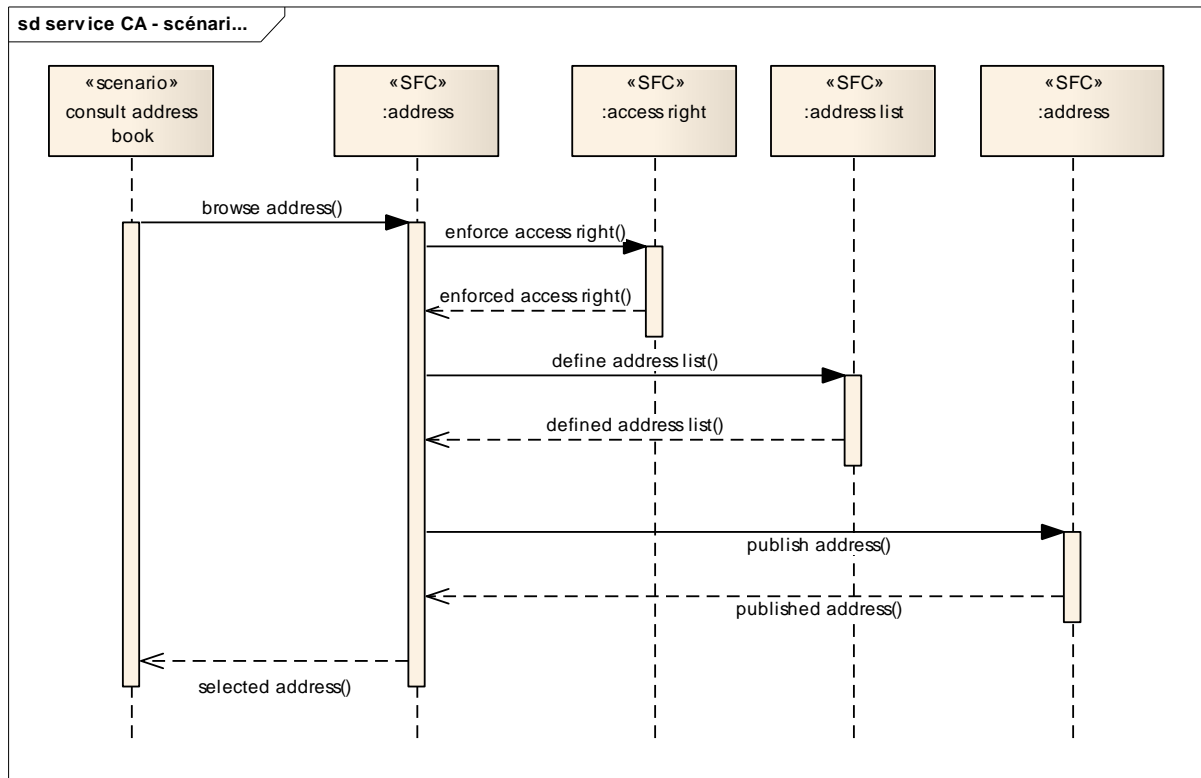


Figure 70 : Séquence fonctionnelle du scénario de consultation du carnet d'adresses

La description d'un scénario de service sous forme de séquence de SFC permet ainsi de comparer entre eux différents services, chaque service étant décrit de la même façon. Des métriques pertinentes peuvent même être définies à ce niveau, comme par exemple une métrique mesurant l'innovation d'un service, ou sa complexité. Nous en proposerons quelques unes au chapitre V.

Nous pouvons formaliser ces nouveaux concepts dans un méta-modèle d'architecture des services, distinct du méta-modèle d'urbanisme du chapitre III. Le travail de description de services avec des SFC est ainsi complètement dissocié du travail de conception de SFC à partir de processus de service.

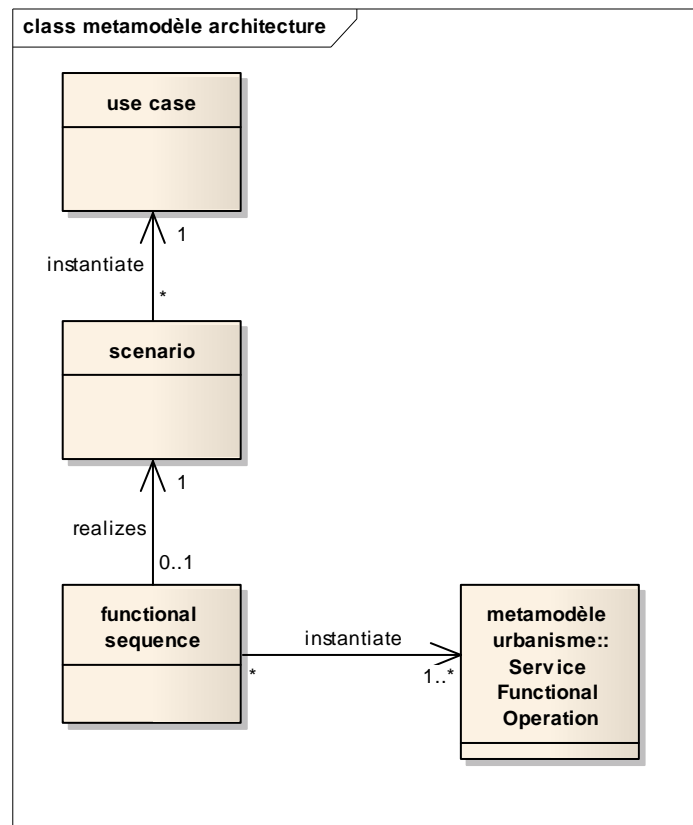


Figure 71 : Méta-modèle de la vue fonctionnelle d'un service

Nous aurions pu également décrire un service sous forme statique, en listant les SFC qu'il réalise. Ce type de représentation présente l'avantage de la simplicité, mais ne permet pas de cerner la dynamique du service. A titre illustratif, le diagramme de classes ci-dessous présente ainsi les composants fonctionnels de notre service de carnet d'adresses et leurs dépendances.

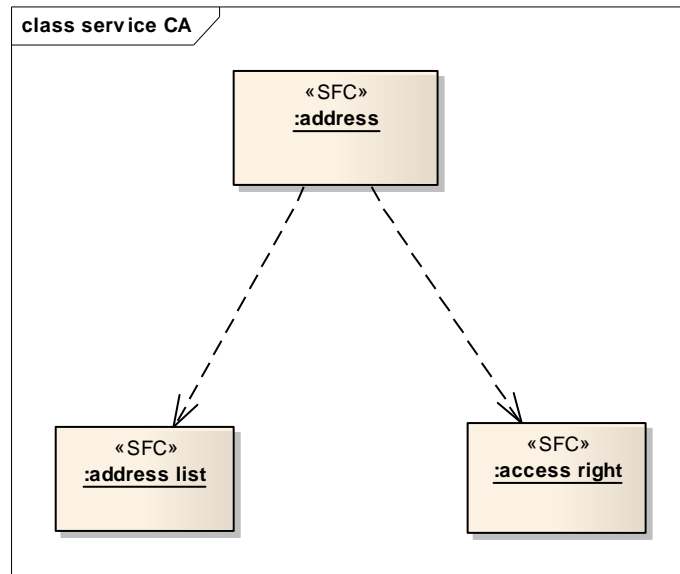


Figure 72 : Dépendances entre composants fonctionnels d'un service de carnet d'adresses

Ce type de représentation ne permet pas de faire le lien avec les scénarios, car il reste statique. Nous ne pouvons ainsi pas justifier les dépendances entre les composants fonctionnels. Ce type de schéma présente toutefois une vue synthétique des différents composants et de leurs dépendances et peut être utilisé en complément des diagrammes de séquence à des fins de communication.

D'autres représentations sont envisageables. Ainsi, les scénarios pourraient être décrits fonctionnellement par des diagrammes d'activité, en représentant les SFC sous forme de couloirs (*swimlane*) et les opérations fonctionnelles sous forme d'activités dans ces couloirs. La représentation en séquence est néanmoins plus conforme aux bonnes pratiques du génie logiciel et de l'urbanisation des systèmes d'information [Simonin, 2009].

IV.1.2 Exemples de services simples

Nous allons maintenant illustrer cette démarche avec trois services basiques :

- Un service de téléphonie, avec comme cas d'utilisation "appeler un correspondant au téléphone"
- Un service de vidéo à la demande, avec comme cas d'utilisation "regarder une vidéo"
- Un service de messagerie, avec comme cas d'utilisation "consulter ses emails"

Dans chaque cas, on considérera tout d'abord une description très simple, probablement équivalente en termes de détails à celle qui serait réalisée en l'absence de vue fonctionnelle de

référence. Puis nous verrons comment cette vue de référence nous guide dans la prise en compte des enjeux fonctionnels de ces cas d'utilisation.

IV.1.2.1 Exemple d'un service de téléphonie

Le premier cas d'utilisation, appeler un correspondant au téléphone, est déclenché par l'appelant.

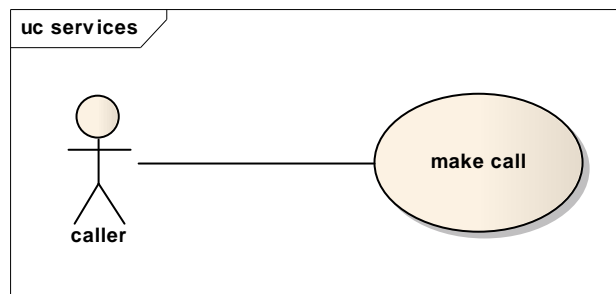


Figure 73 : Cas d'utilisation d'un service de téléphonie

Supposons que ce cas d'utilisation est décrit par une équipe marketing avec le scénario suivant. Alice sélectionne le n° de téléphone Bob dans son service de carnet d'adresses (tel que vu dans la section précédente) et clique sur son numéro pour lancer un appel vers lui ; le téléphone de Bob sonne.

En première approximation, ce scénario peut être décrit fonctionnellement par le diagramme de séquence suivant.

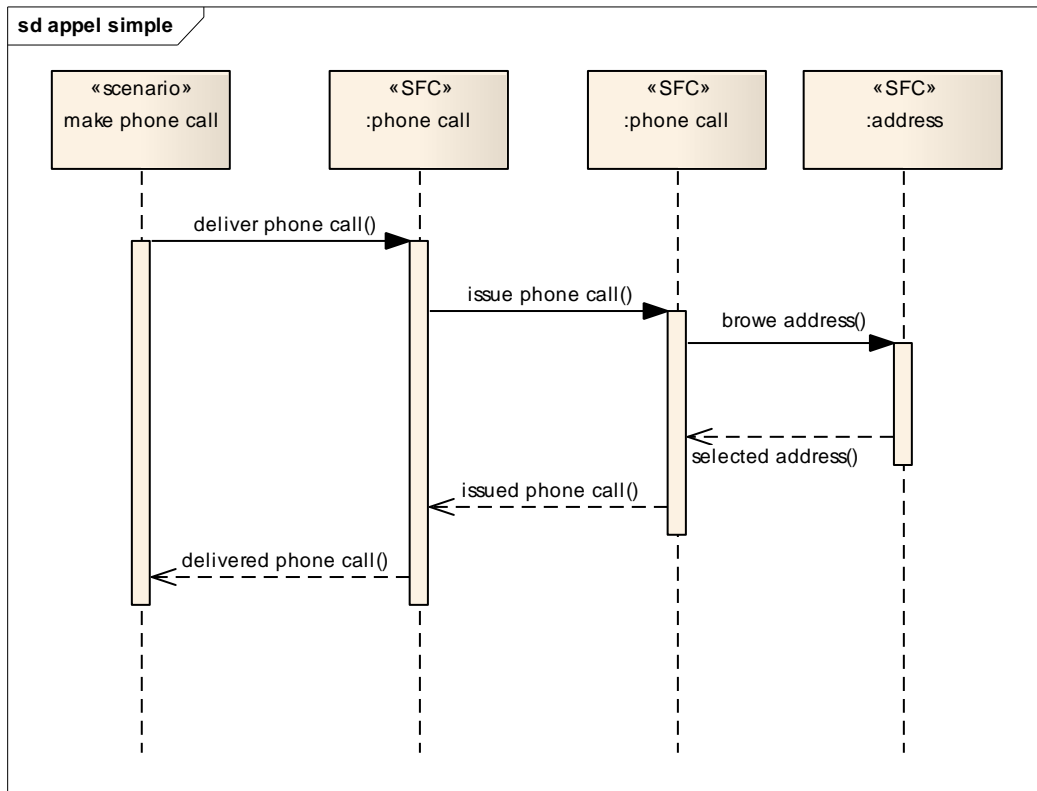


Figure 74 : Séquence fonctionnelle du scénario d'appel téléphonique (simplifiée)

Comme expliqué dans la section précédente, ce diagramme de séquence montre un ordre de prérequis et non un ordre temporel. Le diagramme signifie que le scénario a comme objectif qu'un appel téléphonique soit acheminé jusqu'au téléphone de Bob (acheminement modélisé par *SFC phone call* et invoqué par *SFO deliver phone call*). La réalisation de cet acheminement nécessite qu'un appel téléphonique ait été émis (émission déclenchée via *SFO issue phone call*). La réalisation de cette émission d'appel nécessite qu'Alice puisse rechercher et choisir un numéro de téléphone dans son carnet d'adresses (recherche et sélection modélisées par *SFC address*). Comme vu auparavant, le fait qu'un SFC A nécessite un SFC B est modélisé par l'invocation par le SFC A de l'opération offerte par le SFC B.

Ce diagramme de séquence décrit le scénario d'émission d'appel avec la nomenclature définie dans la vue fonctionnelle de référence vue au chapitre précédent, car il utilise les SFO définis dans cette vue fonctionnelle. Comme mentionné, ceci nous permettra de comparer facilement des scénarios décrits par des équipes projet différentes. Toutefois, ce diagramme de séquence laisse de côté des problématiques fonctionnelles importantes et la vue fonctionnelle de référence va nous aider à les identifier. Considérons les SFO impliquées dans ce diagramme de séquence. Si l'on remplace ces SFO dans la vue fonctionnelle de référence

présentée au chapitre précédent, on voit que certains SFO utilisent des SFD produites par d'autres SFO. Ainsi, *SFO deliver phone call* utilise la donnée *SFD enforced routing rule*, produite par *SFO enforce routing rule*. Nous pouvons ainsi étudier la pertinence de chacun de ces liens d'utilisation dans le cadre du scénario étudié. La vue fonctionnelle de référence devient alors non seulement un moyen de formaliser les scénarios de services avec une nomenclature commune, mais également une aide pour compléter un scénario en identifiant toutes les fonctions nécessaires à sa réalisation, et en se posant les bonnes questions. Dans notre exemple, ce questionnement du scénario par la vue fonctionnelle de référence nous conduit à identifier, en plus des fonctions envisagées en première approximation, de nouvelles fonctions élémentaires à prendre en compte :

- la fonction d'authentification de l'appelant (le *SFC phone call* invoque l'opération *SFO enforce access right*)
- la fonction d'authentification du titulaire du carnet d'adresses, qui doit être cohérente avec celle de l'appelant (le *SFC address* invoque l'opération *SFO enforce access right*)
- la fonction de déclenchement de services supplémentaires, comme par exemple un renvoi sur messagerie ou un blocage de l'identité de l'appelant (le *SFC phone call* invoque l'opération *SFO enforce routing rule*)
- la fonction de routage vers terminal physique correspondant au numéro composé, avec éventuellement des sonneries simultanées (le *SFC phone call* invoque l'opération *SFO enforce registration*)

Ces nouvelles fonctions et leurs relations avec les fonctions précédemment identifiées sont indiquées dans la figure suivante. La vue fonctionnelle de référence nous a ainsi aidés à mettre en évidence dès la phase de conception des scénarios du service des problématiques fonctionnelles non triviales comme la prise en compte des services supplémentaires de l'appelé, ou encore la cohérence nécessaire entre l'authentification et les droits de l'appelant et ceux du titulaire du carnet d'adresse.

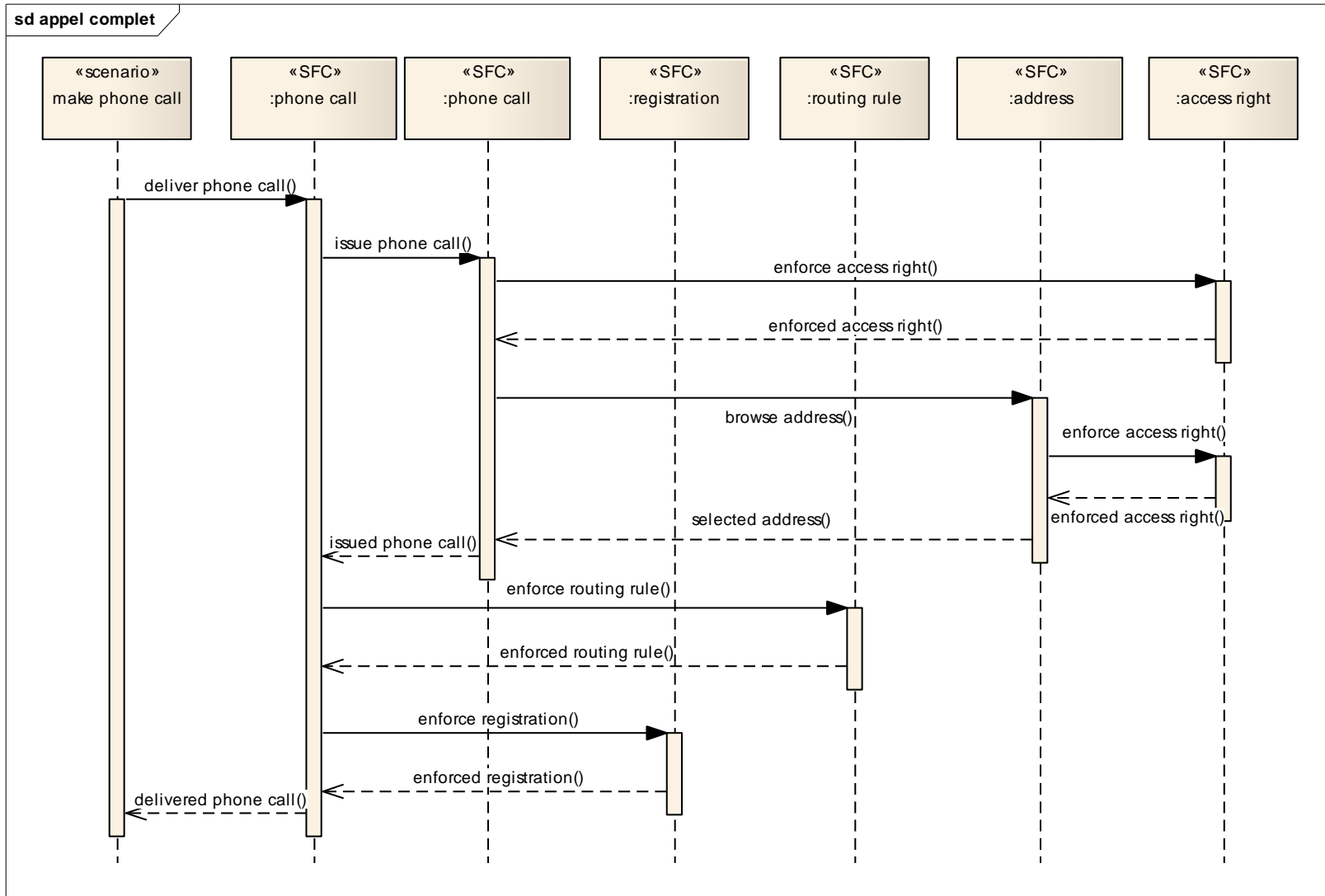


Figure 75 : Séquence fonctionnelle du scénario d'appel téléphonique (complète)

IV.1.2.2 Exemple d'un service de vidéo

Etudions maintenant un autre cas d'utilisation, regarder une vidéo à la demande. Ce cas d'utilisation est déclenché par le consommateur de vidéo.

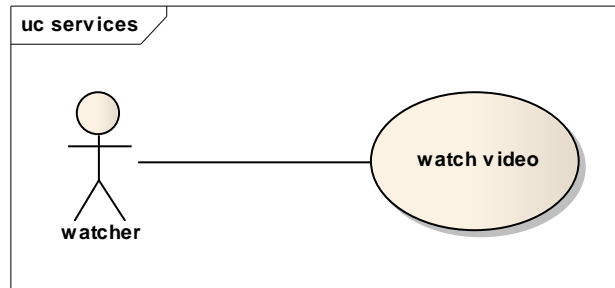


Figure 76 : Cas d'utilisation d'un service de vidéo

Un scénario décrivant ce cas d'utilisation est le suivant : Alice choisit une vidéo dans le catalogue d'Orange de vidéos à la demande (ou VOD pour *Video On Demand*) et regarde cette vidéo. En première approximation, ce scénario peut être décrit fonctionnellement par le diagramme de séquence suivant.

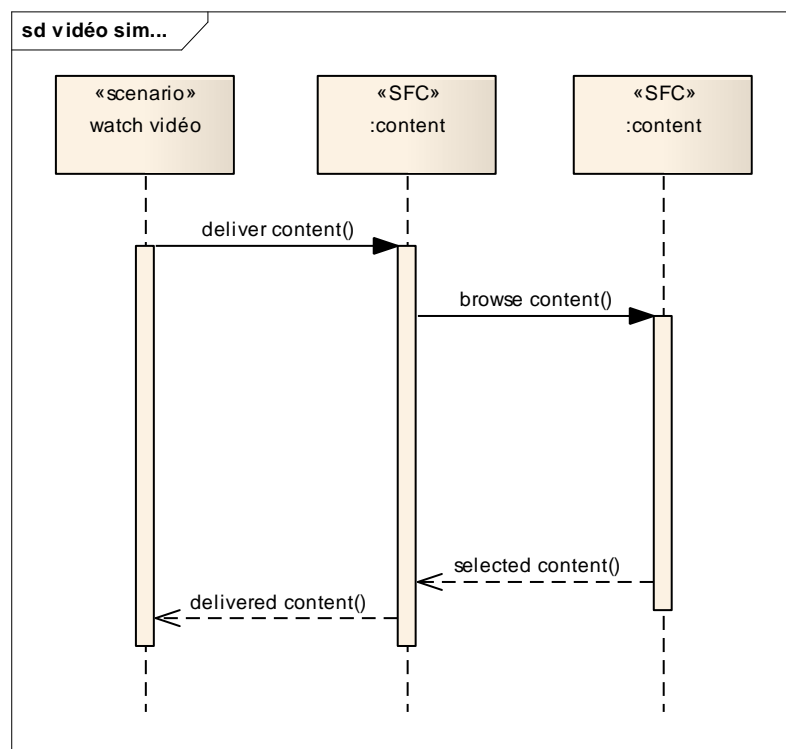


Figure 77 : Séquence fonctionnelle du scénario de consommation de vidéo (simplifiée)

Le scénario nécessite que du contenu soit joué à Alice par un media renderer (livraison et rendu modélisés par l'opération *SFO deliver content* du *SFC content*). Ce jeu du contenu vidéo nécessite qu'Alice puisse rechercher et choisir ce contenu dans un catalogue (recherche et choix modélisés par l'opération *SFO browse content* du *SFC content*). Comme précédemment, nous pouvons rendre ce scénario plus pertinent en le questionnant avec la vue fonctionnelle de référence. Nous pouvons alors identifier des fonctions élémentaires manquantes :

- la fonction de livraison du contenu au media renderer (le *SFC content* invoque l'opération *SFO issue content*)
- la fonction d'authentification d'Alice et la détermination de ses droits pour accéder au catalogue de contenu (le *SFC content* invoque l'opération *SFO enforce access right*)
- la fonction d'authentification d'Alice et la détermination de ses droits pour télécharger une vidéo, qui doit être cohérente avec les droits d'Alice dans le catalogue (le *SFC content* invoque l'opération *SFO enforce access right*)
- la fonction de détermination de l'adresse de livraison des contenus, c'est-à-dire de l'adresse du media renderer qui jouera le contenu, cette adresse devant être corrélée avec l'authentification d'Alice (le *SFC content* invoque l'opération *SFO define delivery device*)

Ces nouvelles fonctions et leurs relations avec les fonctions précédemment identifiées sont indiquées dans la figure suivante. De nouveau, la vue fonctionnelle de référence nous aide à identifier des problématiques fonctionnelles importantes, comme la livraison du contenu (qui sera réalisée de façon très différente pour un contenu en direct ou non), ou la corrélation entre l'authentification d'Alice et l'adresse de son media renderer. Ceci conduira à dialoguer avec l'équipe marketing qui porte ce scénario afin de déterminer comment résoudre ces problématiques fonctionnelles. Par exemple, l'équipe marketing pourra décider que la corrélation entre l'authentification d'Alice et son media renderer se fait par l'association sur la plate-forme de service entre l'identité d'Alice et l'adresse d'une set-top box.

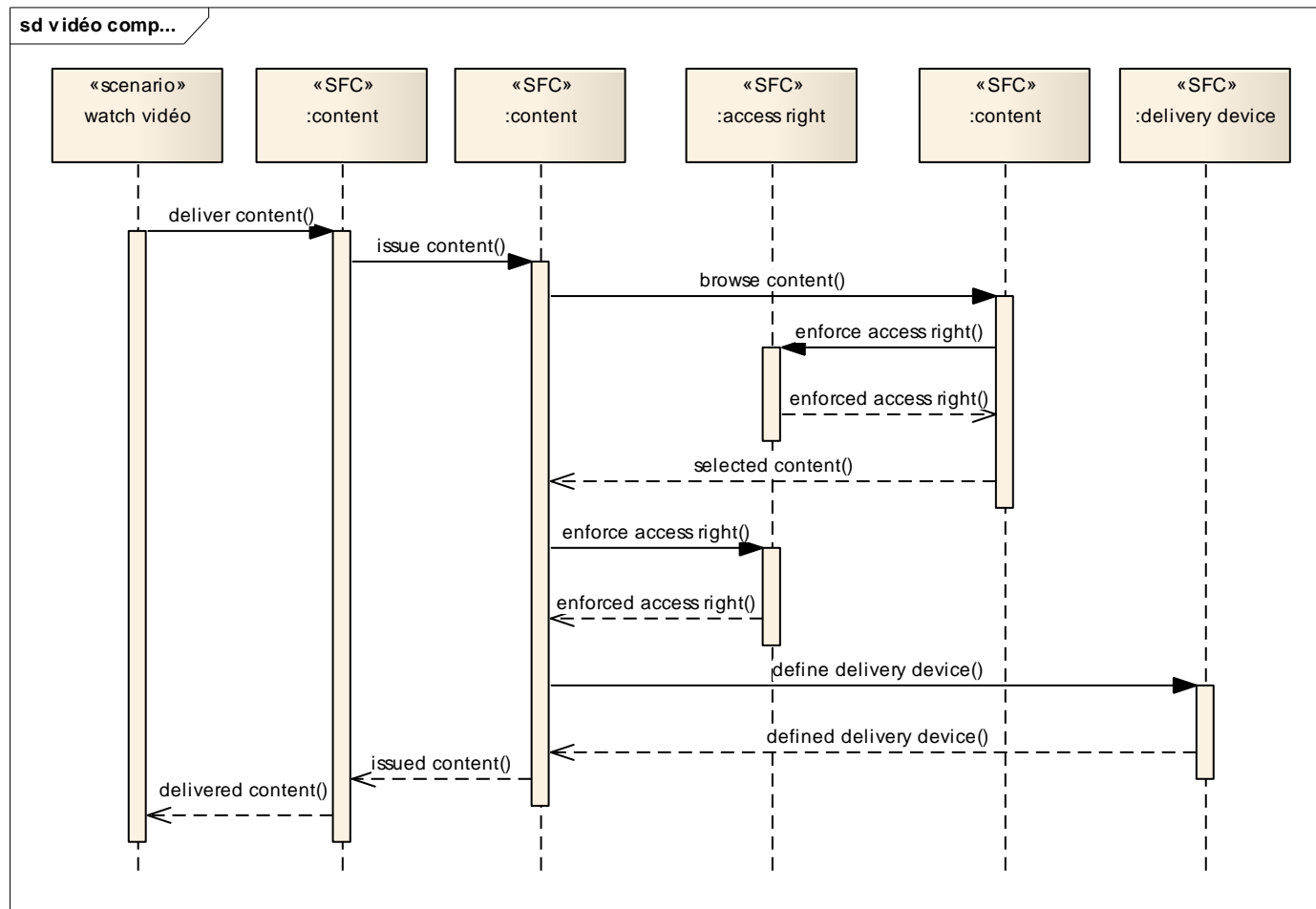


Figure 78 : Séquence fonctionnelle du scénario de consommation de vidéo (complète)

IV.1.2.3 Exemple d'un service de messagerie

Considérons enfin un troisième cas d'utilisation, la consultation d'emails. Ce cas d'utilisation est déclenché par le titulaire de la boîte aux lettres.

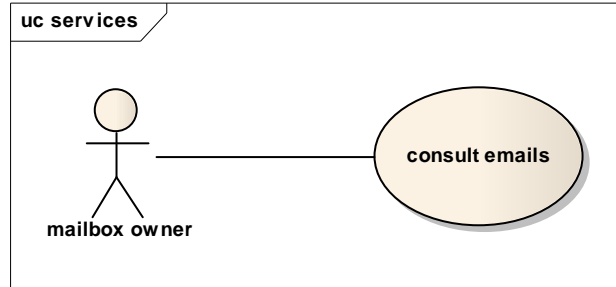


Figure 79 : Cas d'utilisation d'un service de messagerie email

Dans un scénario de ce cas d'utilisation, Alice accède à sa boîte aux lettres et consulte ses emails. Ce scénario peut être simplement représenté fonctionnellement par le diagramme de séquence suivant.

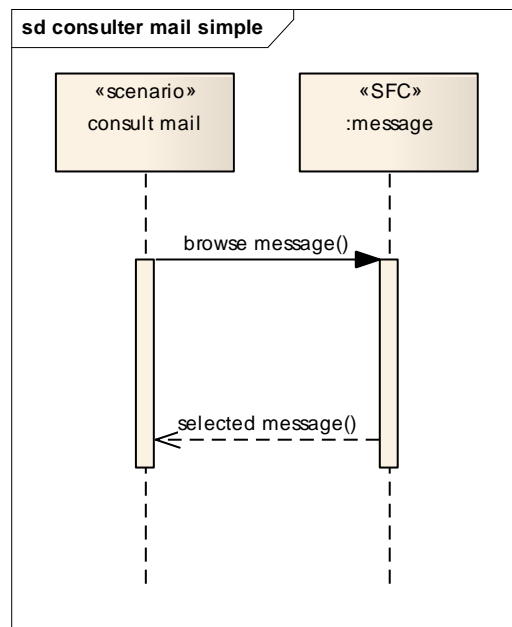


Figure 80 : Séquence fonctionnelle du scénario de consultation d'emails (simplifiée)

Le scénario nécessite qu'Alice puisse rechercher, consulter et sélectionner des messages (possibilités modélisées par la *SFC message*, à travers l'opération *SFO browse message*). En partant de la vue fonctionnelle, nous pouvons comme précédemment enrichir ce diagramme de séquence avec :

- la fonction de mise en place et de gestion de la boîte aux lettres (le *SFC message* invoque l'opération *SFO define message folder*)
- la fonction de gestion des droits d'accès (le *SFC message* invoque l'opération *SFO enforce access right*)
- la fonction de stockage et de classement des messages (le *SFC message* invoque l'opération *SFO publish message*)
- la fonction de routage des messages à partir de l'adresse de destination (le *SFC message* invoque l'opération *SFO deliver message*)

Comme précédemment, le questionnement par la vue fonctionnelle de référence met en lumière des problématiques fonctionnelles importantes. Ces problématiques pourraient tout à fait être identifiées sans utiliser la vue fonctionnelle de référence, uniquement grâce à l'expertise métier des architectes en charge de la réalisation de ces scénarios. Toutefois, la vue fonctionnelle de référence fournit un outillage conceptuel intéressant pour passer d'une démarche de conception de service "artisanale", basée sur l'expertise individuelle, à une démarche de conception de service plus industrielle, basée sur le respect de règles et de processus internes à l'entreprise de services télécoms.

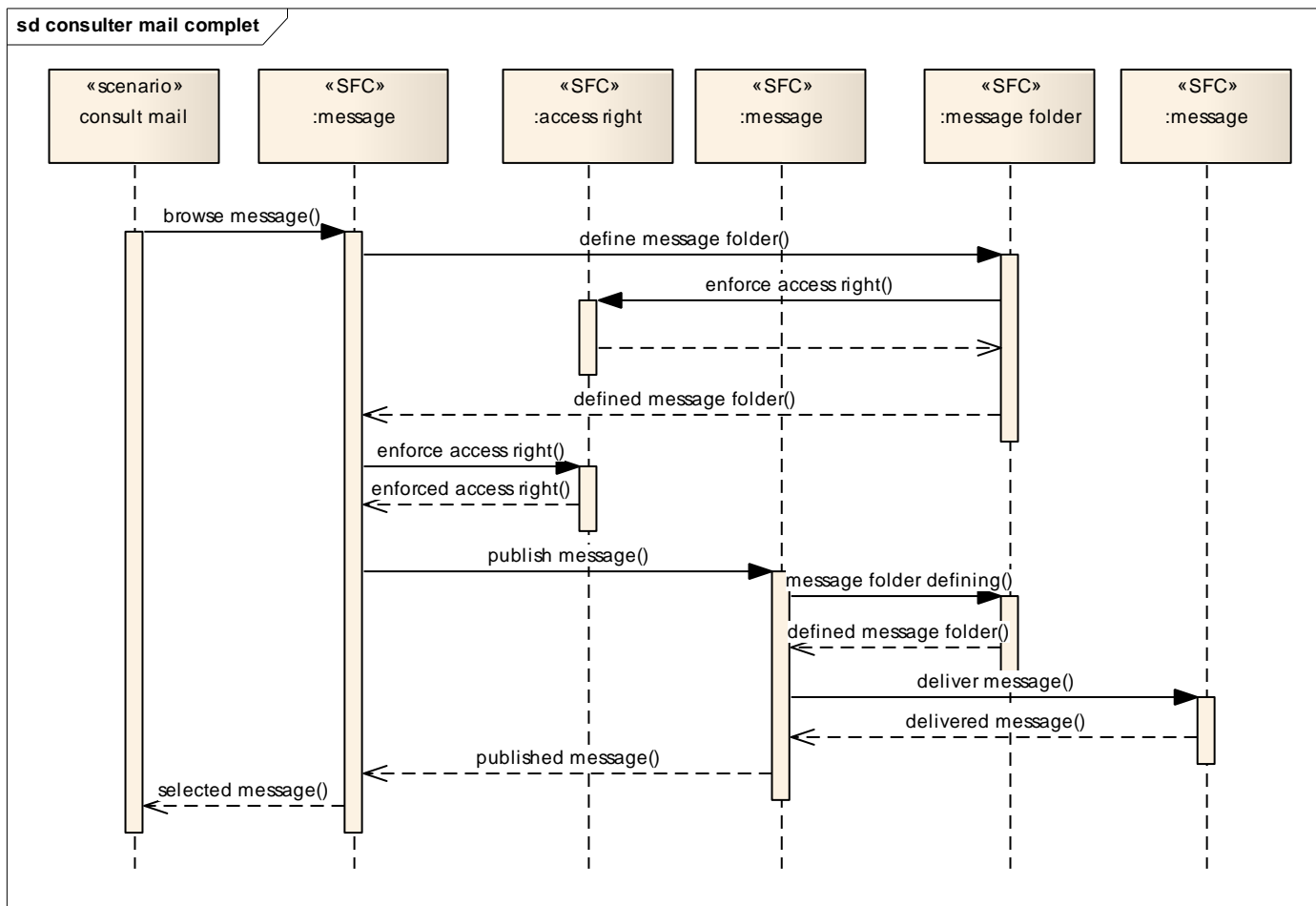


Figure 81 : Séquence fonctionnelle du scénario consultation d'emails (complète)

IV.1.3 Cas d'un service composé : VideoMe

Dans le cas d'un service composé, la vue fonctionnelle nous permet de considérer en détail comment les capacités d'actions de différents services vont se composer. En effet, la composition de service ne consiste pas simplement à juxtaposer des services, ni à les additionner, mais plutôt à les mélanger, les mêler. On parle ainsi de *blended services* [O'Connell, 2008] pour désigner ces services mélangés. Avec la vue fonctionnelle d'un service composé, nous avons une vue de bout en bout sur ce mélange. Les unités de la composition ne sont alors plus des services ou des enablers, chacun avec leur périmètre de responsabilité, mais les opérations fonctionnelles, les capacités d'action permises par ces services ou ces enablers.

Considérons ainsi un service composé mélangeant deux de nos services vus précédemment. Par exemple un service qui permet de diffuser une vidéo à une personne avec qui l'on cherche à établir un appel téléphonique, illustré par le scénario suivant : Alice choisit Bob dans son carnet d'adresse, elle choisit ensuite une vidéo à lui envoyer et clique sur son numéro pour lancer un appel vers lui. Le téléphone de Bob sonne et la vidéo lui est jouée en même temps. Nous appellerons ce service VideoMe. Ce scénario peut être représenté en première approximation par le diagramme de séquence suivant, qui compose les deux diagrammes de séquence vus précédemment.

