

Université Badji Mokhtar Annaba
Département Informatique
Module: Architecture Orientée Services
Dr Klai Sihem

Architecture Orientée Services: MODELISATION

1. Conception du service métier
2. La démarche orientée service

CONCEPTION DU SERVICE METIER

- Pour concevoir les **services métiers**, il faut déterminer et spécifier **les opérations qui les composent** au niveau de la **modélisation des processus**.
- Ceci revient à définir **le périmètre fonctionnel de l'opération** et spécifier ses **conditions de déclenchement** et de restitution des **résultats** (pré-condition et post-condition).
- L'opération peut être utilisée par **plusieurs consommateurs**, il faut donc, tenir compte de **ses multiples contextes d'usage** qui sont souvent **hétérogènes et évolutifs**.
- L'opération est configurable à partir d'un ensemble de paramètres qui définit son comportement selon ses contextes d'utilisation.

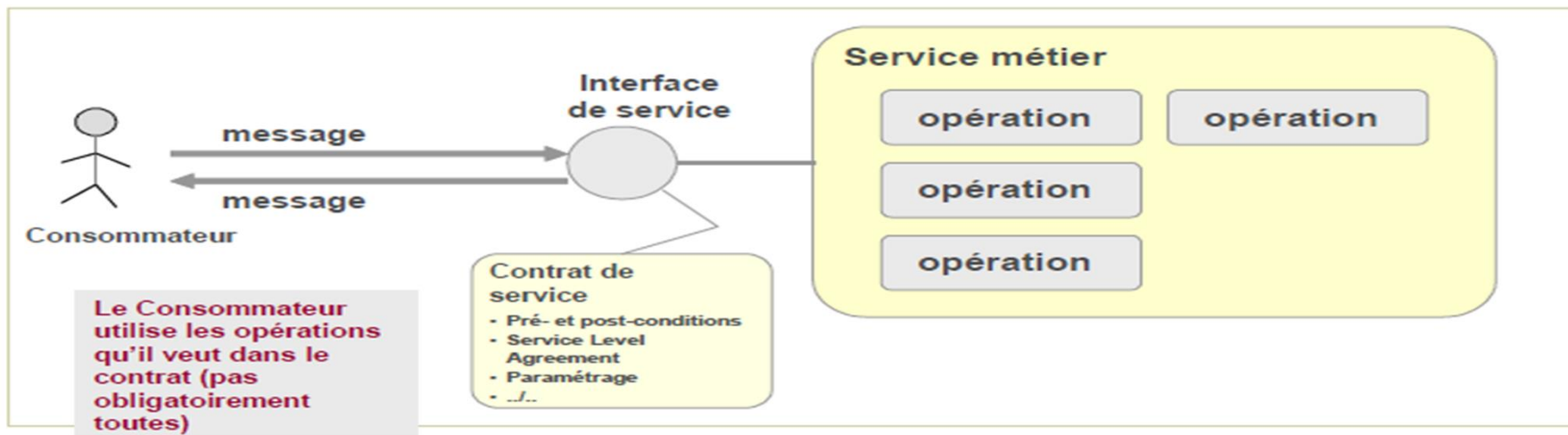
CONCEPTION DU SERVICE METIER

SOA impacte la modélisation des processus au niveau suivant :

- Découverte des opérations qui constituent des sous-ensembles du processus d'origine.
- Spécification des conditions de déclenchement et de restitution des opérations sous la forme de pré-conditions et de post-conditions.
- Spécification des messages d'entrée et de réponse et plus particulièrement, standardisation des types de données qui composent ces messages.
- Spécification du niveau d'adaptabilité du service par la modélisation d'un ensemble de paramètres de configuration (nommé modèle d'adaptation).
- Ces impacts doivent être compris par la **maîtrise d'œuvre** et aussi par la **maîtrise d'ouvrage**.
- Le travail de conception mené en SOA est complémentaire de la modélisation habituelle des processus ; il est déterminant pour **l'exposition des services métiers** ainsi que pour l'obtention d'une **architecture applicative orientée services**.

SERVICE METIER ET OPERATION

- Afin de modéliser les traitements exposés par le service métier, la notion d'opération est introduite dès la conception des processus.
- Un service métier est le regroupement d'une ou plusieurs opérations qui forment une cohérence fonctionnelle;



- **Le Service Métier est un regroupement d'opération (1 à N)**
- **L'interface du service expose la signature des opérations**

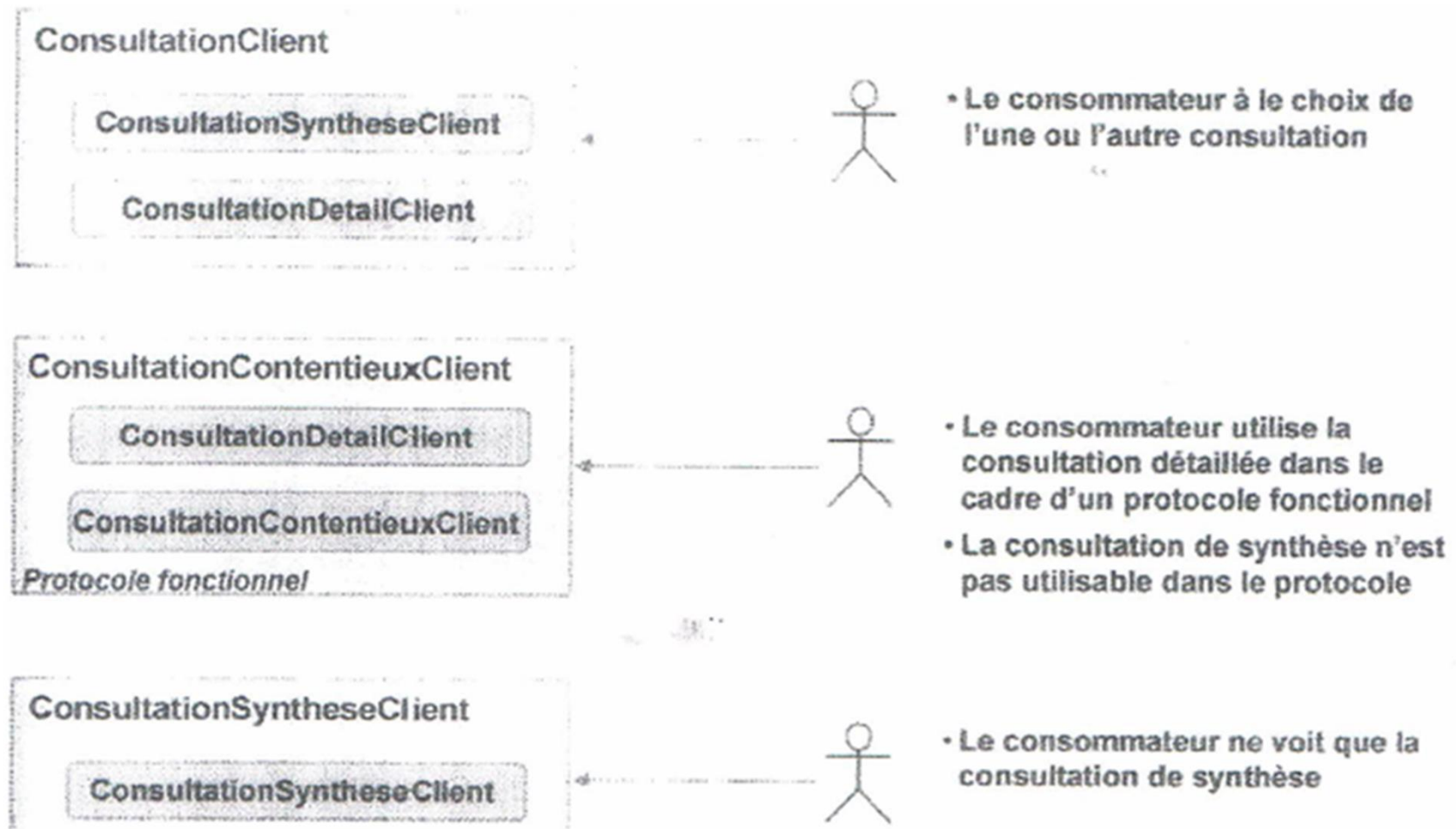
- **Service métier = WSDL**
- **Opération = operation au sens du WSDL**

