



L'ingénierie dirigée par les modèles appliquée à la conception d'un observatoire sous-marin

Mikaël Bourvic

► To cite this version:

Mikaël Bourvic. L'ingénierie dirigée par les modèles appliquée à la conception d'un observatoire sous-marin. Base de données [cs.DB]. 2012. dumas-01063255

HAL Id: dumas-01063255

<https://dumas.ccsd.cnrs.fr/dumas-01063255>

Submitted on 18 Sep 2014

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

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

CONSERVATOIRE NATIONAL DES ARTS & METIERS
Centre Régional associé de Rennes

Mémoire
présenté en vue d'obtenir le
DIPLOME D'INGENIEUR C.N.A.M.
en
informatique
par

Mikaël BOURVIC

L'ingénierie dirigée par les modèles appliquée à la conception d'un
observatoire sous-marin

soutenu le 22 juin 2012

JURY

PRESIDENT : Professeur POLLET

MEMBRES : M. Préaux Charles
M. Champeau Joël
M. Schneider Jean-Philippe

Remerciements

Je remercie, tout d'abord, les membres de l'équipe IDM, Pascale Gautron, Dominique Kerjean, Oussama Kassem Zein et Hanh Nhi Tran qui ont activement participé à la conception et à la réalisation du projet MeDON en travaillant sur les capteurs intelligents, la gestion des données ou encore le développement d'un système d'exploitation dédié aux cartes électroniques que nous concevons.

Tous mes remerciements à Jean-Philippe Schneider, mon collègue de bureau, pour ses conseils autour d'UML, et son expérience en architecture des systèmes.

Je remercie également l'équipe acoustique passive dont Lucia Di Orio, Virginie Jaud, Cyril Chailloux, Cedric Gervaise pour l'expertise du milieu marin et des techniques d'écoutes de milieux à l'aide de méthodes non intrusives qu'ils ont partagé.

Merci à tous les étudiants qui nous ont accompagnés dans différents projets autour de MeDON et sans qui, les expérimentations n'auraient pu se dérouler.

Merci à tous les stagiaires, Mathilde Dhaussy, Fabien Nedellec ainsi que nos partenaires chinois qui ont participé à cette réalisation.

Je tiens à remercier tout particulièrement M. Charles Préaux, M. Joël Champeau, et Jean-Philippe Schneider, mes trois tuteurs pour ce stage, pour m'avoir offert la chance de réaliser ce mémoire, ainsi que pour leur encadrement, leur disponibilité et leur soutien dans ma progression au-delà de ce dernier travail.

Je remercie l'ensemble du personnel du CNAM Bretagne pour les connaissances, compétences qu'ils ont su me transmettre.

Je tiens, enfin, à remercier les membres du jury pour avoir accepté de participer à la soutenance de ce mémoire et M. Paulet pour avoir accepté de présider ce même jury.

TABLE DES MATIERES

INTRODUCTION.....	10
CHAPITRE 1 - INTRODUCTION	11
1.1 Le contexte.....	12
1.2 Présentation du sujet du mémoire CNAM	14
1.2.1 La problématique MeDON	14
1.2.2 Le rôle de ce mémoire.....	14
1.2.3 Les enjeux liés au logiciel / Le sujet	15
1.3 Les objectifs du mémoire.....	25
1.4 Présentation du plan.....	26
1.5 Gestion de projet.....	27
1.5.1 Organisation du projet.....	27
1.5.2 Les charges et les délais.....	28
1.5.3 Mon rôle dans le projet	28
1.5.4 Planning prévisionnel.....	29
CHAPITRE 2 - CADRE GENERAL DE L'ETUDE	31
2.1 Étude de l'existant.....	32
2.2 Spécification techniques de besoins.....	34
2.3 Encadrement de projets étudiants	36
CHAPITRE 3 - ÉTAT DE L'ART.....	38
3.1 L'existant	39
3.1.1 Le monde de l'IDM	41
3.1.2 Les Systèmes d'informations.....	51
3.1.3 Les systèmes d'information dans un contexte d'observatoires sous-marins distribués.....	55
CHAPITRE 4 - ÉTUDE CONCEPTUELLE.....	57
4.1 Étude et conception des modèles UML	58
4.2 Architecture Métiers	61
4.3 L'architecture Fonctionnelle	64
4.4 L'architecture Applicative.....	66
4.5 L'architecture physique et technique.....	68
4.5.1 La virtualisation	69
4.5.2 Système d'opérations	70