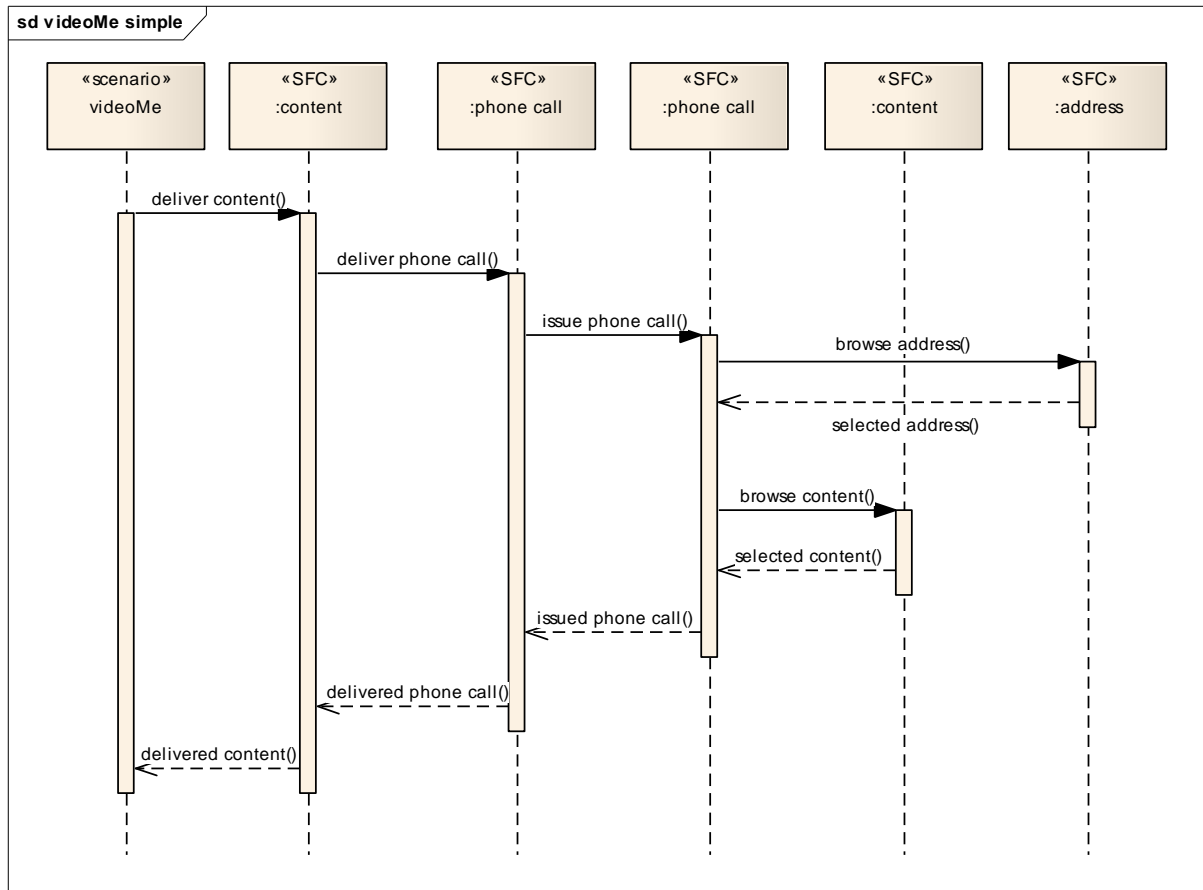


Figure 82 : Séquence fonctionnelle du scénario VideoMe (simplifiée)

De même que précédemment, des problématiques fonctionnelles importantes n'apparaissent pas. On peut alors compléter cette vue simplifiée grâce aux diagrammes de séquence détaillés des scénarios d'appel et de vidéo vus précédemment.

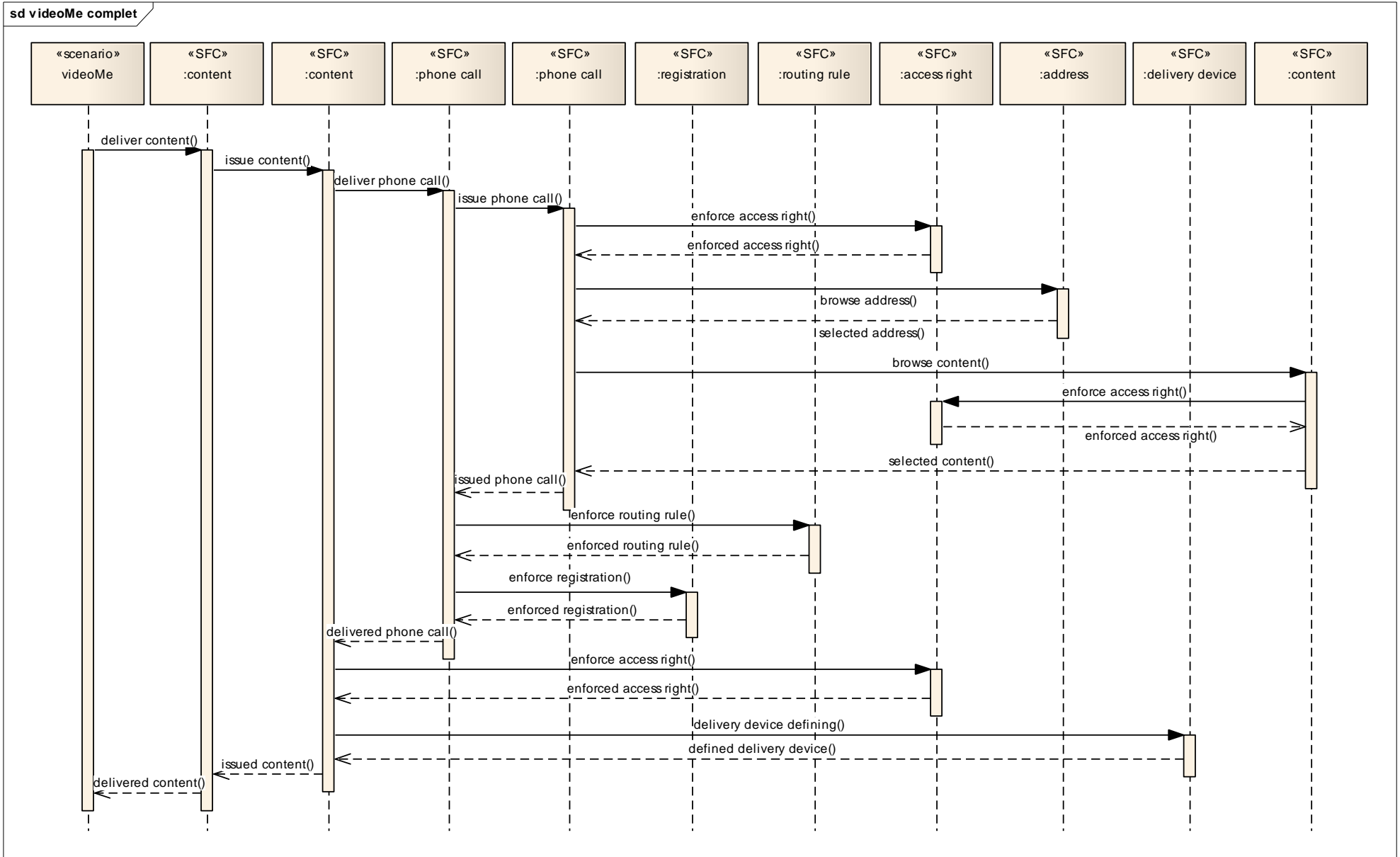


Figure 83 : Séquence fonctionnelle du scénario VideoMe (complète)

Cette figure reprend et combine les figures précédentes. Nous pouvons ainsi visualiser exactement ce qui est combiné des services de téléphonie et de vidéo. Nous voyons ainsi des enchaînements d'opérations repris à l'identique des services précédents, et nous voyons où s'opère la différenciation. Par exemple, la séquence d'opération d'émission d'appel (suite à l'opération *SFO deliver phone call*) est inchangée. Les séquences d'opérations fonctionnelles apparaissant inchangées dans de nombreux services composés nous fourniront des informations intéressantes sur des enablers potentiels. Il y a en effet peut-être un intérêt à définir un enabler réalisant cette séquence.

Soulignons par ailleurs deux nouvelles problématiques fonctionnelles liées à la composition qui apparaissent dans la figure ci-dessus.

D'une part, la nécessité d'une corrélation entre l'authentification d'Alice auprès du service de téléphonie et son authentification auprès du service d'accès aux contenus vidéo (par *SFO enforce access right*).

D'autre part, le lien nécessaire entre l'adresse où distribuer le contenu vidéo et Bob. En effet, comment déterminer l'adresse du media renderer qui va jouer le contenu à Bob ? Le concepteur du service peut par exemple choisir de lier cette adresse de media renderer au numéro de téléphone de Bob, par exemple en lui demandant de télécharger un client qui jouera la vidéo et s'enregistrera auprès du service avec le numéro de téléphone de Bob. C'est cette solution qui est représentée dans le diagramme de séquence précédent avec un lien de *SFC delivery address* vers *SFC reachable address*. La détermination de l'adresse de livraison du contenu nécessite alors l'adresse de joignabilité. Le concepteur du service aurait pu également choisir de lier le media renderer de Bob à son terminal de téléphonie. Le terminal devrait exposer des capacités de vidéo. Ce serait alors par exemple *SFC phone call* qui serait relié à *SFC delivery address*. De même que pour un service unitaire, la vue fonctionnelle de référence permet donc de dégager les grandes problématiques fonctionnelles des services composés.

La vue fonctionnelle du service composé donne une vue de bout en bout du service rendu à travers les capacités d'action mises en œuvre par les acteurs du service, mais ne permet pas de séparer ces fonctions dans les systèmes techniques supportant les services unitaires. Par exemple en indiquant comment la plate-forme de téléphonie et la plate-forme de

vidéo doivent communiquer. Avant de traiter cette question avec la vue applicative d'un service, nous devons déterminer la vue technique de nos services.

IV.2 La vue technique d'un service

Nous venons d'étudier comment décrire la vue fonctionnelle d'un service. Un service peut également être décrit à travers les rôles techniques qu'il met en œuvre. De même que pour la vue fonctionnelle, l'utilisation de la liste des rôles techniques et des points de référence définis dans la vue technique de référence permet à des équipes différentes de décrire les technologies de leurs services avec le même vocabulaire et également d'affiner leur vision technique du service.

IV.2.1 Méthode de description

La description de cette vue technique ne pose pas de difficulté particulière. Il s'agit d'identifier les technologies majeures mises en œuvre dans le service et de les décrire suivant une sémantique commune.

Pour cela, nous devons considérer la solution technique portant le service. Cette solution est ici regardée d'un point de vue uniquement technique, indépendamment du service qu'elle supporte. Prenons l'exemple d'un service élémentaire de téléphonie supporté par un proxy OpenSER et par des terminaux softphone X-Lite. Seront considérés dans la vue technique le proxy OpenSER et le softphone X-Lite, sans considération pour le service rendu. Un service complètement différent du point de vue fonctionnel, comme un service de messagerie instantanée, pourra reposer sur exactement la même solution technique. Par ailleurs, un service fonctionnellement similaire pourra reposer sur une solution technique tout à fait différente, comme un cœur IMS ou même un réseau de pairs, comme dans le cas de Skype.

La solution technique se décompose en nœuds d'exécution, c'est-à-dire en ensemble de matériels et logiciels informatiques sous la responsabilité d'un même projet et se comportant vis-à-vis de l'extérieur comme un tout. Un nœud d'exécution doit ainsi être compris non comme une machine mais comme un périmètre de responsabilité. A ce titre, un nœud d'exécution peut éventuellement être décomposé en nœuds de grain plus fin, détaillant une responsabilité plus finement. Une plate-forme de service Alcatel-Lucent OSP ou un serveur d'application Oracle WebLogic sont par exemple des nœuds d'exécution. De même, le support

d'exécution d'un enabler est un nœud d'exécution, par exemple un serveur XDM [OMA, web] ou un proxy SIP. Dans notre exemple ci-dessus, le proxy OpenSER et le softphone X-Lite sont des nœuds d'exécution. Un composant d'une plate-forme de service pourra également être considéré comme nœud d'exécution si cet élément est suffisamment autonome et possède un périmètre de responsabilité clair. Ces nœuds d'exécution offrent vis-à-vis de l'extérieur des interfaces qui peuvent être caractérisées techniquement par un ou plusieurs points de référence définis dans le référentiel de la vue technique. Le fait qu'un nœud d'exécution expose un point de référence signifie qu'il joue le rôle du rôle technique fournissant ce point de référence. Un AS WebLogic pourra ainsi jouer un rôle de *web service provider*. Si un point de référence relie deux nœuds d'exécution, cela signifie que chacun des nœuds joue le rôle d'un des rôles techniques participant à ce point de référence. Par exemple, le softphone X-Lite et le proxy OpenSER sont reliés par un point de référence SIP client/proxy. Les points de référence permettent de décrire les nœuds d'exécution avec une sémantique commune à tous les services. La vue technique de référence permet également à l'architecte de questionner sa vue technique d'un service, en vérifiant que des nœuds d'exécution jouent bien les rôles de tous les rôles techniques pertinents pour le service.

Nous pouvons formaliser ces concepts dans le méta-modèle de description des services.

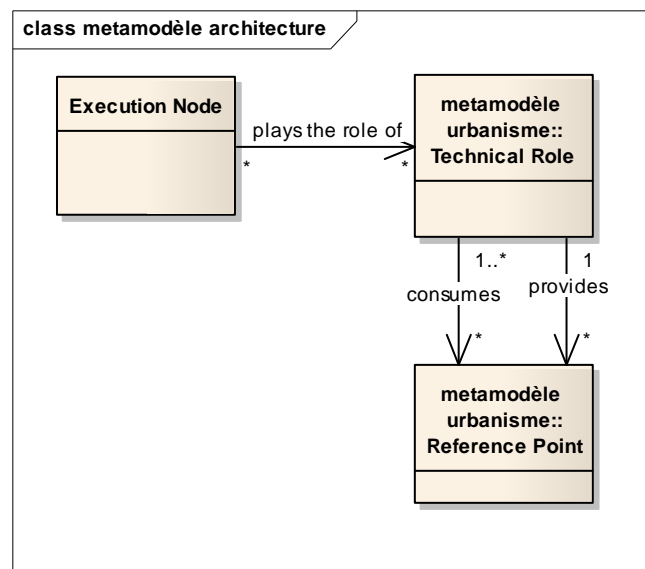


Figure 84 : Méta-modèle de la vue technique d'un service

La méthode de description de la vue technique consiste à identifier les nœuds d'exécution, puis à décrire les liens entre eux ou les interfaces externes qu'ils exposent à travers les points de référence définis dans la vue technique du référentiel. De nouveaux points de référence pourront être ajoutés si nécessaire. En première analyse, on se limitera aux nœuds d'exécution de plus haut niveau. Une découpe fine des différents nœuds d'exécution, par exemple en allant jusqu'au niveau du système d'exploitation d'un équipement, ne sera réalisée que si elle permet de clarifier une problématique architecturale importante pour le projet. Les liens entre nœuds d'exécution ne seront pas nécessairement les liens physiques, mais plutôt les liens logiques, répondant à une problématique architecturale. Par exemple, le lien entre un navigateur et un serveur web pourra être représenté directement, sans l'intermédiaire de routeurs, proxy et firewalls. Une vue plus détaillée pourra ensuite être réalisée en précisant la réalisation précise de chaque point de référence en terme d'infrastructure utilisée. En somme, la vue technique peut être considérée comme une représentation haut-niveau d'une vue détaillée de l'infrastructure.

Reprenons notre exemple d'un service de carnet d'adresses. Techniquement, ce service est réalisé par un enabler OMA de carnet d'adresses, appelé NAB pour *Network Address Book*. Cet enabler étant encore en cours de spécification, nous adopterons une architecture technique simplifiée, où le service est porté par :

- Un utilisateur qui joue le rôle de personne au sens du motif technique IHM. Il est relié au client NAB par le point de référence GUI, implémenté par l'interface utilisateur du client NAB.
- Un client NAB, le terminal qui permet de consulter le carnet d'adresses. Il joue le rôle d'un client XDM au sens du motif technique XDM. Il est relié au serveur carnet d'adresses par le point de référence XDM du motif technique XDM, ce point de référence étant implémenté par le protocole XCAP. Ce client NAB offre un point de référence GUI, du motif IHM, qui permet à un utilisateur de le manipuler. Il joue donc également le rôle d'une machine, au sens du motif IHM.
- Un serveur NAB, qui stocke les listes de contacts de tous les utilisateurs. Il joue le rôle d'un serveur XDM au sens du motif technique XDM présenté au chapitre précédent. Il offre le point de référence XDM-AP pour être interrogé par un client XDM.

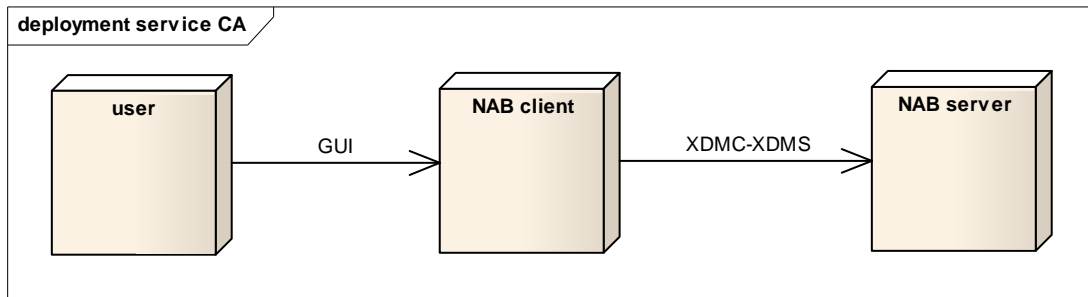


Figure 85 : Vue technique d'un service de carnet d'adresses

Ces nœuds d'exécution et leurs points de référence constituent la vue technique du service de carnet d'adresses. Le sens des flèches indique le sens d'émission des requêtes : l'utilisateur invoque le NAB client via le point de référence GUI.

IV.2.2 Exemples de services simples

Reprenons maintenant nos trois exemples précédents, avec les services de téléphonie, de vidéo et de messagerie.

IV.2.2.1 Exemple d'un service de téléphonie

Le service de téléphonie est par exemple déployé avec une solution technique de type IMS. Les nœuds d'exécution jouent les rôles techniques du motif IMS :

- le terminal téléphonique, qui joue le rôle de l'UE,
- le serveur d'appels, qui joue le rôle de CSCF,
- le serveur de profils, qui joue le rôle de HSS,
- le serveur d'applications de téléphonie, qui joue le rôle d'AS.

Ces nœuds sont reliés par les points de référence tels que définis dans la vue technique de référence. Le service a également besoin d'une plate-forme pour gérer les carnets d'adresses des utilisateurs et utilise pour cela le serveur NAB défini dans la vue technique de notre service de carnet d'adresses. Le rôle de client NAB est ici joué par le terminal téléphonique. Il y a donc un seul nœud d'exécution, le terminal qui joue à la fois le rôle de terminal téléphonique et de client XDM. Ceci est manifesté par les deux points de référence utilisés par le terminal : Gm pour son rôle de terminal téléphonique et XDM pour son rôle de client NAB.

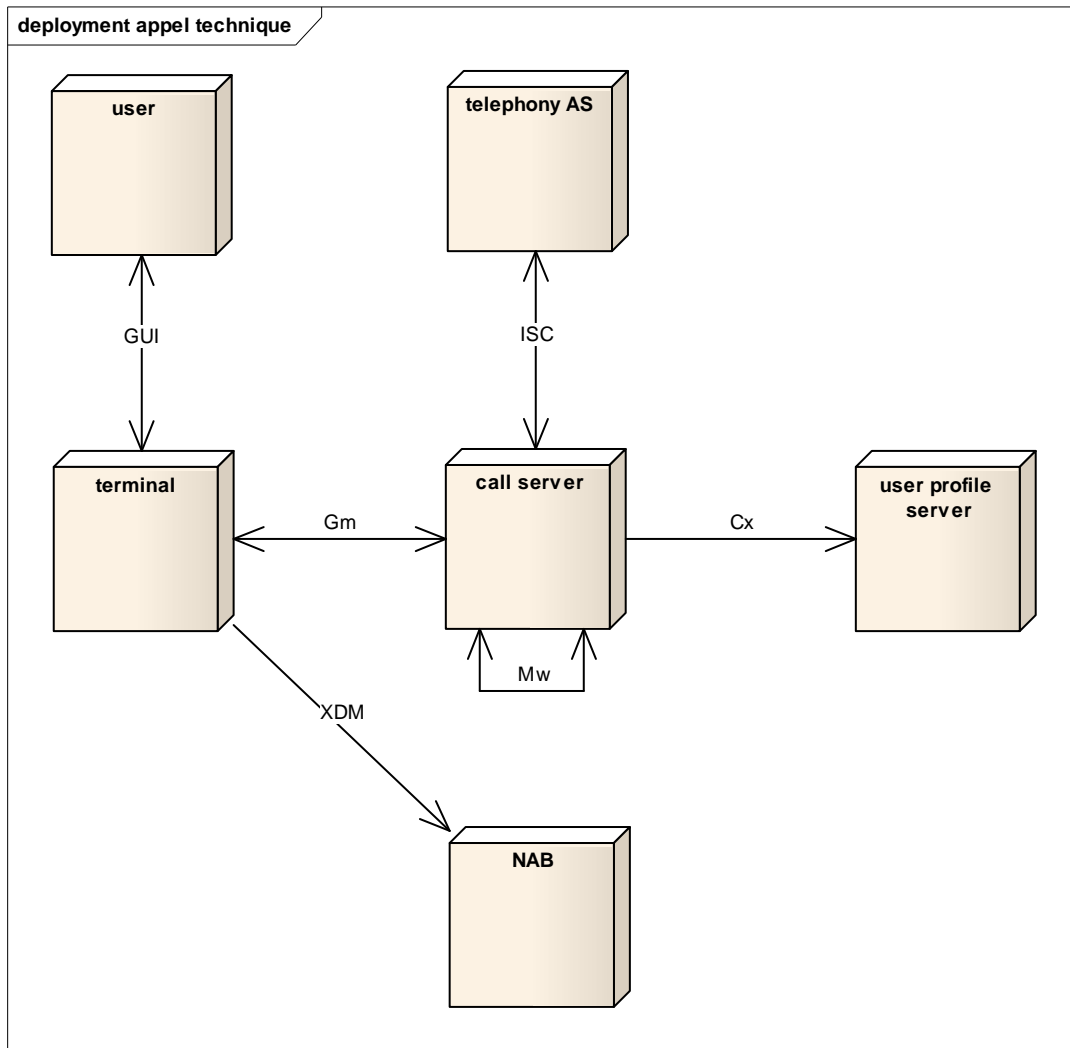


Figure 86 : Vue technique d'un service de téléphonie

Les liens avec les éléments de la vue technique de référence sont indiqués dans le tableau suivant.

Tableau 24 : Nœuds d'exécution d'un service de téléphonie

Nœud d'exécution	Rôle technique joué	Issu du motif technique
user	person	IHM
phone	machine	IHM
	UE	IMS
	XDM client	XDM
call server	CSCF	IMS
telephony AS	AS	IMS

user profile server	HSS	IMS
NAB	XDM server	XDM

On peut également indiquer les liens avec les points de référence de la vue technique de référence.

Tableau 25 : Points de référence d'un service de téléphonie

Nœud d'exécution	Point de référence exposé	Issu du motif technique
user	-	-
terminal	GUI	IHM
	Gm	IMS
call server	Gm	IMS
	ISC	IMS
telephony AS	ISC	IMS
user profile server	Cx	IMS
NAB	XDM	XDM

IV.2.2.2 Exemple d'un service de vidéo

Le service de vidéo est par exemple déployé avec une solution technique de type triple play. Les nœuds d'exécution sont un décodeur vidéo appelé STB pour Set-Top Box, un portail appelé Content Access Portal, une plate-forme de commande vidéo appelée IPTV VOD Controller, et une plate-forme de diffusion de VOD (pour *Video On Demand*, vidéo à la demande) appelée VOD Delivery.

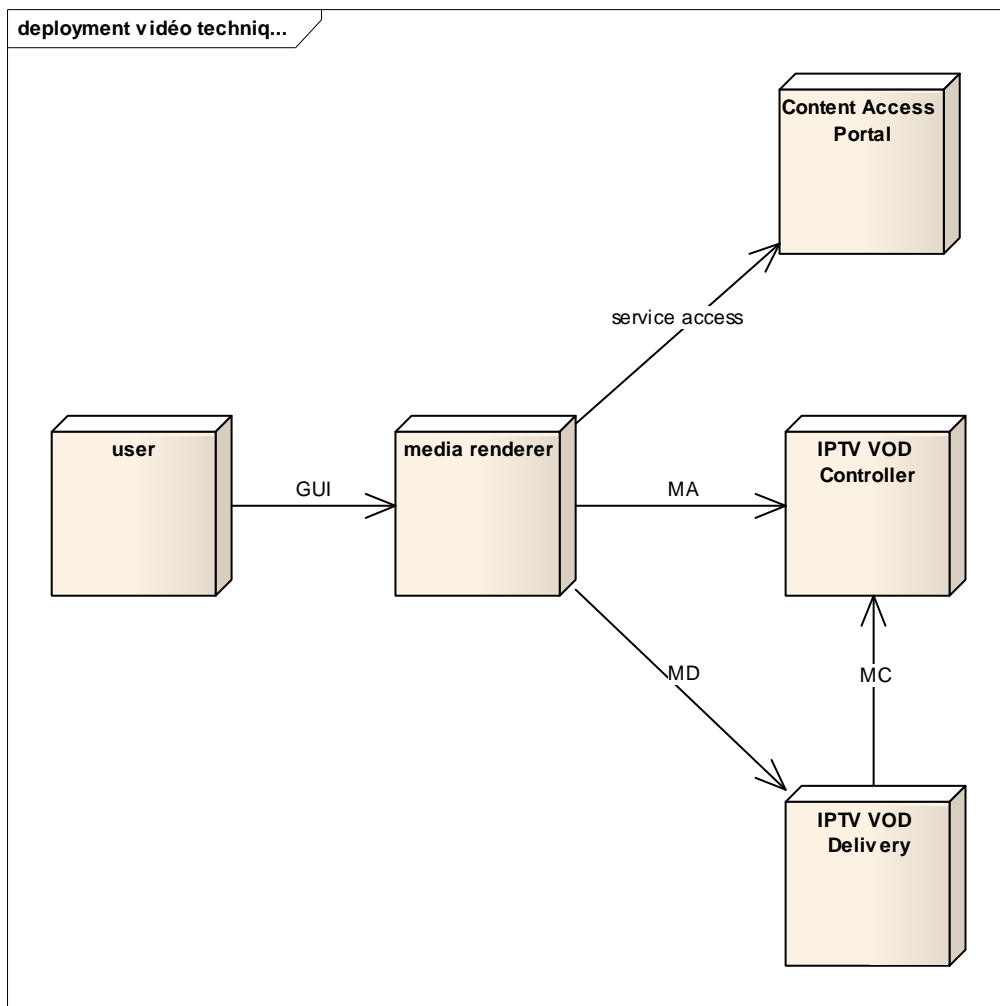


Figure 87 : Vue technique d'un service de vidéo

Les liens avec les éléments de la vue technique de référence sont indiqués dans le tableau suivant.

Tableau 26 : Nœuds d'exécution d'un service de vidéo

Nœud d'exécution	Rôle technique joué	Issu du motif technique
user	person	IHM
media renderer	machine	IHM
	web page	web
	media renderer	IPTV
content access portal	service logic	Web
IPTV VOD controller	media controller	IPTV
IPTV VOD delivery	media delivery	IPTV

On peut également indiquer les liens avec les points de référence de la vue technique de référence.

Tableau 27 : Points de référence d'un service de vidéo

Nœud d'exécution	Point de référence exposé	Issu du motif technique
media renderer	GUI	IHM
Content access portal	service access	web
IPTV VOD controller	MA	IPTV
	MC	IPTV
IPTV VOD delivery	MD	IPTV

IV.2.2.3 Exemple d'un service de messagerie

Le service de messagerie est par exemple déployé avec une solution technique de type webmail. Les nœuds d'exécution sont une page web, une plate-forme de service webmail, un mail back-end contenant les boîtes aux lettres, un serveur SMTP et une plate-forme de gestion de l'identité et de l'authentification, que nous appellerons IDP pour IDentity Provider.

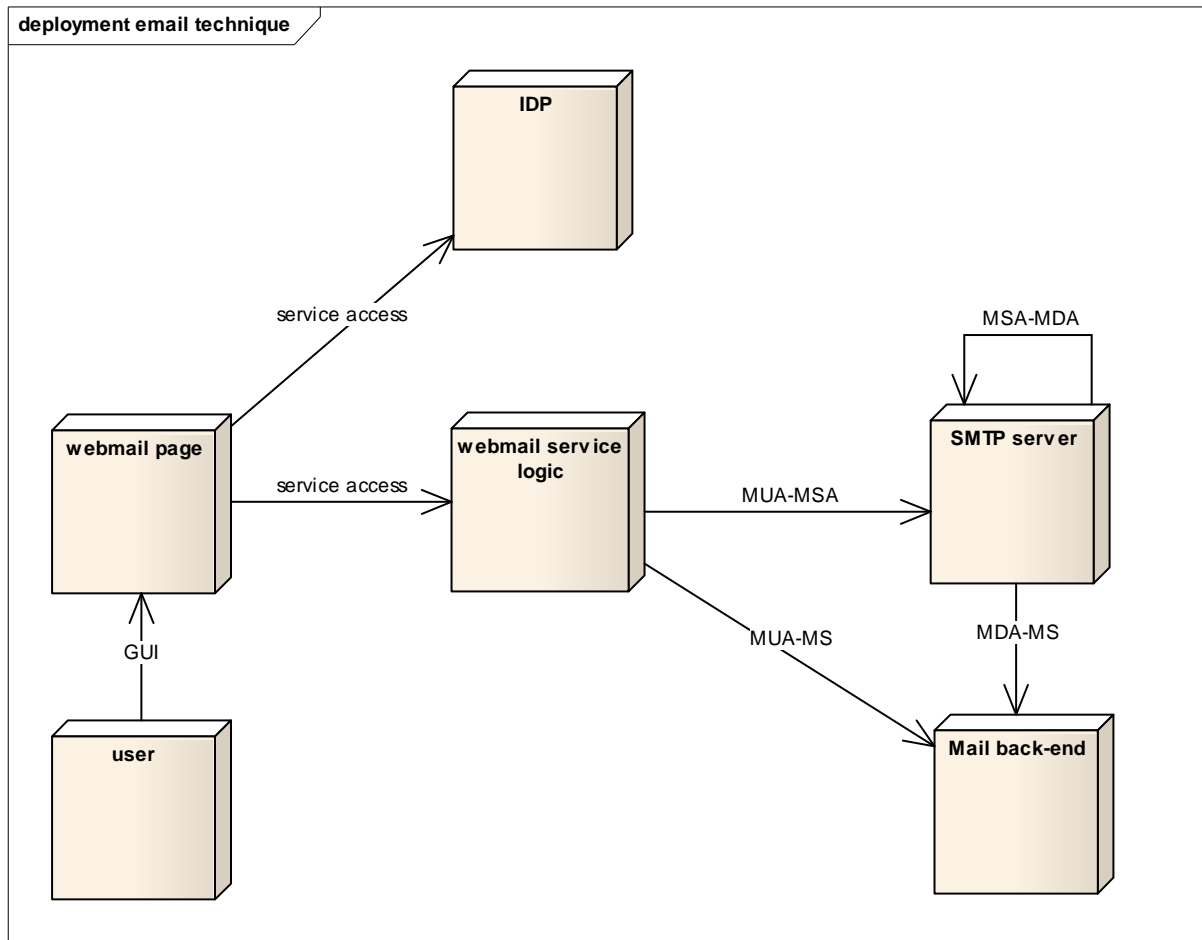


Figure 88 : Vue technique d'un service de messagerie email

Les liens avec les éléments de la vue technique de référence sont indiqués dans le tableau suivant.

Tableau 28 : Nœuds d'exécution d'un service de messagerie email

Nœud d'exécution	Rôle technique joué	Issu du motif technique
user	person	IHM
webmail page	machine	IHM
	web page	web
webmail service logic	service logic	web
	MUA	email
IDP	service logic	web
SMTP server	MSA	email
	MTA	email

	MDA	email
mail back-end	MS	email

On peut également indiquer les liens avec les points de référence de la vue technique de référence.

Tableau 29 : Points de référence d'un service de messagerie email

Nœud d'exécution	Point de référence exposé	Issu du motif technique
webmail page	GUI	IHM
webmail service logic	service access	Web
IDP	service access	Web
SMTP server	MUA-MSA	email
	MSA-MDA	email
mail back-end	MUA-MS	email
	MDA-MS	email

IV.2.3 Cas d'un service composé : VideoMe

Si pour la vue fonctionnelle d'un service composé, le service était envisagé de bout en bout, le service va au contraire être ici découpé selon les systèmes techniques responsables de l'exécution de ce service. Nous aurons donc un nœud d'exécution téléphonie et un nœud d'exécution vidéo, contenant chacun les nœuds d'exécution vus précédemment. La séparation entre ces nœuds d'exécution de haut-niveau se fait également par les points de référence. Cette découpe permet de décrire avec précision les systèmes techniques mis en œuvre dans un service composé, comme une composition des nœuds d'exécution des services unitaires.

La construction de la vue technique est bien sûr liée à des considérations techniques, mais également à des considérations fonctionnelles. Par exemple, dans la vue fonctionnelle de notre service composé, nous nous sommes posé la question du lien entre l'adresse où distribuer le contenu vidéo et Bob. La réponse choisie dans la vue fonctionnelle a un impact sur le point de référence à utiliser entre les deux systèmes, et réciproquement. Nous voyons ainsi combien l'étude de la vue fonctionnelle et celle de la vue technique doivent être menées

concomitamment, même si nous les présentons ici successivement pour clarifier notre propos. Le lien entre ces deux vues sera l'objet de la vue applicative.

Dans notre exemple, le système videoMe jouera les rôles de terminal et de media renderer. La composition entre les services de téléphonie et de vidéo est donc réalisée au niveau *end-point* et non à un niveau serveur. Le fonctionnement technique est donc le suivant :

- Le terminal de téléphonie est spécifique au service VideoMe
- Le media renderer est spécifique au service VideoMe
- Le système de téléphonie est inchangé
- Le système de consultation de vidéo est inchangé

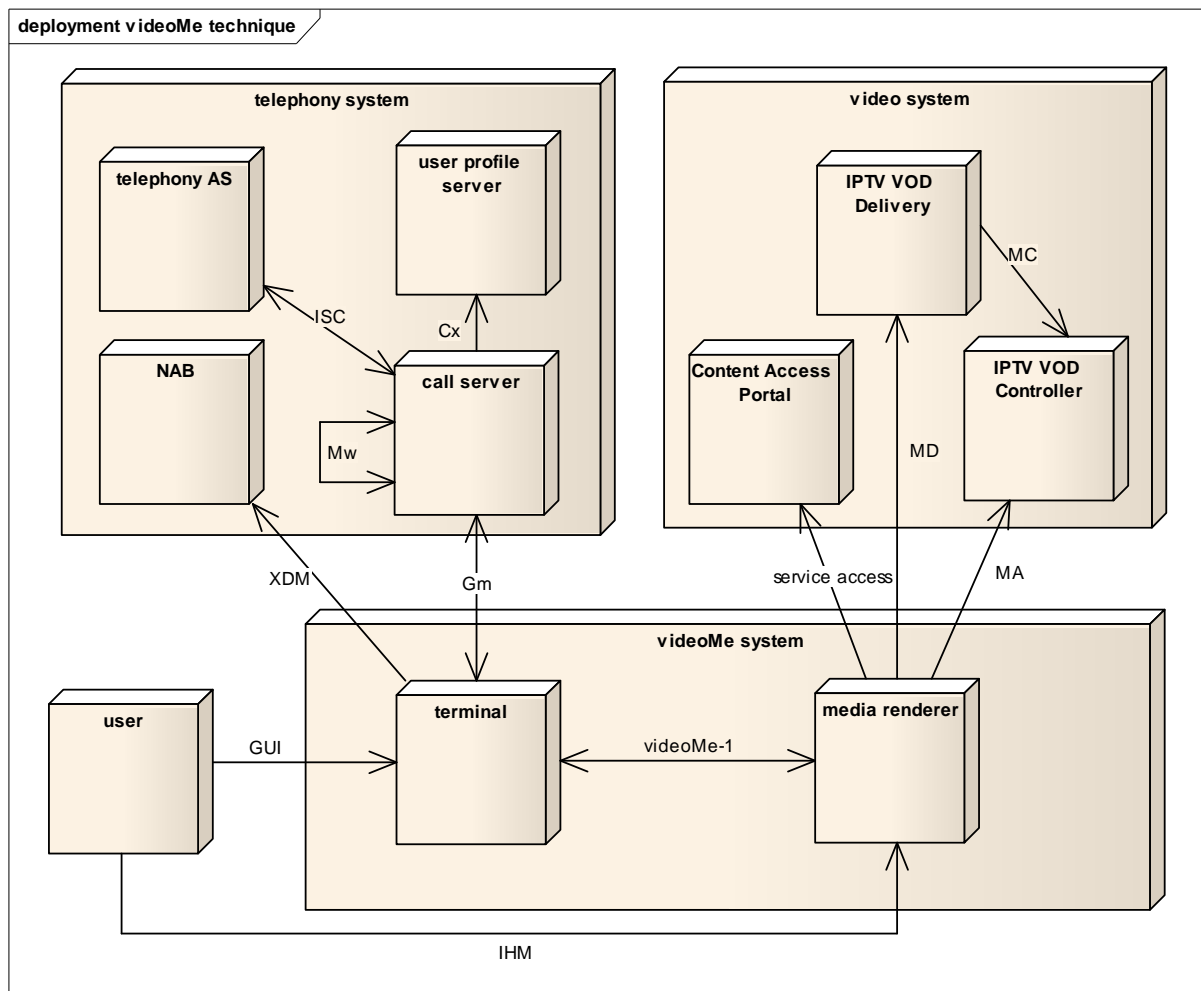


Figure 89 : Vue technique du service VideoMe

IV.3 La vue applicative d'un service

La vue applicative d'un service constitue le lien entre la vue fonctionnelle et la vue technique de ce service. Dans les bonnes pratiques de l'urbanisation des systèmes d'information présentées dans le premier chapitre, cette vue contient les applications du système informatique, ces applications implémentant des fonctions de la vue fonctionnelle sur des nœuds d'exécution de la vue technique. Dans notre cas des systèmes de service télécoms, cette vue contiendra à la fois des applications et des composants plus orientés télécoms tel un serveur d'appels.