SERVICE METIER EN OPERATION

Une même opération peut être exposée dans plusieurs Web services. Les règles retenues pour guider les regroupements sont les suivantes :

- **Publication** : Les services métier expose toutes ses opérations
- **Cohérence** : Les opérations dans un même service métier doivent faire sens fonctionnellement pour le consommateur (ex : protocole fonctionnel, enchaînement d'opérations)
- **Réutilisation** : Une opération peut être réutilisée dans plusieurs services métiers
- **Version** : La gestion de version porte sur le service métier et pas sur l'opération

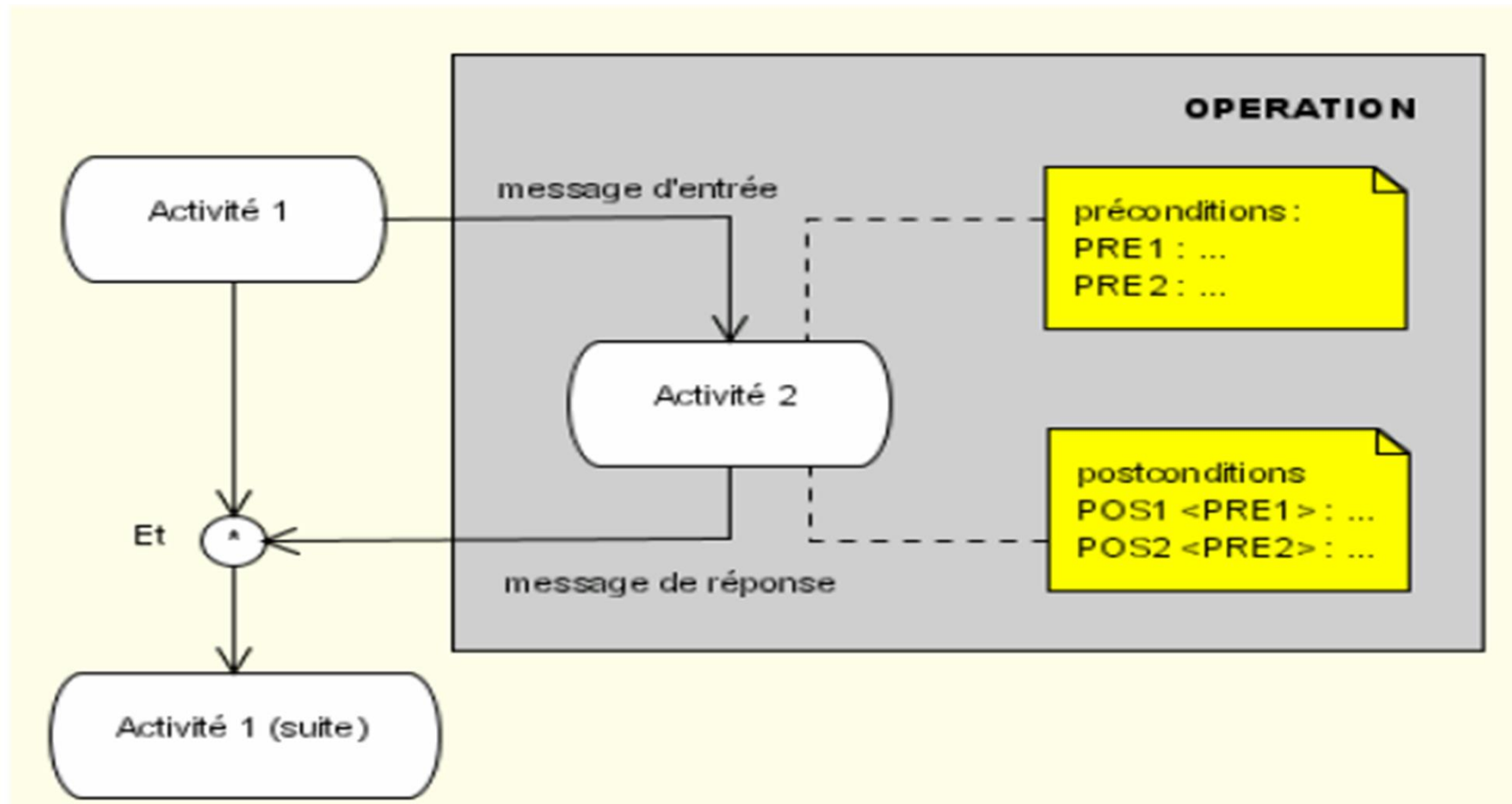
EXEMPLE



MODELISATION AU NIVEAU DU DIAGRAMME D'ACTIVITE

- Les opérations ne se connaissent pas entre elles. La logique d'enchaînement entre les opérations est placée dans une fonction d'orchestration matérialisée ici par le concept de processus.
- Une opération est un regroupement de plusieurs activités d'un processus ou d'un sous processus obligatoirement non interruptibles.
- L'opération est formée par :
 - Un message d'entrée qui contient les paramètres d'appel.
 - Un message de réponse qui contient les paramètres de résultats.
 - Des exceptions de notification des erreurs.