4.5.3	Le SGBD.....	70
4.5.4	Les serveurs Web.....	71
4.6	Conception du métamodèles de l'observatoire.....	72
4.6.1	Meta-modèle du SI MeDON.....	72
CHAPITRE 5 - MODELISATION		76
5.1	Cas d'utilisations.....	77
5.1.1	Cas d'utilisation globale	77
5.1.2	Cas d'utilisation Composant 1	80
5.1.3	Cas d'utilisation Composant 2	80
5.1.4	Cas d'utilisation Composant 3	81
5.1.5	Cas d'utilisation Composant 4.....	82
5.1.6	Cas d'utilisation Composant 5	84
5.1.7	Cas d'utilisation Composant 6	84
5.2	Diagrammes de séquence.....	86
5.2.1	Déploiement de la carte/annuaire du système.....	86
5.2.2	Acquisition de données brutes	87
5.2.3	Traitements des données	89
5.2.4	Déclenchement d'une alerte.....	90
5.3	Transformations	93
5.3.1	Modèle Objet vers Modèle relationnel	93
CHAPITRE 6 - DEVELOPPEMENTS.....		97
6.1	Le cycle de développement.....	98
6.2	Le prototypage	101
6.2.1	La première version	101
6.2.2	La deuxième version	101
6.2.3	La troisième version.....	103
6.3	La génération de code	104
6.3.1	Outils personnels.....	104
6.3.2	RulesComposer	106
6.4	Service d'annuaire	110
6.5	service d'alertes	111
CHAPITRE 7 - INTEGRATION.....		112
7.1	Méthodologie d'intégration suivie.....	113

7.2 L'architecture de développement	114
7.3 La gestion des versions	115
7.3.1 Le code.....	115
7.3.2 La base de donnée.....	115
7.4 Habilitations.....	116
7.4.1 Les services.....	116
7.4.2 Habilitation orientée donné.....	116
7.4.3 Autorisation d'accès à distance.....	116
7.4.4 Format d'échange des données	116
CHAPITRE 8 - DEPLOIEMENT DE L'ARCHITECTURE	118
8.1 Script de déploiement	119
CHAPITRE 9 - TESTS ET ANALYSE DES RESULTATS.....	122
9.1 Phase de développement.....	124
9.2 Phase d'intégration.....	126
9.3 Tests système	128
9.4 Métriques sur les développements	129
CHAPITRE 10 - APPORTS SCIENTIFIQUE.....	131
10.1 De nouveaux objectifs.....	132
10.1.1 Les observatoires	132
10.1.2 L'ingénierie dirigée par les modèles	133
10.2 Collaboration.....	135
10.2.1 Visibilité du laboratoire.....	135
10.2.2 Échange de données.....	135
CHAPITRE 11 - FONCTIONNALITES DE L'OBSERVATOIRE	136
11.1 Recommandations	137
11.2 Les prochaines pistes.....	138
11.2.1 Méthodologique.....	138
11.2.2 Le concept de Cloud Servicing.....	138
11.2.3 Les interfaces.....	139
11.3 Les perspectives d'améliorations	141
11.3.1 La validation par les utilisateurs finaux.....	141
11.3.2 Matériel.....	141
11.3.3 Logiciel.....	141

CHAPITRE 12 - COMMUNICATIONS AUTOUR DU PROJET	144
12.1 A court terme.....	146
12.2 A moyen terme.....	149
12.3 A long terme.....	150
CONCLUSION	151

INTRODUCTION

A l'heure où les effets du réchauffement climatique commencent à être perçus via la fonte des glaces et la désertification de nombreuses régions du globe, il devient urgent de s'équiper de technologies permettant aux scientifiques de mener des études basées sur des données réelles et sur de longues séquences temporelles.

Ce besoin a contribué au développement massif d'observatoires très hétérogènes par leurs fonctionnalités tel que Netpune au Canada, Bonne Bay Observatory ou encore le projet IOOS (Integrated Ocean Observing System).

Dans ce contexte de recherche scientifique, la technologie doit répondre à des besoins concrets d'études de notre environnement. Ces besoins peuvent se décrire en termes de fonctionnalités répondant aux objectifs métiers des utilisateurs de notre système. Dans le cadre de l'étude de notre environnement, les scientifiques ont besoin de faire l'acquisition de grandes quantités de données. Celles-ci peuvent alors faire l'objet de traitements immédiats ou pouvoir être réutilisées pour répondre à des études connexes suite à la détection ou la compréhension de nouveaux phénomènes.

Une des problématiques est donc de doter les observatoires de l'intelligence leur permettant de répondre aux besoins scientifiques actuels et futurs. Pour cela, différents problèmes se doivent d'être résolus et l'on doit bien prendre en compte les environnements considérés.

Chapitre 1 - Introduction

1.1 LE CONTEXTE

Le projet MeDON (Cf figure 1.1.1), pour Marine eData Observatory Network est un projet européen d'observatoires câblés auquel participe l'ENSTA-Bretagne afin d'apporter des éléments de réponse à la mise en place de technologies d'observations sous-marine. Ce projet s'intègre dans la volonté de proposer un type d'observatoire capable d'être déployé en eau peu profonde et répondant aux exigences des scientifiques en termes d'acquisition et de traitements des données et de partage de l'information.