La spécificité de la vue applicative par rapport à la vue technique est moins intuitive dans le monde télécom que dans le monde informatique. Dans les systèmes informatiques, la dissociation entre application et nœud d'exécution est relativement claire. Un Application Server désigne ainsi sans équivoque en informatique un nœud d'exécution qui va supporter des applications, sans présupposé sur ce que réalise cette application. Un Application Server supportera aussi bien une application de réservation de salles de réunion qu'une application de traduction automatique de documents ou une application de poker en ligne. Dans le monde télécom, le terme Application Server désigne à la fois un nœud d'exécution et l'application portée par ce nœud d'exécution. On parle ainsi d'application server de téléphonie ou d'application server de présence au-dessus d'un cœur IMS. Ceci est probablement une conséquence du lien fort entre protocole et service dans le monde télécom, comme nous l'avons évoqué dans la section 3 du chapitre précédent. Autant un protocole comme http est utilisé pour véhiculer tout type de données, et permet même le transport de données d'établissement d'appels téléphoniques [2008-6] ou de données destinées au temps-réel – comme lorsque des applications comme Skype encapsulent des flux audio temps-réels sur http afin de contourner les protections des firewalls. Autant un protocole comme SIP est conçu pour véhiculer des données d'établissement de sessions multimédia, et reste essentiellement utilisé pour transporter des données d'établissement d'appels téléphoniques. Ceci est la conséquence de deux cultures différentes. Dans le monde informatique, la fourniture de la connectivité réseau et la fourniture des applications sont historiquement séparées et relèvent d'acteurs différents. Dans le monde télécom, la fourniture de la connectivité réseau et la fourniture des services télécoms (au premier rang desquels la téléphonie) sont historiquement confondues et relèvent des mêmes acteurs, ce regroupement étant sans doute à l'origine dû aux contraintes du temps-réel et au caractère universel de la téléphonie – un service de

téléphonie doit permettre de joindre n'importe quel connecté au réseau. Cette différence entre culture informatique et télécom est toutefois de moins en moins pertinente avec la convergence entre applications informatiques et applications télécoms, illustrée par exemple par MSN Messenger, Skype ou Google Grand Central. Cela motive notre volonté de promouvoir néanmoins une dissociation forte entre vue fonctionnelle, vue technique et vue applicative, y compris pour les services télécoms. La distinction entre vue applicative et vue technique pour les services télécoms sera alors une question d'angle de vue, plus que de différence de nature. Un application server de téléphonie sera ainsi vu dans la vue technique uniquement à travers les interfaces protocolaires qu'il expose et utilise – décrites par les points de référence –, et à travers son comportement vis-à-vis de ces interfaces – décrites par les rôles techniques. Dans cette vue technique, un application server de téléphonie ou de présence pourront être décrit semblablement si leurs interfaces protocolaires sont identiques. Alors que dans la vue applicative, ces mêmes application server seront vus comme des applications réalisant des fonctions différentes, quoique portées par des nœuds d'exécution similaires.

IV.3.1 Comment identifier et décrire les composants applicatifs d'un service ?

La vue applicative d'un service décrit l'implémentation de la vue fonctionnelle de ce service sur la vue technique. Cette vue est constituée de composants applicatifs, exposant des opérations applicatives, reliés entre eux par des connecteurs, et gérant des données applicatives. Nous appellerons ces composants applicatifs SAC pour *Service Applicative Component*.

Les composants applicatifs peuvent être par exemple un serveur d'appel ou un serveur de présence. Un composant applicatif est décrit par le ou les opérations fonctionnelles SFO qu'il implémente, et par le ou les nœuds d'exécution utilisés à cette fin. Comme nous l'avons évoqué précédemment, la découpe en composants applicatifs permet le découplage entre les différents modules d'un projet (pour le développement logiciel) ou entre les différentes sphères de responsabilité (pour un déploiement de plates-formes télécom), par exemple entre réseau et plates-formes de service, ou entre différents enablers. L'objectif n'est pas de décrire en détail le fonctionnement interne du logiciel ou des plates-formes ; cela sera du ressort de l'équipe chargée du développement logiciel ou de l'équipe chargée de l'intégration de la plate-forme. Cette description détaillée peut d'ailleurs être vue comme un raffinement, un "zoom" sur chaque composant applicatif. La structuration en composants applicatifs est ainsi un

moyen de séparer les principaux modules du service en précisant les responsabilités de chacun, avant d'entrer plus avant dans une conception détaillée. Les données applicatives sont les données majeures manipulées par le composant applicatif. Chaque composant applicatif est ainsi responsable de données applicatives. Un composant applicatif peut contenir des composants applicatifs de plus bas niveau, ce qui permet de détailler une architecture en "zoomant" sur chaque composant applicatif.

La description de l'architecture logicielle en composants applicatifs, données applicatives, opérations applicatives et connecteurs est classique en génie logiciel et dans l'urbanisation des systèmes informatiques. L'architecture logicielle est ainsi définie par le Software Engineering Institute de l'université Carnegie Mellon [SEI, web] de la façon suivante.

"The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationship among them"

Nous voyons apparaître dans cette définition les notions de composants (*software elements*) et de connecteurs portant les interactions entre composants (inclus dans les *relationship among them*). Nous les représenterons par des classes UML stéréotypées :

- des composants UML stéréotypés SAC, *Service Applicative Component* pour les composants applicatifs ;
- des opérations UML appelées SAO, *Service Applicative Operation* pour les opérations applicatives de service, qui sont les opérations portées par les composants applicatifs SAC ;
- des ports de composants SAC, exprimant les rôles techniques réalisés par ces composants SAC, c'est-à-dire les rôles techniques dont le rôle est joué par le nœud d'exécution dont découle le composant ;
- des classes d'association pour les connecteurs (selon la proposition de [Ivers, 2004]), qui relient les ports des composants applicatifs entre eux via un point de référence, et portent une ou plusieurs opérations applicatives SAO. Le sens de l'association est celui de l'invocation des opérations. Chaque classe d'association est nommée par le nom du point de référence qu'elle implémente.

Comment identifier les composants applicatifs d'un service ? Trois approches différentes sont envisageables : laisser le périmètre des composants applicatifs à l'appréciation

des personnes manipulant la vue applicative, ou les circonscrire à partir de la vue fonctionnelle, ou à partir de la vue technique.

La première approche, dans un esprit bottom-up, serait de laisser le soin à l'architecte d'un projet de service le soin d'identifier les composants applicatifs de son service, en lien avec l'équipe responsable du développement (dans le cadre d'un projet axé développement informatique) ou de l'intégration (dans le cadre d'un projet axé intégration télécom). Cette approche présente l'avantage d'une très bonne appropriation de l'architecture par l'équipe (de développement ou d'intégration) en charge de la mettre en œuvre. Les composants applicatifs seraient alors les composants intéressants du point de vue du développement ou de l'intégration. Cette approche présente au moins deux inconvénients. D'une part, les architectures applicatives de différents projets seront peu comparables, car élaborées par des équipes différentes selon des critères différents, cette diversité pouvant même se retrouver au sein du même service, lorsque des équipes différentes interviennent ou pour les services composés. D'autre part, il y a un risque que les composants soient d'une granularité excessivement fine afin de refléter les spécifications de conception détaillée. Cette approche ne permet donc pas de garantir une bonne séparation entre la logique de l'architecture et celles du développement ou de l'intégration.

La deuxième approche, guidée par le fonctionnel, serait d'identifier un composant applicatif par fonction élémentaire SEF ou par composant fonctionnel SFC. On aurait ainsi par exemple un composant applicatif *SAC Call issuing* ou un composant *SAC Call* qui serait supporté dans le cas de l'IMS par les nœuds d'exécution UE, CSCF, HSS et AS. Le composant applicatif *SAC Call issuing* représenterait alors l'ensemble des interactions entre UE, CSCF, HSS et AS. Cette approche présente de façon très claire le lien entre la vue fonctionnelle et la vue technique, à travers le lien entre un composant applicatif (ici directement issu d'un composant fonctionnel) et un nœud d'exécution. Un composant applicatif sera fortement cohérent, car incluant tous les éléments nécessaires à la réalisation du composant fonctionnel auquel il correspond. Il sera faiblement couplé car ces interfaces externes correspondront à des interactions entre composants fonctionnels différents. Par contre, cette approche reproduit in fine dans la vue applicative la logique de la vue fonctionnelle. Or, les fonctions élémentaires SEF ou les composants fonctionnels SFC de la vue fonctionnelle sont par nature des fonctions de bout et bout, qui vont mobiliser une chaîne de nœuds d'exécution pour être réalisées. Ainsi, dans notre exemple IMS, le *SFC call issuing* est implémenté par un

enchaînement d'applications, applications supportées par les nœuds d'exécution UE, CSCF, HSS, AS. Cette approche conduirait à une granularité de trop haut niveau dans la vue applicative ; un seul composant applicatif regroupant UE, CSCF, HSS et AS serait de trop haut niveau pour présenter de façon compréhensible par les équipes de développement ou d'intégration l'architecture applicative qu'ils auront à mettre en œuvre. De plus, un autre composant applicatif reposant sur les mêmes nœuds d'exécution, par exemple pour la publication de présence, serait issu d'un autre composant fonctionnel et donnerait donc lieu à un autre composant applicatif, sans lien avec le premier dans la vue applicative. A l'opposé de la première approche, ceci conduit à une vue applicative trop peu efficace pour les équipes de développement ou d'intégration du service, et qui serait sans doute perçue comme trop théorique et peu pertinente vis-à-vis de leurs préoccupations.

La troisième approche, qui nous semble un compromis intéressant, comme nous l'illustrerons dans la section suivante, est de définir :

- un composant applicatif SAC pour chaque nœud d'exécution du service ;
- un connecteur entre SAC pour chaque point de référence entre nœud d'exécution ;
- une opération applicative SAO pour chaque opération fonctionnelle SFO au rendu duquel ce composant applicatif SAC participe, cette opération applicative étant portée par un connecteur ;
- une donnée fonctionnelle SAD pour chaque donnée fonctionnelle SFD produite par une opération fonctionnelle SFO au rendu duquel ce composant applicatif SAC participe.

Dans notre exemple IMS, cela donne un composant applicatif UE, un composant CSCF, un composant HSS et un composant AS, chacun de ces composants exposant une opération issue de SFO issue phone call. Ces opérations sont respectivement déployées sur des connecteurs issus des points de référence Gm, Cx et ISC de la vue technique du service. Ces opérations applicatives ne réalisent pas chacune la totalité du SFO, mais participent à la réalisation de ce SFO, chaque opération représentant une étape dans une réalisation du SFO de bout en bout. Chaque opération a donc une valeur ajoutée propre. Cette valeur ajoutée sera vérifiée en lien avec l'équipe de développement afin de ne pas confondre trajet des flux de données et plus-value fonctionnelle (un composant applicatif n'offre une opération applicative que s'il est réellement impliqué dans le rendu de l'opération fonctionnelle dont cette opération applicative est issue, et pas uniquement s'il est présent sur le trajet des flux de données). Cette approche présente l'avantage d'un alignement, par construction, entre la vue applicative et la vue

technique, permettant d'homogénéiser cette vue applicative. Elle permet également de prendre en compte la réalisation de bout en bout d'un composant fonctionnel par un enchaînement de nœuds d'exécution. La méthode proposée pour identifier les composants applicatifs est donc :

1. A partir des séquences fonctionnelles décrivant le service dans la vue fonctionnelle, identifier les nœuds d'exécution impliqués dans la réalisation de chaque opération fonctionnelle invoquée dans cette séquence. On obtient n nœuds d'exécution, chacun étant impliqué dans o_i opérations fonctionnelles. Ces n nœuds d'exécution sont reliés par m points de référence.
2. Créer n composants applicatifs, un par nœud d'exécution.
3. Créer m connecteurs entre composants applicatifs, un par point de référence entre nœuds d'exécution dont sont issus les composants applicatifs, l'orientation éventuelle de ce connecteur découlant de celle du point de référence. Un connecteur donne lieu à un port dans chaque composant applicatif que relie ce connecteur, un port représentant le rôle technique joué par le nœud d'exécution dont découle le composant.
4. Pour chaque composant applicatif, créer o_i opérations applicatives correspondant aux opérations fonctionnelles dans la réalisation desquelles est impliqué le nœud d'exécution dont le composant applicatif est issu. Une opération applicative peut se référer à plusieurs opérations fonctionnelles lorsque ces dernières sont fortement liées.
5. Valider avec les équipes en charge du développement ou de l'intégration la valeur ajoutée de chaque opération applicative créée et la décrire. Cela pourra conduire à fusionner des opérations applicatives ou à en dissocier.
6. Rattacher chaque opération applicative au connecteur sur lequel elle est déployée.
7. Pour chaque composant applicatif, créer o_i données applicatives, une par donnée fonctionnelle constituant le paramètre de retour des opérations fonctionnelles dont sont issues les opérations applicatives du composant applicatif.
8. Valider avec les équipes en charge du développement ou de l'intégration la valeur ajoutée de chaque donnée applicative créée et la justifier textuellement.

Les opérations applicatives modélisent ainsi la participation du composant applicatif à la réalisation d'une opération fonctionnelle, chaque opération applicative jouant un rôle particulier dans cette réalisation. Cette méthode fournit un moyen simple pour appliquer le principe de faible couplage et celui de forte cohérence. Les composants applicatifs qui participent à la réalisation de la même opération fonctionnelle forment un ensemble de composants fortement cohérent (cohésion fonctionnelle). Dans un système de téléphonie, tous les éléments impliqués dans la réalisation de la SFO issue phone call forment un ensemble fortement cohérent, comme par exemple UE, CSCF HSS et AS de téléphonie. Les liens de communication entre rôles techniques pour implémenter la SFO (par exemple les messages SIP circulant entre ces éléments pour réaliser l'émission d'un appel) peuvent être présentés par

un diagramme de séquence, éventuellement en indiquant les connecteurs portant ces opérations. Les flux de données précis (par exemple les messages SIP utilisés) seront plutôt détaillés par les équipes en charge de ce composant applicatif. Ces concepts sont récapitulés dans le méta-modèle ci-dessous. Le lien entre connecteurs et points de référence reprend la proposition de F. Menai dans [Menai, 2005].

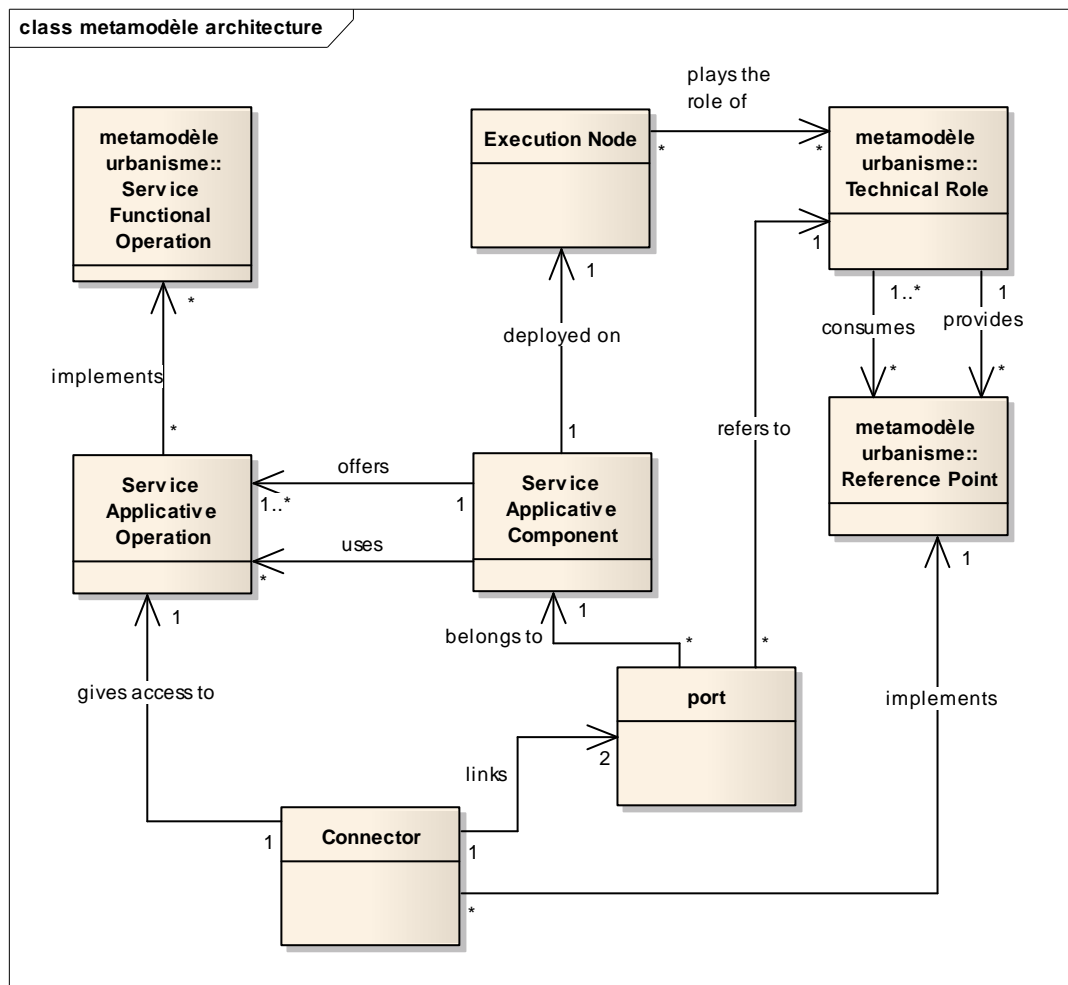


Figure 90 : Méta-modèle de la vue applicative d'un service

Reprenons notre exemple d'un service de carnet d'adresse. Les quatre opérations fonctionnelles du système de service sont : *SFO Enforce access right*, *SFO Define contact list*, *SFO Publish reachable address*, et *SFO Browse reachable address*. Les nœuds d'exécution sont l'utilisateur, le client NAB et le serveur NAB, qui sont a priori chacun impliqués dans les quatre opérations fonctionnelles.

Nous en déduisons donc trois composants applicatifs : *SAC user*, *SAC NAB client* et *SAC NAB serveur*. Les connecteurs entre ces composants SAC sont les suivants :

- Un connecteur entre *SAC user* et *SAC client NAB*, qui implémente le point de référence GUI
- Un connecteur entre *SAC client NAB* et *SAC serveur NAB*, qui implémente le point de référence XDM

SAC user ne possède aucune opération applicative. Chacun des deux autres composants applicatifs possède quatre opérations applicatives, correspondant aux quatre opérations fonctionnelles, dont la valeur ajoutée peut être justifiée. Concernant le composant applicatif *SAC NAB client*:

- L'opération applicative *SAO Launch terminal* permet d'initialiser le terminal et d'authentifier l'utilisateur. Elle correspond à l'opération fonctionnelle *SFO Enforce access right*. Elle est portée par le point de référence GUI du client NAB.
- L'opération applicative *SAO Create list* permet à l'utilisateur de demander la création d'une liste de contacts. Elle correspond à l'opération fonctionnelle *SFO Define address list*. Elle est portée par le point de référence GUI du client NAB.
- L'opération applicative *SAO Create contact* permet à l'utilisateur d'entrer un nouveau contact à travers un formulaire. Elle correspond à l'opération fonctionnelle *SFO Publish address*. Elle est portée par le point de référence GUI du client NAB.
- L'opération applicative *SAO Select contact* permet de sélectionner un contact dans une liste. Elle correspond à l'opération fonctionnelle *SFO Browse address*. Elle est portée par le point de référence GUI du client NAB.

Concernant le composant applicatif *SAC NAB server* :

- L'opération applicative *SAO Authenticate* permet d'initialiser le terminal et d'authentifier l'utilisateur. Elle correspond à l'opération fonctionnelle *SFO Enforce access right*. Elle est portée par le point de référence XDM du serveur NAB.
- L'opération applicative *SAO Set list* permet de créer une liste de contacts (que ce soit par l'utilisateur via son client NAB ou directement par un autre composant logiciel, par exemple du système d'information de prise de commande). Elle correspond à l'opération fonctionnelle *SFO Define address list*. Elle est portée par le point de référence XDM du serveur NAB.

- L'opération applicative *SAO Set contact* permet de créer un contact. Elle correspond à l'opération fonctionnelle *SFO Publish address*. Elle est portée par le point de référence XDM du serveur NAB.
- L'opération applicative *SAO Browse NAB* permet de naviguer dans le carnet d'adresse. Elle correspond à l'opération fonctionnelle *SFO Browse address*. Elle est portée par le point de référence XDM du serveur NAB.

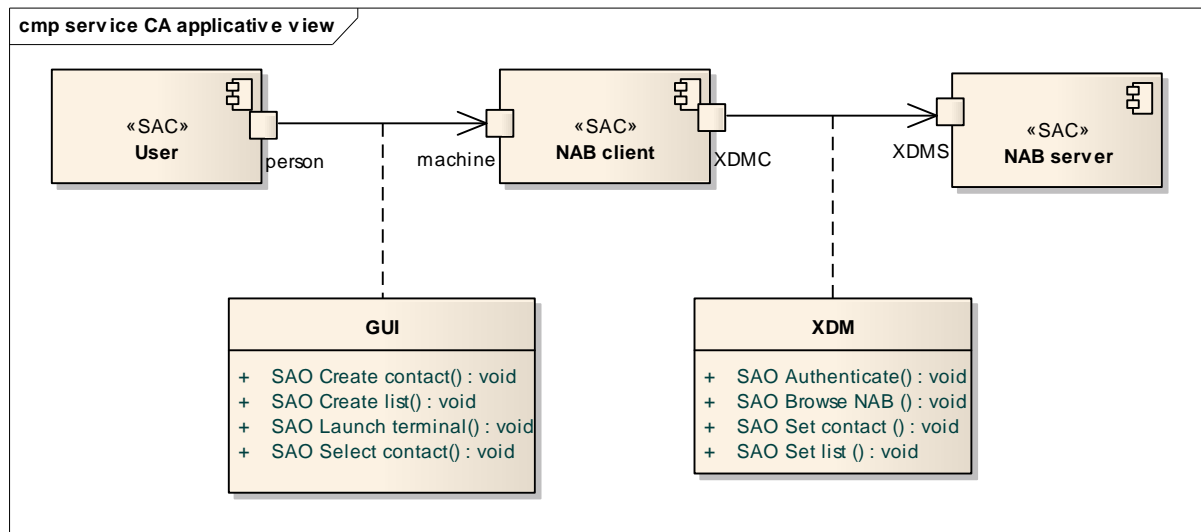


Figure 91 : Vue applicative d'un service de carnet d'adresse

Nous avons ainsi décrit la vue applicative de notre service de carnet d'adresse, en corrélant sa vue fonctionnelle avec sa vue technique. Les opérations applicatives offertes par chaque composant applicatif présentent effectivement une valeur ajoutée. Et cette vue applicative est claire, simple et univoque pour une équipe de développement, qui pourra aisément se l'approprier.

IV.3.2 Exemples de services simples

Détaillons maintenant les vues applicatives de nos trois exemples de service.

IV.3.2.1 Exemple d'un service de téléphonie

Dans le cas du service de téléphonie, les composants applicatifs sont donc l'utilisateur, le terminal, le serveur d'appel, l'AS de téléphonie, le serveur de profils et le NAB, tels que vus

précédemment. Ces composants offrent les opérations applicatives décrites dans le tableau ci-dessous, issues des opérations fonctionnelles indiquées.

Tableau 30 : Lien applicatif-fonctionnel pour un service de téléphonie

Composant applicatif	Opération Applicative offerte	Issue de l'opération fonctionnelle
user	Alert user	Deliver phone call
terminal	Launch terminal	Enforce access right
	Select contact	Browse address
	Launch call	Issue phone call
	Deliver call	Deliver phone call
call server	Register	Enforce access right
	Route originating call	Issue phone call
	Route terminating call	Enforce Registration
		Deliver phone call
user profile server	Assignment request	Enforce access right
	Get profile	Enforce registration
		Enforce routing rule
telephony AS	Trigger supplementary services	Enforce routing rule
NAB	Browse NAB	Browse address

L'exposition d'une opération par le composant applicatif représentant l'utilisateur peut paraître abusive. Néanmoins, si l'on considère l'utilisateur comme un nœud d'exécution, source et destination de flux, alors ces flux peuvent réaliser dans la vue applicative une opération fonctionnelle. Ici, la remise d'un appel (*phone call delivery*) implique une notification vers l'utilisateur, manifestée typiquement par une sonnerie de téléphone. Cette notification peut donc être représentée par une opération applicative exposée par le composant *SAC user*, comme pour un autre composant applicatif.

La réalisation applicative de chaque opération fonctionnelle peut alors être décrite par un diagramme de séquence, par exemple pour *SFO issue phone call*.

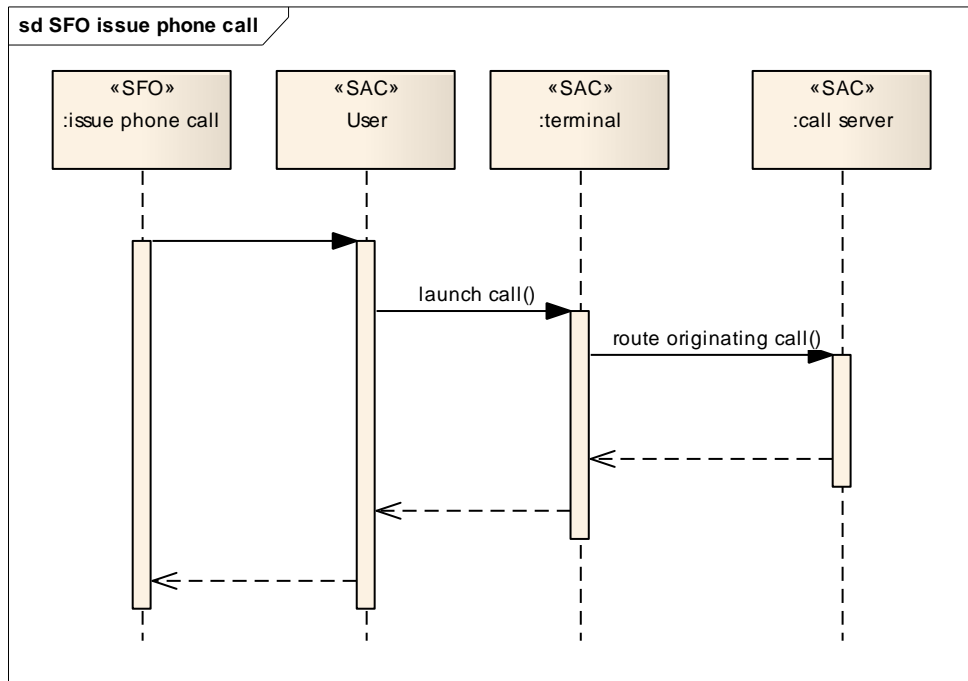


Figure 92 : Exemple d'implémentation applicative de *SFO issue phone call*

Ici, l'opération applicative *SAO launch call* du composant applicatif *SAC terminal* permet d'initier un appel depuis le terminal. Elle est habituellement portée par une IHM du terminal. L'opération applicative *SAO route originating call* du composant applicatif *SAC call server* permet de demander au serveur d'appel d'acheminer un appel.

Chaque opération applicative est supportée par un point de référence de la vue technique du service. Dans le cas de notre service de téléphonie, ces liens sont repris dans le tableau suivant.

Tableau 31 : Lien applicatif-technique pour un service de téléphonie

Composant applicatif	Opération Applicative	Portée par le connecteur issu du point de référence
User	Alert user	GUI
Terminal	Launch terminal	GUI
	Select contact	
	Launch call	
	Deliver call	Gm
call server	Register	Gm
	Route originating call	

	Route terminating call	Mw
user profile server	Assignment request	Cx
	Get profile	
telephony AS	Trigger supplementary services	ISC
NAB	Browse NAB	XDM

L'architecture applicative statique peut alors être représentée de manière synthétique dans la figure ci-dessous.

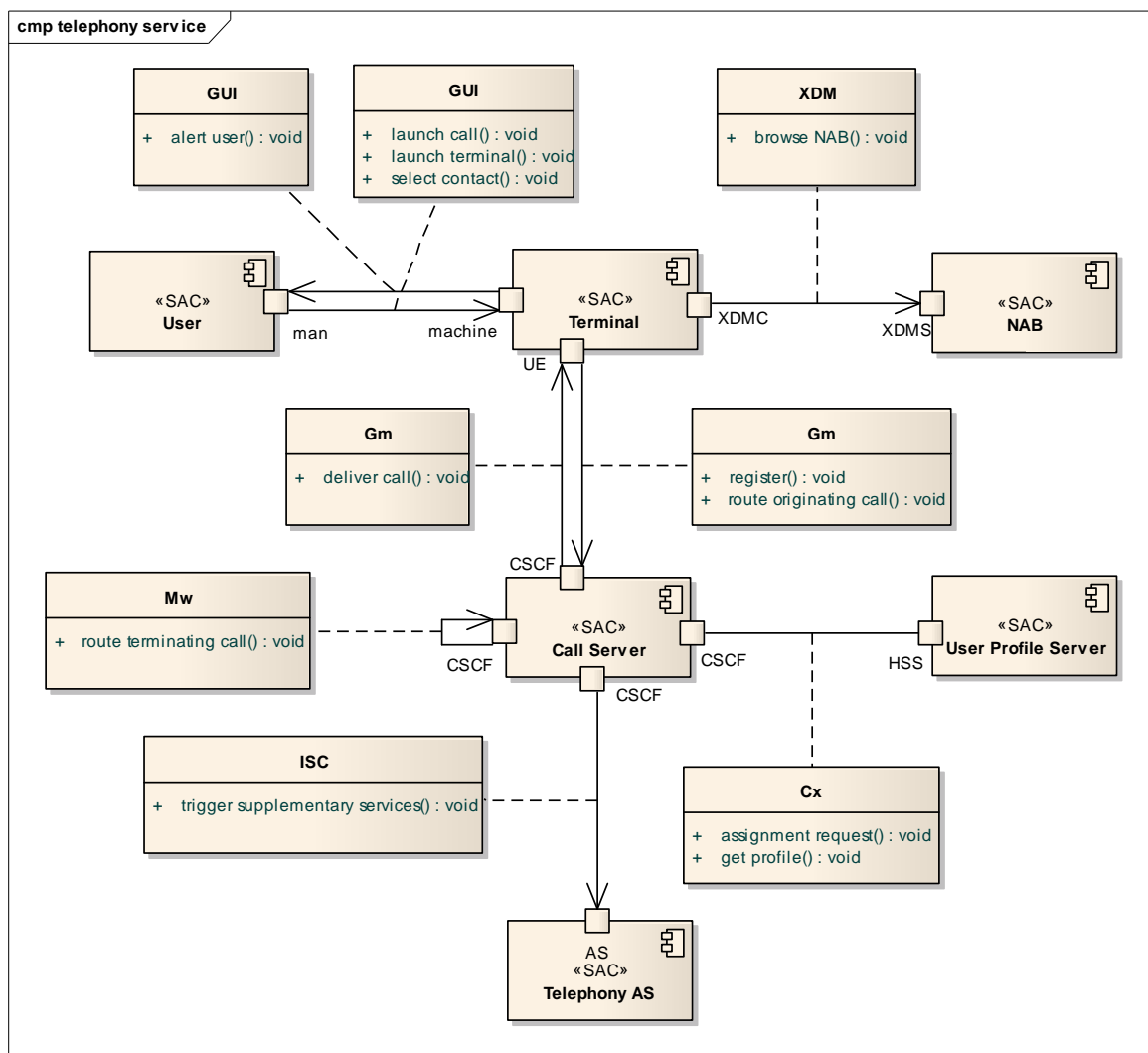


Figure 93 : Vue applicative d'un service de téléphonie

Une fois que la réalisation de chaque opération fonctionnelle a été décrite par une séquence d'opérations applicatives (éventuellement réduite à une seule opération applicative),

on peut en déduire la réalisation des scénarios entiers. Pour cela, on place dans la séquence applicative l'ensemble des composants applicatifs identifiés précédemment. Chaque message entre composants fonctionnels dans la séquence fonctionnelle est alors traduit en un ou plusieurs messages entre composants applicatifs. Cette séquence de composants applicatifs suit cette fois un ordre temporel. Le sens des messages n'est donc pas le même que dans la vue fonctionnelle.

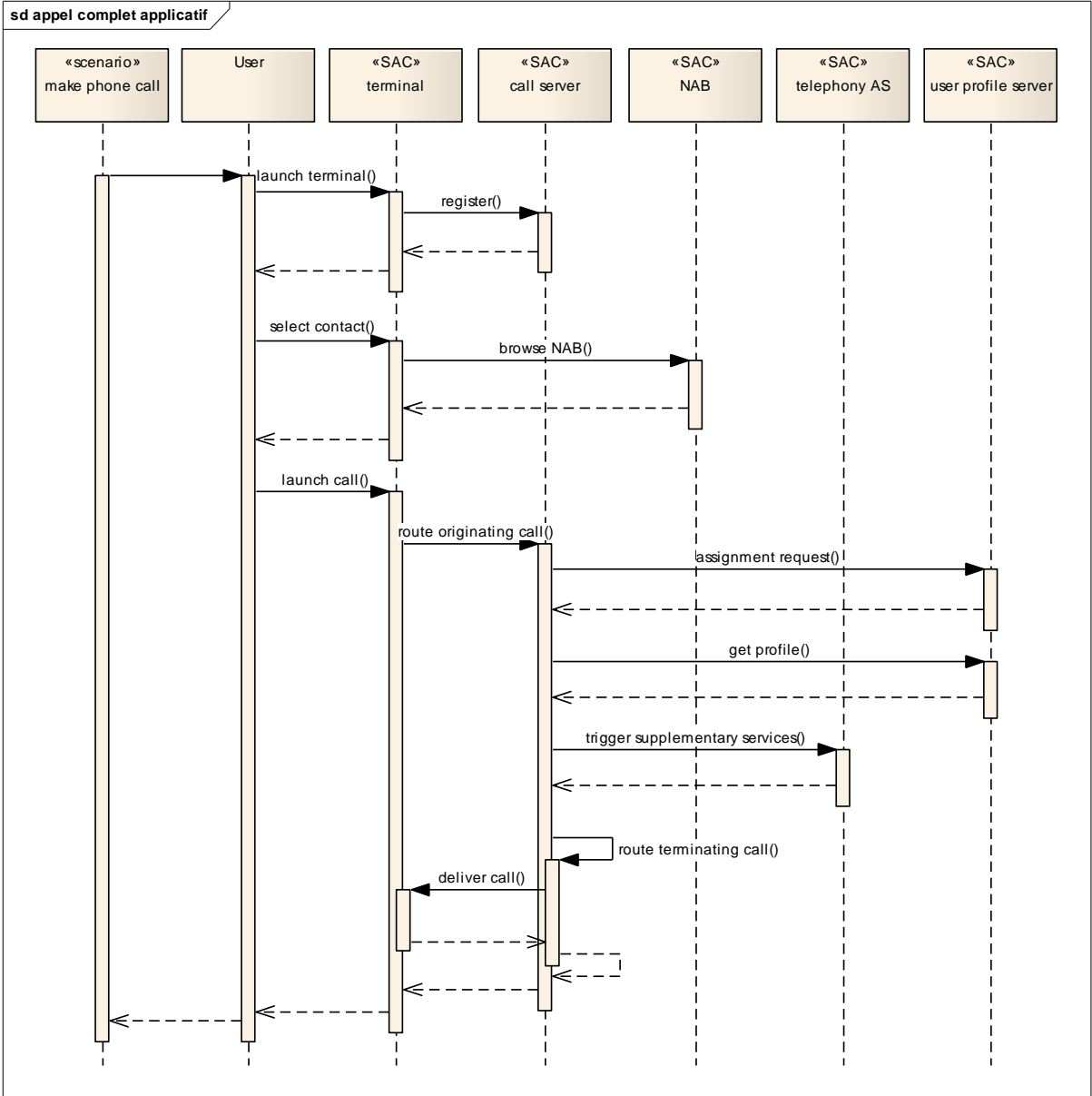


Figure 94 : Séquence applicative du scénario d'appel téléphonique

IV.3.2.2 Exemple d'un service de vidéo

Dans le cas du service de vidéo, les composants applicatifs sont l'utilisateur, le media renderer, le serveur d'appel, l'AS de téléphonie, le serveur de profils et le NAB. Ces composants offrent les opérations applicatives décrites dans le tableau ci-dessous, issues des opérations fonctionnelles indiquées.