EXEMPLE DE REPRESENTATION DE PRE-POST CONDITION



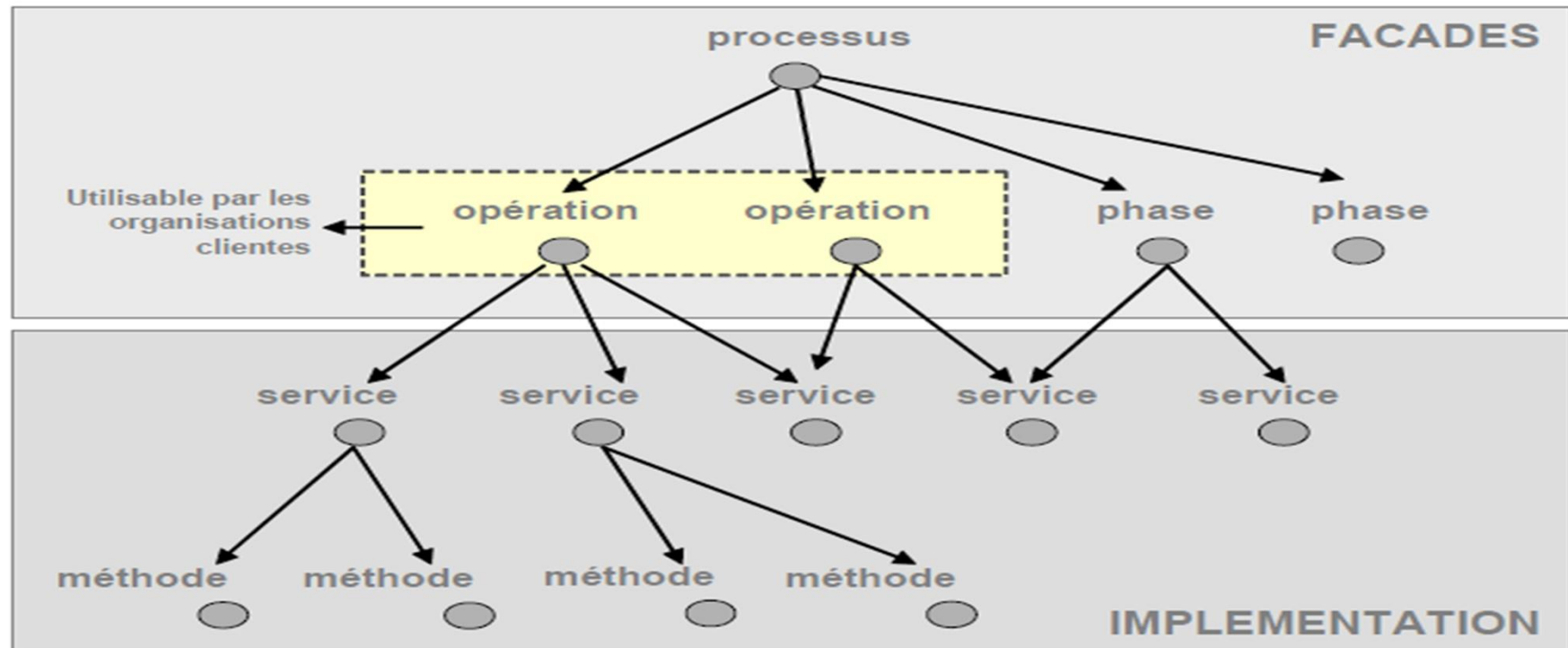
- Une opération est formée par un seul message d'entrée et un seul message de réponse. Ces messages correspondent à un ensemble de types simples ou complexes dont la modélisation est issue d'un diagramme de classes UML ou d'une représentation en XML.
- Puisque ces types de données sont partagés entre le fournisseur et le consommateur, il est déterminant de favoriser la standardisation aux niveaux suivants :
 - Codification : un même type ne doit pas être codifié de manière différente d'une opération à l'autre. Le respect de ce principe peut conduire à la mise en place de meta-type qui masque des codifications hétérogènes préexistants à la construction des opérations.
 - Sémantique : une administration des données doit garantir l'homogénéisation de la terminologie.
 - Implémentation : l'usage de XML Schéma s'impose comme standard pour la gestion des types.
- Une opération peut être exposée à plusieurs consommateurs dont les contextes d'exécution sont hétérogènes. Pour être en mesure de gérer de manière rationnelle ces contextes il faut :
 - Modéliser une structure de paramètre qui permette de configurer l'opération de manière adaptée selon les contextes d'exécution. Ces paramètres concernent la présentation ergonomique de l'opération, son comportement fonctionnel, sa qualité de service.
 - Exemple : pour une opération de consultation détaillée d'un portefeuille de bourse, on peut disposer des paramètres suivants :
 - Nombre de ligne maximum par consultation
 - Format des montants
 - Présentation d'une marque
 - Présentation des clauses juridiques personnalisées
 - Heures d'ouverture de service.
 -
- Le contrat d'utilisation du service métier et plus précisément de ses opérations est un document rédigé par le fournisseur et remis au consommateur. Ce document exprime les engagements du fournisseur et les devoirs du consommateur pour l'utilisation de chaque opération.

PHASE

- Lors de la modélisation du processus, les activités qui ne participent pas à la conception d'une opération sont regroupées sous la forme de phase.
- Une phase est un regroupement d'activités qui ne participent pas aux opérations.
- La phase est un traitement qui pourra être exposé ultérieurement sous la forme d'un service métier.

PROCESSUS

- Un processus est représenté par un **diagramme d'activités qui modélise un traitement de bout en bout** pour une préoccupation de gestion.
- **Enchaîne le déclenchement d'opérations et de phases** et assure un rôle d'orchestrateur.
- Le processus orchestre les appels d'opérations et de phases.
- L'opération et la phase orchestrent les appels de services.
- Le service orchestre les appels de méthodes.



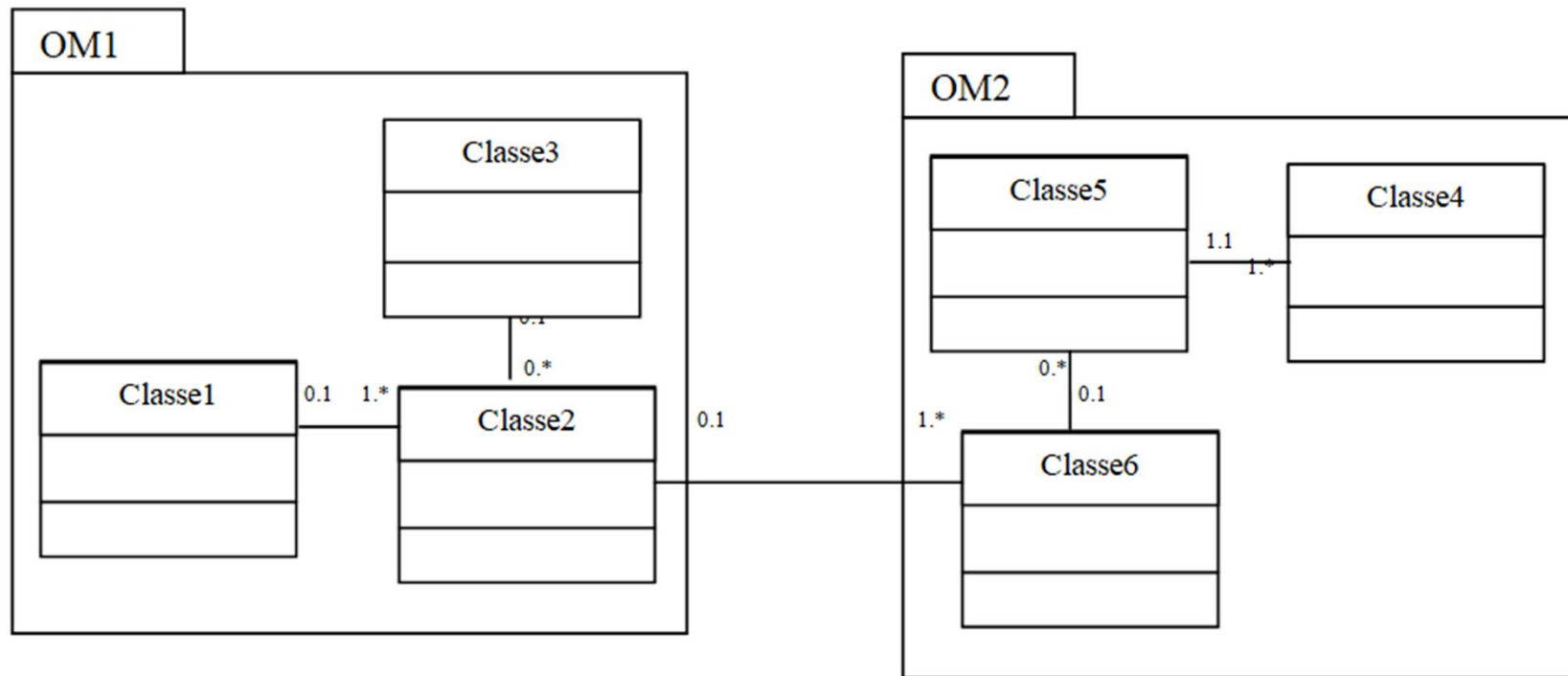
LA DEMARCHE ORIENTEE SERVICES

- La démarche SOA conduit à la modélisation d'une **cartographie** des données sous la forme de **Catégories**. Chaque catégorie représente **un objet métier ou sujet métier** à partir de laquelle on construit les services.
- Les opérations des services métiers échangent des messages dont le format d'implémentation est le plus souvent XML.
- **La démarche d'architecture orientée service** décompose les traitements sous la forme de services rattachés à des unités cartographiques du modèle de données (méthode Orchestra). Ces unités correspondent au concept **Catégorie**.
- Cette approche forme **le pattern d'architecture SOA** qui est une propriété obligatoire que le concept de service doit respecter.