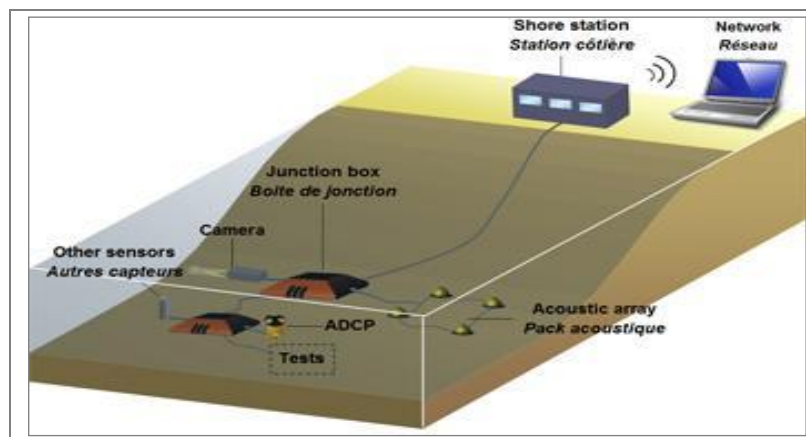


FIG. 1.1.1: Observatoire MeDON

Qui sont les acteurs de la conception ?

Le projet MeDON est le fruit du travail collaboratif de différents acteurs bretons et anglais. Il s'agit de l'équipe IDM et de l'équipe AP (Acoustique Passive) du pôle STIC de l'ENSTA-Bretagne en collaboration avec l'IFREMER et Océanopolis de Brest. Outre manche, le National Marine Aquarium et l'University of Plymouth et Plymouth Marine Laboratory(PML) se préparent à déployer un second observatoire. PML pilote le management du projet. Le rôle de chacun de ces acteurs est décrit en annexe dans le tableau « Liste des actions, Work Package(WP) » (Annexes 4,5 et 6). Notre équipe est chargée de l'action 3.3 *DataManagement and archiving system* (annexe tableau 6).

Un « Work Package » est une phase du projet global. Les Work Package sont composés d'un ensemble d'actions réparties entre les différents partenaires.

Au sein de l'équipe IDM, nous sommes cinq à avoir travaillé sur le projet MeDON.

Joël Champeau s'est occupé du management interne du projet et j'ai travaillé avec Pascale Gautron sur le modèle de base de données.

J'ai également travaillé avec Oussama Kassem Zein sur la conception et la réalisation de services Web pour les capteurs intelligents (SmartSensor).

Jean-Philippe Schneider et Oussama Kassem Zein ont travaillé sur la définition du langage Smart Modelling for Sensors, SMS (langage spécialisé pour les réseaux de capteurs intelligents) pour ces capteurs.

Dominique Kerjean développe quant à lui, un système d'exploitation temps réel pour ces cartes électroniques (SmartSensor également).

Enfin, j'ai travaillé avec Jean-Philippe Schneider sur la réalisation du dernier prototype du système d'information.

Oussama Kassem Zein a quitté le projet il y a un an. Plusieurs stagiaires se sont relayés sur le projet et nous ont accompagnés dans les développements.

Pour ma part, je travaille avec Joël Champeau, Pascale Gautron et Dominique Kerjean depuis son lancement et j'ai veillé au développement et à l'alignement entre la base de données, les services Web et les cartes électroniques. J'ai également communiqué avec nos partenaires et modélisé le système. J'ai donc endossé les rôles de concepteur, urbaniste et développeur.

Qui finance ?

Le financement du projet est assuré à 50% par Interreg IV pour chaque partenaire. Les 50% restants sont sur les fonds propres de chaque partenaire. L'enveloppe de l'ENSTA-Bretagne est de 309 374,42€.

Interreg IV A France(Manche) /Angleterre, est un programme de coopération transfrontalière de 2007 à 2013 et financé par les Fonds Européens de Développement Régional (FEDER).

1.2 PRESENTATION DU SUJET DU MEMOIRE CNAM

1.2.1 La problématique MeDON

MeDON (Marine eData Observatory Network) a pour but de développer, tester et disséminer un nouveau concept de surveillance du littoral s'appuyant sur un réseau de capteurs sous-marins câblés. Ces observatoires permettent de fournir un suivi en temps-réel du fond marin. Leurs fonctionnalités doivent être adaptées aux besoins des utilisateurs et les outils d'acquisition de données doivent être non intrusifs pour l'environnement. La composante écologique est prise en compte et appuie nos solutions d'acquisition passive de données en milieu naturel protégé.

Ce projet présente deux phases :

- La première, dans laquelle nous sommes actuellement, a pour objectif de mettre en commun les connaissances, de développer des techniques, de trouver des financements et de sensibiliser le public.
- La seconde sera consacrée à l'installation de deux observatoires. Le pilote sera déployé en mer d'Iroise, et son clone dans la baie de Plymouth en Angleterre. Ce concept pourra être disséminé dans d'autres sites pour développer le réseau d'observatoires.

La région brestoise, et en particulier l'ENSTA-Bretagne, sont des partenaires privilégiés de ce projet. En effet, la rade de Brest est une baie semi-fermée, sujette à de fortes pressions humaines (zone urbaine et agriculture intensive sur des bassins versants relativement restreints), et ouverte sur un écosystème côtier très riche (Mer d'Iroise). Elle est par conséquent un lieu incontournable pour des initiatives ambitieuses afin de protéger la biodiversité.

Le Parc Naturel Marin d'Iroise a notamment pour objectif de gérer durablement et de protéger les ressources marines et espèces protégées, de réduire les pollutions maritime, et d'approfondir et disséminer la connaissance des écosystèmes marins.