Tableau 32 : Lien applicatif-fonctionnel pour un service de vidéo

Composant applicatif	Opération Applicative offerte	Issue de l'opération fonctionnelle
user		
media renderer	Select content in catalog	Browse content
	Play content	Deliver content
content access portal	Authenticate user	Enforce access right
	Get catalog	Browse content
IPTV VOD controller	Order content	Define delivery device
	Check rights	Enforce access right
IPTV VOD delivery	Start streaming	Issue content

La réalisation applicative de chaque opération fonctionnelle peut alors être décrite par un diagramme de séquence, par exemple pour *SFO browse content*.

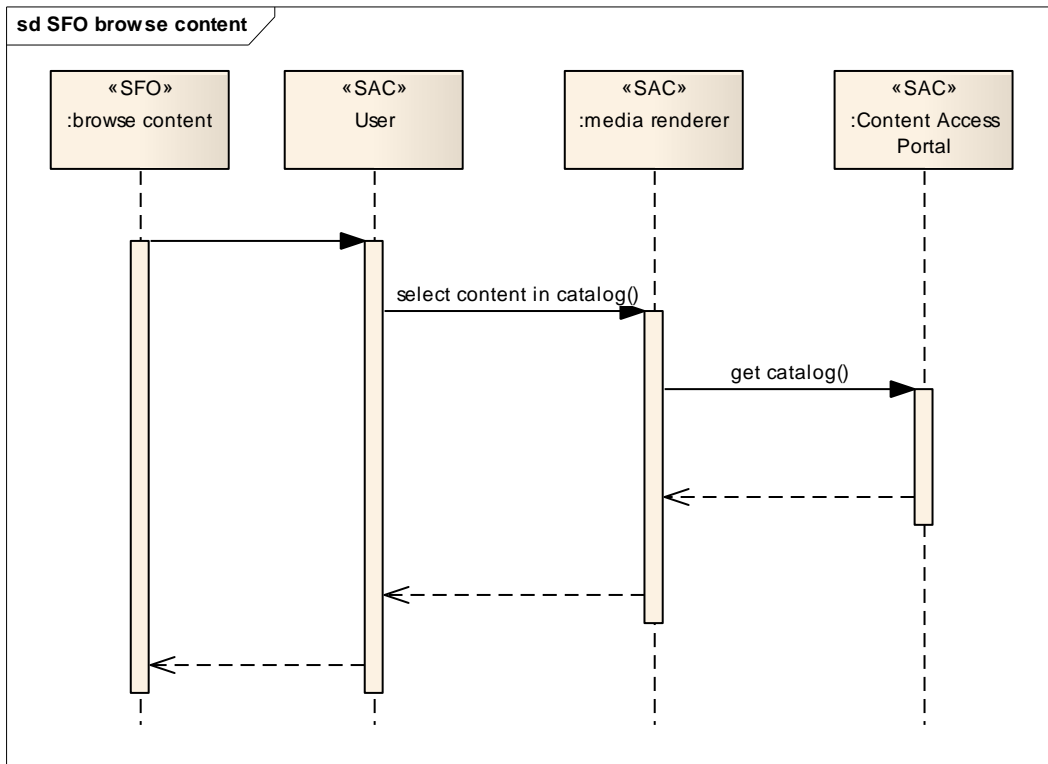


Figure 95 : Exemple d'implémentation applicative de *SFO browse content*

Ici, l'opération applicative *SAO select content in catalog* du composant applicatif *SAC media renderer* permet de consulter le catalogue. Elle est habituellement portée par une IHM du media renderer. L'opération applicative *SAO get catalog* du composant applicatif *SAC content access portal* permet de récupérer le catalogue à consulter.

Dans le cas de notre service de vidéo, les liens applicatifs-technique sont repris dans le tableau suivant.

Tableau 33 : Lien applicatif-technique pour un service de vidéo

Composant applicatif	Opération Applicative	Portée par le connecteur issu du point de référence
user		
media renderer	Choose content	GUI
	Play content	
content access portal	Authenticate user	Service access
	Get content page	
IPTV VOD controller	Order content	MA

	Check rights	MC
IPTV VOD delivery	Start streaming	MD

L'architecture applicative statique peut alors être représentée de manière synthétique dans la figure ci-dessous.

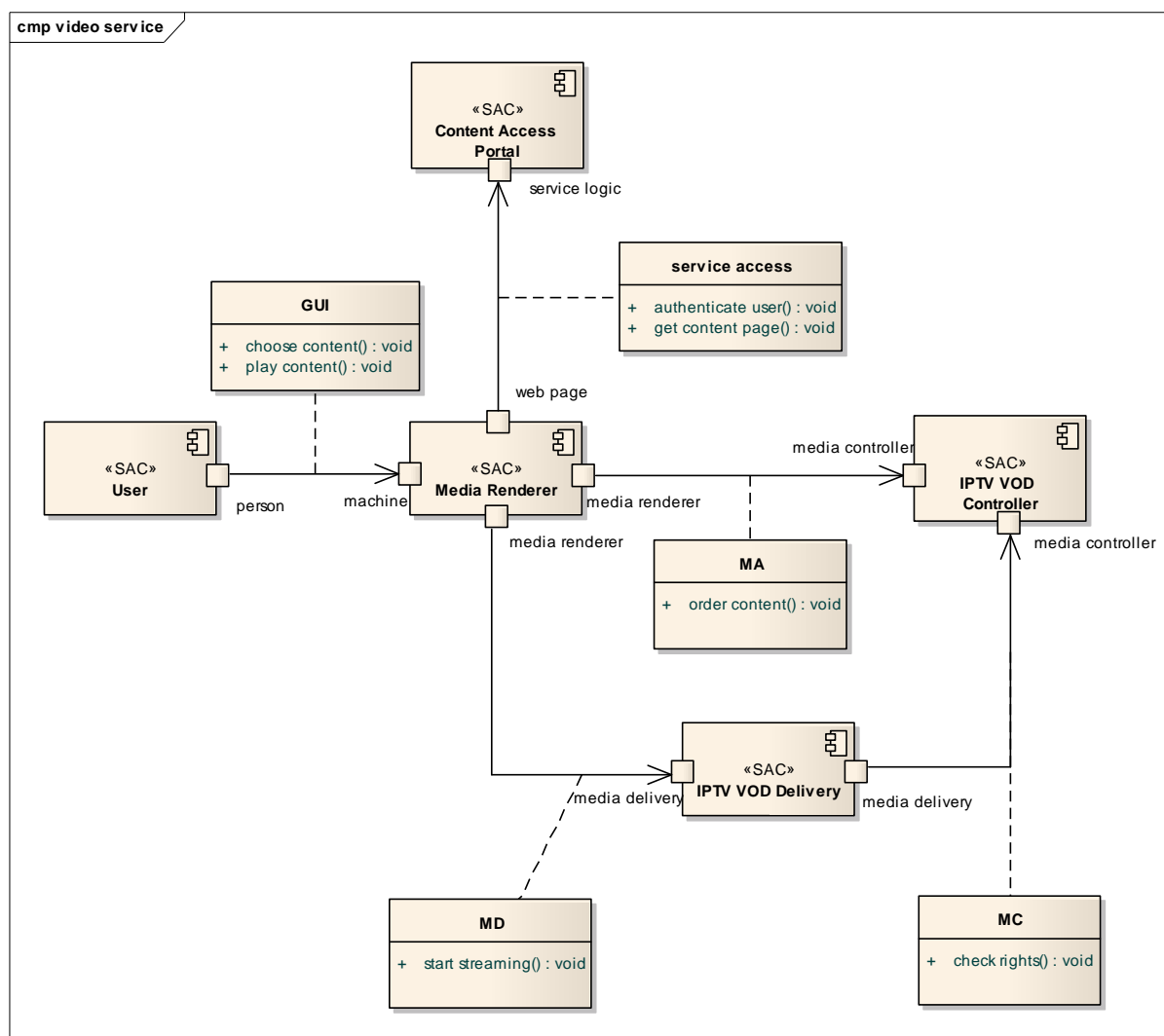


Figure 96 : Vue applicative d'un service de vidéo

Une fois que la réalisation de chaque opération fonctionnelle a été décrite par une séquence d'opérations applicatives (éventuellement réduite à une seule opération applicative), on peut en déduire la réalisation des scénarios entiers.

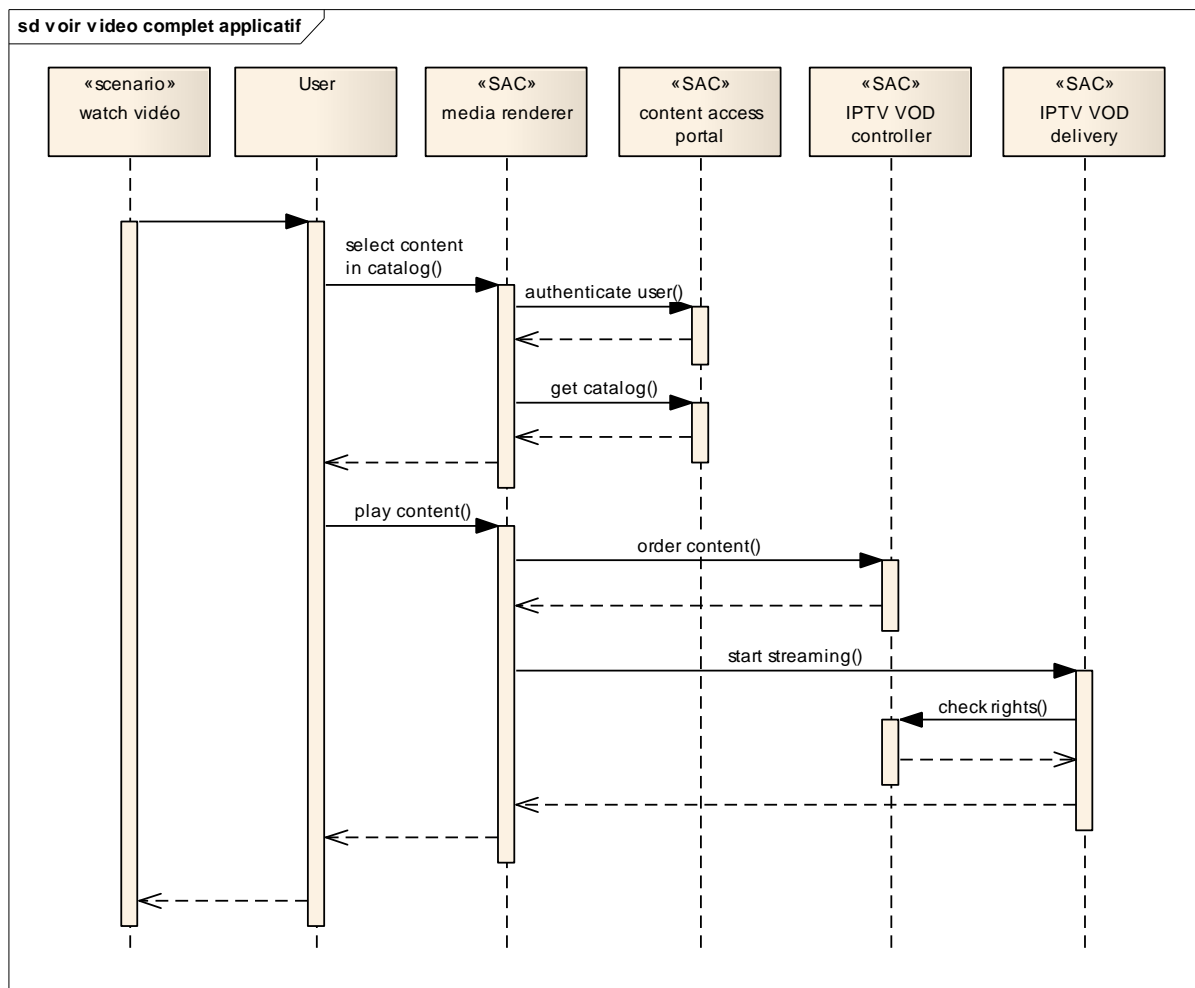


Figure 97 : Séquence applicative du scénario de consommation de vidéo

IV.3.2.3 Exemple d'un service de messagerie

Dans le cas du service de messagerie, les composants applicatifs sont l'utilisateur, la page webmail, la logique de service webmail, la plate-forme de gestion de l'identité et de l'authentification IDP, le serveur SMTP et le back-end de messagerie (message store). Ces composants offrent les opérations applicatives décrites dans le tableau ci-dessous, issues des opérations fonctionnelles indiquées.

Tableau 34 : Lien applicatif-fonctionnel pour un service de messagerie email

Composant applicatif	Opération Applicative offerte	Issue de l'opération fonctionnelle
user		
webmail page	Display mailbox	Browse message

	Create folder	Define message folder
webmail service logic	Manage emails	Browse message
	Manage folders	Define message folder
IDP	Authenticate	Enforce access right
SMTP server	Receive email	Deliver message
Mail back-end	Store email	Publish message
	Get email	Browse message
	Set folder	Define message folder

La réalisation applicative de chaque opération fonctionnelle peut alors être décrite par un diagramme de séquence, par exemple pour *SFO browse message*.

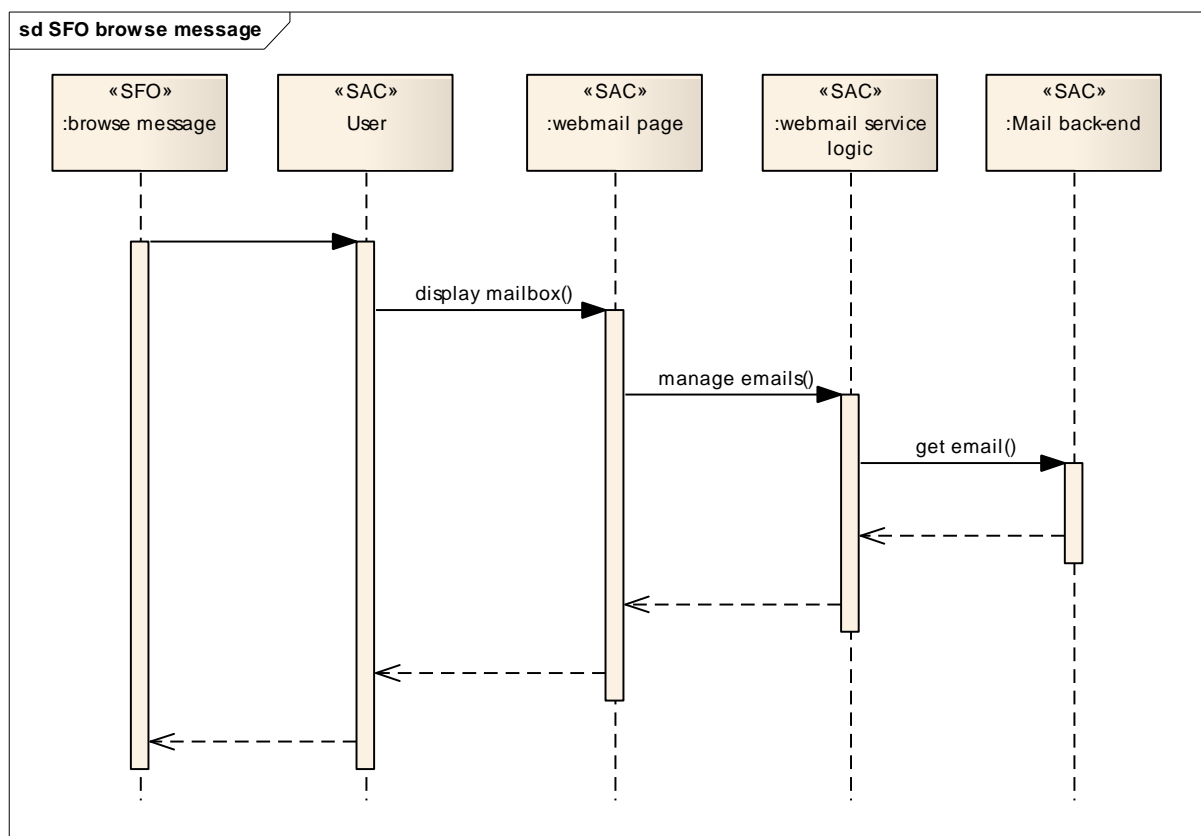


Figure 98 : Exemple d'implémentation applicative de *SFO browse message*

Ici, l'opération applicative *SAO display mailbox* du composant applicatif *SAC webmail page* permet de demander la consultation de la boîte email. Elle est initiée par l'utilisateur depuis la page webmail, qui joue un rôle d'IHM vis-à-vis de l'utilisateur. L'opération applicative *SAO*

manage emails du composant applicatif *SAC webmail service logic* permet à la page de récupérer les données à afficher. L'interface *SAO get email* du composant applicatif *SAC mail back-end* permet d'obtenir les données brutes, typiquement par des messages IMAP. Les liens applicatif-technique sont repris dans le tableau suivant.

Tableau 35 : Lien applicatif-technique pour un service de messagerie

Composant applicatif	Opération Applicative	Portée par le connecteur issu du point de référence
User		
webmail page	Display mailbox	GUI
	Create folder	
webmail service logic	Manage emails	Service access
	Manage folders	
IDP	Authenticate	Service access
SMTP server	Receive email	MTA-MDA (SMTP)
mail back-end	Store email	MDA-MS (SMTP)
	Get email	MUA-MS (IMAP)
	Set folder	

L'architecture applicative statique peut alors être représentée de manière synthétique dans la figure ci-dessous.

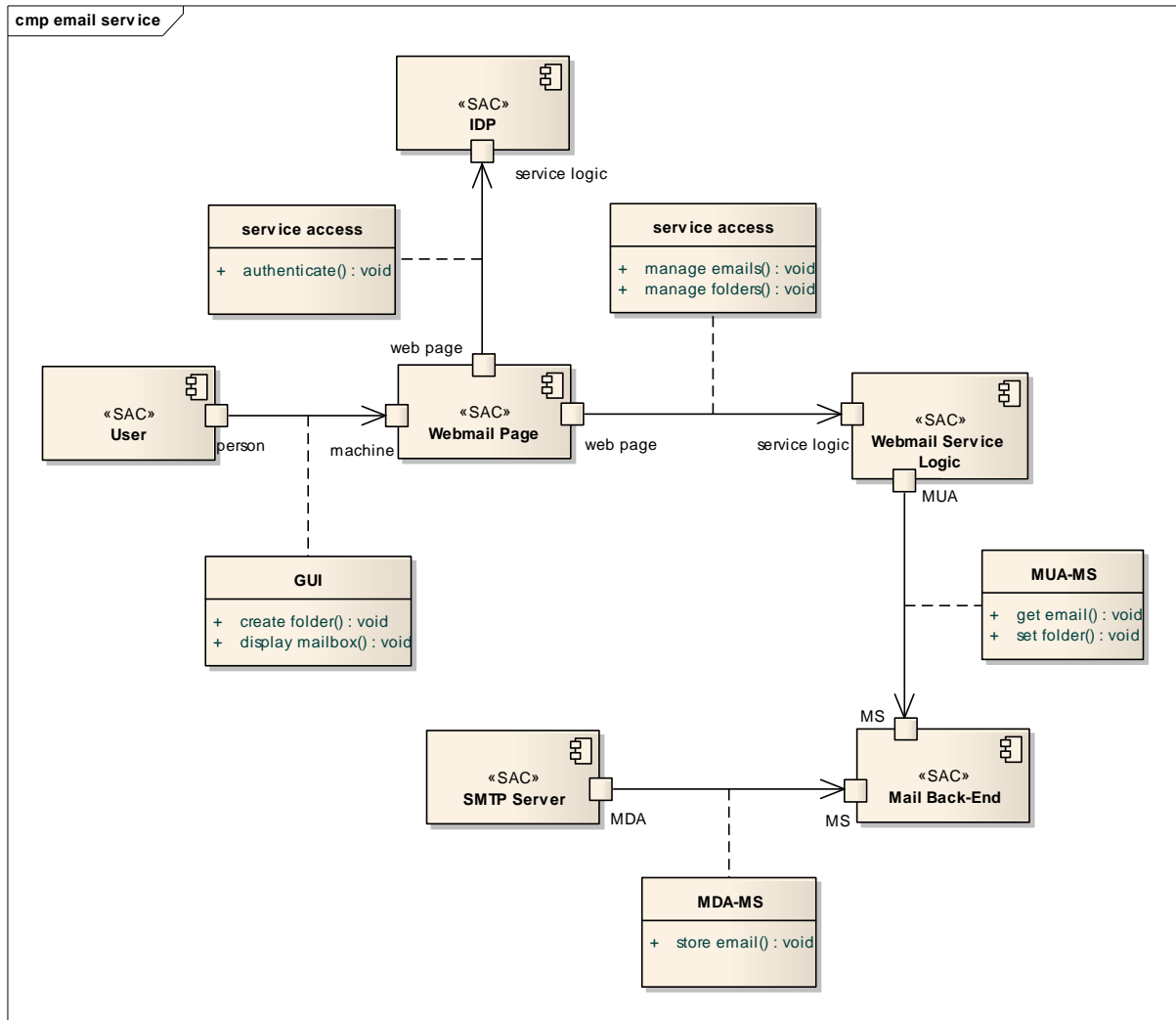


Figure 99 : Vue applicative d'un service de messagerie

Une fois que la réalisation de chaque opération fonctionnelle a été décrite par une séquence d'opérations applicatives (éventuellement réduite à une seule opération applicative), on peut en déduire la réalisation des scénarios entiers.

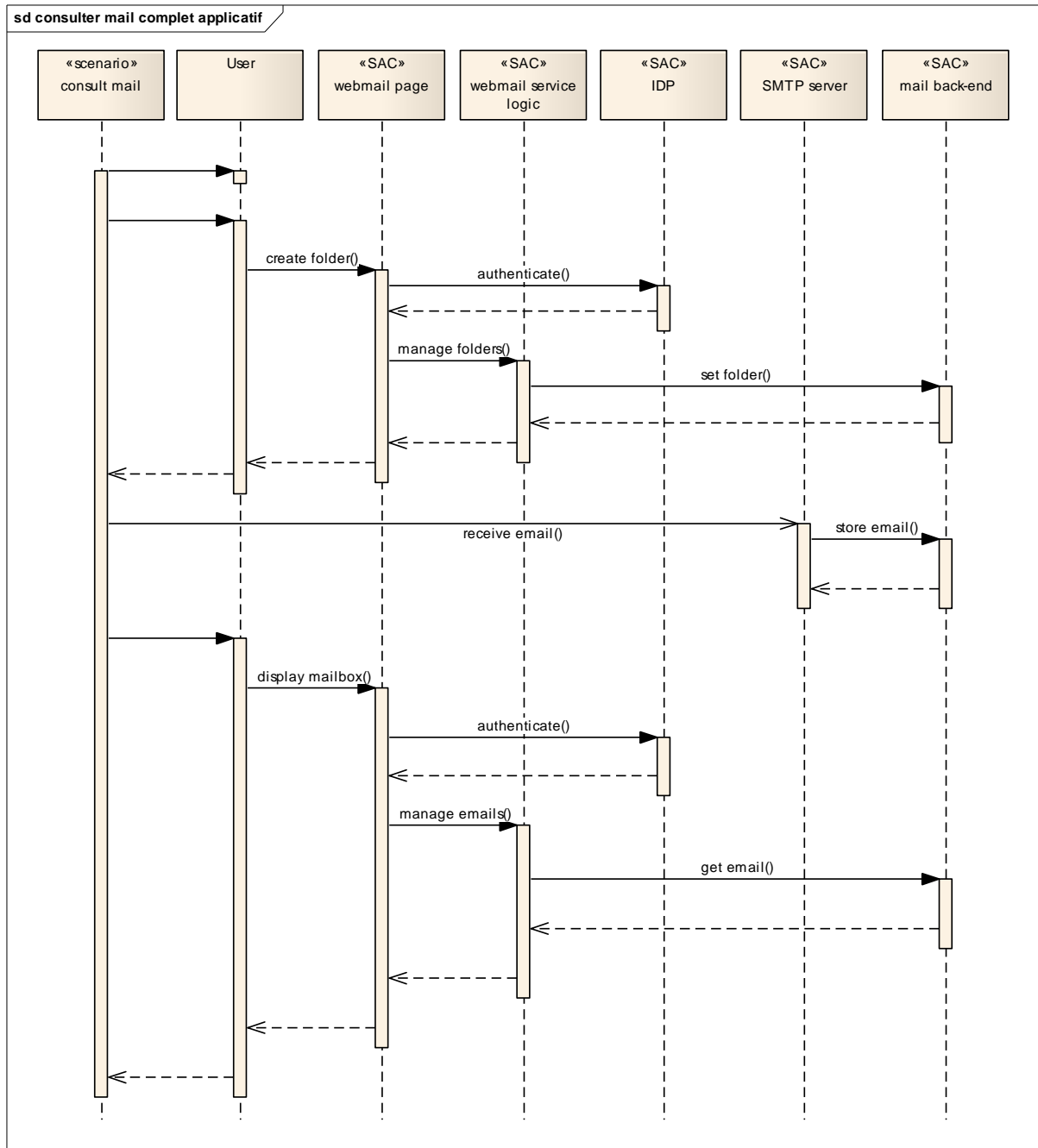


Figure 100 : Séquence applicative du scénario de consultation d'emails

IV.3.3 Cas d'un service composé

Le principe est similaire dans le cas d'un service composé. Dans le cas du service videoMe, les composants applicatifs sont donc le terminal, le serveur d'appel, l'AS de téléphonie, le

serveur de profils et le NAB. Ces composants offrent les opérations applicatives décrites dans le tableau ci-dessous, issues des opérations fonctionnelles indiquées.

Tableau 36 : Lien applicatif-fonctionnel pour le service VideoMe

Composant applicatif	Opération Applicative offerte	Issue de l'opération fonctionnelle
user		
VideoMe system	Launch videoMe terminal	Enforce access right
	Launch videoMe call	Issue phone call
		Browse address
		Browse content
	Deliver videoMe call	Deliver phone call
		Deliver content
Define delivery device		
Telephony system	Register	Enforce access right
	Route originating call	Issue phone call
	Route terminating call	Enforce Registration
		Deliver phone call
	Assignment request	Enforce access right
	Get profile	Enforce registration
		Enforce routing rule
Trigger supplementary services	Enforce routing rule	
Browse NAB	Browse address	
Video system	Authenticate user	Enforce access right
	Get catalog	Browse content
	Order content	Define delivery device
	Check rights	Enforce access right
	Start streaming	Issue content

Les liens applicatif-technique sont repris dans le tableau suivant.

Tableau 37 : Lien applicatif-technique pour le service VideoMe

Composant applicatif	Opération Applicative	Portée par le connecteur issu du point de référence
User		
VideoMe system	Select callee	GUI
	Launch call	
	Select content in catalog	
	Deliver call	Gm
	Play content	videoMe-1
Telephony system	Register	Gm
	Route originating call	
	Route terminating call	Mw
	Assignment request	Cx
	Get profile	
	Trigger supplementary services	ISC
	Browse NAB	XDM
Video system	Authenticate user	Service access
	Get catalog	
	Order content	MA
	Check rights	MC
	Start streaming	MD

L'architecture applicative statique peut alors être représentée de manière synthétique dans la figure ci-dessous, en séparant clairement les responsabilités et les interfaces de chaque composant. Cette figure fait clairement apparaître les différents systèmes composants le service composé et les liens entre eux, liens à la fois fonctionnels et techniques.

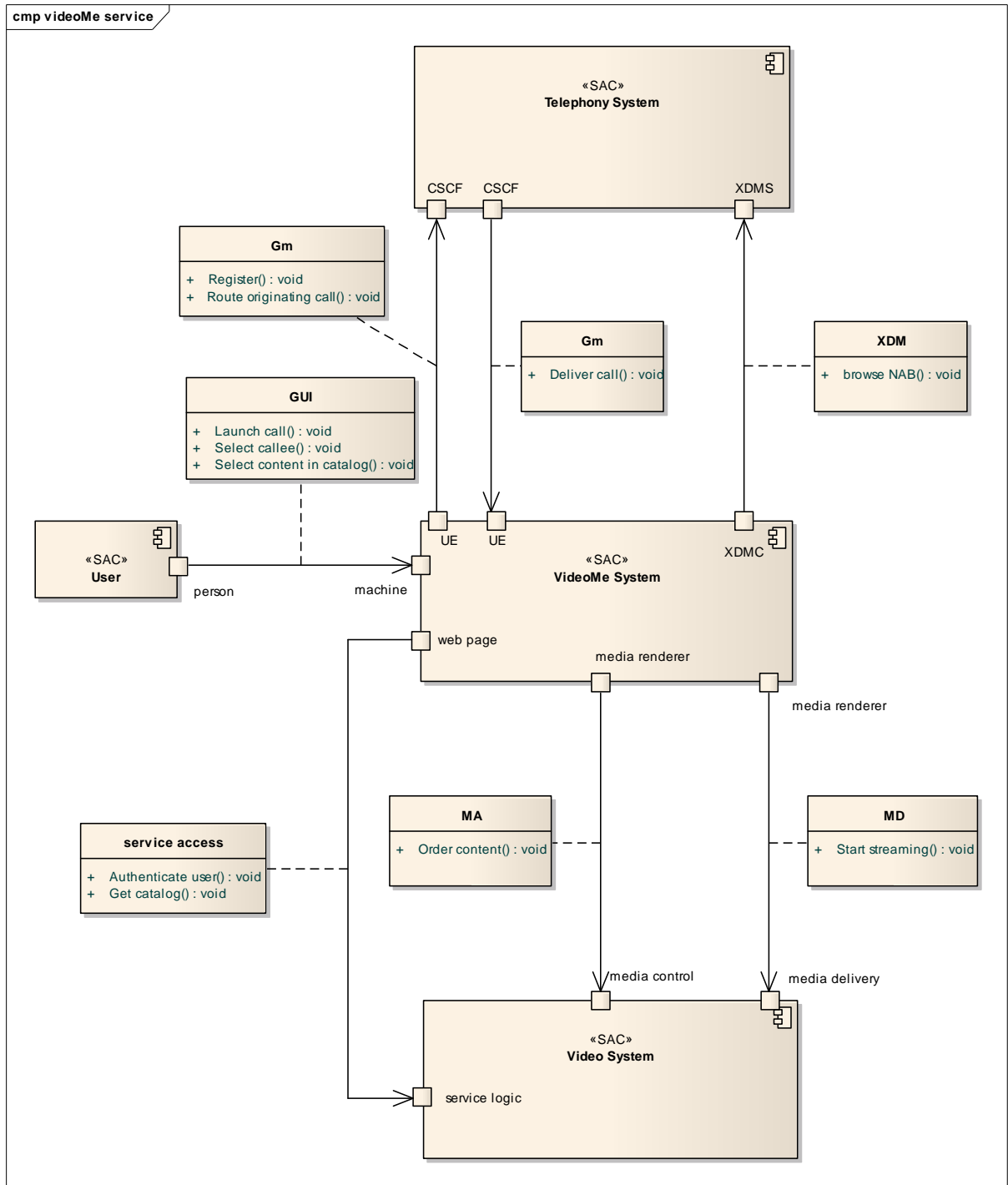


Figure 101 : Vue applicative du service VideoMe

Synthèse

Durant ce chapitre, nous avons proposé une approche pour décrire les services de communication, en nous basant sur les vues fonctionnelle et technique de référence définies au chapitre précédent.

La vue fonctionnelle d'un service décrit la réalisation d'un scénario du service par ses parties-prenantes, à travers un enchaînement d'opérations fonctionnelles. Cet enchaînement est *seamless*, il ne considère dans la réalisation du service ni de frontières techniques (comme le terminal, la plate-forme ou le réseau utilisé), ni de frontières de responsabilité (comme par exemple les services de base dans le cas d'un service composé). Cette vue fonctionnelle correspond ainsi à une vision de bout en bout du service.

La vue technique d'un service décrit l'environnement technique du service, à travers un ensemble de nœuds d'exécution reliés par des points de référence. Un nœud d'exécution n'est pas à comprendre uniquement comme un matériel technique, mais comme un périmètre de responsabilité sur des composants matériels et logiciels. Une découpe de responsabilité comme le motif *front-office / back-office* apparaîtra ainsi dans la vue technique du service. La vue technique n'est donc pas *seamless*, mais permet au contraire d'identifier formellement les hétérogénéités, et notamment les différents systèmes (services de base) formant un service composé.

La vue applicative d'un service décrit l'implémentation du déroulement du service (vue fonctionnelle) dans un environnement technique donné (vue technique). Il s'agit ici d'identifier les opérations applicatives, issues d'opérations fonctionnelles, offertes par chacun des nœuds d'exécution via un point de référence donné. Cette vue permet ainsi de visualiser à la fois la découpe du service en composants responsabilisés sur leur périmètre et à la fois la façon dont ces composants supportent les actions des parties-prenantes du service.

Dans le chapitre suivant, nous verrons diverses applications de ces résultats pour faciliter la convergence des services.

Chapitre V Applications et perspectives organisationnelles et logicielles

Nous allons maintenant étudier quelques applications réalisées à partir de la démarche de description des architectures de service exposée au cours des chapitres précédents.

V.1 Une aide pour la construction des offres de service

Tout d'abord, la vue fonctionnelle de référence est une aide pour la conception, l'évaluation et la gestion des services télécoms et des enablers.

V.1.1 La conception de services

L'activité de conception de services télécoms est usuellement pilotée par un département marketing. Suite à une expression de besoin initiale, de nombreux échanges ont lieu avec les équipes techniques en charge du développement des services afin de traduire ce besoin marketing en une solution technique. Ce mécanisme reste souvent relativement artisanal et peu outillé [Kellogg, 2006], le marketing rechignant à utiliser des méthodes strictes et des artefacts formels. La traduction réalisée sur un projet donné ne sera pas nécessairement réutilisée pour des besoins similaires, à moins que les équipes projet ne soient constituées des mêmes personnes. La vue fonctionnelle de référence présentée au chapitre III constitue une aide pour ce travail de traduction. Les fonctions élémentaires SEF sont un support pour organiser le besoin marketing. La plupart des besoins, fonctionnels mais aussi non fonctionnels, peuvent être rattachés à une fonction élémentaire. Par exemple, pour un service de messagerie, la taille de stockage de la boîte aux lettres peut être rattachée à la fonction *SEF message folder defining capability*. La proximité des SEF avec la vue métier les rend intelligibles pour une équipe marketing. Et leur caractère formel et structuré les rend utiles comme base de conception d'une architecture applicative, comme nous l'avons vu avec la vue applicative d'un service au chapitre IV.

Ces SEF permettent de formaliser un besoin marketing dans un vocabulaire standardisé. Ceci facilitera ensuite l'étude de ce besoin et le passage à une solution technique.

A titre illustratif, les figures suivantes présentent quelques captures d'écran d'un outil de spécification du contour d'une offre, développé à titre expérimental par un département marketing de France Telecom. L'exemple illustré est celui d'un service de messagerie email. La première figure présente le contour fonctionnel d'une offre d'email avec des fonctions élémentaires SEF (sous un nom simplifié). La deuxième figure présente le détail d'une de ces fonctions élémentaires, enrichie d'options de l'offre qui en complètent la description. Par exemple, une fonction élémentaire d'authentification peut être réalisée par mot de passe, être implicite suite à une première authentification ou bien effectuée via le réseau d'accès. De même, la visualisation des messages peut inclure l'affichage des images, la recherche dans les messages reçus par sujet ou par recherche d'une chaîne de caractères. Le stockage des messages peut être caractérisé par un nombre maximal de dossiers, peut inclure la possibilité d'appliquer des règles de filtrage des messages, celle de déplacer des messages par *drag&drop*, ou encore la vérification des messages par un antivirus ou un antispam.

De plus, les liens de dépendance entre SEF peuvent aider les équipes marketing à élargir leur questionnement aux besoins fonctionnels connexes. Une fonction élémentaire peut se référer à un autre service, par exemple, dans la première figure, la fonction de sélection d'un contact nécessite le service de carnet d'adresses.