CATEGORIE: OBJET METIER: SUJET METIER

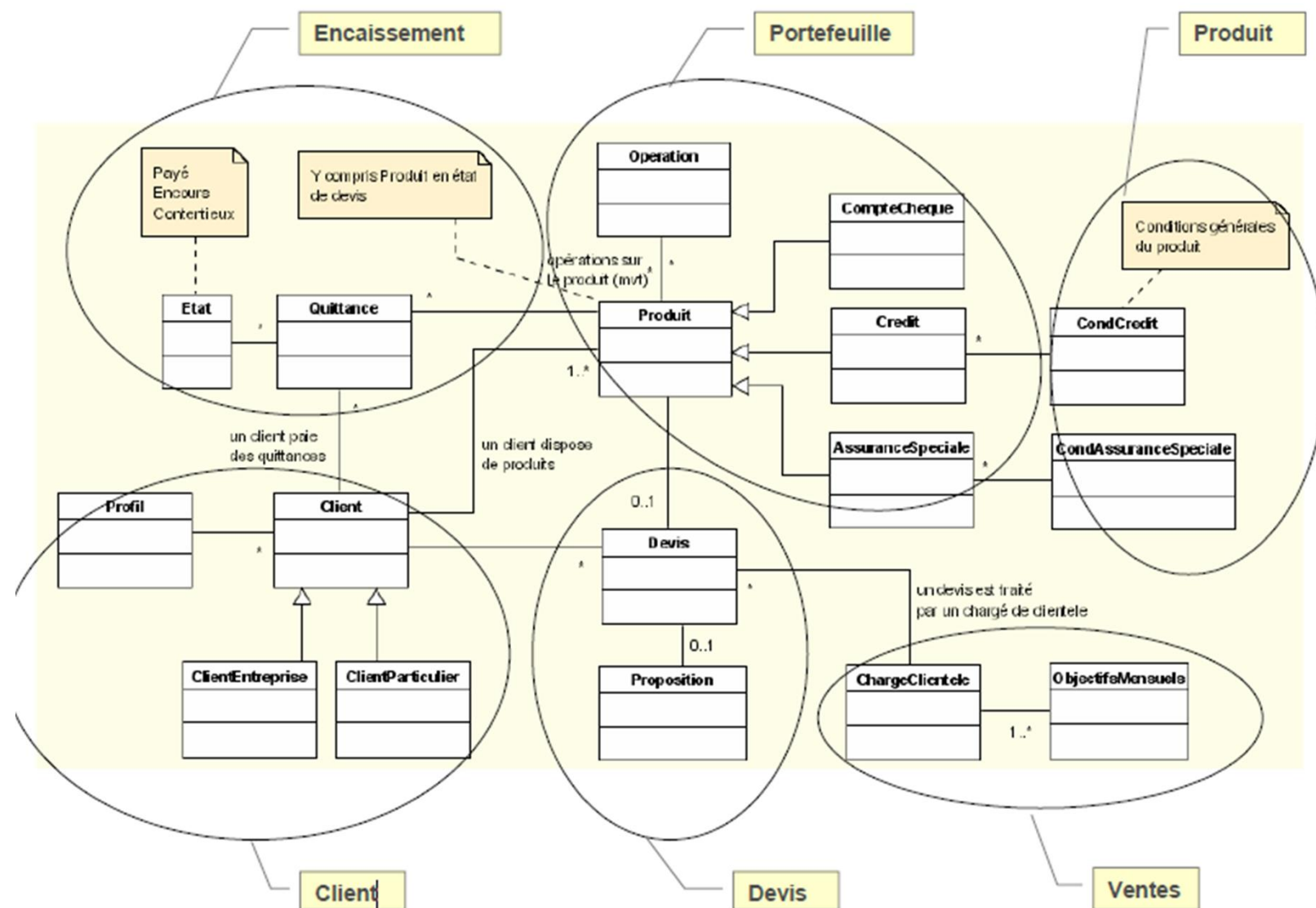
- On regroupe les classes du diagramme de classe selon les grands objets de gestion du SI. Chaque un respecte obligatoirement les cinq propriétés suivantes :
- **Stable** : sa frontière ne varie pas en fonction des évolutions des traitements
- **Consistante** : Elle est composée d'une classe pivot qui représente son concept métier et des classes dérivée qui la décrit.
- **Mono préoccupation** : Elle ne contient que les classes qui décrivent son seul et unique concept métier.
- **Contiguë** : des classes composent la catégorie sont toutes en relation entre elles. Il n'y a pas de classes isolées.
- **Nommé** : Chaque catégorie dispose d'un nom unique la définissant.
- **La catégorie** est représentée sous la forme d'un package UML avec un stéréotype l'indiquant.

CARTOGRAPHIE EN CATEGORIES



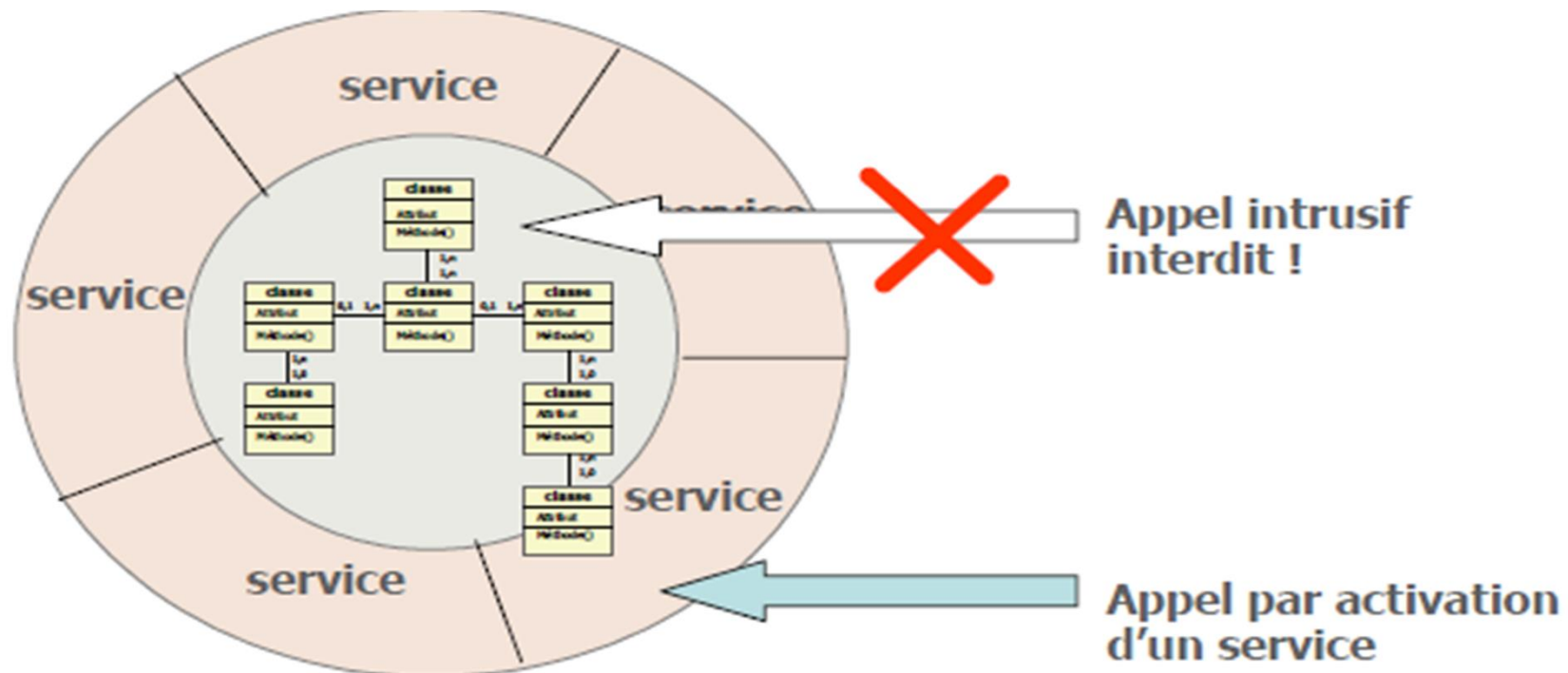
REGLES DE DECOUPE DU DIAGRAMME DE CLASSE