En résumé, la problématique consiste à concevoir un observatoire sous-marins qui respecte l'environnement, qui soit re-déployable et qui réponde aux besoins scientifiques en matière d'acquisition de données.

1.2.2 Le rôle de ce mémoire

Ce mémoire s'intègre dans le cycle de validation du diplôme d'ingénieur CNAM. Il reflète un ensemble de connaissances acquises lors de notre formation. L'attention donnée au fond et la forme de son contenu doit rendre possible sa diffusion auprès d'un large public.

Il synthétise six mois de travail en entreprise et capitalise un ensemble de documents et de recherches lié au projet MeDON.

A la différence du véritable travail fourni durant les six mois, son contenu est contraint par son volume et nécessite donc un effort de synthèse conséquent.

Par ailleurs, le mémoire pouvant faire l'objet d'une publication ou d'une communication, il est nécessaire de donner au lecteur toutes les explications ou références nécessaires à une bonne compréhension de son contenu. Cet effort de vulgarisation permet de prendre du recul par rapport au travail réalisé. L'un des buts du mémoire est donc à la différence des objectifs projets, de pouvoir être analysé par un large public et pas seulement par les clients du projet.

1.2.3 Les enjeux liés au logiciel / Le sujet

Les enjeux liés au logiciel sont importants car ce démonstrateur pose les premières briques de ce que seront les observatoires du futur. Il fournira également des enseignements pour les prochains observatoires câblés côtiers.

Par ailleurs, par rapport à la problématique de la propriété des recherches effectuées sur ces installations, de l'argent public investi servant à sa construction et enfin des coûts de maintenance prévisibles, se posent différentes questions politiques, économiques et techniques pour lesquelles nos choix de conception logicielle jouent actuellement un rôle non négligeable.

En effet, j'ai pris en compte ces différents aspects pour le respect des coûts, des délais et la solution finale actuelle répond aux exigences précitées. Parmi les choix réalisés nous pouvons citer les protocoles et technologie logiciels utilisés, le déploiement des composants ou encore le choix de l'utilisation de solutions libres de droits (système d'exploitation, certains algorithmes, ...).

Enfin, le troisième enjeu est de permettre au grand public d'accéder à des outils qui sont à ce jour réservés à des scientifiques. L'objectif est de le sensibiliser à la cause du réchauffement climatique et aux problèmes de dérèglement de certains mécanismes marins qui perturbent cet environnement. Les réseaux de communications ne cessent de se développer. A l'heure du Web 2.0 et des terminaux mobiles connectés en illimité, notre technologie pourrait se diffuser rapidement et il peut être important de nous positionner sur ce marché en pleine croissance.

Que souhaitons-nous faire ?

Nous souhaitons appliquer une méthodologie d'ingénierie dirigée par les modèles (IDM), pour la conception du système d'information d'un observatoire sous-marin. L'IDM est un des axes de recherche du laboratoire. Nous cherchons à améliorer cette méthodologie et l'utiliser lors de projets innovant. Nous avons choisi le projet MeDON car il répondait à des besoins transverses de notre laboratoire. La thématique « observation sous-marine » est l'un des fils d'ariane de l'Ensta-Bretagne.

Cette participation permettra de fournir un observatoire opérationnel pour l'archipel de Molène (cf carte Fig 1.2.3.2).

Nos compétences en méthodologie IDM dans un cadre R&D où il est nécessaire de concevoir des solutions mettant en jeu différents cœurs de métier, nous ont permis d'intégrer ce projet. Des alternatives à la méthodologie IDM auraient pu être envisageables comme les méthodologies d'ingénierie classique. En revanche, une autre approche telle que la méthodologie SCRUM n'aurait pas été adaptée à ce projet en raison de la taille de nos équipes, de notre éclatement géographique et principalement du caractère « recherche » du projet par exemple.

Les modèles permettent aux différents partenaires de s'entendre sur les perspectives. L'objectif est d'intégrer dans ces modèles le vocabulaire permettant aux acteurs de s'organiser. Partant de rien pour aboutir à un système complexe et non compliqué, l'IDM, méthodologie que nous défendons, permet d'atteindre nos objectifs à travers tout le cycle de développement logiciel.

L'argument principal du choix de la méthodologie IDM réside dans la volatilité des spécifications du projet. Dans le cadre d'un projet de recherche, celles-ci peuvent évoluer très fréquemment et une approche IDM est suffisamment souple et robuste pour permettre de générer de nouvelles architectures de façon optimale et validée.

Au cours du projet, les spécifications n'ont cessé d'évoluer impliquant l'évolution de la conception du système. Nos modèles ont été mis à jour en continu sans que nos développements antérieurs ne subissent trop d'impacts. Au final nous avons convergé vers une solution qui doit, pour cette raison offrir des facilités de maintenabilité, d'évolution par sa modularité et donc de redéploiement conforme à l'objectif global du projet MeDON et aux besoins de nos partenaires et utilisateurs finaux.

De manière générale, l'objectif du projet pilote MeDON, est de répondre à différentes questions telles que :

- Que pouvons-nous faire, quel savoir faire pouvons nous apporter au projet ?
- Quelle est l'audience ?
- Quelles sont les technologies utilisables ?
- Quel est le modèle économique ?
- Comment interagir avec les autorités ?