A la suite de cette définition du contour fonctionnel d'une offre, une équipe marketing peut décomposer le service en une séquence de tâches à effectuer par l'utilisateur. Cette démarche est classique dans le domaine de l'ergonomie des relations homme-machine, connue sous le nom de méthode HTA pour *Hierarchical Task Analysis* [Annett, 2004]. Chaque tâche peut alors être traduite en une fonction élémentaire SEF, et la séquence de tâches peut être traduite en séquence d'opérations fonctionnelles SFO. Ceci fournit à l'équipe en charge du développement ou de l'intégration un cadre formalisé. Comme présenté au chapitre précédent, chaque SFO peut être reliée à un nœud d'exécution au sein d'un composant applicatif. Les séquences de SFO fournissent alors les scénarios de test de bout en bout du service.

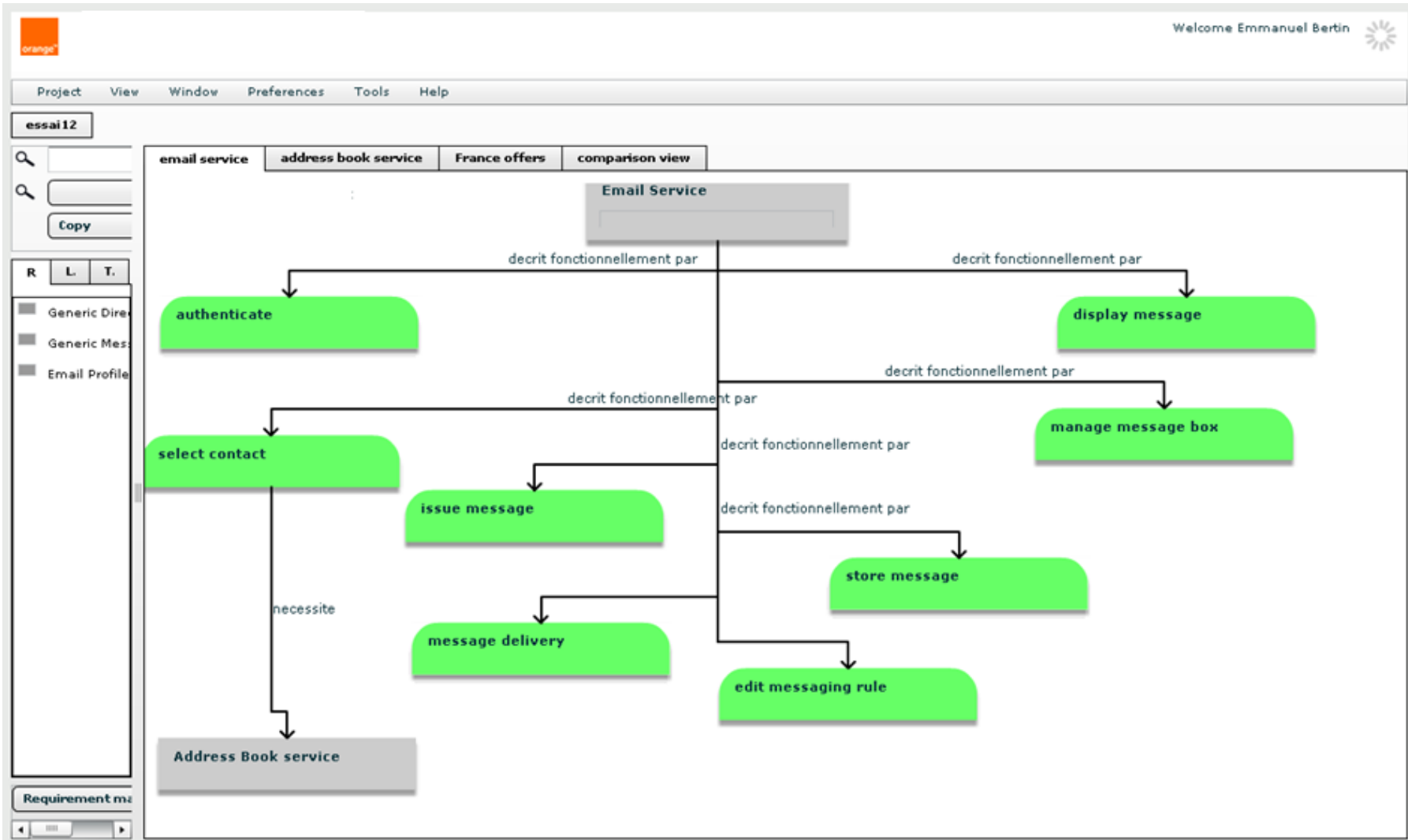


Figure 102 : Contour fonctionnel d'une offre d'email dans l'outil

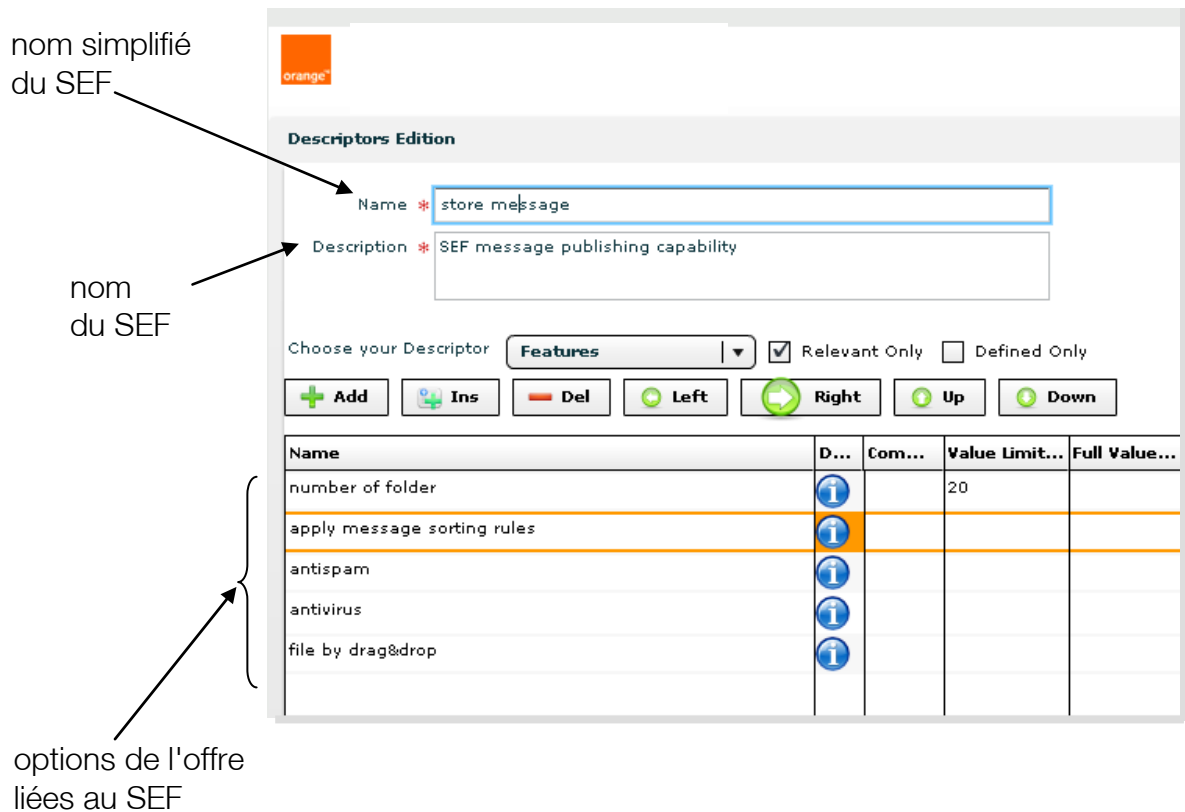


Figure 103 : Exemple de descripteur

V.1.2 L'évaluation et la comparaison de services

Une fois formalisé un service selon les vues fonctionnelle, technique et applicative, il devient beaucoup plus aisé de partager ces architectures au sein d'une organisation comme un opérateur télécom. Des sites web internes peuvent par exemple être mis en place pour donner accès à l'architecture des différents projets de service pour les personnes intéressées. Les architectures techniques peuvent, par exemple, être partagées pour étudier les bonnes pratiques et définir de nouveaux motifs techniques. De même pour les architectures applicatives, une équipe de développement peut consulter les solutions techniques ayant été apportées à un besoin fonctionnel, et réutiliser les mêmes solutions pour un besoin similaire. Le besoin fonctionnel recherché peut être caractérisé de façon précise, par exemple en ne recherchant que les solutions techniques implémentant une séquence donnée d'opérations fonctionnelles. Ce formalisme peut également servir de support à des comités de validation, vérifiant notamment que les solutions préconisées pour répondre à un type de besoin ont bien été utilisées. Ainsi, un comité peut prescrire que l'implémentation d'une opération fonctionnelle comme *SFO deliver phone call* doit impérativement utiliser le motif technique

IMS (et donc le protocole SIP) et faire appel à une solution technique de type IMS fournie par un équipementier donné. Les procédures d'évaluation et de validation des architectures définies dans l'urbanisation des systèmes d'information peuvent ainsi être appliquées aux services télécoms, grâce aux invariants définis dans les vues fonctionnelle et technique de référence.

De plus, cette description suivant les vues classiques de l'urbanisme rend possible l'application de vérifications automatiques de cohérence, pouvant servir de support à ces comités de validation. On peut ainsi envisager de mesurer l'alignement entre la vue fonctionnelle d'un service et la vue fonctionnelle de référence, ou avec la vue applicative. De telles mesures d'alignement ont notamment été développées par Jacques Simonin dans [Simonin, 2007].

Enfin, des métriques permettant de qualifier numériquement un aspect particulier des services télécoms peuvent être définies. Nous allons en proposer maintenant quelques unes, à considérer plus comme des pistes de travail que comme des résultats définitifs. Il convient tout d'abord de se demander quels aspects nous souhaitons qualifier. Nous retiendrons ici quatre aspects :

- la complexité d'usage d'un service ;
- sa complexité de développement ;
- la nouveauté d'un service ;
- sa complexité d'intégration.

La complexité d'usage peut être caractérisée par le nombre de composants fonctionnels SFC présents dans la séquence. Avec notre convention d'un SFC par entité de service, le nombre de SFC signifie donc le nombre de concepts du service que les parties-prenantes du service vont devoir manipuler pour le réaliser. La complexité d'un service peut également être mesurée à partir de la complexité des séquences fonctionnelles qui le décrivent. Elle peut être caractérisée par le nombre d'invocations d'opérations fonctionnelles de la séquence. Ce nombre signifie le nombre de manipulations que les parties-prenantes du service vont devoir effectuer sur les concepts du service pour réaliser la séquence. La métrique de complexité d'usage doit respecter les propriétés suivantes :

- A nombre de SFC constant dans la séquence, le service est plus complexe lorsque le nombre d'opérations SFO augmente ;
- A nombre d'opération SFO constant dans la séquence, le service est plus complexe lorsque le nombre de composants SFC augmente ;
- En première approximation, on peut considérer que l'ajout d'un composant ou l'ajout d'une opération augmente de la même façon la complexité d'usage.

Une métrique de complexité d'usage satisfaisante peut alors être la moyenne entre le nombre de SFC et le nombre de SFO. Dans nos exemples du chapitre IV, les valeurs correspondantes sont indiquées dans le tableau suivant. Le service VideoMe apparaît logiquement comme le plus complexe à utiliser.

Tableau 38 : Métrique de complexité d'usage

Service	Nombre de SFC	Nombre de SFO	Moyenne
Téléphonie	5	7	6
Vidéo	3	5	4
Messagerie	3	7	5
VideoMe	7	12	9.5

La complexité de développement d'un service peut être abordée à travers la complexité de traitement des composants qui vont le réaliser. Les composants applicatifs les plus complexes à réaliser sont souvent ceux qui demandent le maintien d'une session au cours de laquelle diverses requêtes auront lieu dont les résultats devront être corrélés. Cet aspect peut être abordé à travers le nombre d'invocations sortantes d'un composant, c'est-à-dire le nombre d'opérations que ce composant va invoquer, y compris vers lui-même. Pour certains composants, cette valeur sera nulle. Ce sont des composants en bout de chaîne, qui effectuent un traitement ne nécessitant pas d'autre opération. Ces composants seront usuellement plus simples à développer. La métrique de complexité de développement doit augmenter lorsque le nombre d'invocation sortante d'un des composants de la séquence augmente. Une métrique de complexité de développement satisfaisante peut alors être la somme des invocations sortantes de chaque composant de la séquence. Dans nos exemples du chapitre IV, les valeurs correspondantes sont indiquées dans le tableau suivant. Le service VideoMe est le plus

complexe à développer, ce qui est effectivement le cas puisqu'il nécessite le développement des services le composant.

Tableau 39 : Métrique de complexité de développement

Service	Nombre total d'invocations sortantes
Téléphonie	$5+0+0+1+0=6$
Vidéo	$5+0+0=5$
Messagerie	$5+0+1=6$
VideoMe	$5+6+0+0+0+0+0=11$

L'innovation d'un service peut être mesurée par rapport à la vue fonctionnelle de référence. Si un service reprend exactement les composants et les opérations de la vue fonctionnelle de référence, il ne présente pas d'innovation. L'innovation peut venir soit d'un nouveau composant fonctionnel, soit de l'ajout d'une invocation sortante à un composant fonctionnel. Dans nos exemples du chapitre IV, les valeurs correspondantes sont indiquées dans le tableau suivant. Logiquement, les services de téléphonie, de messagerie ou de visionnage de vidéo n'apparaissent pas comme innovants. Par contre, le service VideoMe est un service innovant.

Tableau 40 : Métrique d'innovation

Service	Nombre de nouveau SFC	Nombre de nouvelles invocations sortantes	Moyenne
Téléphonie	0	0	0
Vidéo	0	0	0
Messagerie	0	0	0
VideoMe	0	2	1

En ce qui concerne la complexité d'intégration, nous pouvons la mesurer à partir du nombre de composants impliqués dans la réalisation d'une même opération fonctionnelle. En effet, les composants applicatifs implémentant une même opération fonctionnelle sont par nature fortement couplés. Leur intégration sera donc un enjeu important, qui devra être validé par des tests d'intégration poussés. Les valeurs pour nos exemples du chapitre IV sont indiquées

dans le tableau suivant. Dans la métrique, seuls doivent être considérés les cas où le nombre de composants applicatifs SAC participant à la réalisation d'une même opération fonctionnelle SFO est supérieur ou égal à 2. Nous pouvons donc retrancher 1 au nombre de SAC par SFO.

Tableau 41 : Métrique de complexité d'intégration

Service	Somme du nombre de SAC par SFO	Somme du nombre de SAC par SFO moins 1
Téléphonie	$2+3+3+2+2+2=14$	$1+2+2+1+1+1=8$
Vidéo	$1+2+1+2+1=7$	$0+1+0+1+0=2$
Messagerie	$1+1+1+3+3=9$	$0+0+0+2+2=4$
VideoMe	$1+2+2+1+1+2+2+2+3+1=17$	$0+1+1+0+0+1+1+1+2+0=7$

La téléphonie apparaît de façon intéressante comme le service le plus complexe à intégrer. Ceci rejoint l'expérience de terrain dans l'architecture des services télécoms. Les composants d'un service de téléphonie ne sont pas excessivement complexes à développer, mais leur mise en œuvre nécessite un important travail d'intégration et de tests de bout en bout.

V.1.3 La gestion de services

Comme nous l'avions annoncé dans le chapitre I, en section 3.3, nous n'avons pas traité ici des processus de gestion des services comme la prise de commande ou la facturation, mais de l'usage de ces services. Néanmoins, les processus de gestion peuvent devoir prendre en compte les dépendances entre services, notamment dans le cas de la gestion de services composés. Par exemple, dans le cas d'un service composé comme VideoMe, des ressources de téléphonie ainsi que des ressources de choix et de visualisation de vidéo devront être provisionnées. Notre travail sur l'usage des services peut aider à formaliser ces dépendances.

Dans les systèmes de gestion de service tels que définis au TMF, une partie importante, localisée au niveau Service Management and Operations de la cartographie de processus eTOM, consiste à passer de la définition de l'offre telle que souscrite par le client à l'identification des ressources à provisionner pour réaliser cette offre. Cette activité est appelée *Service Configuration and Activation*. Elle nécessite bien sûr une description complète des ressources à configurer et activer pour chaque offre. Dans l'état de l'art [2006-7], deux approches coexistent. D'une part, une configuration centralisée, à travers des moteurs de

workflow tels qu'IBM WebSphere Business Integration Collaborations for Telecommunications. Ce type de produit permet de définir un enchaînement de ressources à provisionner lors de la souscription d'une offre, à l'aide d'objets métier abstrayant les offres comme les ressources. Un objet métier abstrayant une offre pointera vers des objets métier abstrayant les ressources à configurer pour mettre en place cette offre. D'autre part, des solutions décentralisées, comme proposé dans [2008-3] ou dans [Schreiner, 2008], où les plates-formes de service peuvent prendre elles-mêmes en charge une grande partie de ces opérations en établissant un module "*operation manager*" par offre, capable de provisionner les ressources nécessaires à cette offre. Dans les deux cas, la liste des ressources à provisionner pour chaque offre est un pré-requis. Ces listes sont souvent établies directement dans le formalisme de l'application utilisée, par exemple IBM WebSphere Business Integration Collaborations. Elles sont difficiles à réutiliser d'une offre à l'autre. Leur définition est donc une charge importante lors du développement d'un service télécom.

La description d'une offre de service sous forme de séquence d'opérations fonctionnelles permet de faciliter leur établissement. En effet, chaque opération fonctionnelle rendue par le service peut être décrite dans la vue applicative par un enchaînement de composants applicatifs SAC (voir par exemple les figures 92, 95 ou 98) ; ces composants applicatifs étant précisément les ressources nécessaires pour la réalisation de cette opération fonctionnelle. Ces enchaînements nous donnent donc pour chaque opération fonctionnelle du service les ressources (plates-formes de service, équipements réseau) à provisionner pour que cette opération puisse être réalisée dans le cadre du service. Ensuite, ces enchaînements peuvent ainsi être repris et modifiés durant le cycle de vie de l'offre. En cas de modification du contour fonctionnel de l'offre par ajout ou suppression d'une opération fonctionnelle dans la séquence d'opérations fonctionnelles la décrivant, les impacts applicatifs peuvent être immédiatement cernés, en termes de composant à ne plus provisionner et de nouveau composant à provisionner. Ces enchaînements peuvent également être réutilisés pour définir une offre de service similaire. Ces mécanismes ont notamment été étudiés dans [2008-3].

De plus, ces séquences applicatives liant opération fonctionnelle et ressources peuvent également être utilisées dans le processus de supervision (*assurance*). En effet, en cas de défaillance d'une ressource (détectée par les logiciels de supervision des équipements réseau ou des plates-formes de service), elles permettent de déterminer quelles sont les opérations fonctionnelles impactées par cette défaillance, et par suite quelles sont les actions que les

parties-prenantes des services télécoms ne vont plus pouvoir réaliser. Cela permet alors de remonter des informations pertinentes pour juger de la gravité de la défaillance.

En conclusion, les travaux présentés aux chapitres III et IV constituent une aide pour la construction des offres de service télécoms. Ils permettent d'établir une démarche cohérente de construction d'une offre, depuis sa conception par le marketing jusqu'à sa gestion, en passant par sa validation par des décideurs ad-hoc et par son développement ou son intégration. Une architecture de service décrite selon la démarche présentée dans le chapitre IV constitue bien ainsi un lieu de délibération entre les différentes parties-prenantes dans la construction d'un offre de services, les marketeurs, les décideurs, les développeurs, les intégrateurs, les exploitants. Nous allons maintenant voir comment la vue fonctionnelle de référence nous fournit en outre une aide pour identifier le périmètre fonctionnel des enablers ou des services logiciels.

V.2 Une aide pour la rationalisation des services

Durant le premier chapitre, nous avons distingué différents types d'éléments applicatifs dont la vocation est d'être réutilisés lors de la construction de services télécoms : les enablers, les services logiciels SOA. Nous avons pointé la difficulté de déterminer clairement le périmètre fonctionnel souhaitable de ces éléments et comment ce travail était mené soit dans un processus de standardisation pour les enablers, soit dans un processus dit de gouvernance SOA au sein des entreprises, ces processus étant peu outillés. Nous avons également vu comment le paradigme web sémantique appliqué aux services logiciels butait sur le problème classique du manque de vocabulaire commun aux parties-prenantes des services. Nous allons maintenant étudier comment la vue fonctionnelle de référence permet de clarifier ces enjeux.

V.2.1 Le cas des enablers

Comme nous l'avons vu aux chapitres I et II, la question de l'identification des enablers et de leur périmètre fonctionnel n'a pas de réponse claire, la meilleure solution résidant actuellement dans un processus de standardisation afin qu'une pluralité de points de vue puisse se confronter sans primauté hiérarchique.

A la suite des chapitres précédents, nous pouvons proposer une autre voie. En effet, un enabler peut être décrit, comme un service, à travers une vue fonctionnelle, une vue technique et une vue applicative. Elaborer un enabler consiste alors à identifier son périmètre

fonctionnel et son périmètre technique. Les enablers spécifiés à l'OMA comme les capacités de service de l'ETSI privilégient soit la dimension fonctionnelle (l'enabler va rendre un service de base à l'utilisateur, lequel pourra être réutilisé par d'autres services), soit la dimension technique (l'enabler va encapsuler une solution technique, laquelle pourra être ainsi réutilisée pour implémenter des services). Bien sûr, les enablers privilégiant la dimension fonctionnelle spécifient également une vue technique indiquant dans quel environnement technique l'enabler pourra s'intégrer. Par contre, les enablers privilégiant la dimension technique ne définissent qu'à minima une vue fonctionnelle.

Un enabler privilégiant la vue fonctionnelle doit être en correspondance avec la stratégie fonctionnelle de l'entreprise et évoluer selon cette stratégie. Les éléments de la vue fonctionnelle de référence constituent justement des blocs stables définis à partir de la stratégie de l'entreprise et devant évoluer avec celle-ci. Dans notre perspective, un tel enabler devra donc naturellement s'inscrire comme réalisant un ensemble de fonctions élémentaires de la vue fonctionnelle de référence. A titre illustratif, nous pouvons étudier comment les quartiers que nous avons proposés comme regroupement logique de fonctions élémentaires peuvent correspondre à des enablers existants.

Tableau 42 : Quartiers fonctionnels et capacités de service TISPAN

Quartier fonctionnel de référence	OMA Enabler [OMA, web]	TISPAN Service capability [ETSI, 2006]
address book	OMA Converged Address Book	
communication setup	OMA IP Multimedia Subsystem	Communication Control
content publication and access	OMA Mobile Broadcast Services	
identity and rights	OMA Identity Management Framework	Logon/Sign-on
message publication and access	OMA Converged IP Messaging	MMS-session control

Un enabler privilégiant la dimension technique doit être en correspondance avec la stratégie technique de l'entreprise. Il tire sa légitimité non du service rendu aux parties-prenantes du service, mais de la technologie qui le sous-tend. Par exemple à l'OMA, l'enabler OMA Data Synchronization spécifie la synchronisation de données entre un client mobile et un serveur par le protocole SyncML ; l'enabler OMA Download spécifie le téléchargement

depuis un client mobile vers un serveur ; l'enabler XDM la gestion de documents XML. De même, à l'ETSI, les capacités de service MMS-forwarding, Bidirectional Broadband conversational audio streaming ou Media gateway correspondent également à une réalisation de points de références. Un tel enabler devra donc dans notre perspective réaliser un point de référence de la vue technique de référence.

Nous avons ainsi déterminé des critères pour identifier et délimiter les enablers. Il semble pertinent de distinguer entre les enablers à dimension fonctionnelle et ceux à dimension technique. Des noms différents pourraient même être employés, comme enabler fonctionnel et enabler technique. Les enablers fonctionnels doivent respecter l'architecture fonctionnelle de référence de l'entreprise, en implémentant un ensemble cohérent de fonctions élémentaires. Un enabler implémentant une seule fonction élémentaire sera sans doute de grain trop fin, car trop dépendant d'autres composants. La bonne échelle sera sans doute le quartier fonctionnel, qui comme nous l'avons vu regroupe des fonctions homogènes quand à la nature de l'information traitée. Le tableau ci-dessus nous montre que ce niveau est cohérent avec les standards existants. Les enablers techniques doivent respecter l'architecture technique de référence. Ils devront donc se positionner comme réalisant un ou plusieurs points de référence.

Dans les deux cas, avant de spécifier un enabler, il conviendra de déterminer les fonctions ou les points de référence qu'il réalisera. Si ces fonctions ou ces points de référence n'existent pas dans l'architecture de référence, il conviendra tout d'abord de valider l'opportunité de les y ajouter en étudiant leur poids, fonctionnel ou technique, vis-à-vis de la stratégie fonctionnelle ou technique de l'entreprise. Ces étapes permettent ainsi de formaliser le périmètre d'un enabler et son importance pour l'entreprise.

V.2.2 Le cas des services logiciels

Nous avons vu au cours du chapitre I que l'identification des services SOA posait question. Comme pour les enablers, la vue fonctionnelle de référence peut nous aider à identifier ces services logiciels. Alors que les enablers privilégient un périmètre fonctionnel relativement large, afin d'être le plus possible indépendant les uns des autres, les services SOA ont habituellement un périmètre fonctionnel plus limité. Comme nous l'avons vu au cours du premier chapitre, un service SOA expose une interface, par exemple en WSDL, indiquant ses opérations avec leurs paramètres typés d'entrée et de sortie. Un service SOA peut donc être

indépendant d'autres services tout en ayant un périmètre fonctionnel restreint car il recevra en entrée les informations qui lui sont nécessaires. Cela implique bien sûr, comme nous le développerons ci-dessous, que les composants logiciels invoquant ce service partagent avec lui la même sémantique et la même syntaxe pour ces paramètres d'entrée. Les appels à différents services SOA peuvent ensuite être chaînés pour réaliser un service plus complexe, cet enchaînement étant par exemple réalisé avec un moteur d'orchestration, comme par exemple dans [Margaria, 2008] ou [Blum, 2008a].

Le travail d'identification de fonctions élémentaires à partir de processus métiers et de fonctions élémentaires mené dans le chapitre III nous donne un outil pour identifier ces services SOA dans le cas des services télécoms. Nous avons identifié les fonctions élémentaires à partir de deux critères (la cohérence avec les processus de service et la cohérence avec la stratégie fonctionnelle de l'entreprise), valables également pour identifier les services SOA.

Dans le domaine des services télécom, les services SOA sont principalement utilisés pour offrir des interfaces (API) à des partenaires, ces API permettant d'exposer des capacités des services télécoms. Ceci est par exemple le cas pour France Telecom avec le programme Orange Partner¹ ou pour Telefonica avec le programme Open Movilforum². Les opérations fonctionnelles telles que proposées au chapitre III nous fournissent ainsi une liste de services SOA typiques pour les télécoms. Comme nous l'avons évoqué, il est usuel dans les entreprises de nommer les services SOA selon un vocabulaire commun, en utilisant typiquement une liste restreinte de verbes pour les opérations. Notre démarche va plus loin en permettant de caractériser le service rendu par une opération selon une sémantique commune, celle de la vue fonctionnelle de référence.

Le tableau ci-dessous illustre la correspondance entre des services SOA du monde télécom et des opérations fonctionnelles SFO présentées au chapitre III. Les services SOA sont issus des travaux du groupe Parlay X, qui a justement pour mission de définir des API de services web au-dessus des réseaux de télécommunication. Ces spécifications ont été ensuite reprises par l'ETSI dans le cadre de OSA (*Open Service Access*).

¹ <http://www.orangepartner.com>

² <http://open.movilforum.com>

Tableau 43 : Opérations fonctionnelles et web service Parlay X

Exemple de Web service Parlay X et standard ETSI OSA correspondant	Opération SFO correspondante
sendSms (ETSI ES 202 504-4)	SFO issue message
makeCallSession (ETSI ES 202 504-2)	SFO issue phone call
setRules (ETSI ES 202 504-10)	SFO define routing rule
startMediaStream (ETSI ES 202 504-19)	SFO deliver content
createGroup (ETSI ES 202 504-13)	SFO define address list
addMember (ETSI ES 202 504-13)	SFO publish address

Les opérations fonctionnelles SFO ne spécifient bien sûr pas le détail d'un service SOA, mais sont un outil intéressant pour définir son contour fonctionnel.

Cette approche est également applicable aux services logiciels de type REST (pour *Representational State Transfer*). Pour les promoteurs de ce style d'architecture applicative, les données publiées par les services logiciels sont une caractéristique essentielle, au point que ce style repose principalement sur l'identification de ces données et sur leur manipulation via les opérations du protocole http (GET, POST, PUT, DELETE) [Fielding, 2000]. L'enjeu majeur est alors non plus l'identification des services, mais l'identification des données. Par exemple, dans le cas des services télécoms étudié dans [Lozano, 2008], les données manipulées sont les sessions d'appel IMS ou les états de présence. Dans ce cas, ce sont les données fonctionnelles SFD qui nous serviront de base pour périmétrer ces services logiciels.

V.2.3 Le cas des services sémantiques

Face à la multitude de services SOA disponibles dans une entreprise ou sur Internet, le catalogage de ces services devient un enjeu majeur. Comme évoqué au chapitre I, les travaux actuels, notamment dans des projets européens, s'orientent vers la construction de places de marché (*marketplace*) de services, à travers lesquelles on peut accéder à une grande variété de services suivant des procédures simples. En plus d'un point d'accès unique aux services, les places de marché en cours d'étude dans les projets académiques intègrent généralement des mécanismes sémantiques pour décrire les services d'une façon compréhensible par les machines. Des travaux axés sur les services dans les réseaux ad-hoc tels que [Maghmoumi, 2008] présentent le même besoin d'une description commune des services pour détecter des

affinités. Grâce à ces descriptions sémantiques, la proposition automatique de services en fonction d'un besoin devient possible. L'objectif est alors de trouver le service SOA le plus adapté à un besoin, exprimé par exemple en langage naturel (on pourra par exemple se référer à [Tarkoma, 2007]). Des mécanismes plus légers comme le classement par mots-clés sont insuffisants car ils permettent uniquement une catégorisation de haut-niveau et arborescente des services.

Pour cela, des ontologies, telles qu'introduites au premier chapitre, ont été développées, afin de faire le lien entre les annotations sémantiques des services et les besoins des utilisateurs. Mais, comme exposé par exemple dans [Hepp, 2007], ces ontologies s'avèrent très délicates à construire. Il s'agit en effet, comme nous l'avons vu, de relier de façon systématique le vocabulaire que l'utilisateur emploie pour décrire ses besoins avec le vocabulaire technique dont les développeurs se servent pour caractériser leurs services. Les technologies basées sur un framework sémantique pour la composition automatique de services Web, SWS (*Semantic Web Service*), comme WSMF (*Web Service Modeling Framework*), permettent de définir un objectif (*goal*), mais qui reste très difficile à lier de façon complètement automatique à un service. Ces solutions restent très complexes au niveau de l'implémentation et n'ont pas débouché sur des résultats généralisables pour l'instant. Elles se heurtent notamment à la difficulté à mettre en relation les aspects usage et les aspects applicatifs d'un service. Comme expliqué dans [Roche, 2007], il ne suffit pas de mettre sous forme d'ontologie le vocabulaire employé pour décrire un service, mais un véritable travail de conceptualisation est indispensable.

La description fonctionnelle des services telle que nous l'avons vue apporte justement des outils pour fonder une ontologie des services télécoms. Une opération fonctionnelle est en effet un pivot entre l'aspect usage d'un service (les actions des parties-prenantes du service) et ses aspects applicatifs (les composants applicatifs et leurs opérations applicatives portées par des points de référence). Nous avons commencé à mettre en œuvre ces notions dans le projet européen Seryery en y introduisant la notion de service abstrait [2009-4], un service abstrait étant la description fonctionnelle d'un service, en dehors de toute considération non-fonctionnelle, comme présenté dans [Boussard, 2009]. En se basant sur la vue fonctionnelle de référence, un service abstrait peut être défini comme un ensemble d'opérations fonctionnelles SFO. Dans le cas le plus simple, un service abstrait peut être unitaire, c'est-à-dire correspondre à une seule opération fonctionnelle SFO. Un service abstrait peut également

être plus complexe, composé de plusieurs SFO enchainées les unes à la suite des autres avec des embranchements logiques. Dans les deux cas, les opérations fonctionnelles SFO et leurs dépendances sont représentées par des concepts dans une ontologie. Les développeurs de services peuvent alors rattacher leurs services à ces services abstraits. Par exemple, à un service abstrait défini par l'opération *SFO send message* pourront correspondre des services logiciels concrets, tel un service web d'envoi de SMS d'Orange ou un service REST d'envoi de SMS de Telefonica. Ces services abstraits peuvent être découverts, partagés, composés par des utilisateurs, au même titre que des services logiciels classiques. Notre objectif est que la traduction d'un service abstrait en service logiciel concret soit uniquement réalisée au *runtime*, lors de l'exécution du service.

La vue fonctionnelle de référence est ainsi une aide pour identifier des éléments réutilisables pour les architectures de services télécoms, que ce soient des enablers, des services logiciels ou des concepts d'une ontologie. Nous allons maintenant voir comment utiliser ces résultats pour faciliter la convergence de service, notamment à travers l'exemple de la convergence web/IMS.

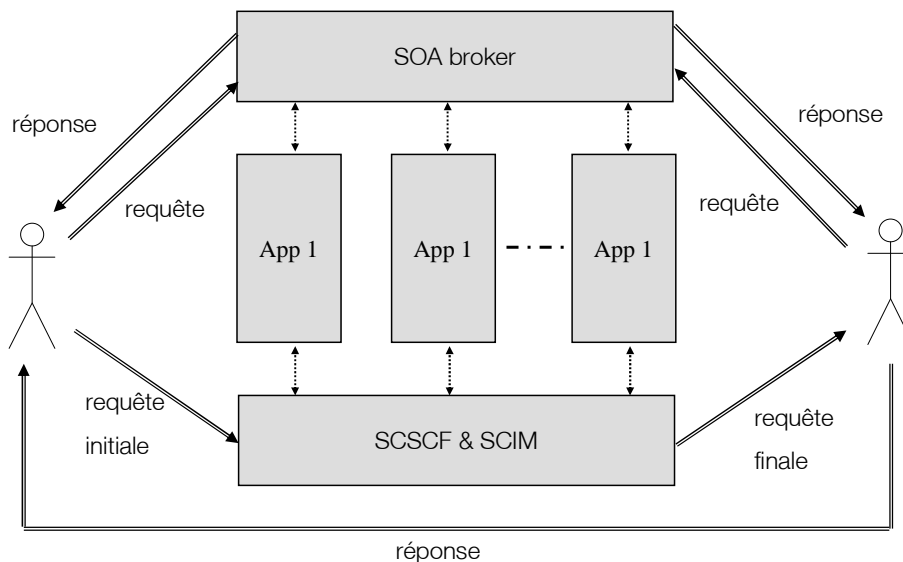
V.3 Une aide pour réaliser la convergence de service

Nous avons développé le long du chapitre II les enjeux de la convergence de service. Nous allons maintenant regarder comment la démarche de description des architectures de service des chapitres III et IV constitue un outil pour réaliser cette convergence, à travers plusieurs exemples. Nous rechercherons d'abord un motif technique pour clarifier la question de la convergence web/IMS selon la découpe front-office / back-office introduite aux sections 2.1 et 3.1 du chapitre II. Puis, nous verrons comment, au sein de ce front-office, diverses capacités de service peuvent être présentées aux utilisateurs et composés par eux en réponse au besoin d'un consommateur, comme évoqué à la section 3.3 du même chapitre II, et comment ces capacités présentées aux utilisateurs peuvent suivre une sémantique commune avec les composants back-office. Cela nous conduira à généraliser à d'autres environnements de services convergents en étudiant comment la vue fonctionnelle de référence peut constituer un soutien pour la communication entre services.

V.3.1 Un motif technique pour la convergence web/IMS

Par rapport à notre réflexion du chapitre II, la distinction entre vue fonctionnelle et vue technique permet de clarifier les architectures. La distinction entre protocoles comme SIP ou http relève de la vue technique. Les différents services à composer pour répondre à un besoin relèvent de la vue fonctionnelle. Les figures 22 ou 23 du chapitre II sont des modes de représentation usuels des architectures de convergence. Ces figures restent de haut-niveau. Nous pouvons les affiner et les rendre plus utiles en suivant la démarche de description de l'architecture d'un service présentée au long du chapitre IV.

Ainsi, la figure 22 rappelée ci-dessous est suivant notre cadre la vue technique d'un service, c'est-à-dire un ensemble de nœuds d'exécution reliés entre eux par des points de référence.



Les nœuds d'exécution correspondant à cette figure seront :

- l'utilisateur, en tant qu'émetteur et destinataire de flux via des IHM (notons que le point de référence GUI entre le terminal UE et l'utilisateur est bidirectionnel, tout comme le point de référence Gm, car l'utilisateur émet des requêtes vers, mais reçoit également des notifications du terminal UE) ;
- les interfaces de présentation, avec un terminal pour l'IMS (UE pour *User Equipment*) et une interface utilisateur pour les services SOA (UI pour *User Interface*) ;
- les courtiers, avec le CSCF pour l'IMS et le *broker* SOA ;

- les serveurs d'applications avec des AS pour *Application Server*, à la fois pour l'IMS et pour SOA.