- Déterminer la **classe maîtresse**, elle est considérée maîtresse s'il est **possible d'agréger autour d'elle un ensemble d'autres classes** qui se trouvent structurellement sous son contrôle.
- Chaque classe doit **appartenir de manière exclusive à une seule et unique catégorie**
- Voici quelques règles de découpe en fonction des cardinalités entre les classes :
 - Soient A et B deux classes en relation
 - Si la cardinalité du côté A est la même que celle du côté B alors les deux classes appartiennent à la même catégorie
 - Cas (0,1 ;0,n) ou (0,1 ;1,n) ou (1,1 ;0,n) ou (1,1 ;1,n) : on peut couper du côté de B ; c.à.d. que A est l'association appartiennent à une même catégorie
 - Cas (0,n ;1,n) : on peut couper du côté de A



SERVICE ET CATEGORIE

- Un service est un traitement qui est rattaché à une seule catégorie
- Le service respecte les principes du couplage faible avec les autres services
- La catégorie est donc composée de classes encapsulées par ses services, autrement dit, les services forment une façade d'accès aux classes de la catégorie et assurent la propriété d'encapsulation.



DECOMPOSITION DE L'ACTIVITE EN SERVICE

- La logique de l'activité est décomposée sous la forme de services rattachés aux catégories.
- Voici un exemple qui montre que l'activité 2 est éclatée en deux services respectivement placés dans les catégories nommées objet métier OM1 et objet métier OM2