Cette liste de questions touche à différents domaines qu'il faut classer, analyser et que je dois traiter de façon transversale par rapport à mon activité. Les réponses à ces questions nous permettent de modéliser le système et son environnement. L'idée est ensuite de générer une partie du code de l'architecture modélisée puis de permettre son déploiement sur une architecture physique spécifiée.

Quelle est l'audience ?

Un système doit être réalisé avec des objectifs précis. Identifier les utilisateurs est la première étape permettant de contextualiser les différents types d'interactions avec ce que nous pouvons à ce stade considérer comme une « boîte noire ».

D'après les spécifications réalisées dans le WP1 (Work Package 1), trois classes d'utilisateurs se démarquent :

- Les Administrateurs et techniciens
- Les scientifiques
- Le grand public

Le projet concerne tous les scientifiques qui ont un intérêt à étudier les océans. Les utilisateurs finaux identifiés se sont réunis afin que nous puissions entendre leurs besoins. Il ne faut pas oublier le grand public, qui fait partie de notre cible et pour qui cette source de données permettra, nous l'espérons, une forte sensibilisation.

La liste des utilisateurs scientifique du démonstrateur se trouve en Table 1.2.3.1.

TABLE. 1.2.3.1: Liste des partenaires scientifiques

Area	Name	Project or Organisation
Technical review	Jean-Francois Rolin	European Seafloor Observatory Network (ESONET)
Technical review	Paolo Favali	European Multidisciplinary Seafloor Observatory (EMSO)
Technical review	Adrian Round	Victoria Experimental Network Under the Sea (Venus)
Technical review	Jaume Piera	Expandable Seafloor Observatory (OBSEA)
End-user services	Laurent Chauvaud	Institut Universitaire Européen de la Mer (l'IUEM)
End-user services	Dickon Howell	Marine Management Organisation (MMO)
End-user services	Yann Stéphan	Service Hydrographique et Océanographique de la Marine (SHOM)
Integration into coastal management	Kaja Curry	Tamar Estuaries Consultative Forum (TECF)
Integration into coastal management	P. Watremez et P. Le Niliot	Agence des Aires Marines Protégées
Commercialisation potential	Attilio Francois	Littoralis Plouzané
Commercialisation potential	Jeremy C Thomason	Marine Ecological Services

Ce tableau a été construit à partir de la liste des participants à nos différentes réunions de travail. Il fait apparaître quatre centres d'intérêt pour le projet (colonne *Area* du tableau 1.2.3.1).

1. Technical review

Les aspects techniques vont permettre aux organisations partenaires qui conçoivent des observatoires marins de profiter de nos études et de faire évoluer leurs solutions vers des standards permettant davantage de communications entre les centres de données et les ressources matérielles sous-jacentes.

2. End-user service

Les utilisateurs finaux offrant des services pourront profiter de nouvelles sources de données pour corroborer leurs propres études.

3. Integration into coastal management

Les partenaires chargés du management de l'intégration de nouvelles technologies sur des aires protégées, veillent au respect de l'environnement en nous faisant profiter de leur expertise en matière de nuisances des différents choix technologiques que nous faisons.

4. Commercialisation potential

Enfin, différentes équipes suivent nos développements en analysant les potentiels commerciaux inhérents aux nouveaux services proposés ce qui permettra à moyen terme d'avoir des solutions de viabilisation pour le maintien en condition opérationnelle des infrastructures déployées.

Que pouvons-nous faire ?

Un système d'information (SI) est un ensemble organisé de ressources (matériels, logiciels, personnels, données et procédures) qui permet de regrouper, de classer, de traiter et de diffuser de l'information sur un environnement donné. (Source Wikipédia)

A la différence des observatoires terrestres, un observatoire sous-marin est une infrastructure immergée. Les observatoires peuvent être classés en deux catégories. Ceux qui sont immergés en eau peu profonde¹ et ceux qui sont déployés en eau profonde². Concernant le littoral breton et plus précisément la mer d'Iroise, les eaux profondes se situent environ à plus de 20 km de la côte. MeDON entre dans la première catégorie car nous souhaitons simplement apporter une preuve de faisabilité liée à la technologie particulière que nous déployons. Nous travaillons donc sur la méthodologie et l'approche, et ne cherchons pas à concevoir des équipements devant répondre à des exigences physiques liées au grand fond.

Dans le cadre de notre observatoire, le SI est constitué de composants communicants via un système de services Web. Ces composants sont déployés sur une architecture physique distribuée. Notre démonstrateur pilote est déployé sur l'île de Molène. Les serveurs de l'ENSTA-Bretagne sont eux installés sur le continent. Le logiciel est constitué de composants logiques qu'il nous appartient de déployer de manière optimale en fonction des contraintes d'espace de stockage, de flux d'informations et de débit réseau.

Le personnel responsable de l'infrastructure physique est une équipe de l'IFREMER. Le personnel responsable du déploiement logiciel est notre équipe de l'ENSTA-Bretagne. Les données appartiennent aux scientifiques ayant un accès à notre réseau et aux capteurs. Les protocoles de communications sont définis conjointement entre l'IFREMER et l'ENSTA-Bretagne en fonction de nos besoins logiciels spécifiés en phase 1.