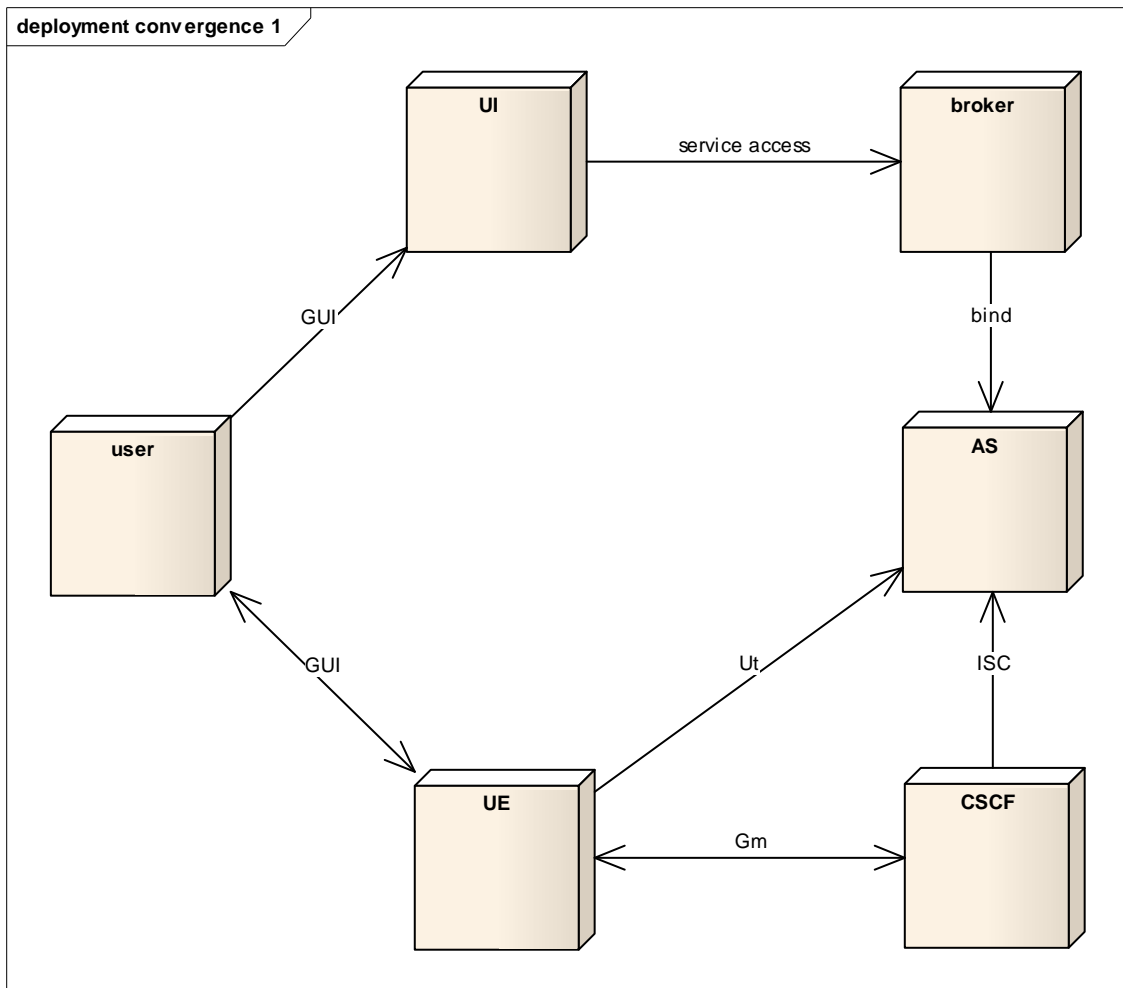
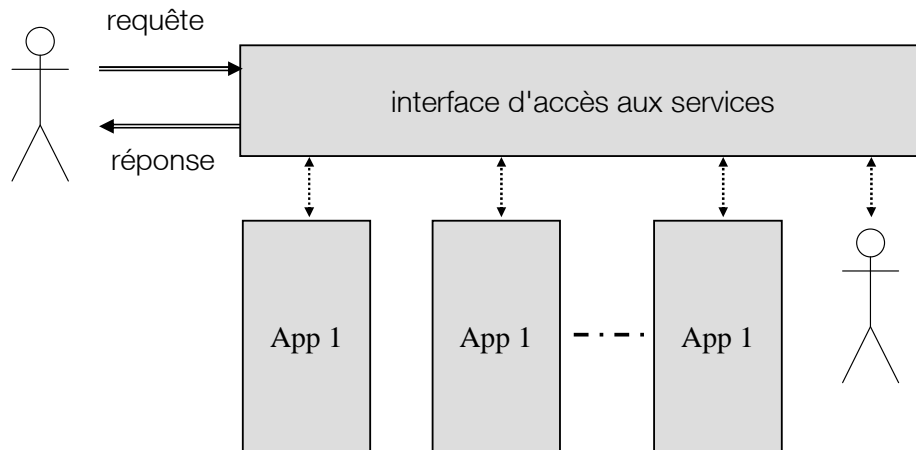


Figure 104 : Vue technique d'un environnement de services convergents SOA+IMS

Une telle découpe avait déjà été proposée, notamment dans des articles tels [2006-5] ou [Bachmann, 2008]. Néanmoins, le mode de représentation adapté ici présente plusieurs avantages. Tout d'abord, il permet de se placer explicitement dans une vue technique. Ensuite, les points de référence de l'architecture technique de référence sont une aide précieuse pour le réaliser de façon exacte et sans omission. Enfin, il permet de travailler facilement les évolutions possibles (vers vue applicative ou en modifiant les rôles techniques).

La figure 23, rappelée également ci-dessous, constituait une proposition d'architecture pour une convergence web/IMS s'inscrivant pleinement dans le motif front-office / back-office présenté à la fin de la section 2.2 du chapitre II.



Nous pouvons représenter cette architecture technique de façon plus formelle, en ajoutant un nœud d'exécution "service access interface" et en supprimant le nœud d'exécution UI. Il nous suffit pour cela de reconfigurer les points de référence entre nœuds d'exécution. Le terminal UE est dépouillé de son interface utilisateur, puisque toutes les interfaces des services sont centralisées dans le nœud d'exécution "service access interface". Et il doit fournir un point de référence "service access" du motif technique web. Son nom est donc changé en "web-enabled UE". Ce nœud d'exécution pourrait correspondre à une passerelle SIP / web services, telle que décrite dans [Verdot, 2009]. Cette architecture est présentée dans la figure ci-dessous.

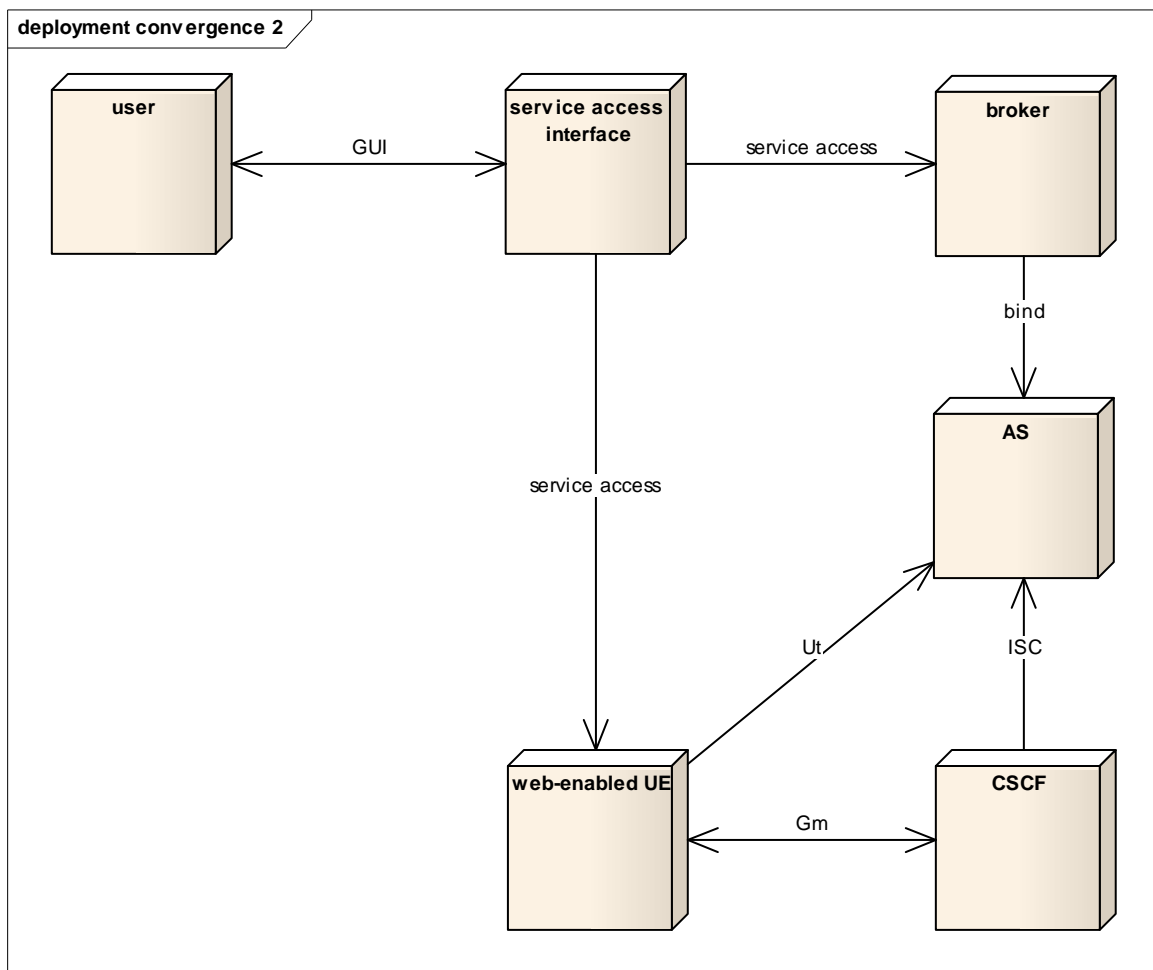


Figure 105 : Vue technique d'un environnement de services convergents sans couture (provisoire)

Grâce à cette formalisation, nous détectons immédiatement une incohérence. Alors que les points de référence GUI et Gm sont bidirectionnels, le point de référence service access ne l'est pas. Le nœud d'exécution "web-enabled UE" ne peut alors pas avertir l'utilisateur d'un appel entrant. Pour cela, nous devons ajouter un point de référence, que nous appellerons "RP web push" du rôle technique "TR service logic" vers le rôle technique "TR web page". Ce point de référence peut typiquement être réalisé en utilisant les mécanismes de la solution technique Flash de Adobe (via le protocole RTMP – *Real Time Messaging Protocol*), comme dans [Verdot, 2009], ou bien par la solution exposée dans [2008-6], basée sur la technologie Cometd de push via le protocole http. Cette architecture corrigée est représentée dans la figure ci-dessous. Cette figure constitue un motif technique pour la convergence web/IMS pleinement conforme au motif front-office / back-office.

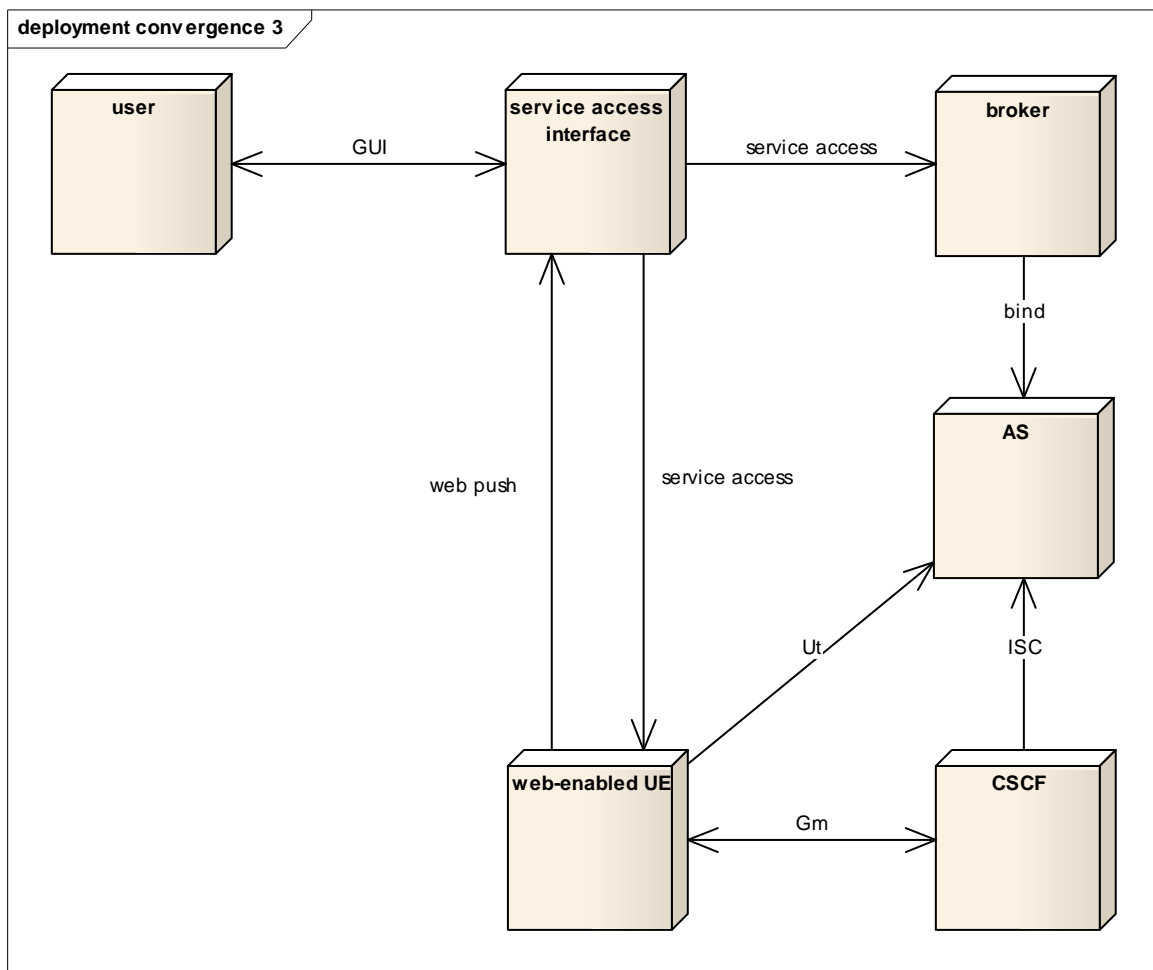


Figure 106 : Vue technique d'un environnement de services convergents sans couture (finale)

Les nœuds d'exécution broker, AS et CSCF correspondent à des équipements existants et bien connus dans le domaine SOA ou IMS. Le nœud d'exécution web-enabled UE correspond à des produits en cours d'étude tels que présentés dans [Verdot, 2009] ou dans [2008-6]. Des produits industriels commencent à être présent sur le marché, comme Flash Media Gateway de Adobe¹, NeXpresso de Nexcom² ou Intellivic d'Androme³. Quant à l'interface d'accès au service, *service access interface*, elle pourrait notamment être réalisée par des widgets au sein d'un agrégateur de services, tel que nous l'avons développé dans [2008-4]. Un agrégateur est une page web capable d'afficher de multiples interfaces web indépendantes, les widgets ou gadgets. Chaque service y est ainsi présenté graphiquement sous la forme d'une widget, définie par le W3C (*World Wide Web Consortium*) de la manière suivante.

¹ <http://www.adobe.com/products/flashmediainteractive/>

² <http://www.nexcom.fr/nexpresso-flash.pdf>

³ <http://www.intellivic.com/>

"Widgets are small client-side Web applications for displaying and updating remote data that are packaged in a way to allow a single download and installation on a client machine, mobile phone, or mobile Internet device"

De telles interfaces sont également en cours d'investigation dans le projet CELTIC Servery [Boussard, 2009] sous le nom de PSE pour *Personal Service Environment*. A titre illustratif, la figure suivante présente une capture d'écran de l'interface d'accès unifié aux services présentée dans [2008-4].

Une fois résolue la question du motif technique pour la convergence web/IMS, il nous reste à nous demander comment une sémantique commune peut faire le lien entre utilisateur, front-office (interface d'accès au service) et back-office (AS).

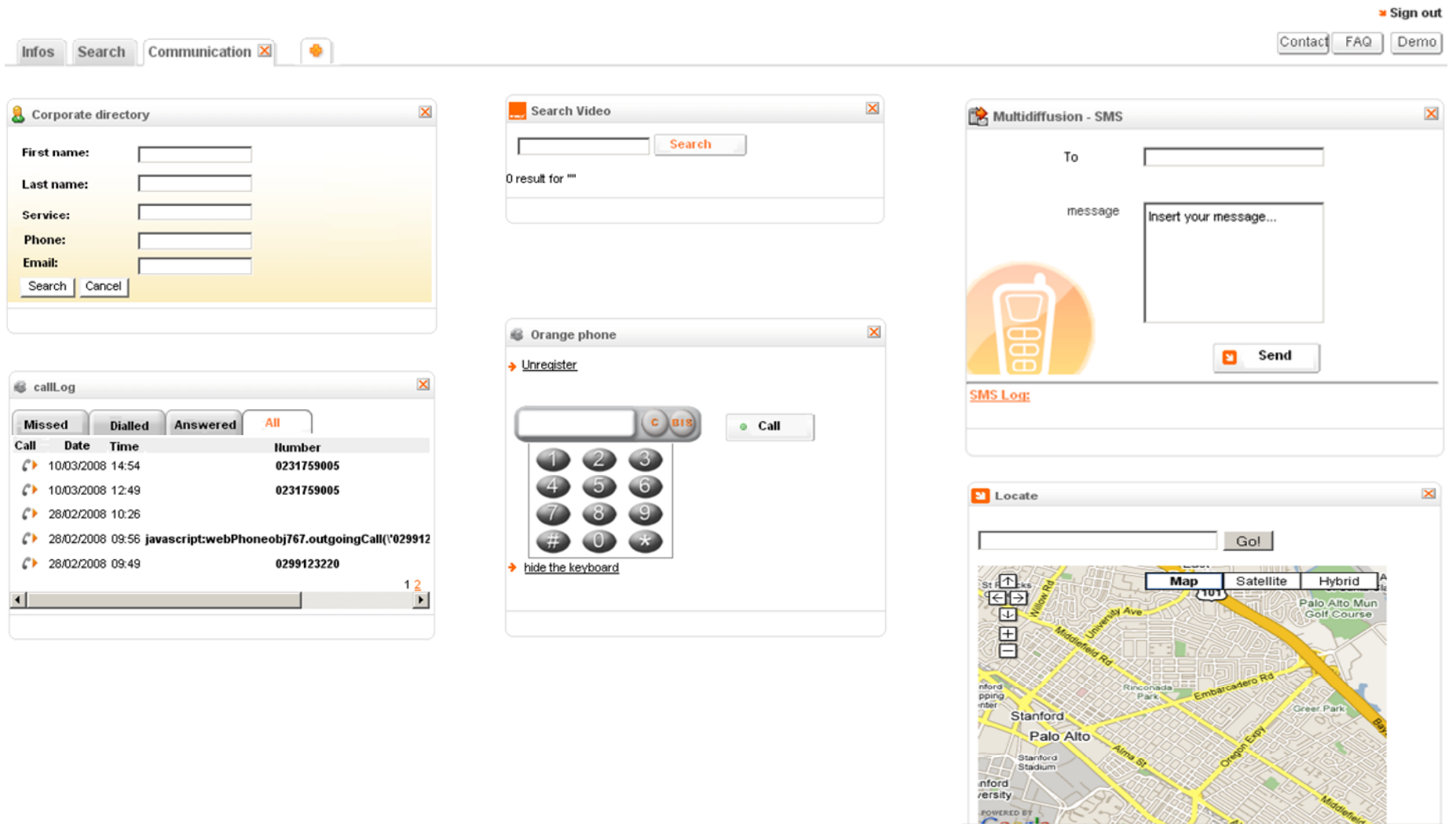


Figure 107 : Exemple d'interface d'accès unifié aux services

V.3.2 Un accès unifié aux services

Pour instancier ce motif technique, nous devons étudier la vue applicative correspondante. Les nœuds d'exécution user, broker, web-enabled UE et CSCF donneront lieu à un seul composant applicatif SAC, ces composants étant communs aux différents services fournis. Par contre, les nœuds d'exécution service access interface et AS sont spécifiques à chaque service. Il y aura donc de multiples composants AS et service access interface (par exemple sous forme de widgets), selon les services proposés au sein de cette architecture. Le composant service access interface constitue la partie front-office d'un service, et le composant AS en constitue la partie back-office, selon le motif présenté à la fin de la section 2.2 du chapitre II. La question que nous soulevions à la fin du chapitre II était celle d'une sémantique commune au front-office et au back-office, donc ici entre widgets et AS. Les fonctions élémentaires SEF et les opérations SFO définies dans la vue fonctionnelle de référence répondent à ce besoin.

Dans la vue applicative, le terme d'AS est un terme générique pouvant recouvrir des enablers, des services logiciels SOA ou des services SIP. Ces AS peuvent donc être décrits, comme vu au chapitre IV, par des opérations applicatives, c'est-à-dire des opérations fonctionnelles SFO portées par des points de référence. Le point de référence bind portera alors les opérations fonctionnelles offertes par les services SOA et le point de référence ISC celles offertes ou utilisées par les services SIP. D'autres points de référence pourront éventuellement être impliqués dans le cas des enablers.

Par ailleurs, nous avons vu dans la deuxième section de ce chapitre que les enablers ou les services logiciels avaient besoin d'être rationalisés par une sémantique commune, afin d'éviter les recouvrements fonctionnels et faciliter la composition de service. Cette problématique est la même pour les widgets, interfaces d'accès aux services. Par exemple, dans l'agrégateur de service illustré dans la figure précédente, comment assurer une cohérence entre les fonctionnalités rendues par les différents éléments ? Les widgets présentées dans cette interface peuvent en effet être redondants, ou implémenter des fonctionnalités trop complexes. Comme présenté dans [2009-2], les widgets ne doivent implémenter que des fonctionnalités atomiques pour pouvoir être aisément intégrées dans un environnement de services convergents. Nous proposons ainsi dans cet article la définition suivante des widgets (nous soulignons en gras la différence avec la définition du W3C).

"Widgets are small client-side web applications for offering atomic functionalities of an enterprise application that are packaged in a way to allow a single download and installation on a client machine, mobile phone, or mobile Internet device" [2009-2]

Ceci est en cohérence avec les travaux développés par J. Raskin, concepteur de l'IHM du Macintosh, dans [Raskin, 2000]

"Instead of providing application programs, software vendors will provide command sets that offer a collection of related operations. For example, instead of a program for doing photographic manipulation, a vendor would provide a number of independent commands that, collectively, provide exactly the same range of operations. The user would install as many or as few as she finds useful rather than having to install a huge application, of which only a fraction would be used."

Comment déterminer ces fonctionnalités atomiques ? A nouveau, la vue fonctionnelle de référence peut nous guider, comme pour les enablers ou les services logiciels. Ces fonctionnalités atomiques qui sont habituellement déterminées de façon intuitive se retrouvent facilement dans la vue fonctionnelle de référence. La figure suivante présente ainsi la correspondance entre widgets et fonctions élémentaires SEF. La seule widget sans correspondance est celle d'affichage de plan. Nous n'avons en effet pas considéré le service de cartographie dans notre analyse du chapitre II sur les processus de service télécoms. Une analyse plus exhaustive conduirait assurément à le prendre en compte, ce qui donnerait sans doute dans la vue fonctionnelle de référence une fonction élémentaire comme *SEF location browsing capability*. Quant à la *SEF call log browsing capability*, elle correspond à la navigation dans un journal d'appels.

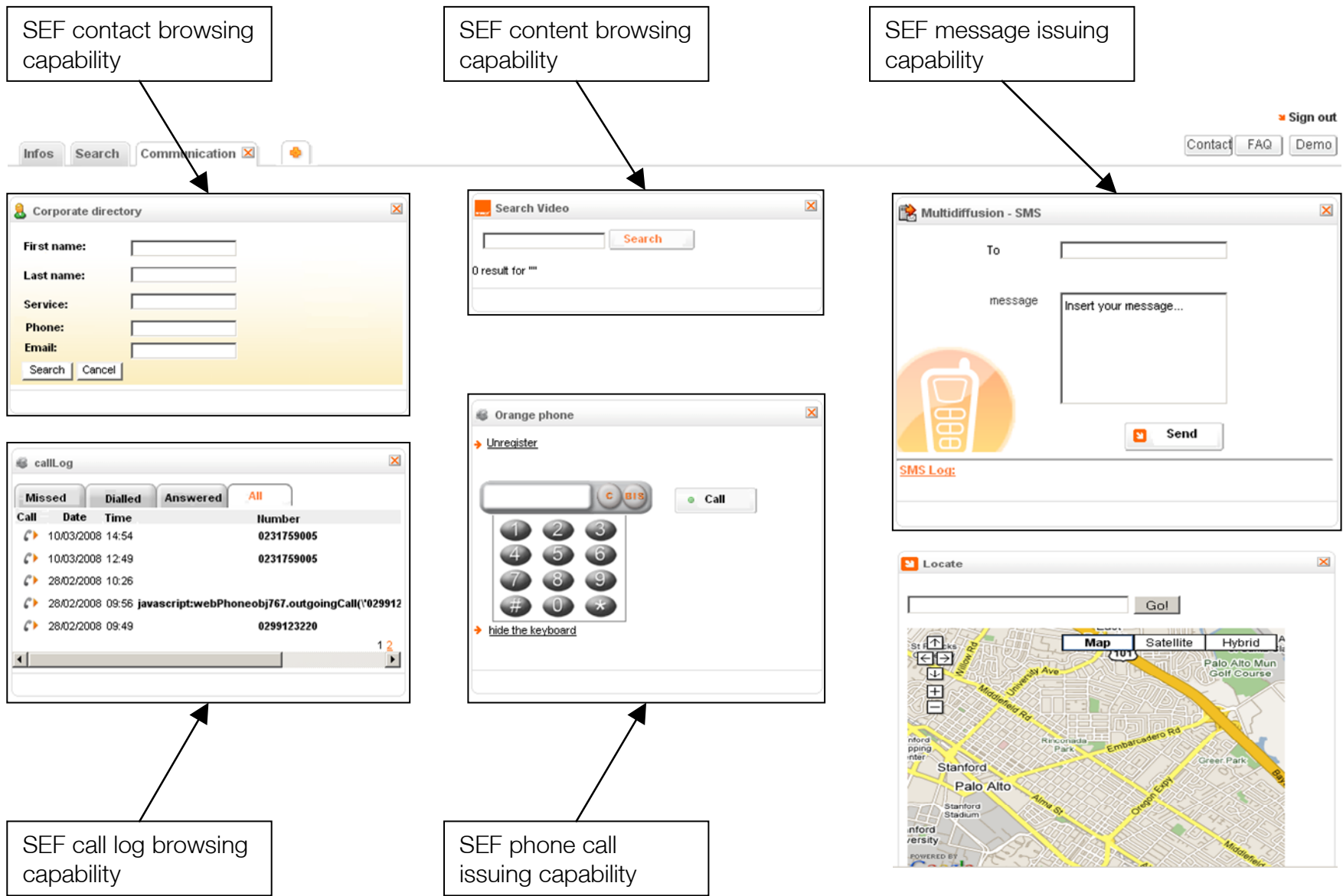


Figure 108 : Widgets et fonctions élémentaires

La description de widgets, de services SOA et d'enablers à partir de la vue fonctionnelle de référence permet d'envisager une plus grande cohérence des architectures de service. Une widget, un service SOA, un enabler peuvent être décrits de façon uniforme par la même opération fonctionnelle, cette opération étant portée par des points de référence différents – RP GUI pour les widgets, RP bind pour les services SOA et divers points de référence (en fonction des protocoles) pour les enablers. Ceci permet de remplir le besoin exprimé dans [Yelmo, 2008] d'une plus grande cohérence entre le front-end et le back-end des services dans les environnements de service convergents.

Les services peuvent être enchaînés les uns avec les autres en réponse à un besoin précis d'un utilisateur. Cet enchaînement peut soit être construit par un développeur, par exemple à l'aide de langages tels BPEL, soit généré à la volée, par exemple en réponse à un besoin exprimé en langage naturel.

Cette description unifiée ouvre la voie à une composition de service unifiée, telle qu'évoquée par exemple dans [Lavinal, 2009] ou [Arbanowski, 2004]. Un utilisateur pourra naviguer dans une liste de capacités de services ou saisir une demande en langage naturel. S'il existe déjà, un service complet (front-end et back-end) sera mis à sa disposition. S'il existe uniquement sous forme de service back-end, une widget pourra être automatiquement générée, à partir de la signature du service. Le style de l'interface sera adapté en fonction de l'opération fonctionnelle demandée. Si aucun service ne correspond à cette demande, un nouveau service pourra être créé par composition de services existants. Comme les entrées/sorties des services logiciels sont décrites à travers les opérations fonctionnelles qu'ils réalisent, la construction d'un enchaînement entre services est facilitée, notamment par rapport à des approches comme [Lécué, 2009] qui ne définissent pas de modèle pivot. Cette composition automatique de services suite à une requête en langage naturel est étudiée dans le projet collaboratif Servedy. La description fonctionnelle des services servira de base à cette composition.

La figure suivante présente ainsi une widget dans laquelle l'utilisateur peut entrer son besoin en langage naturel, aux côtés d'une widget d'affichage du calendrier (correspondant à *SEF appointment browsing capability*) et d'une widget d'affichage de mails (correspondant à *SEF message browsing capability*).

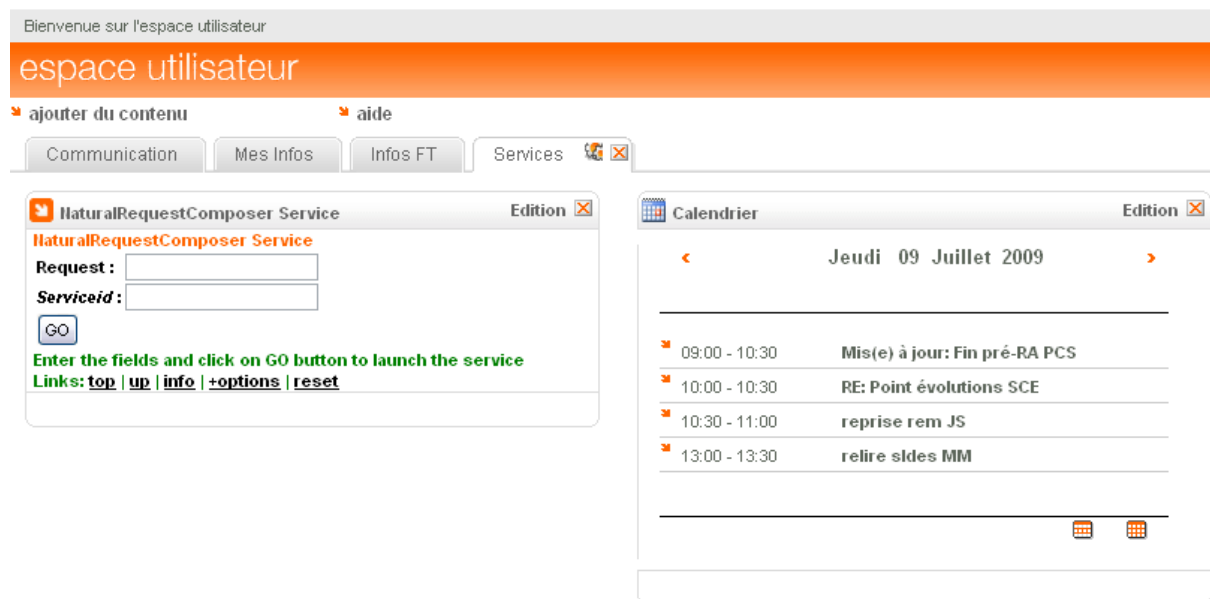


Figure 109 : Widget de construction de composition de service

La requête en langage naturel (typiquement exprimé sous forme de verbes d'action et de compléments) est transformée en une description fonctionnelle du service, en prenant en compte les différentes tournures et synonymes pour exprimer la même fonction. Cette description fonctionnelle est ensuite traduite en un enchaînement de services back-end. Une nouvelle widget est générée pour demander les inputs nécessaires et afficher le résultat si nécessaire. A titre d'exemple, la widget suivante a été générée en réponse à une demande en langage naturel "send sms". Des champs de saisie sont générés pour la saisie du n° de mobile et du message.

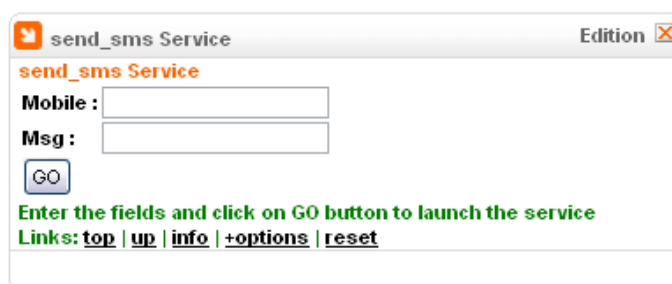


Figure 110 : Résultat d'une composition de service

Ces travaux sont notamment développés par Mariano Belaunde [Belaunde, 2008] dans le cadre du projet Servery.

V.3.3 La communication entre services

Dans les exemples présentés, la composition de service est conçue à l'avance comme un enchaînement de services donnés, cet enchaînement ayant été spécifié soit par un développeur, soit par un utilisateur [2008-9]. Mais nous pourrions aussi, dans une perspective autonome telle que présentée par exemple dans [2008-3], laisser chaque service invoquer lui-même d'autres, sans que ces invocations soient prédéterminées par un script de composition. Au lieu de composer des services à l'avance, il s'agirait ici de permettre à chaque service de coopérer avec d'autres services au sein d'un environnement de services convergents. Cette coopération est généralement codée dans le comportement du service par son développeur, mais pourrait aussi être à l'initiative de l'utilisateur, en réponse à des requêtes de sa part. Ceci élargit le cadre de la composition de service vers la communication entre services. Il n'y a plus de services unitaires ou composés, mais des services capables de s'invoquer les uns les autres sans se connaître au préalable. Cela peut être vu comme la réalisation cible de l'environnement de services convergents que nous visions au chapitre II.

Nous avons vu au premier chapitre comment le paradigme SOA impliquait la décomposition des applications en fonctionnalités élémentaires, les services SOA. Nous avons également vu dans la section précédente qu'une telle décomposition pouvait également être valable pour les interfaces graphiques, à travers le concept de widget. La communication entre services permet d'envisager la construction ad-hoc d'un ensemble cohérent de fonctionnalités, ce que nous appelons usuellement une application, par l'assemblage de composants de service hétérogènes – services SOA, widgets – communiquant les uns avec les autres.

Il s'agit donc de laisser, dans une perspective plus décentralisée, le soin à chaque composant de service d'invoquer d'autres composants de services. Ainsi, un composant de service de carnet d'adresses peut invoquer un composant de téléphonie pour lancer un appel vers un contact sélectionné. L'appel vers un composant de téléphonie prédéfini ne doit bien sûr pas être inscrit dans le code du composant de carnet d'adresses, sinon nous revenons à un mode centralisé où tous les composants sont connus à l'avance. Il s'agit au contraire que le composant de carnet d'adresses soit capable de faire appel à un composant de téléphonie lorsque ce dernier devient présent dans l'environnement. Chaque composant de service

devient alors pour ainsi dire adaptatif, c'est-à-dire capable de s'adapter aux capacités de service présentes dans son environnement. Cet environnement, que l'on peut rapprocher de l'environnement de service unifié présenté à la figure 13 est le cadre d'exécution des composants de service. Alors que les modes d'invocation classiques de composants de service ne permettent que des connections prédéfinies et fixes entre les composants, une telle approche permet a contrario de combiner dynamiquement les composants de service, sans que celui qui effectue une requête ne connaisse au préalable les composants disponibles au sein de l'environnement unifié de services.

Cet environnement peut être par exemple un système d'exploitation. Ainsi Google Android, un des systèmes d'exploitation les plus récents, permet justement une telle communication entre composants de service. Chaque composant peut exprimer une intention, *intent*, définie ainsi :

"An intent is an abstract description of an operation to be performed. (...) An intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed. The primary pieces of information in an intent are: action (the general action to be performed (...)) and data (the data to operate on, such as a person record in the contacts database, expressed as a Uri).

Some examples of action/data pairs are ACTION_VIEW content://contacts/1 (display information about the person whose identifier is "1") or ACTION_DIAL content://contacts/1 (display the phone dialer with the person filled in)."

Le système d'exploitation détermine le composant de service le mieux à même de réaliser cette intention. Par exemple, lorsque l'intention est d'appeler un contact, c'est un composant de téléphonie qui sera invoqué. Cette résolution n'est pas déterminée une fois pour toute, mais dynamique. Chaque composant de service déclare en effet auprès du système d'exploitation quelles sont les actions qu'il peut réaliser et sur quels types de données. Lorsqu'une intention comporte cette action et une donnée de ce type, ce composant sera alors invoqué. Pour plus de détail, on pourra se référer à [Android, web], mais aussi à [Rolland, 2009] pour un background théorique sur la notion d'intention pour le paradigme SOA.