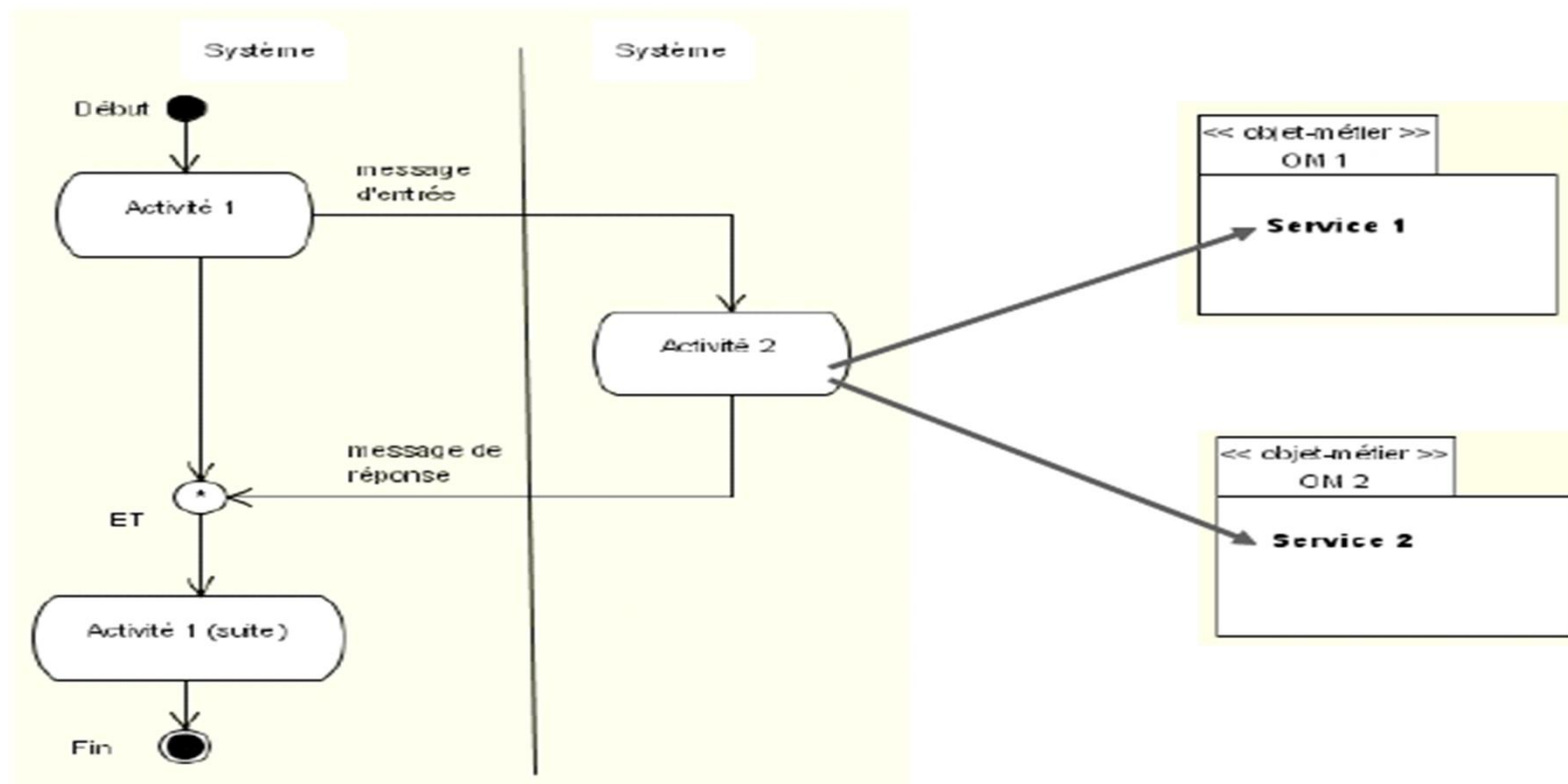
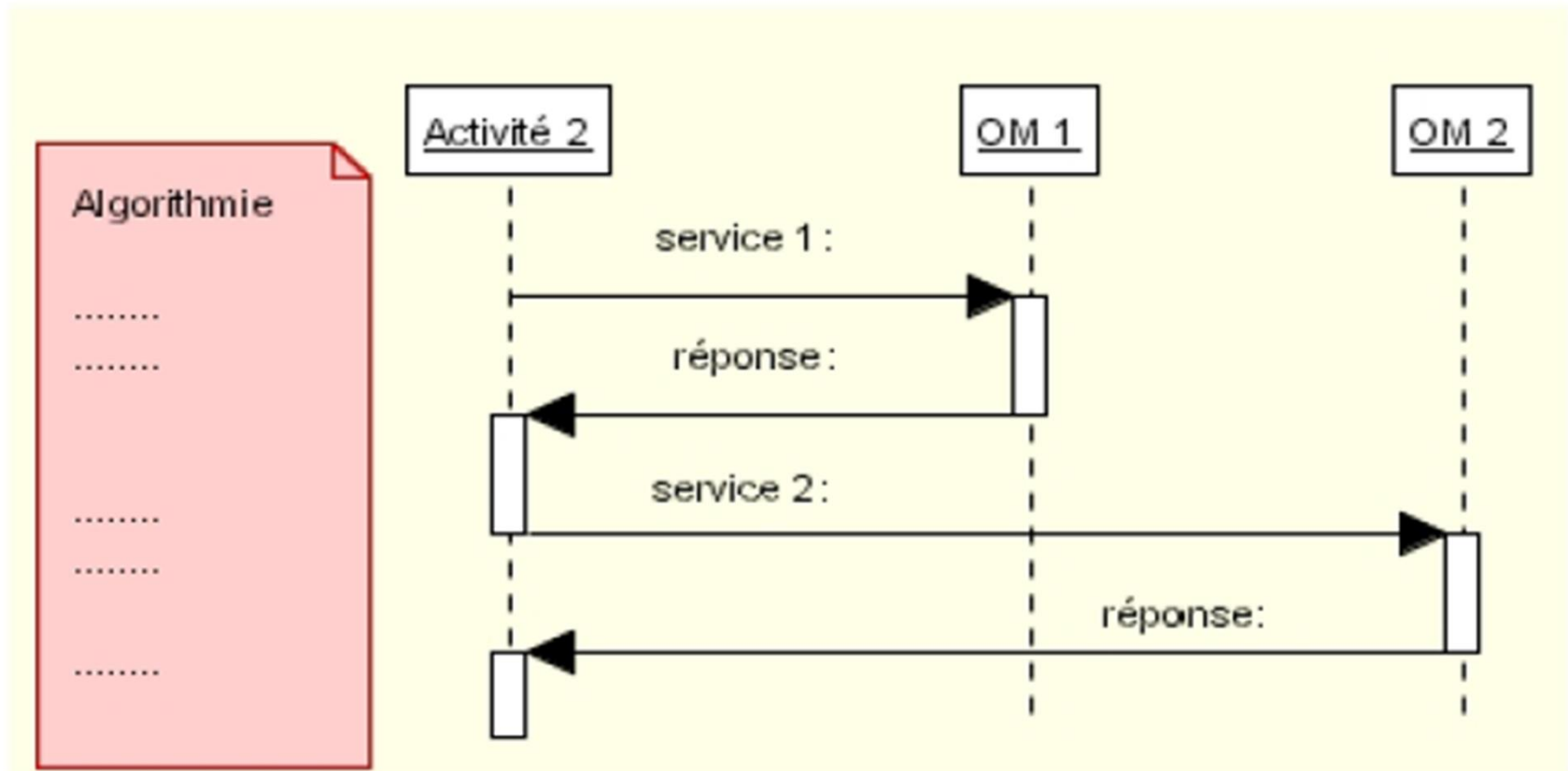
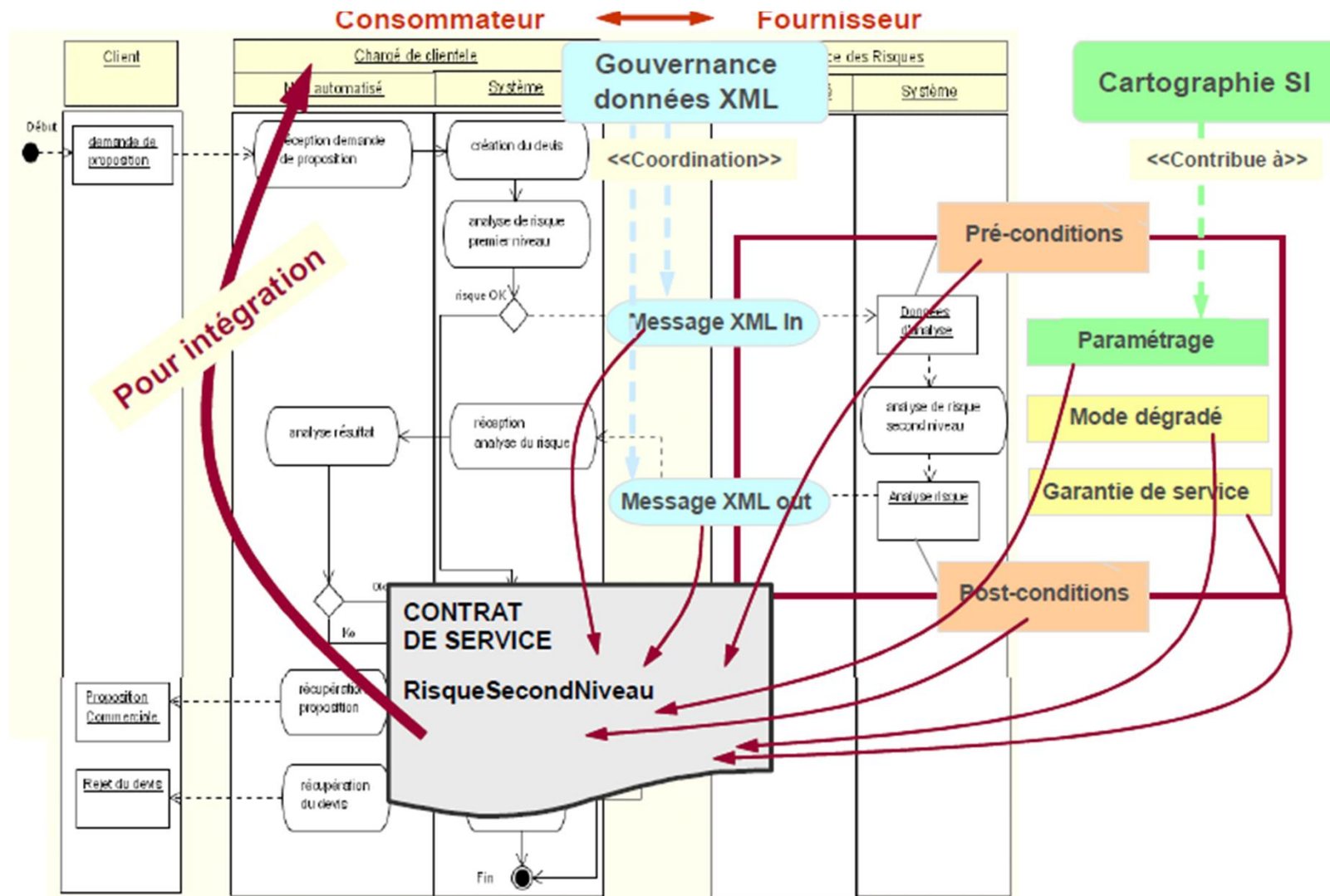