Dans le cadre de mon travail, j'ai effectué une modélisation de notre système d'information. Cette modélisation s'intègre dans la documentation que l'on doit fournir comme livrable du WP3.3.

Ces cartographies dans le domaine de l'informatique peuvent emprunter le vocabulaire de l'urbanisme et nous pouvons considérer notre système comme un ensemble complexe d'éléments hétérogènes.

¹ Eau peu profonde : Eau dont la profondeur est comprise entre -6 et 0m : fonds sableux/fonds rocheux

(Source : <http://www.brgm.fr/>, établissement public de référence dans le domaine des sciences de la Terre)

² Eau profonde : Eau dont la profondeur est inférieure à -6m, le zéro étant le 0 NGF

En termes d'acteurs nous retrouvons les clients, produits, fournisseurs, partenaires qui vont s'animer suivant différents processus d'organisation et répondant à différents objectifs métiers. Mon travail a consisté à partir de ces objectifs métiers, à suivre une démarche d'urbanisation du système, et à partir de spécifications précises que nous avons fait apparaître, proposer différentes vues de notre système de façon à le rendre plus évolutif et efficace pour exécuter les tâches qui lui seront soumises.

En quelques mots concernant l'urbanisation du système, chaque couche est représentée par un point de vue, et chaque point de vue regroupe plusieurs diagrammes : couche métier (entité, persistance, accès aux données), couche composant (traitements, services, urbanisation) et couche présentation (cinématique de navigation, IHM). Les fonctions de l'observatoire sont l'acquisition, le traitement, la sauvegarde et la diffusion de données à nos partenaires et au grand public tel que représenté dans le cas d'utilisation simplifié figure 1.2.3.2.

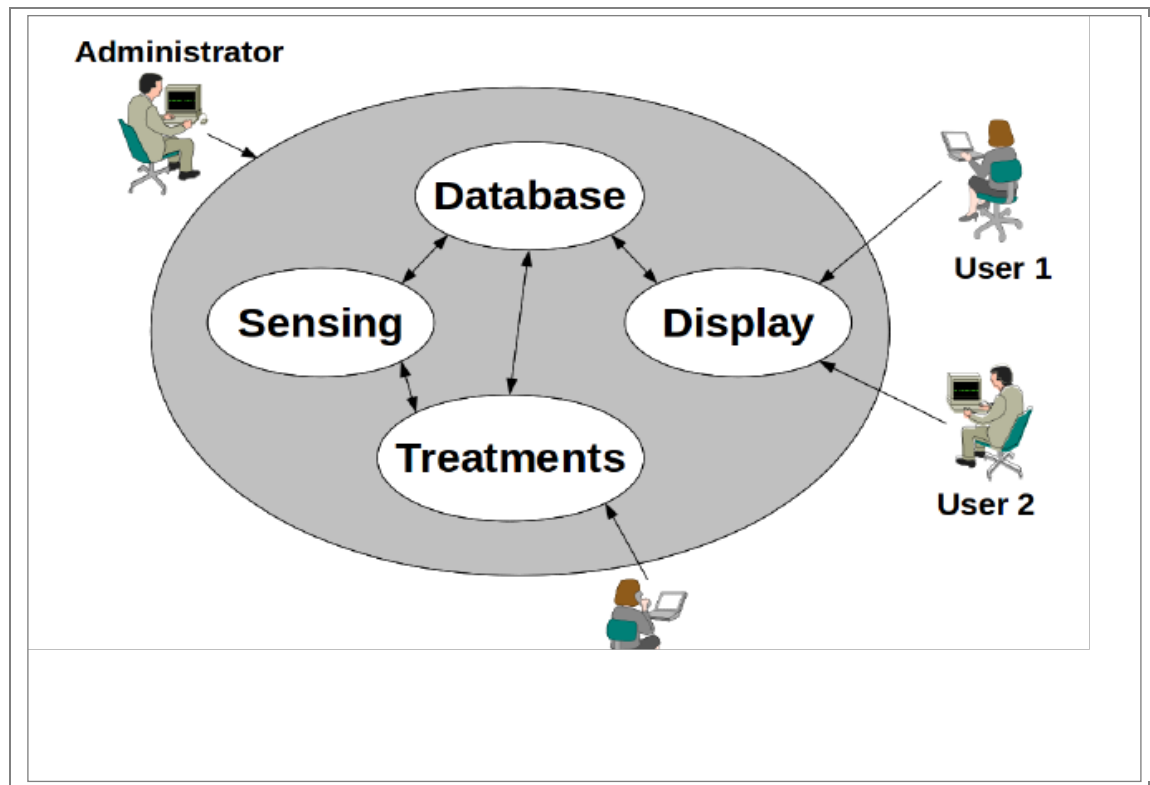


FIG. 1.2.3.2: Besoins de l'observatoire

Comment ?