Cette approche a été étendue aux services télécoms par Niklas Blum dans [Blum, 2008b] avec InSerT, pour *Intent-based Service Request*. L'environnement de services unifié est alors un ensemble de plates-formes de services télécoms. Une intention sera par exemple

l'envoi de SMS. Des composants applicatifs peuvent par exemple se déclarer comme envoyant des SMS et d'autres déclarer leur besoin d'envoyer des SMS.

Mais une telle approche est également applicable aux widgets, comme nous l'avons montré avec Nassim Laga dans [2009-2] ou [2009-3]. L'environnement de services unifié est alors un agrégateur de services, présenté dans une page web. Les widgets peuvent déclarer les données qu'elles produisent et les données qu'elles acceptent en input. Par exemple, une widget d'annuaire pourra déclarer sa capacité à accepter une donnée de type contact (afin d'afficher sa fiche annuaire). Et une widget de journal d'appel ou de consultation d'emails pourra déclarer sa capacité à produire une donnée de type contact. La mise en relation entre ces widgets peut alors se faire d'au moins deux manières.

- Premièrement, le fait que l'environnement unifié de services soit constamment présenté à l'utilisateur sous forme de page web permet de laisser à cet utilisateur l'initiative des communications entre widgets. Ce n'est pas le composant de service qui est programmé pour émettre une intention qui sera résolue dans l'environnement, mais c'est l'environnement qui offre à l'utilisateur la possibilité d'exécuter le service offert par un composant à partir de données produites par un autre composant. Cette possibilité peut être rendue graphiquement par divers procédés. Un mécanisme de drag and drop peut être mis en place entre widgets. Par exemple, on pourra glisser un contact depuis une widget de journal d'appel vers une widget d'annuaire. Ce mécanisme est détaillé dans [2009-2]. Un autre mécanisme peut consister en l'ajout automatique d'icônes au sein de chaque widget. Chaque icône symbolise une autre widget de la page susceptible d'être invoquée à partir d'informations produites par la première widget. Ce mécanisme est détaillé dans [2009-3].
- Secondement, des liens entre types de widgets peuvent être mis en place par un rôle d'administrateur pour tous les utilisateurs dont cet administrateur est responsable. Par exemple, un lien entre les widgets de type journal d'appel et les widgets de type annuaire. Ces liens sont ensuite automatiquement mis à jour en fonction des widgets présentes dans la page, c'est-à-dire dans l'environnement unifié de services. Ceci facilite notamment l'outillage des processus métiers lorsque cet environnement est utilisé dans le cadre d'une entreprise.

Dans les trois cas que nous venons de parcourir – système d'exploitation, plates-formes de service télécom, agrégateur de services –, se pose la question de la correspondance entre la requête d'un composant de service et les capacités déclarées par les autres composants de service. Comme nous l'avons discuté dans [2008-9], cette mise en correspondance peut se faire diversement. Reprenons nos trois exemples. Dans le cas de Google Android, la mise en correspondance est faite via un vocabulaire limité et prédéfini par Google : les actions demandées dans un message d'intention – comme ACTION_VIEW ou ACTION_DIAL. Dans le cas des widgets communicantes, nos implémentations actuelles se basent sur un formatage commun des données générées et acceptées par les widgets. Par exemple, un format commun sera utilisé pour les données de contact. Nous avons pour cela privilégié les microformats [Microformats, web], formatage de données à l'aide d'attributs XHTML sur lequel existe un certain consensus dans la communauté web. Par exemple, les données de contact sont formatées selon le microformat hCard. Cette approche pose toutefois un problème de complétude. Toutes les données nécessaires ne sont pas spécifiées sous forme de microformats. Dans le cas de la proposition InSerT pour les plates-formes de services télécoms, il n'y a pas de vocabulaire ou de format de données prédéfini, mais la mise en correspondance serait réalisée par des mécanismes sémantiques. Une ontologie des services de télécommunication permettrait de faire le lien entre les demandes d'un composant de service, demandes exprimées dans un vocabulaire propre à ce composant (ou plutôt à ses développeurs), et les capacités offertes par les autres composants de service, capacités exprimées elles aussi dans un vocabulaire propre à chaque composant. Si ceci fonctionnera dans des cas simples – par exemple les synonymes comme envoyer un SMS et envoyer un texto –, il semble délicat, comme nous l'avons discuté au cours du premier chapitre, de construire une ontologie des services télécoms sans partir d'une base stable, à laquelle pourront se rattacher les diverses variations sémantiques. Dans les trois cas, nous voyons donc le besoin d'un vocabulaire de référence. Comme sémantique partagée dans les deux premiers et comme base de construction d'une ontologie dans le dernier. Dans les trois cas, le mode de construction de ce vocabulaire de référence n'est pas abordé. Il est à la discrétion de Google dans le premier cas. Il suit dans le deuxième cas les bonnes pratiques établies par la communauté des microformats, mais n'est pas extensible à de nouveaux besoins (aucun microformat n'est ainsi défini pour les appels téléphoniques). Repose sur une ontologie des services de communication dont le mode de construction n'est pas décrit dans le dernier cas.

Notre travail du chapitre III sur un modèle fonctionnel de référence fournit justement une base pour ce vocabulaire de référence, l'avantage étant que le mode de construction de ce vocabulaire est entièrement explicité. Il se base sur des éléments considérés comme des invariants sur un intervalle de temps suffisamment long : les entités manipulées par les parties-prenantes des services, leurs actions sur ces entités et les motifs fonctionnels du domaine des télécoms. Nous avons également explicité le cycle de vie de ce vocabulaire, notamment en fonction de la stratégie d'un opérateur télécoms.

La vue fonctionnelle de référence permet de formaliser les capacités offertes et les capacités requises par un composant de service. Elle peut même permettre de contrôler des liens entre composants de service en fonction des liens entre opérations fonctionnelles SFO et données fonctionnelles SFD. Ainsi, l'on peut envisager qu'un composant de service implémentant une opération fonctionnelle SFO-A ne puisse invoquer que les composants qui produisent les données fonctionnelles SFD-X qui sont des paramètres d'entrée de SFO-A.

Pratiquement, nous investiguerons dans nos travaux ultérieurs l'utilisation de cette vue fonctionnelle de référence dans plusieurs directions :

- La construction d'une ontologie dans le domaine particulier des fonctions des services de communication, qui pourra ensuite être utilisée soit pour résoudre le lien entre une requête en langage naturel et les capacités des composants de service, soit résoudre les demandes de capacité d'un composant de service ;
- La caractérisation de widgets d'un agrégateur de service à travers les opérations fonctionnelles SFO, c'est-à-dire à la fois les entités de service qu'elles manipulent et les actions qu'elles permettent de réaliser sur ces entités de service. On pourra alors envisager de mettre à profit les dépendances de la vue fonctionnelle de référence (le fait qu'une opération fonctionnelle nécessite en paramètre d'entrée un paramètre de sortie produit par une autre opération fonctionnelle) pour proposer des liens privilégiés entre widgets.

Ces travaux seront notamment menés dans le cadre du projet collaboratif Servery.

Synthèse

Nous avons vu dans ce chapitre diverses applications et perspectives d'application des résultats des chapitres III et IV, applications organisationnelles et applications logicielles.

Les applications organisationnelles sont des bénéfices directs de notre démarche pour décrire de façon univoque les architectures de service. Cela facilite en effet les échanges entre les acteurs de la conception d'un service : le marketing qui conçoit le périmètre fonctionnel du service, les prescripteurs techniques qui spécifient les technologies à utiliser, les comités d'architecture qui poussent à la réutilisation de composants applicatifs existants, les équipes de développement qui implémentent le service, les équipes d'intégration qui le mettent en place, les équipes du système d'information qui en développent la prise de commande, le service après-vente... Les architectures de services différents peuvent être plus facilement comparées, y compris par des moyens formels tels des métriques. Outre une sémantique commune pour décrire les architectures de service, la vue fonctionnelle et la vue technique de référence constituent également un outil pour identifier des composants applicatifs réutilisables (enablers, services SOA, widgets) et nous aident à distinguer entre réutilisabilité fonctionnelle et réutilisabilité technique. Ces applications organisationnelles peuvent être immédiatement appliquées pour expliciter les enjeux de la convergence de service introduite au chapitre II.

Les applications logicielles consistent en l'intégration de notre sémantique de référence au sein des applications informatiques, afin de réaliser plus simplement des services convergents. Faciliter la gestion des services en embarquant dans les plates-formes de service une référence aux opérations applicatives réalisées par ces plates-formes et aux points de référence exposés. Faciliter la découverte de services en identifiant des concepts de base pour mettre en place une ontologie cohérente des services télécoms. Faciliter la communication entre services en embarquant dans chaque widget ou plate-forme de service une référence aux opérations applicatives réalisées et utilisées, afin que des relations entre ces composants puissent se mettre en place dynamiquement en fonction de leur présence dans un même environnement de service.

Conclusion

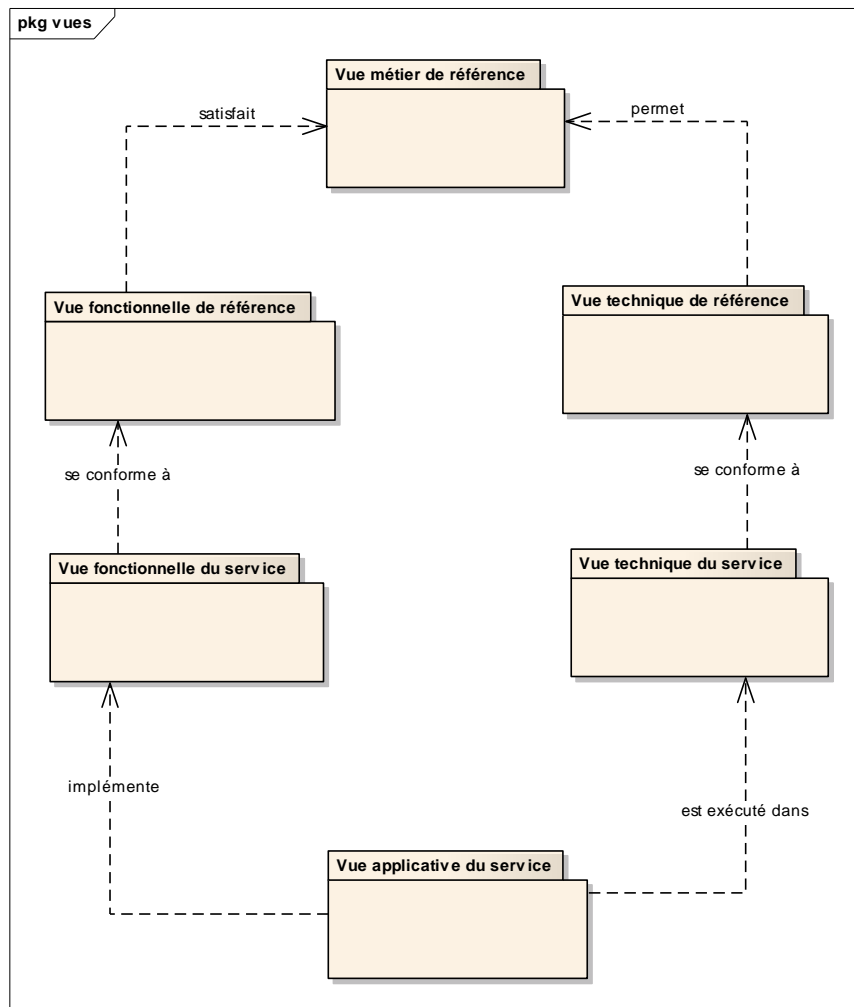
Au terme de cette thèse, le lecteur se demandera peut-être si nous y avons cédé à une "*passion française de la généralité et de l'abstraction*"¹ en réduisant abusivement la complexité, la diversité, le foisonnement des services de communication. A la différence d'un travail d'ingénieur, un travail de recherche consiste sans doute moins à apporter une solution clé en main qu'à creuser une question jusqu'à en obtenir une problématique claire, à en dérouler les conséquences en suivant une approche particulière, puis à valider la valeur ajoutée de cette approche – sans que cela invalide nécessairement d'autres approches. Nous sommes partis ici d'une question simple : comment décrire l'architecture des services de communication, alors que la convergence des services pousse à sortir d'un mode artisanal, au cas par cas, de construction de ces architectures pour aller vers une démarche commune aux divers services ? L'idée même d'architecture implique un cadre structuré pouvant servir de support aux échanges entre les multiples parties-prenantes dans la conception d'un service.

Notre question de départ nous a poussés dans deux directions. D'une part, une architecture étant définie comme la façon dont un système est constitué par des éléments en vue de son fonctionnement, quels sont les éléments à considérer lorsque le système est un service de communication ? D'autre part, comment définir une base comparative, des invariants stables, si nous voulons pouvoir exprimer ces éléments non seulement selon un même formalisme, une syntaxe commune, mais en respectant une sémantique commune, afin de pouvoir les comparer, identifier les réutilisations possibles, valider la pertinence de leur composition et finalement les intégrer dans un environnement de services convergents ?

Nous avons tout d'abord discuté de la tendance à la convergence dans les services de communication, en distinguant la convergence de service de la convergence de réseau, puis en replaçant cette convergence de service dans le contexte général du secteur des services.

Nous avons ensuite adopté la méthodologie de l'architecture d'entreprise et de l'urbanisation des systèmes d'information pour considérer les éléments constitutifs d'un service selon quatre points de vue : métier, fonctionnel, technique et applicatif. Une vue

métier de référence, une vue fonctionnelle de référence et une vue technique de référence peuvent contenir une sémantique de référence qui servira à exprimer la vue fonctionnelle, la vue technique et la vue applicative d'un service², suivant la figure 27 rappelée ci-dessous.



Nous avons montré que ces vues étaient adaptées aux services de communication. Un formalisme a été défini pour chacune, avec par exemple des fonctions élémentaires SEF, des points de référence RP ou des composants applicatifs SAC. Nous avons formulé une vision claire de termes comme service ou utilisateur, qui sont utilisés habituellement dans un flou sémantique certain, leur sens étant souvent différent à la fois selon la communauté de recherche (télécom, web, IT) et selon le métier de la personne qui les emploie (marketing service, marketing réseau, architecte logiciel, architecte technique, développeur de plateforme de service, développeur d'application de prise de commande ou de facturation...).

¹ Selon les mots de l'historien Pierre Rosanvallon dans [Rosanvallon, 2004]

² Ces vues pouvant bien sûr inclure des éléments spécifiques au service, non définis dans une vue de référence

Reste la question des invariants. Qu'il y a-t-il de suffisamment stable dans les architectures des services télécoms pour constituer une référence ? Ayant explicité cette question, souvent implicite dans les travaux existants sur les architectures de services dans la communauté télécom, nous avons proposé trois types d'invariants :

- des invariants langagiers décrivant les actions conjointes des parties-prenantes du service (typiquement l'utilisateur et le fournisseur de service) pour le rendre,
- des invariants stratégiques décrivant les grandes fonctions du ressort d'un opérateur télécom (communication, diffusion, configuration d'informations),
- des invariants techniques décrivant les protocoles et les modèles d'architecture technique (par exemple l'architecture dite *3-tiers*).

Nous avons ensuite construit la vue métier de référence avec les invariants langagiers, la vue fonctionnelle de référence en déclinant les invariants stratégiques selon les invariants langagiers, et la vue technique de référence avec les invariants techniques.

Notre ambition n'est pas tant d'établir une méthodologie pour généraliser, abstraire le foisonnement des services de communication que de monter l'intérêt et la possibilité de caractériser rigoureusement les fonctionnalités des services de communication pour distinguer et relier :

- une architecture fonctionnelle
- une architecture technique habituellement privilégiée par les architectes télécoms,
- et une architecture applicative habituellement privilégiée par les architectes informatiques.

Nous avons montré l'intérêt pour les services convergents d'une vue technique définie de façon plutôt lâche par rapport aux modèles télécoms en couches, les notions de rôle technique et de point de référence permettant de prendre en compte aussi bien des motifs techniques télécoms comme l'IMS que des motifs techniques IT comme les architectures SOA. La vue applicative est en conséquence assez efficace et novatrice car elle permet de représenter dans un formalisme clair le fonctionnement de bout en bout d'un service, en incluant à la fois les composants télécoms et IT, les organes réseau et les plates-formes de service, et en indiquant dans la même figure la façon dont ils contribuent à la réalisation du service par les opérations applicatives portées par des points de référence.

Ces résultats ne sont pas à considérer comme un tout inséparable, mais plutôt comme une boîte à outils, mobilisable dans des contextes divers et notamment dans le contexte de la convergence des services. Nous avons ainsi évoqué comment la vue métier de référence pouvait servir de support au marketing pour établir des parcours utilisateurs communs à différentes offres de services et comment la vue fonctionnelle pouvait servir à ce même marketing pour la conception et la formalisation d'offres de service. Nous avons indiqué que la vue fonctionnelle de référence ou la vue technique de référence étaient utilisables indépendamment pour aider à identifier des composants applicatifs visant la réutilisabilité, comme les enablers ou les services SOA, ainsi que pour aider à constituer une ontologie des services télécoms. Nous avons vu enfin que l'intégration de descriptions sémantiques du service partagées au sein des applications informatiques ouvrait la voie à des liens dynamiques entre ces applications, soit en réponse à la demande d'un utilisateur, soit à l'initiative des applications elles-mêmes ou de leur environnement d'exécution, cette perspective pouvant être étendue vers les applications de gestion des services.

En conclusion, retenons peut-être les trois points suivants :

- la bonne adéquation des quatre vues de l'urbanisme pour les services de communication, à condition de définir précisément ce que représentent les vues métier et fonctionnelle dans ce cadre ;
- l'intérêt de formaliser, dans la vue applicative, le lien entre protocoles (ou modèles d'architecture technique) et fonctionnalités de bout en bout dans le rendu desquelles ces protocoles sont impliqués, à travers les notions de composant applicatif et d'opération applicative, ces notions convenant particulièrement pour les services convergents car valables à la fois pour les composants télécoms et IT ;
- la possibilité et l'intérêt de déterminer des invariants dans les services télécoms, d'une part pour améliorer la conception des architectures et leur validation, d'autre part pour faciliter la communication entre services et la construction automatisée de services composés au sein d'un environnement de services convergents.

Ces travaux offrent plusieurs perspectives de recherche, dans différents domaines. Tout d'abord dans le domaine de l'ingénierie des modèles, dans le cadre de l'assistance à l'urbanisation des systèmes d'information (comme par exemple la démarche EA4UP, développée dans [Simonin, 2008]). En prolongement de l'outil de spécification du contour

d'une offre présenté à la section V.1.1, une suite possible de cette thèse est de proposer un outil directement utilisable par les parties-prenantes du développement des services télécoms (marketing, décideurs, architectes, développeurs...) pour concevoir et consulter l'architecture d'un service. Cet outil maintiendrait automatiquement une cohérence entre les différentes vues d'architecture d'un service, et avec les vues de référence, tout en proposant une assistance à la conception de ces vues. La réflexion s'orienterait au moins dans quatre directions :

- la recherche des modes de représentation les plus pertinents pour matérialiser les architectures de services, selon la vue et selon les parties-prenantes (on pourra par exemple évaluer l'intérêt des diagrammes d'activité à la place des diagrammes de séquence dans la vue fonctionnelle d'un service) ;
- l'outillage de la démarche et des notions présentées ici, en se basant sur les technologies de transformation de modèles (décrites par exemple dans [Belaunde, 2008]) ;
- la prise en compte des aspects non-fonctionnels comme le coût, la qualité de service, la fiabilité ; ces aspects ont été volontairement laissés de côté au cours de cette thèse, sachant qu'ils sont étudiés dans de nombreux autres travaux, mais leur intégration est nécessaire pour un outillage complet ;
- la recherche de métriques basées sur ces vues (à la suite de la section V.1.2), métriques qui seraient intégrées dans cet outil.

Ensuite dans le domaine de la contextualisation des services (*context awareness*), nos travaux pourraient être poursuivis par une recherche sur la corrélation entre les activités concrètes des utilisateurs et les opérations fonctionnelles SFO. Il s'agirait de détecter à partir de traces d'activités les opérations réalisées. Cette analyse pourrait être effectuée en temps différé, par exemple pour mesurer le taux d'usage des opérations fonctionnelles et aider à gérer leur cycle de vie. Elle pourrait également être réalisée en temps-réel pour caractériser l'activité courante de l'utilisateur à l'aide des opérations fonctionnelles SFO en train d'être réalisées. Cette caractérisation permettrait de lui proposer automatiquement des services pour poursuivre cette activité courante (un utilisateur sélectionnant un contact se verrait ainsi proposer un service d'email ou de téléphonie vers ce contact). Mais aussi d'améliorer sa joignabilité en déterminant automatiquement s'il est opportun ou non de l'interrompre durant cette activité (une mesure d'interruptibilité pourrait ainsi être définie). Ou encore en adaptant automatiquement ses services à l'activité en cours (la consommation d'un contenu audiovisuel

pourrait par exemple conduire à notifier visuellement plutôt qu'auditivement un appel téléphonique).

Dans le domaine des plates-formes de service, nos travaux pourraient conduire, en prolongement des sections V.2 et V.3, à mettre en place une description fonctionnelle des services basée sur une sémantique commune, en utilisant les mécanismes du Web sémantique. Comme nous l'avons discuté, ceci rendrait possible plusieurs types d'applications : par exemple, permettre à des utilisateurs de choisir, ou à des développeurs d'invoquer dans leur code, un service abstrait, caractérisé uniquement par des critères fonctionnels, la détermination du service concret à utiliser se faisant seulement au moment de l'exécution du service (au *runtime*). Ou encore, réaliser un environnement de services convergents, supportant la communication entre services, avec la vue fonctionnelle de référence comme sémantique pivot. Ici encore, les aspects non-fonctionnels devront être pris en compte.

Enfin, dans le domaine de l'ingénierie des services (*service science* ou SSME), une étude pourrait être menée pour évaluer la pertinence d'étendre les principes proposés ici à d'autres secteurs économiques que la fourniture de services de communication. Des secteurs où la notion de service rendu prime sur la notion de donnée à traiter, et où l'informatisation – donc la conception d'architectures – est nécessaire. Ainsi, dans le domaine médical, une coordination informatisée entre les services rendus par les différentes professions de santé devient peut-être impérative pour fournir au patient un "environnement de services convergent".

Bibliographie

Publications

Articles de revues internationales avec comité de lecture

- [2009-1] E. Bertin and N. Crespi, "Service Business Processes for the Next Generation of Services: a Required Step to Achieve Service Convergence," *Annals of Telecommunications*, Vol 64, No 3-4, Apr 2009, pp 187-196, Springer-Verlag
- [2007-1] E. Bertin, I. Grida Ben Yahia, N. Crespi, "Modeling IMS Services", *Journal of Mobile Multimedia*, Vol.3, 2007, pp 150-167, Rinton Press

Contributions à des ouvrages collectifs

- [2008-1] E. Bertin and N. Crespi, "Next Generation Communication Services: A usage perspective", *ServiceWave 2008*, LNCS Vol. 5377, pp86-97, 2008, Springer-Verlag
- [2008-2] E. Bertin and N. Crespi, "Chapter 1: IMS Service, Models and Concepts," in *IP Multimedia Subsystem (IMS) Handbook*, 2008, CRC Press
- [2007-2] A. Gouya, N. Crespi, E. Bertin, L. Oueslati, "Managing Service Capability and Service Feature Interactions in the IMS of UMTS", in "Business Models and Drivers for Next Generation IMS Service", 2007, International Engineering Consortium
- [2006-1] E. Bertin, N. Crespi, I. Grida Ben Yahia, article NGN in *Encyclopédie de l'informatique et des systèmes d'information*, ISBN 2711748464, Nov2006, Vuibert

Conférences internationales avec comité de lecture

- [2009-2] N. Laga, E. Bertin, and N. Crespi, "A web based framework for rapid integration of Enterprise applications," in *ACM International Conference on Pervasive Services*, Imperial College, London, UK, July 2009
- [2008-3] I. Grida Ben Yahia, E. Bertin, and N. Crespi, "Self-Managed Next Generation Services: How autonomic principles address the challenge of NGS management?" in *ICIN*, Bordeaux, October 2008

- [2008-4] N. Laga, E. Bertin, and N. Crespi, "A unique interface for web and telecom services: from feeds aggregator to services aggregator," in ICIN, Bordeaux, October 2008
- [2007-3] A. Gouya, N. Crespi E. Bertin, "Service Invocation Issues within the IP Multimedia Subsystem," in International Conference on Networking and Services, ICNS'07, Greece, Athens, June 2007
- [2007-4] I. Grida Ben Yahia, E. Bertin, N. Crespi, and J. Simonin, "Self Management Models for the Next Generation Service: Analysis Stage," in International Conference on Networking and Services, ICNS'07, Greece, Athens, June 2007
- [2007-5] I. Grida Ben Yahia, E. Bertin, and N. Crespi, "Ontology-based Management Systems for the Next Generation Services," in International Conference on Networking and Services, ICNS'07, Greece, Athens, June 2007
- [2006-2] I. Grida Ben Yahia, E. Bertin, N. Crespi (INT), JP.Deschrevel, "Service Definition for NGN " , International Conference on Networking ICN06, Mauritius, April 2006
- [2006-3] A. Gouya, N. Crespi, E. Bertin, "SCIM (Service Capability Interaction Manager) Implementation Framework in IMS " , IEEE International Conference on Communications ICC'06, Istanbul, Turkey, June 2006
- [2006-4] A. Gouya, N. Crespi, E. Bertin, L. Oueslati, "Managing Service Capability and Service Feature Interactions in the IMS of UMTS " , International Conference on Networking and Services ICNS'06, Silicon Valley, USA, July 2006
- [2006-5] E. Bertin, P. Lesieur, "Which architecture for integrated communication services ? " , International Conference on Networking and Services ICNS'06, Silicon Valley, USA, July 2006
- [2006-6] I. Grida Ben Yahia, E. Bertin, N. Crespi, "Towards autonomic management for NGN Services" , International Conference on Networking and Services ICNS'06, Silicon Valley, USA, July 2006
- [2006-7] I. Grida Ben yahia, E. Bertin, N. Crespi, "New/Next Generation Networks Services and Management" , International Conference on Networking and Services ICNS'06, Silicon Valley, USA, July 2006
- [2005-1] E. Bertin, "Stakes of next-generation communication services" , Advanced International Conference on Telecommunications AICT'05, Lisbon, Portugal, July 2005

- [2004-1] E. Bertin, E. Bury, P. Lesieur, "Operator Services Deployment with SIP: Trends of Telephony over IP", 3rd European Conference on Universal Multiservice Networks (ECUMN 04), Porto, October 2004
- [2004-2] E. Bertin, E. Bury, P. Lesieur, "Intelligence distribution in next-generation networks: An architectural framework for multimedia services", IEEE International Conference on Communications (ICC 2004), Paris, June 2004
- [2003-1] E. Bertin, E. Bury, P. Lesieur, "Operator service deployments with SIP: Wireline feedback and Wireless perspectives", 8th International Conference on Intelligence in Networks (ICIN 2003), Bordeaux, April 2003
- [2001-1] E. Bertin, E. Bury, P. Lesieur, "Next Generation Service Creation: a Comparison of Architectural Approaches", 7th International Conference on Intelligence in Networks (ICIN 2001), Bordeaux, October 2001

Workshops internationaux avec actes et comité de lecture

- [2009-3] N. Laga, E. Bertin, and N. Crespi, "Building a user friendly service dashboard: Automatic and non-intrusive chaining between widgets," in IEEE International Workshop on Web X.o in IEEE Congress on Services (SERVICES 2009), Los Angeles, USA, July 2009
- [2008-5] N. Laga, E. Bertin, and N. Crespi, "User-centric services and service composition, a survey", in Special Track on Service Engineering in a converging Telecommunications/Web 2.0 World at the 3rd International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (Isola 2008), Greece, Kassandra, October 2008
- [2007-6] K. Hamadache, E. Bertin, A. Bouchacourt, I. Ben Yahia, "Context-Aware Communication Services: an Ontology Based Approach," in International Workshop on Context Modeling and Management for Smart Environments (CMMSE'07) in Conjunction with the IEEE ACM Second International Conference on Digital Information Management ICDIM 2007, France, Lyon, October 2007
- [2007-7] E. Bertin, I. Fodil, N. Crespi, "Business view for NGN user relationship management", IEEE/IFIP International Workshop on Broadband Convergence Networks (BcN2007), In Conjunction with IM'07, Munich, May 2007

Brevets

- [2009-4] I. Bedini, E. Bertin, N. Laga, "Procédé d'exécution d'un service applicatif dans un environnement web", n° 09 54427, déposé le 29 juin 2009
- [2008-6] E. Bertin, JP Deschrevel, "Téléphonie en réseaux à modules séparés", n° 08 56321, déposé le 19 septembre 2008
- [2008-7] E. Bertin, JP Deschrevel, N. Laga, "Procédé et système de communication entre applications Web distinctes", n° 08 56592, déposé le 30 septembre 2008
- [2008-8] E. Bertin, JP Deschrevel, K. Henry, "Procédé et système d'accès par un utilisateur à au moins un service offert par au moins un autre utilisateur", n° 08 53442, déposé le 27 mai 2008
- [2007-8] E. Bertin, J. Van Den Bossche, JP Deschrevel, "Procédé et dispositif de communication entre applications Web distinctes", n° 07 55315, déposé le 29 mai 2007
- [2007-9] J. Simonin, JP Deschrevel, E. Bertin, "Procédé de détermination de données dédiées à un service de télécommunications", n° FR2007/052587 du 20.12.2007
- [2007-10] J. Simonin, JP Deschrevel, E. Bertin, "Procédé de détermination de quartiers fonctionnels service pour des services de télécommunications", n° FR2007/052601 du 21.12.2007
- [2005-2] E. Bertin, L. Lancieri, "Procédé et système d'accès à des informations relatives à une pluralité d'utilisateurs et mémorisées dans un module de stockage", n° 05 09607, déposé le 20 septembre 2005
- [2004-3] E. Bertin, L. Lancieri, "Procédé de traitement d'informations reçues et/ou demandées par au moins un utilisateur au sein d'un système de télécommunication", n° 04 08569, déposé le 3 août 2004
- [2004-4] S. Polouchkine, E. Bertin, "Procédé et système de gestion d'appels téléphoniques", n° 04 03385, déposé le 31 mars 2004
- [2003-2] S. Corre, V. Gouraud, E. Bertin, M. Letouzey, "Procédé et système de contrôle de communication multi-terminaux", n° 03 11928, déposé le 13 octobre 2003
- [2003-3] E. Bertin, C. Roussel, S. Polouchkine, "Dispositif et procédé de traitement de données de présence", n° 03 08423, déposé le 9 juillet 2003
- [2002-1] E. Bertin, E. Bury, M. Letouzey, S. Polouchkine, "Procédé et système d'acheminement de flux d'informations", n° 02 14688, déposé le 22 novembre 2002

- [2002-2] E. Bertin, I. Grida, C. Poirier, "Procédé et système d'avertissement et de diffusion d'informations par un réseau public de transmission de données numériques", n° 02 09216, déposé le 19 juillet 2002
- [2002-3] E. Bertin, M. Letouzey, "Routage d'appel personnalisé", n° 02 02976, déposé le 8 mars 2002

Autres communications

- [2008-9] E. Bertin and N. Crespi, "Composing services: services as music," presented in 4th International FOKUS IMS Workshop, Berlin, November 2008
- [2004-5] E. Bertin, E. Bury, P. Lesieur, "SIP and UMTS: Which architecture for global service deployments?", presented in 5th International SIP summit, Paris, January 2004
- [2003-4] E. Bertin, E. Bury, P. Lesieur, "SIP services in a distributed architecture : Feedback from implementation and trial", presented in 4th International SIP summit, Paris, January 2003
- [2002-4] E. Bertin, E. Bury, P. Lesieur, "Next generation architectures: which roles for an incumbent operator?", presented in Eurescom Annual Summit, Heidelberg, October 2002

Références

- [3GPP, 2001] 3GPP, TS 03.40, "Technical realization of the Short Message Service (SMS)", V7.5.0 (2001-12)
- [3GPP, 2007a] 3GPP, "Presence service using the IP Multimedia (IM) Core Network (CN) subsystem"; TS 24.141, version 7.4.0, sept 07
- [3GPP, 2007b] 3GPP, "Messaging using the IP Multimedia (IM) Core Network (CN) subsystem; TS 24.247, version 7.2.0, jun 07
- [3GPP, 2007c] 3GPP, "Conferencing using the IP Multimedia (IM) Core Network (CN) subsystem", TS 24.147, version 7.6.0, sept 07
- [3GPP, 2009] 3GPP TS 22.101, "Service principles (Release 8)", 2009
- [AFNOR, 1991] NF X50-151, "Analyse de la valeur – Analyse fonctionnelle – Expression fonctionnelle du besoin et cahier des charges fonctionnel", Dec. 1991
- [Alexander, 2004] I. Alexander, N. Maiden, "Scenarios, stories, and use cases: the modern basis for system development," Computing & Control Engineering Journal , vol.15, no.5, Oct.-Nov. 2004, pp. 24-29
- [Android, web] <http://developer.android.com/index.html>
- [Annett, 2004] J. Annett, "Hierarchical Task Analysis", in "The handbook of Task Analysis for Human-Computer Interaction", Lawrence Erlbaum Associates, Mahwah, New Jersey, 2004, pp 67-82
- [Anscombe, 1957] E. Anscombe, "L'intention", Paris, Gallimard, 2002 (1^{re} éd. : 1957)
- [Arbanowski, 2004] S. Arbanowski et al., "I-centric communications: personalization, ambient awareness, and adaptability for future mobile services", IEEE Communications Magazine, Volume 42, Issue 9, September 2004, pp 63-69
- [Arroyo, 2006] S. Arroyo, J.M. Lopez-Cobo, "Describing web services with semantic metadata", Int. J. Metadata Semant. Ontologies, Vol 1, Jan. 2006, pp. 76-82.
- [Asatani, 1996] K. Asatani, S. Nogami, "Trends in the standardization of telecommunications on GII, multimedia, and other network technologies and services," Communications Magazine, IEEE , vol.34, no.6, Jun. 1996
- [Bachmann, 2008] A. Bachmann, A. Motanga, T. Magedanz, "Requirements for an extendible IMS client framework", ACM International Conference Proceeding Series, vol. 278, Feb. 2008

- [Barcet, 1987] A. Barcet, "La Montée des services, vers une économie de la servuction", Thèse de doctorat, Université de Lyon II, 1987
- [Bedini, 2008] I. Bedini, G. Gardarin, B. Nguyen, "Deriving Ontologies from XML Schema", Proceedings 4èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2008), Toulouse, France, June 5 – 6, 2008
- [Belaunde, 2008] M. Belaunde and P. Falcarin, "Realizing an MDA and SOA Marriage for the Development of Mobile Services", Proceedings of the 4th European Conference on Model Driven Architecture: Foundations and Applications, Berlin, Germany, June 09 - 13, 2008
- [Berners-Lee, 2001] T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web", Scientific American Magazine, 2001
- [Bleistein, 2006] S.J. Bleistein , K. Cox , J. Verner, "Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams", Journal of Systems and Software, v.79 n.3, p.362-378, March 2006
- [Blum, 2008a] N. Blum, T. Magedanz, F. Schreiner, "Definition of a Service Delivery Platform for Service Exposure and Service Orchestration in Next Generation Networks", UbiCC Journal, Volume 3, Number 3, 2008
- [Blum, 2008b] N. Blum, S. Dutkowski, T. Magedanz, "InSeRt - An Intent-based Service Request API for Service Exposure in Next Generation Networks ", Special Track on "Service Engineering in a converging Telecommunications / Web 2.0 World" at the 3rd International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (Isola 2008), Greece, October 2008
- [Booch, 1999] G. Booch, J. Rumbaugh, I. Jacobson, "The Unified Modeling Language – User Guide". Addison – Wesley, 1999
- [Boussard, 2009] M. Boussard, S. Fodor, N. Crespi, V. Iribarren, J.P. Le Rouzic, I. Bedini, G. Marton, D. Moro Fernandez, O. Lorenzo Duenas, B. Molina "SERVERY: the web-telco marketplace," ICT-Mobile Summit 2009, Santander, Spain, June 2009
- [Braun, 2005] C. Braun, R. Winter, "A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform", in Enterprise Modelling and Information Systems Architectures, Gesellschaft für Informatik, Bonn, 2005, pp. 64-79

- [Broadbent, 2008] S. Broadbent, V. Bauwens, "Understanding convergence", ACM Interactions, Vol 15 No 1, pp. 23-27, 2008
- [Caetano, 2007] J. Caetano, P. Santos, P. Renditore, M. Demartini, P. Falcarin, C. Baladrón, J. Aguiar, B. Carro, "Introducing the user to the service creation world: concepts for user centric service creation, personalization and notification", 16th IST Mobile & Wireless Communications Summit, Budapest, Hungary, July 2007
- [Cai, 2007] H. Cai, JY Chung, H.Su, "Relooking at Services Science and Services Innovation", Proc. IEEE International Conference on e-Business Engineering (ICEBE'07), 2007, pp.427-432
- [Carugi, 2005] M. Carugi, B. Hirschman and A. Narita, "Introduction to the ITU-T NGN focus group release 1: target environment, services, and capabilities", IEEE Communication Magazine, Volume 43, Issue 10, October 2005, pp 42-48
- [Channabasavaiah, 2003] K. Channabasavaiah, K. Holley, and E. Tuggle, "Migrating to a Service-Oriented Architecture," IBM Developer-Works, 16 Dec. 2003
- [Chesbrough, 2006] H. Chesbrough, J. Spohrer, "A research manifesto for services science", Communications of the ACM, Vol 49 No 7, Jul. 2006, pp 35-40
- [Child, 2001] J. Child, R. G. McGrath, "Organizations unfettered: Organizational form in an information-intensive economy", Acad. Management Journal, Vol 44 No 6, 2001, pp. 1135-1148.
- [Cochennec, 2002] JY Cochennec, "Activities on next-generation networks under Global Information Infrastructure in ITU-T" Communications Magazine, IEEE , vol.40 no.7, Jul 2002, pp.98-101
- [Cockburn, 1997] A. Cockburn, "Goals and Use Cases", JOOP vol 10 No 5, 1997, pp. 35-40
- [Contini, 2002] I. Contini, "L'apport historique de l'urbanisme des villes pour l'urbanisme des systèmes d'information", Colloque Urbanisation des Systèmes d'Information, Paris, 2002.
- [Davidson, 1963] D. Davidson, "Actions, Reasons, and Causes", Journal of Philosophy, Vol 60, 1963
- [Day, 2006] G.S. Day, "Aligning the Organization with the Market", MIT Sloan Management Review, Vol. 48, No. 1, Fall 2006, pp. 41-49.