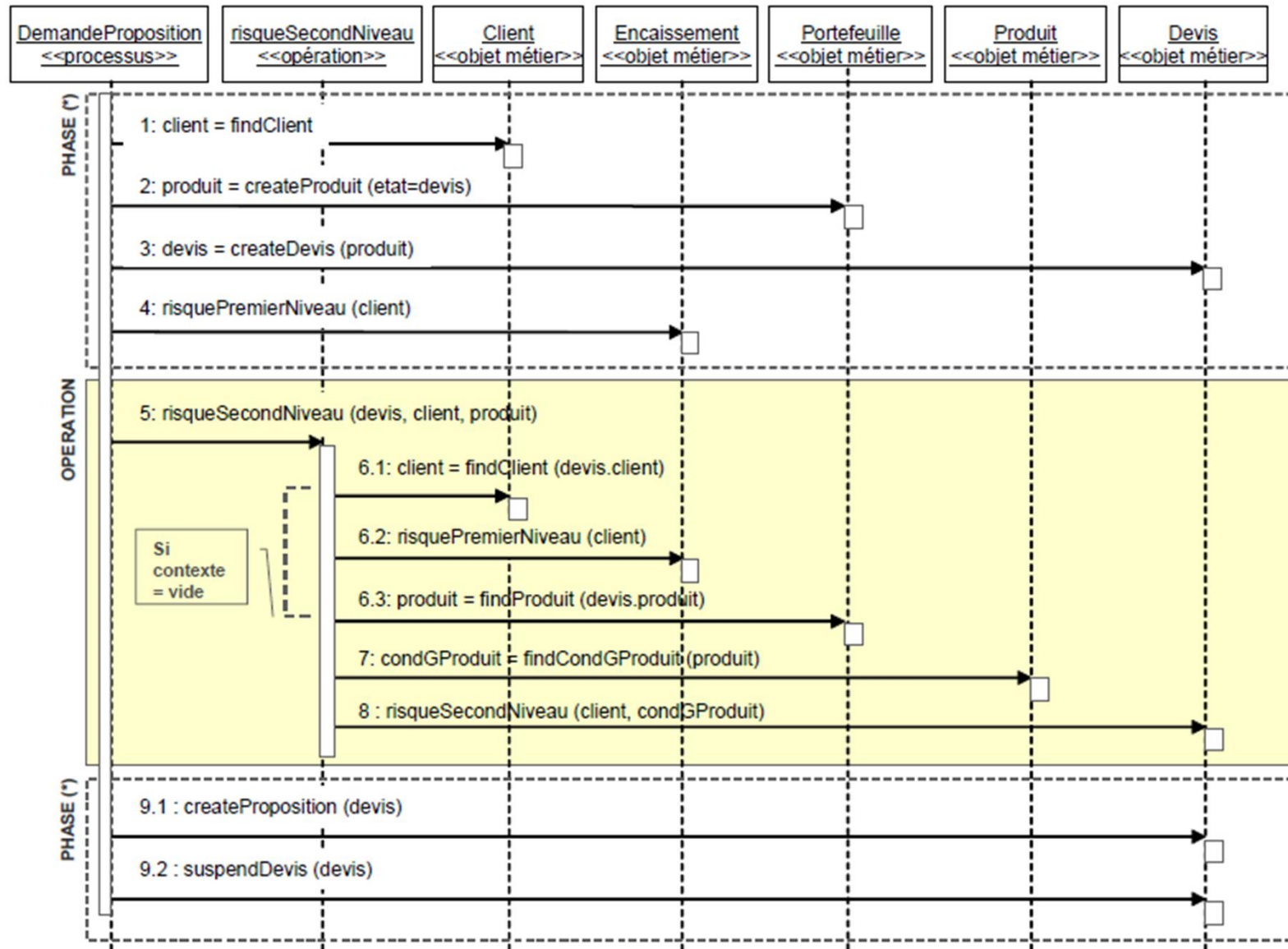
DIAGRAMME DE SEQUENCE DE DECOMPOSITION DE L'ACTIVITE EN SERVICE



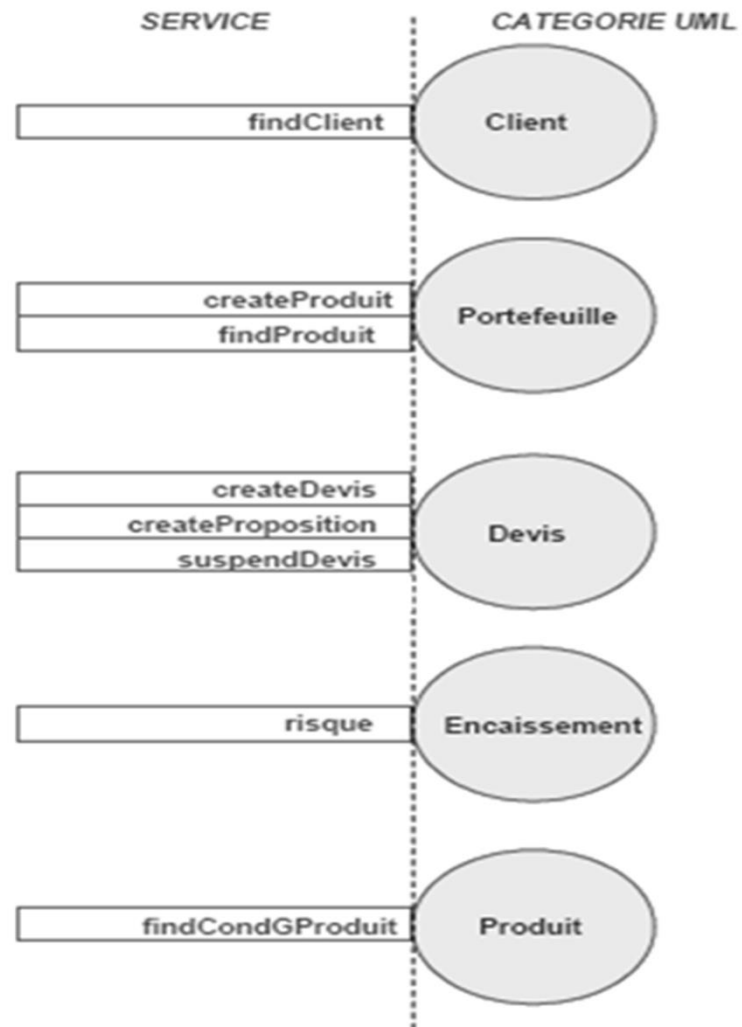
Exemple de décomposition en services(1)



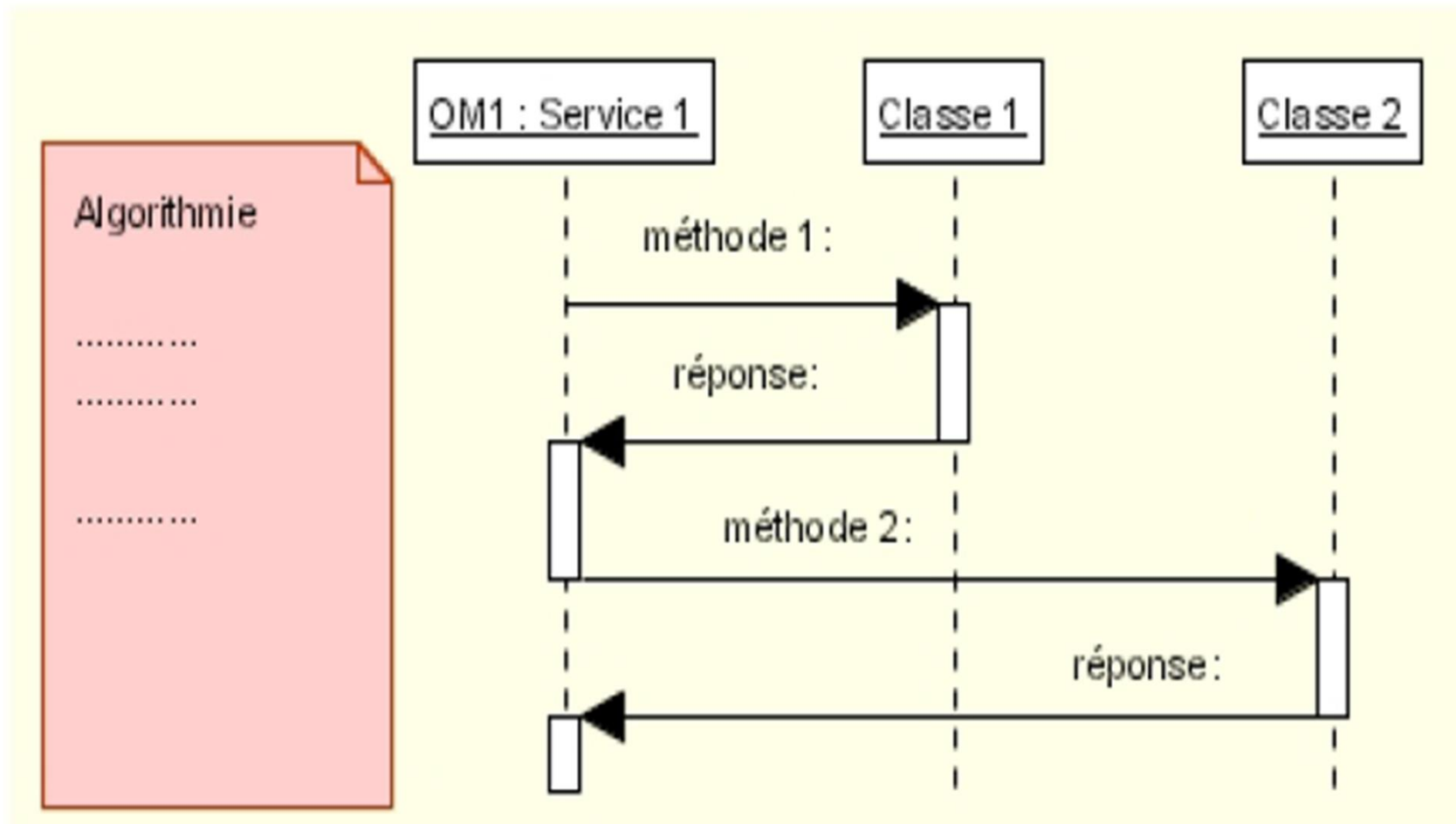
Exemple de décomposition en services(2)