Nous avons suivi les préceptes d'architecture d'entreprise. Ceux-ci permettent de coordonner l'étude et l'intégration présente et future d'un système d'information. Par ailleurs, l'ingénierie dirigée par les modèles appliquée aux observatoires marins nous permet de produire une solution intéressante pour différentes raisons :

- ↳ Rendre accessible les technologies liées aux fonctions d'observation sous-marine pour les utilisateurs finaux
- ↳ Réduire le temps de conception pour une fonctionnalité donnée quelle que soit l'architecture de l'observatoire
- ↳ Rendre plus facile les changements dans la configuration des capteurs (nombre, localisation, etc.)

Avec quoi ?

Je suis chargé de concevoir et d'intégrer les composants logiciels du système d'information. Techniquement, pour la réalisation du prototype, nous nous reposons sur une architecture de plusieurs machines virtuelles mises en réseau. D'un point de vue logique nous pouvons considérer que chaque machine virtuelle représente un composant communicant qui pourrait être déployé sur un serveur indépendant. Ainsi le déploiement sur une architecture distribuée dont nous ignorons encore les détails restera envisageable. Il s'avère que la configuration physique de notre réseau a déjà évolué depuis le WP1 et notre approche porte ses fruits car nos composants restent opérationnels et nous pouvons adapter notre déploiement aux nouvelles contraintes. A la base, il était prévu de rapatrier toutes les données à travers la liaison Molène-continent. Cela ne peut se faire pour des raisons de débits réseaux et donc il nous suffit de déployer le composant de traitement directement sur l'île.

Où ?

L'observatoire est donc réparti entre la mer d'Iroise et la côte ouest bretonne. Des capteurs intelligents sont immergés et communiquent avec nos serveurs.

Les données brutes doivent alors transiter via une connexion ADSL jusqu'au continent pour être stockées sur le serveur de l'IFREMER. Le serveur de l'ENSTA-Bretagne peut alors proposer un accès Web 2.0 pour diffuser les données traitées à nos partenaires et au grand public.

Techniquement, les échanges entre l'ENSTA-Bretagne et l'IFREMER doivent se faire via un VPN. L'île de Quemenes était l'une des îles candidates pour accueillir les serveurs de l'Ifremer. Cependant, il s'est avéré que les contraintes énergétiques ne nous permettent pas d'y déployer suffisamment d'électronique. L'étude menée par l'IFREMER a conclu que les capteurs seraient directement câblés sur l'île de Molène. L'état actuel de l'architecture physique est représenté en figure 1.2.3.4 et la localisation du câble en figure 1.2.3.3.