- [Delaunay, 1987] JC Delaunay, J. Gadrey, "Les enjeux de la société de service", Presses de Sciences Po, Paris, 1987
- [DoDAF, web] DoDAF, "Department of Defense Architecture Framework", http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf
- [Dubberly, 2008] H. Dubberly, S. Evenson, "On modeling the experience cycle", ACM Interactions, Vol 15 No 3, May. 2008, pp. 11-15
- [eTOM, web] TMForum, "enhanced Telecom Operations Map", <http://www.tmforum.org/browse.aspx?catID=1648>
- [ETSI, 2005] ES 282 001, " Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN Functional Architecture Release 1", 2005
- [ETSI, 2006] TISPAN TR 181 004, "NGN Generic capabilities and their use to develop services", 2006
- [Fielding, 2000] R. T. Fielding and R. N. Taylor, "Principled Design of the Modern Web Architecture", Proceedings of the 22nd international conference on Software engineering, June 04-11, 2000, Limerik, Ireland, pp.407-416
- [Flichy, 2008] P. Flichy, "Technique, usage et représentations", revue Réseaux, n°148-149, volume 26, Lavoisier, 2008
- [Ghadiri, 2007] N. Ghadiri, M. A. Nematbakhsh, A. Baraani-Dastjerdi, N. Ghasem-Aghae, "A Context-Aware Service Discovery Framework Based on Human Needs Model", ICSOC 2007, pp. 404-409
- [Griffin, 2007] D. Griffin and D. Pesch, "A Survey on Web Services in Telecommunications", Communications Magazine, IEEE , vol.45, no.7, July 2007, pp.28-35
- [Gronroos, 2007] C. Grönroos, "In Search of New Logic for Marketing. Foundations of Contemporary Theory", John Wiley & Co., Chichester, West Sussex, England, January 2007
- [Gruber, 1993] T. R. Gruber, "Towards Principles for the Design of Ontologies Used for Knowledge Sharing in Formal Ontology in Conceptual Analysis and Knowledge Representation", Kluwer Academic Publishers, 1993

- [Henderson, 1993] J.C. Henderson, N. Venkatraman, "Strategic Alignment: Leveraging Information Technology for Transforming Organizations", IBM Systems Journal, vol. 32, No. 1, 1993, pp. 4-16
- [Hepp, 2007] M. Hepp, "Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies", IEEE Internet Computing Vol 11 No1, 2007, pp. 90-96
- [Hesselman, 2008] C. Hesselman, P. Cesar, I. Vaishnavi, M. Boussard, R. Kernchen, S. Meissner, A. Spedalieri, A. Sinfreu, C. Räck, "Delivering interactive multimedia services in dynamic pervasive computing environments", Proceedings of the 1st international conference on Ambient media and systems, February 11-14, 2008, Quebec, Canada, p.1-8
- [IEEE, 2000] IEEE, Standard 1471 "Recommended Practice for Architectural Description of Software-Intensive Systems", September 2000
- [ILO, 2008] Global Employment Trends: January 2008, International Labour Office, 2008
- [Inoue, 1998] Y. Inoue, D. Cuha, H. Berndt, "The TINA Consortium," Communications Magazine, IEEE , vol.36, no.9, Sep 1998, pp.130-136
- [ITAA, web] <http://www.ita.org/>
- [ITIL, web] ITIL, "The IT Infrastructure Library", <http://www.itil-officialsite.com/home/home.asp>
- [ITU, 1992a] ITU-T Q.1201, "Principles of intelligent network architecture", 1992
- [ITU, 1992b] ITU-T Z.100, "Specification and Description Language (SDL)", 1992
- [ITU, 1993a] ITU-T Q.1204, "Intelligent network – Distributed functional plane architecture", 1993
- [ITU, 1993b] ITU-T Q.1205, "Intelligent network – Physical plane architecture", 1993
- [ITU, 1993c] ITU-T Q.1211, "Introduction to intelligent network capability set 1", 1993
- [ITU, 1997a] ITU-T Q.1202, "Intelligent network – Service plane architecture", 1997
- [ITU, 1997b] ITU-T Q.1203, "Intelligent network – Global functional plane architecture", 1997
- [ITU, 1998] ITU-T Y.110, " Infrastructure mondiale de l'information: Principes et architecture générale", 1998
- [ITU, 2008] ITU-T Y.1910, "IPTV Functional Architecture", 2008.
- [Ivers, 2004] J. Ivers, P. Clements, D. Garlan, R. Nord, B. Schmerl and J.R.O. Silva, "Documenting Component and Connector Views with UML 2.0" Technical report CMU/SEI-2004-TR-008, Carnegie Mellon University, April 2004

- [Kaisler, 2005] S.H. Kaisler, F. Armour, M. Valivullah, "Enterprise Architecting : Critical Problems", International Conference on System Sciences, Hawai, 2005
- [Karunamurthy, 2007] R. Karunamurthy, F. Khendek, R.H. Glitho, "A Business Model for Dynamic Composition of Telecommunication Web Services", IEEE Communications Magazine, vol.45, no.7, July 2007, pp.36-43
- [Kaufmann, 2005] L. Kaufmann, F. Clément, "Le monde selon John Searle", Paris, Cerf, 2005
- [Keck, 1998] D. O. Keck and P. J. Kuehn, "The Feature and Service Interaction Problem in Telecommunications Systems: A Survey", IEEE Transactions on Software Engineering, Vol. 24, No 10, Oct. 1998, pp 779-796
- [Kellert, 2006] P. Kellert, F. Toumani, "Les web services sémantiques", Revue I2 (Information-Interaction-Intelligence), Paris, France, 2006
- [Kellogg, 2006] K.C. Kellogg, W.J. Orlikowski, J. Yates, "Life in the Trading Zone: Structuring Coordination across Boundaries in Postbureaucratic Organizations", Organization Science journal, Vol 17 No 1, Jan. 2006, pp. 22-44.
- [Knightson, 2005] K. Knightson, N. Morita and T. Towle, "NGN architecture: generic principles, functional architecture, and implementation", IEEE Communications Magazine, Volume 43, No 10, October 2005, pp 49-56
- [Kotler, 1988] P. Kotler, "Marketing Management: Analysis, Planning, Implementation and Control", 6th edition. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [Kruchten, 1995] P. Kruchten, "Architectural Blueprints – The "4+1" View Model of Software Architecture", IEEE Software, vol. 12, No.6, 1995, pp. 42-50
- [Kwan, 2008] S.K. Kwan, J.H. Min, "An Evolutionary Framework of Service Systems", International Conference on Service Science (ICSS 2008), April 2008, Beijing, China
- [Lamb, 2003] R. Lamb, R. Kling, "Reconceptualizing Users as Social Actors in Information Systems Research", Management Information Systems Quarterly, Vol. 27 No 1, Article 2, June 2003
- [Lavinal, 2009] E. Lavinal, N. Simoni, M. Song, "A next-generation service overlay architecture", Annals of telecommunications, Vol 64, No 3-4, Apr 2009, Springer-Verlag
- [Lécué, 2009] F. Lécué, O. Boissier, A. Delteil, A. Léger, "Web Service Composition as a Composition of Valid and Robust Semantic Links", International Journal of Cooperative Information Systems (IJCIS), Vol 18 No 1, March 2009, World Scientific, 2009

- [Liberty, web] <http://www.projectliberty.org/>
- [Liu, 2006] L. Liu; C. Chi; J. Jin, E. Yu, "Strategic Capability Modelling of Services", 2nd International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements, SOCCER '06, Minneapolis, USA, Sept. 2006
- [Lizcano, 2008] D. Lizcano, J. Soriano, M. Reyes, J.J. Hierro, "EzWeb/FAST: Reporting on a Successful Mashup-Based Solution for Developing and Deploying Composite Applications in the Upcoming "Ubiquitous SOA"", International Conference on Mobile Ubiquitous Computing Systems Services and Technologies, UBICOMM '08, Sept. 29 2008-Oct. 4 2008, Valencia, Spain, pp.488-495
- [Lo, 2007] A. Lo, E. Yu, "From Business Models to Service-Oriented Design: A Reference Catalog Approach", ER 2007, LNCS 4801, Springer-Verlag, pp. 87—101
- [Lombard, 2008] D. Lombard, "Le village numérique mondial – La deuxième vie des réseaux", Odile Jacob, Paris, Avril 2008
- [Longépé, 2006] C. Longépé, "Projet d'urbanisation du système d'information – Démarche pratique avec cas concret", 3è édition, Dunod, Paris, 2001
- [Lozano, 2008] D. Lozano, L.A. Galindo, L. Garcia, "WIMS 2.0: Converging IMS and Web 2.0. Designing REST APIs for the Exposure of Session-Based IMS Capabilities," International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST '08, 16-19 Sept. 2008, pp.18-24
- [Lusch, 2006] R. F. Lusch, S. L. Vargo, "Service-Dominant Logic: Reactions, Reflections and Refinements", Marketing Theory, Vol 6 No. 3, 2006, pp. 281-288
- [Lusch, 2008] R. F. Lusch , S. L. Vargo , G. Wessels, "Toward a conceptual foundation for service science: contributions from service-dominant logic", IBM Systems Journal, v.47 no1, January 2008, p.5-13
- [Magedanz, 2007] T. Magedanz, N. Blum, S. Dutkowski, "Evolution of SOA Concepts in Telecommunications – A Déjà vu?", Special Issue on Service Oriented Architectures, IEEE Computer, Vol 40 No 11, Nov. 2007, pp. 46-50
- [Maghmoumi, 2008] C. Maghmoumi, T.A. Andriatrimoson, J. Gaber, P. Lorenz, "A Service Based Clustering Approach for Pervasive Computing in Ad Hoc Networks", Global Telecommunications Conference, IEEE GLOBECOM 2008, New Orleans, USA, Nov. 30 2008-Dec. 4 2008

- [Maglio, 2006] P. Maglio, S. Srinivasan, J.T. Kreulen, J. Spohrer, "Service systems, service scientists, SSME, and innovation", Communications of the ACM, Vol 49 No 7, Jul. 2006, pp. 81-85
- [Margaria, 2008] T. Margaria, "The Semantic Web Services Challenge: Tackling Complexity at the Orchestration Level", 13th IEEE International Conference on Engineering of Complex Computer Systems, ICECCS 2008, Belfast, UK, March 31 2008-April 3 2008, pp.183-189
- [Menai, 2005] M.F. Menai, G. Fromentoux, D. Gäiti, M. Lemerrier, "A UML Tool for Urbanism and Control Architecture Design Applied to 3GPP Based Architectures", Proceedings of the international Tele-traffic Congress, vol. 6-b, 2005, pp 1987-1996
- [Mentzas, 2007] G. Mentzas, K. Kafentzis, P. Georgolios, "Knowledge services on the semantic web", Communications of the ACM, Vol 50 No 10, Oct. 2007, pp. 53-58
- [Metcalf, 2001] J.S. Metcalfe, "Modern evolutionary economic perspectives: An overview", in "Frontiers of Evolutionary Economics", Edward Elgar, 2001
- [Microformats, web] <http://microformats.org/>
- [Moerdijk, 2003] AJ. Moerdijk, L. Klostermann, "Opening the networks with Parlay/OSA: standards and aspects behind the APIs", IEEE Network, Vol.17 No.3, May-June 2003, pp. 58-64
- [Morley, 2003] C. Morley, B. Defude, F. Butelle, D. Lang, "Enabling tools for e-commerce", Annals of telecommunications, Vol. 58 No 1-2, Jan.-Feb 2003
- [Morre, 1998] B.W. Morre, "The ITU's role in the standardization of the GII," IEEE Communications Magazine, vol.36, no.9, Sep 1998, pp.98-106
- [Mulliez, 1994], G. Mulliez, R. Whiteley, " La dynamique du client - Une révolution des services", Maxima, Paris, 1994
- [Naics, 2007] North American Industry Classification System 2007 (Naics), US Dept. of Commerce, Sept 2007
- [Najm, 1999] E. Najm, F. Olsen, S. Vignes, "From SIBS to Distributed Objects: A Transformation Approach for Service Creation", IFIP Fifth International Conference on Intelligence in Networks, SMARTNET'99, Pathumthani, Thailand, November 22-26, 1999, pp. 403-417

- [Nuffel, 2007] D. Nuffel, "Towards a Service-Oriented Methodology: Business-Driven Guidelines for Service Identification", On the Move to Meaningful Internet Systems 2007, OTM 2007, Vilamoura, Portugal, Nov. 25-30 2007, Springer
- [Nygaard, 1986] K. Nygaard, "Program Development as a Social Activity", INFORMATION PROCESSING 86, Elsevier Science, IFIP, 1986 (Proceedings from the IFIP 10th World Computer Congress, Dublin, Ireland, September 1-5, 1986), pp. 189-198
- [O'Connell, 2008] J. O'Connell, "Federation of User Data for Multi-Domain Blended Services", ICIN conference, Bordeaux, October 2008
- [O'Brien, 1997] J. O'Brien, "Les systèmes d'information de gestion", Ed. De Boeck Université, 1997.
- [OMA, 2007a] OMA, "Dictionary for OMA specifications", Approved Version 2.6, Jun. 2007, OMA-ORG-Dictionary-V2_6-20070614-A
- [OMA, 2007b] OMA, "OMA Service Environment", Approved Version 1.0.4, 01 Feb 2007, OMA-AD-Service-Environment-V1_0_4-20070201-A
- [OMA, web] <http://www.openmobilealliance.org/>
- [Parlay, web] <http://portal.etsi.org/docbox/TISPAN/Open/OSA/ParlayX30.html>
- [Patel-Schneider, 2007] P.F. Patel-Schneider, I. Horrocks, "A comparison of two modelling paradigms in the Semantic Web", Web Semantic, Vol 5 No 4, Dec. 2007, pp. 240-250
- [Pinhanez, 2008] C. Pinhanez, "Service Systems as Customer-Intensive Systems and Its Implications for Service Science and Engineering", Proceedings of the 41st Annual Hawaii international Conference on System Sciences, January 07 - 10, 2008, IEEE Computer Society, Washington, USA
- [Pressman, 1992] R.S. Pressman, "Software Engineering: A Practitioner's Approach", Third Edition. McGraw-Hill. Chapitre 10, 1992
- [Raskin, 2000] J. Raskin, "The Humane Interface – New directions for Designing Interactive Systems", ACM Press, Addison-Wesley, 2000
- [Ricoeur, 1990] P. Ricoeur, "Soi-même comme un autre", Le Seuil, Paris, 1990
- [Roche, 2007] C. Roche, "Saying is not modelling", NLPCS 2007 (Natural Language Processing and Cognitive Science), ICEIS 2007, Funchal, Portugal, June 2007, pp. 47-56
- [Rolland, 2000] C. Rolland, N. Prakash, "From conceptual modelling to requirements engineering", Annals of Software Engineering, Vol 10 No 1-4, 2000, pp.151-176

- [Rolland, 2009] C. Rolland, C. Souveyet, "Service Oriented Computing: an Intentional Approach", to appear in IEEE Transactions on Service Computing, Special issue on REFS (Requirements Engineering for Services), IEEE, 2010
- [Rosanvallon, 2004] P. Rosanvallon, "Le Modèle politique français, La société civile contre le jacobinisme de 1789 à nos jours", Paris, Le Seuil, 2004, p.123
- [Rust, 2006] R.T. Rust, C. Miu, "What academic research tells us about service", Communications of the ACM, Vol 49 No 7, Jul. 2006, pp. 49-54
- [Salinesi, 2008] C. Salinesi, L.H. Thévenet, "Enterprise Architecture, des problèmes pratiques à l'innovation", Ingénierie des systèmes d'information, vol. 13, Lavoisier, 2008, pp. 75-105.
- [Sassoon, 1998] J. Sassoon, "Urbanisation des systèmes d'information", Hermès, 1998.
- [Schekkerman, 2006] J. Schekkerman, "How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework", Trafford Publishing, 2003
- [Schreiner, 2008] F. Schreiner, N. Blum, P. Jacak, P. Weik, T. Magedanz, "Towards standardized NGN OSS mechanisms for automated service provisioning and fault management for OSIMS-based NGNs", Journal of Network and Systems Management, Vol 16 no 1, March 2008, Springer, pp. 63-91
- [Searle, 2005] J. R. Searle, "What is an institution?", Journal of Institutional Economics, Volume 1, Vol 1, June 2005, pp 1-22
- [SEI, web] <http://www.sei.cmu.edu/>
- [Seryery, web] <http://projects.celtic-initiative.org/seryery/>
- [Shah, 2006] D. Shah, RT. Rust, A. Parasuraman, R. Staelin, G.S. Day, "The Path to Customer Centricity", Journal of Service Research, Vol 9 No 2, 2006, pp.113-124
- [Shiaa, 2008] M. Shiaa, P. Falcarin, A. Pastor, F. Lécué, E. Silva, LF. Pires, "Towards the Automation of the Service Composition Process: Case Study and Prototype Implementations", Proceedings of the ICT-Mobile Summit 2008, June 10-12, 2008, Stockholm, Sweden, pp. 1-8
- [Simoni, 2007] N. Simoni, "Des réseaux intelligents à la nouvelle génération de services", Lavoisier, Paris, Feb. 2007

- [Simonin, 2004] J. Simonin, F. Alizon, "Démarche d'urbanisation métier de type organisationnel", 17th International Conference Software and Systems Engineering and their Applications, Paris, France, 2004
- [Simonin, 2007] J. Simonin, Y. Le Traon, J.M. Jézéquel, "An Enterprise Architecture Alignment Measure for Telecom Service Development", Proceedings of the 11th IEEE International EDOC Conference, Annapolis, USA, 2007, pp. 476-483.
- [Simonin, 2008] J. Simonin, F. Alizon, J.P. Deschrevel, Y. Le Traon, J.M. Jézéquel, B. Nicolas, "EA4UP: an Enterprise Architecture-Assisted Telecom Service Development Method", Proceedings of the 12th IEEE International EDOC Conference, München, Germany, 2008
- [Simonin, 2009] J. Simonin, "Conception de l'architecture d'un système dirigée par un modèle d'urbanisme fonctionnel", Thèse de doctorat, Université de Rennes 1, janvier 2009
- [SPICE, web] <http://ontology.ist-spice.org/index.html>
- [Spohrer, 2007] J. Spohrer, P. Maglio, J. Bailey, D. Gruhl, "Steps Toward a Science of Service Systems", IEEE Computer, vol. 40 No. 1, Jan. 2007, pp. 71-77
- [Spohrer, 2008] J. Spohrer, S.L. Vargo, N. Caswell, P. Maglio, "The Service System Is the Basic Abstraction of Service Science", Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 7-10 Jan. 2008, pp.104-104
- [Tarkoma, 2007] S. Tarkoma, B. Bharat, E. Kovacs, H. van Kranenburg, E. Postmann, R. Seidl, A. Zhdanova, "SPICE: A Service Platform for Future Mobile IMS Services", Proceedings of IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), Helsinki, Finland, June 2007
- [Tien, 2003] J.M. Tien, D. Berg, "A Case for Service Systems Engineering", Journal of Systems Science and Systems Engineering, Vol 12 No 1, Mar. 2003, pp.113-128
- [Tien, 2008] J.M. Tien, "Services: A System's Perspective," IEEE Systems Journal, Vol 2 No 1, March 2008, pp.146-157
- [TOGAF, web] TOGAF, "The Open Group Architecture Framework", <http://www.opengroup.org/togaf/>.
- [UML, web] UML, "Unified Modelling Language", <http://www.omg.org/uml/>
- [Urba-EA, 2006] Club Urba-EA, "Urbanisme des SI & gouvernance", Dunod, 2006
- [Vargo, 2004] S. L. Vargo, R. F. Lusch, "Evolving to a New Dominant Logic for Marketing", Journal of Marketing, Vol 68 No 1, 2004, pp. 1-17

- [Verdot, 2009] V. Verdot, M. Boussard, N. Bouché, S. Shanmugalingam, L. Fournigault, "The Bridging of Two Worlds: A Web-IMS Communication Solution", The Fifth IASTED European Conference on Internet and Multimedia Systems and Applications (EuroIMSA 2009), Cambridge, UK, July 2009
- [Vu Duong, 2003] T. Vu Duong, G. Fromentoux, JL. Le Roux, "A global framework for architecture analysis in telecommunications", IEEE Global Telecommunications Conference GLOBECOM '03, 1-5 Dec. 2003, pp. 2896-2902
- [W3C, web] <http://www.w3.org/TR/ws-gloss/>
- [Whiteley, 1991] R. Whiteley, "The customer-driven company: Moving from talk to action", Perseus Books, 1991
- [Yelmo, 2008] J.C. Yelmo, J. M. Del Alamo, R. Trapero, P. Falcarin, J. Yu, B. Carro, C. Baladron, "A User-Centric Service Creation Approach for Next Generation Networks", Proceedings of ITU-T/IEEE Innovations in NGN Future Network and Services Conference, Geneva, Switzerland, May 2008
- [Zachman, 1987] J. A. Zachman, "A Framework for Information Systems Architecture", IBM Systems Journal, Vol 26, No. 3, 1987, pp. 276-292
- [Zaha, 2008] J.M. Zaha, M. Dumas, A.H.M. Hofstede, A. Barros, and G. Decker, "Bridging Global and Local Models of Service-Oriented Systems", IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, Vol 38 No 3, May 2008, pp. 302-318
- [Zeithaml, 1985] V. Zeithaml, A. Parasuraman, and L. L. Berry, "Problems and Strategies in Services Marketing", Journal of Marketing, Vol 49, Spring 1985, pp. 33-46
- [Zeithaml, 1996] V. Zeithaml, M.J. Bitner, "Services Marketing", McGraw-Hill, New York, 1996
- [Zeithaml, 2006] V. Zeithaml, M.J. Bitner, D. Gremler, "Services Marketing: Integrating Customer Focus Across the Firm", 2006, 4th edition, McGraw-Hill, New York, USA
- [Zeng, 2001] H. Zeng, T.C. Son, "Semantic Web Services", IEEE Intelligent Systems, Vol 16 No 2, March 2001, pp. 46-53
- [Zhixiong, 2007] J. Zhixiong, Q. Leqiu, P. Xin, "A Formal Framework for Description of Semantic Web Services", Proceedings of the 7th IEEE international Conference on Computer and information Technology, October 16 - 19, 2007, IEEE Computer Society, Washington, USA, pp. 1065-1070

Annexes

Acronymes des vues fonctionnelle, technique et applicative

GFD	Generic Functional Data
GFO	Generic Functional Operation
RP	Reference Point
SAC	Service Applicative Component
SAD	Service Applicative Data
SAO	Service Applicative Operation
SEF	Service Elementary Function
SFC	Service Functional Component
SFD	Service Functional Data
SFO	Service Functional Operation
TR	Technical Role

Autres acronymes

AFNOR	Association française de normalisation
API	Application Programming Interface
AS	Application Server
BPEL	Business Process Execution Language
BPML	Business Process Modeling Language
CAMEL	Customized Applications for Mobile Network
CSCF	Call Session Control Function
DoDAF	Department of Defense Architecture Framework
DRM	Digital Rights Management
EA	Enterprise Architecture
eTOM	Enhanced Telecom Operation Map
ETSI	European Telecommunications Standards Institute
FAST	Functional Analysis System Technique
GII	Global Information Infrastructure
GSM	Global System for Mobile communications
GUI	Graphical User Interface
HSS	Home Subscriber Server
IDP	Identity Provider
IETF	Internet Engineering Task Force
IHM	Interface Homme-Machine
IMS	IP Multimedia Subsystem
IN	Intelligent Network
INCM	Intelligent Network Conceptual Model
InSerT	Intent-based Service Request
IPTV	IP Television
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITU	International Telecommunication Union
MDA	Mail Delivery Agent
MMS	Multimedia messaging service

MS	Message Store
MS	Mobile Station
MSA	Mail Submission Agent
MTA	Mail Transfer Agent
MUA	Mail User Agent
NAB	Network Address Book
NGN	Next Generation Network
NGS	Next Generation Services
OMA	Open Mobile Alliance
OMG	Object Management Group
OSA	Open Service Access
OWL	Ontology Web Language
OWL-S	OWL for Services
PFS	Plate-forme de service
PLU	Plan Local d'Urbanisme
RCS	Rich Communication Suite
RDF	Resource Description Framework
RI	Réseau Intelligent
RTC	Réseau Téléphonique Commuté
RTP	Real Time Protocol
SADT	Structured Analysis and Design Technique
SAWSDL	Semantic Annotations for WSDL
SCIM	Service Capability Interaction Management
S-CSCF	Serving Call Session Control Function
SDP	Service Delivery Platform
SI	Système d'information
SIB	Service Independent Building Block
SIM	Subscriber Identity Module
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SME	Short Message Entity

SMS	Short Message Service
SMSC	SMS Center
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SPD	Session Description Protocol
SSME	Service Science, Management and Engineering
STB	Set-Top Box
TINA-C	Telecommunications Information Networking Architecture Consortium
TMF	TeleManagement Forum
TOGAF	The Open Group Architecture Framework
UE	User Equipment
UGS	User Generated Services
UML	Unified Medeling Language
URI	Uniform Resource Identifier
VOD	Video On Demand
W3C	World Wide Web Consortium
WS	Web Service
WSC	Web Service Consumer
WSD	Web Service Directory
WSDL	Web Service Description Language
WSP	Web Service Provider
XDM	XML Document Management
XDMC	XDM Client
XDMS	XDM Server

Table des figures

Figure 1 : Séparation entre service et accès	20
Figure 2 : Structure en éléments de service.....	22
Figure 3 : structure en SIB	23
Figure 4 : Architecture de session TINA-C	25
Figure 5 : Typologie des fonctions GII	26
Figure 6 : Architecture de référence des services web.....	33
Figure 7 : Principe d'une place de marché dans le projet CELTIC Servery.....	37
Figure 8 : Points de vue du cadre 4+1	44
Figure 9 : Les quatre vues d'un système d'information.....	47
Figure 10 : Cadre de référence eTOM	49
Figure 11 : Architecture NGN.....	55
Figure 12 : La convergence de réseau	58
Figure 13 : La convergence de service.....	59
Figure 14 : Approche standardisée de la production de service.....	65
Figure 15 : Approche individualisée de la production de service	66
Figure 16 : Approche front-office / back-office de la production de service.....	67
Figure 17 : Démarche d'analyse fonctionnelle	69
Figure 18 : Le service comme fourniture d'un bien immatériel	72
Figure 19 : Le service comme système	72
Figure 20 : Environnement de service IMS	73
Figure 21 : Environnement de service SOA.....	74
Figure 22 : Environnement de services convergents SOA + IMS	74
Figure 23 : Environnement de services convergents sans couture.....	75
Figure 24 : Déroulement d'un service télécom.....	79
Figure 25 : Fonctions techniques et fonctions de service pour les services télécoms.....	80
Figure 26 : Les quatre dimensions d'un service télécom.....	82
Figure 27 : Méthode de description de l'architecture d'un service télécom	83
Figure 28 : Processus d'un service postal	96
Figure 29 : Processus et entités de service	98
Figure 30 : Relations entre entités de service.....	98
Figure 31 : Méta-modèle de la vue métier de référence.....	99

Figure 32 : Processus d'envoi et de consultation de messages.....	102
Figure 33 : Relations entre entités du service de messagerie.....	103
Figure 34 : Processus d'émission et de réception d'appels.....	104
Figure 35 : Relations entre entités du service de téléphonie.....	105
Figure 36 : Processus de sélection et de jeu d'un média.....	106
Figure 37 : Relations entre entités du service audiovisuel.....	107
Figure 38 : Processus d'accès à un service.....	108
Figure 39 : Relations entre entités d'accès à un service.....	109
Figure 40 : Notions de SEF, SFO et SFD.....	114
Figure 41 : Méta-modèle de la vue fonctionnelle de référence.....	115
Figure 42 : Notion de motif fonctionnel.....	119
Figure 43 : Motif fonctionnel de communication.....	120
Figure 44 : Motif fonctionnel de diffusion.....	120
Figure 45 : Motif fonctionnel de configuration.....	121
Figure 46 : Lien entre vue fonctionnelle de référence et vue métier de référence.....	124
Figure 47 : Lien avec les motifs fonctionnels.....	125
Figure 48 : Modification d'un processus de service.....	127
Figure 49 : Construction d'une opération fonctionnelle et de la donnée fonctionnelle produite	128
Figure 50 : Construction d'une donnée fonctionnelle utilisée.....	129
Figure 51 : Amélioration du processus de sélection et de jeu d'un média.....	135
Figure 52 : Opérations offertes par <i>SFC Phone call</i>	142
Figure 53 : Opérations offertes par <i>SFC Message</i>	143
Figure 54 : Opérations offertes par <i>SFC Content</i>	144
Figure 55 : Quartier Communication setup.....	147
Figure 56 : Quartier Address book.....	147
Figure 57 : Partition et décomposition en couches.....	153
Figure 58 : Méta-modèle de la vue technique de référence.....	154
Figure 59 : Motif technique IMS.....	156
Figure 60 : Motif technique SOA.....	158
Figure 61 : Motif technique 3-tiers.....	160
Figure 62 : Motif technique Email.....	162
Figure 63 : Motif technique Email d'après l'IETF.....	163

Figure 64 : Motif technique SMS.....	165
Figure 65 : Motif technique OMA XDM.....	167
Figure 66 : Motif technique IPTV.....	168
Figure 67 : Motif technique IHM.....	170
Figure 68 : Cas d'utilisation d'un service de carnet d'adresses.....	175
Figure 69 : Séquence fonctionnelle du scénario de gestion de contacts.....	178
Figure 70 : Séquence fonctionnelle du scénario de consultation du carnet d'adresses.....	179
Figure 71 : Méta-modèle de la vue fonctionnelle d'un service.....	180
Figure 72 : Dépendances entre composants fonctionnels d'un service de carnet d'adresses..	181
Figure 73 : Cas d'utilisation d'un service de téléphonie.....	182
Figure 74 : Séquence fonctionnelle du scénario d'appel téléphonique (simplifiée).....	183
Figure 75 : Séquence fonctionnelle du scénario d'appel téléphonique (complète).....	185
Figure 76 : Cas d'utilisation d'un service de vidéo.....	186
Figure 77 : Séquence fonctionnelle du scénario de consommation de vidéo (simplifiée).....	186
Figure 78 : Séquence fonctionnelle du scénario de consommation de vidéo (complète).....	188
Figure 79 : Cas d'utilisation d'un service de messagerie email.....	189
Figure 80 : Séquence fonctionnelle du scénario de consultation d'emails (simplifiée).....	189
Figure 81 : Séquence fonctionnelle du scénario consultation d'emails (complète).....	191
Figure 82 : Séquence fonctionnelle du scénario VideoMe (simplifiée).....	193
Figure 83 : Séquence fonctionnelle du scénario VideoMe (complète).....	194
Figure 84 : Méta-modèle de la vue technique d'un service.....	197
Figure 85 : Vue technique d'un service de carnet d'adresses.....	199
Figure 86 : Vue technique d'un service de téléphonie.....	200
Figure 87 : Vue technique d'un service de vidéo.....	202
Figure 88 : Vue technique d'un service de messagerie email.....	204
Figure 89 : Vue technique du service VideoMe.....	206
Figure 90 : Méta-modèle de la vue applicative d'un service.....	213
Figure 91 : Vue applicative d'un service de carnet d'adresse.....	215
Figure 92 : Exemple d'implémentation applicative de <i>SFO issue phone call</i>	217
Figure 93 : Vue applicative d'un service de téléphonie.....	218
Figure 94 : Séquence applicative du scénario d'appel téléphonique.....	219
Figure 95 : Exemple d'implémentation applicative de <i>SFO browse content</i>	221
Figure 96 : Vue applicative d'un service de vidéo.....	222

Figure 97 : Séquence applicative du scénario de consommation de vidéo	223
Figure 98 : Exemple d'implémentation applicative de <i>SFO browse message</i>	224
Figure 99 : Vue applicative d'un service de messagerie.....	226
Figure 100 : Séquence applicative du scénario de consultation d'emails.....	227
Figure 101 : Vue applicative du service VideoMe.....	230
Figure 102 : Contour fonctionnel d'une offre d'email dans l'outil.....	235
Figure 103 : Exemple de descripteur.....	236
Figure 104 : Vue technique d'un environnement de services convergents SOA+IMS	250
Figure 105 : Vue technique d'un environnement de services convergents sans couture (provisoire).....	252
Figure 106 : Vue technique d'un environnement de services convergents sans couture (finale)	253
Figure 107 : Exemple d'interface d'accès unifié aux services	255
Figure 108 : Widgets et fonctions élémentaires	258
Figure 109 : Widget de construction de composition de service	260
Figure 110 : Résultat d'une composition de service.....	260

Table des tableaux

Tableau 1 : Points de vue du cadre 4+1	44
Tableau 2 : Part sectorielle dans l'emploi global dans les économies développées	62
Tableau 3 : Exemples de verbes à privilégier	116
Tableau 4 : Liste des opérations fonctionnelles génériques (GFO)	121
Tableau 5 : Liste de SFO correspondant aux processus de services précédents	133
Tableau 6 : Description des SFO et des SFD obtenus à partir des processus de service précédents.....	136
Tableau 7 : Données fonctionnelles utilisées par chaque SFO	140
Tableau 8 : Fonctions élémentaires correspondant à chaque SFO.....	141
Tableau 9 : Regroupement en quartiers et zones fonctionnelles	145
Tableau 10 : Rôles techniques du motif technique IMS	156
Tableau 11 : Protocoles usuels du motif technique IMS.....	157
Tableau 12 : Rôles techniques du motif technique SOA	158
Tableau 13 : Protocoles usuels du motif technique SOA.....	159
Tableau 14 : Rôles techniques du motif technique 3-tiers	160
Tableau 15 : Protocoles usuels du motif technique 3-tiers.....	161
Tableau 16 : Rôles techniques du motif technique Email.....	164
Tableau 17 : Protocoles usuels du motif technique Email	164
Tableau 18 : Rôles techniques du motif technique SMS	165
Tableau 19 : Protocoles usuels du motif technique SMS.....	166
Tableau 20 : Rôles techniques du motif technique OMA XDM.....	167
Tableau 21 : Protocoles usuels du motif technique OMA XDM	167
Tableau 22 : Rôles techniques du motif technique IPTV.....	168
Tableau 23 : Protocoles usuels du motif technique IPTV	169
Tableau 24 : Nœuds d'exécution d'un service de téléphonie.....	200
Tableau 25 : Points de référence d'un service de téléphonie.....	201
Tableau 26 : Nœuds d'exécution d'un service de vidéo.....	202
Tableau 27 : Points de référence d'un service de vidéo.....	203
Tableau 28 : Nœuds d'exécution d'un service de messagerie email.....	204
Tableau 29 : Points de référence d'un service de messagerie email.....	205

Tableau 30 : Lien applicatif-fonctionnel pour un service de téléphonie.....	216
Tableau 31 : Lien applicatif-technique pour un service de téléphonie	217
Tableau 32 : Lien applicatif-fonctionnel pour un service de vidéo.....	220
Tableau 33 : Lien applicatif-technique pour un service de vidéo	221
Tableau 34 : Lien applicatif-fonctionnel pour un service de messagerie email.....	223
Tableau 35 : Lien applicatif-technique pour un service de messagerie	225
Tableau 36 : Lien applicatif-fonctionnel pour le service VideoMe	228
Tableau 37 : Lien applicatif-technique pour le service VideoMe	229
Tableau 38 : Métrique de complexité d'usage.....	238
Tableau 39 : Métrique de complexité de développement	239
Tableau 40 : Métrique d'innovation	239
Tableau 41 : Métrique de complexité d'intégration.....	240
Tableau 42 : Quartiers fonctionnels et capacités de service TISPAN.....	243
Tableau 43 : Opérations fonctionnelles et web service Parlay X.....	246