EXEMPLE DE CATALOGUE DE SERVICES



Décomposition de service en méthodes (diagramme de séquence)



GRANULARITE DES SERVICES (1)

- Il est plus aisé d'assembler des services fins pour en composer de plus large que de décomposer un service existant en plus petits éléments, cela favorise l'approche Bottom-up.
- Par ailleurs, il est difficile d'administrer un nombre important de petits services. Il est alors préférable d'en avoir moins avec un grain plus large, cela favorise l'approche Top-down.
- Cette contradiction est en partie résolue par l'architecture SOA (modélisation Orchestra).
- Le travail top-down est mené grâce à la modélisation des processus et à la cartographie du diagramme de classes par catégorie.
- Les catégories forment un concept de haut niveau dont la granularité est bien adaptée à l'administration du système d'information et limite le couplage entre les services.

GRANULARITE DES SERVICES (2)

- Le travail **bottom-up** est mené grâce à la décomposition des services en méthodes attachées aux classes. Les paquets de classes sont encapsulés par la catégorie propriétaire et le chevauchement entre les méthodes est limité au périmètre de la catégorie.
- Le service est rattaché à une catégorie UML. Une granularité pertinente des catégories assure un premier niveau de régulation du grain des services.
- Le service métier offre un traitement ayant un sens métier pour les organisations clientes. Sa granularité ne doit pas être trop large sous peine de perturber la compréhension fonctionnelle et les performances d'exécution. Elle ne doit pas être trop faible sous peine de ne pas offrir de valeur métier suffisante.
- En cas d'implémentation sous la forme de web service, les principes de paramétrage (granularité dynamique) sont intéressants pour faire varier le WSDL en fonction des besoins de chaque organisation cliente.

Exercice

- Des hôtels proposent leurs services pour les réservations de chambres en ligne exposés à une agence de voyage. Le client fait sa demande de réservation en ligne à partir du site web de l'agence, il doit remplir un formulaire où il doit insérer son nom, prénom, âge, date début, date fin de la période, le type et le nombre de chambres à réserver après avoir sélectionner l'hôtel concerné.
- Selon la disponibilité des chambres et une fois la réservation confirmée, le client doit recevoir le montant qu'il doit payer. Le montant est calculée de la façon suivante: $mt = (\text{nbre-chambre1} * \text{pu1} + \text{nbre-chambre2} * \text{pu2} + \dots) * \text{nombre-jour}$ et le montant tout taxe comprise $mtTTC = mt + mt * 0,05$ (on suppose que la taxe est de 5%).
- Elaborez les diagrammes de séquence pour la décomposition de l'opération « réservation » en services et la décomposition des services en méthodes.
- On suppose que le modèle de données est le suivant et on suppose aussi qu'il forme une seule catégorie sans la classe client:
- Une classe réservation(idreserv, date-deb, date-fin)
- Une classe chambre (idchambre, type)
- Une classe saison(saison)
- Une classe client(idpersonne, nom, prénom, âge)
- Une relation « faire » entre réservation et client
- Une relation « concerne » entre réservation et chambre
- Une relation « appartient » entre chambre et hôtel
- Une relation « evaluer » entre saison et chambre cette relation porte la propriété « prix-unitaire » qui change pour une chambre donnée d'une saison à une autre.

Références Bibliographiques

- [1] James P., H. Howell-Barber: « Service-Oriented Architecture – SOA Strategy, Methodology, and Technology» Auerbach Publication 2008.
- [2] P. Bonnet ; " Cadre de référence Architecture SOA. Meilleures Pratiques". Montreuil, 23 Février 2005. [http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20\(part1\).pdf](http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20(part1).pdf)
- [3] M. Champion, C. Ferris, E. Newcomer, Iona, and D. Orchard ; "Web Services Architecture". 14 Novembre 2002. <http://www.W3.org/TR/2002/WD-ws-arch-20021114/>.
- [4] V. Monfort, S. Goudeau ; " Web services et interopérabilité des systèmes d'information". Dunod, Paris, 2004.
- [5] D. Chappell ; " Entreprise Service Bus". O'Reilly media Inc., 2005.
- [6] B. Lublinsky and D. Tyomkin ; " Dissecting Service Oriented Architectures". Business Integration Journal 2003.
- [7] D. Berardi, D. Clavanes, G. D. Giacomo, M. Lenzerini, and M. Mecella ; " A foundational vision of e-services". In C. Bussler, D. Fensel, M. E.Orlowska, and J. Yang, editors, WES, Volume 3095 of Lecture Notes in Computer Science, pages 28-40. Springer, 2003.
- [8] M. Mrissa ; " Médiation Sémantique Orientée Contexte pour la Composition de Services Web". Thèse de doctorat de l'université Claude Bernard Lyon I. 15 Novembre 2007.
- [9] S. Rampacek ; "Sémantique, interactions et langages de description des services web complexes". Thèse de doctorat. Université de Reims Champagne-Ardenne. Le 10 Novembre 2006.
- [10] F. Leyman ; " Web Services Flow Language (WSFL 1.0)". Mai 2001. <http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>.
- [11] I. A. Arkin ; "Business Process Modeling Language", 13 Novembre 2002. <http://www.bpmi.org/bpml-spec.esp>.
- [12] S. Thatte ; "XLANG - Web Services For Business Process Design", 2001.http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm.
- [13] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana ; "Business Process Execution Language for WebServices (BPEL4WS), Version 1.1". May 2003.
- [14] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau ; " Extensible Markup Language (XML) 1.0". Editors : World Wide Web Consortium, 4 February 2004. <http://www.w3.org/TR/2004/REC-xml-20040204/>.
- [15] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language". Editors: R. Chinnici, Sun Microsystems ; M. Gudgin, Microsoft ; J.J. Moreau, Canon ; J. Schlimmer, Microsoft ; S. Weerawarana, IBM Research. W3C Working Draft 26 March 2004. <http://www.w3.org/TR/2004/WD-wsdl20-20040326>
- [16] M. Chevassus ; "Enterprise Application integration : EAI". Techniques de l'ingénieur, H2915, pp.1-20, 2005.
- [17] Softeam ; http://www.softeam.fr/technologies_web_services.php.