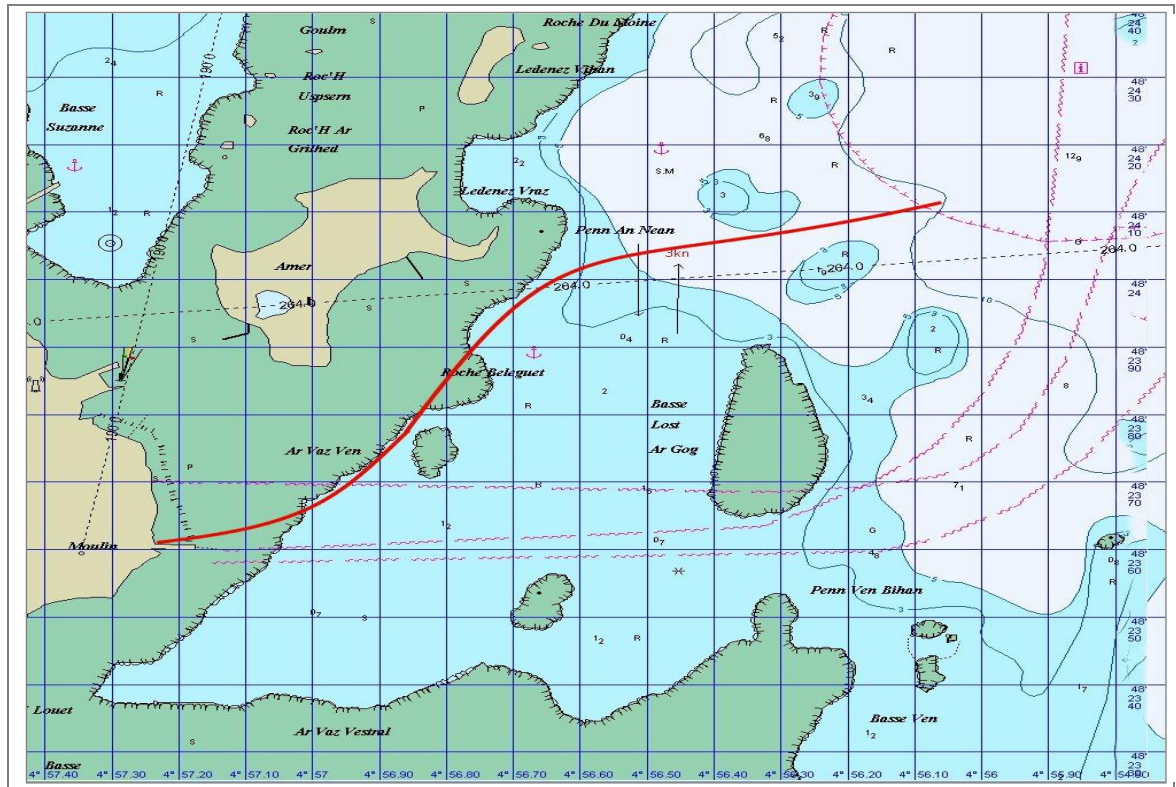


FIG. 1.2.3.3: Localisation du câble

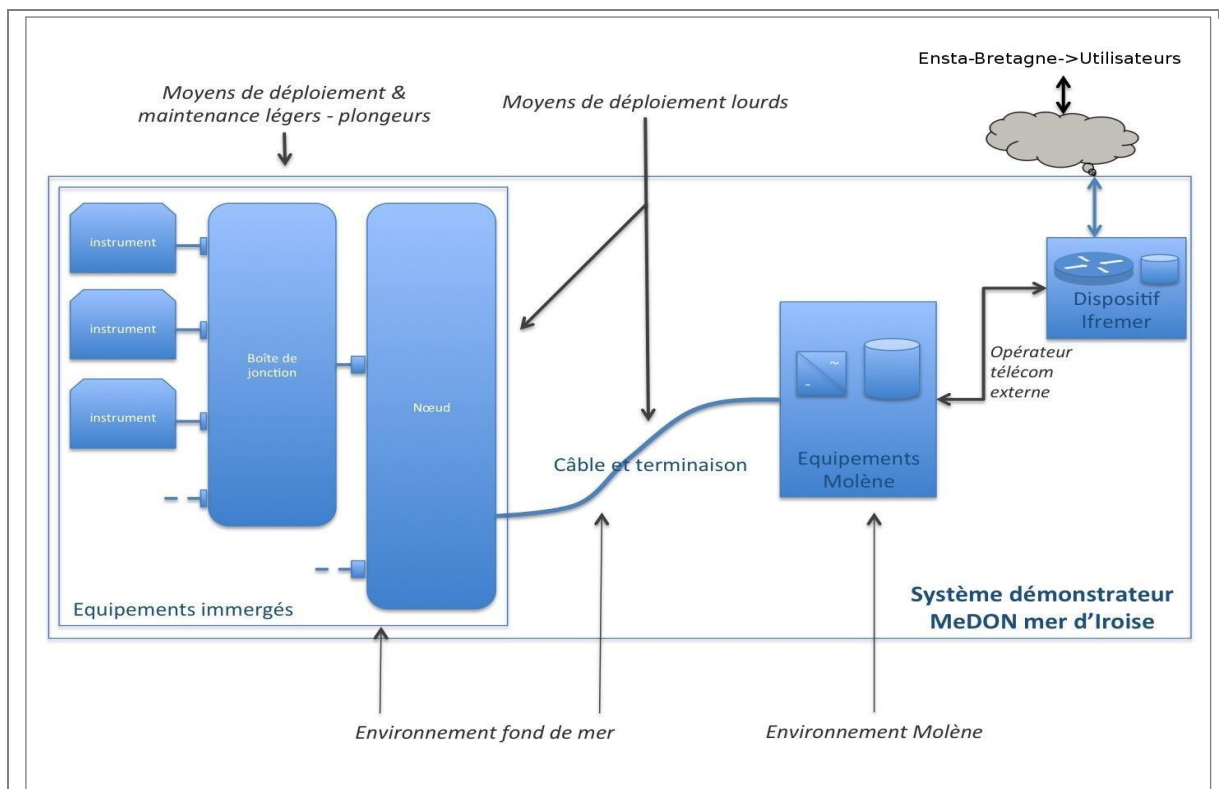


FIG. 1.2.3.4: Architecture physique

Quand ?

Le projet MeDON a débuté en février 2009. Les premiers câbles devaient être immergés fin juin 2011 mais cette action a été reportée à septembre/octobre 2011 par nos partenaires pour finalement n'être réalisée qu'au début 2012 (mars/avril). Nos tests en situation réelle seront donc décalés de quelques mois, qui profiteront à l'équipe pour développer notre documentation et renforcer nos tests unitaires.

Vers où ?

Le projet MeDON est la première étape de l'étude et de la réalisation d'observatoires dits « intelligents » destinés à aider les scientifiques dans l'étude des phénomènes naturels marins. A court terme, il s'agit de capitaliser un savoir faire en terme de déploiement technologique. A moyen terme l'objectif est de déployer d'autres observatoires en des lieux stratégiques. 70% de la surface de la planète sont couverts par les océans. Si tous les reliefs de la planète étaient érodés, l'eau liquide formerait un manteau de 3 km d'épaisseur. L'océan a été très faiblement exploré et le choix de ces lieux stratégiques peut à lui seul faire l'objet de longues études.

L'objectif à long terme de MeDON est de permettre une surveillance globale de zones côtière en interconnectant les observatoires les uns avec les autres.

Ça dépend de quoi ?

PML a la charge de la coordination du projet MeDON. Cédric Gervaise est le responsable scientifique de l'équipe Acoustique Passive du pôle STIC de l'ENSTA-Bretagne et guide l'expression de nos exigences en termes de besoins scientifiques. Joël Champeau est le responsable de l'équipe IDM du pôle STIC et guide notre conception et nos réalisations. Quant à moi, je réalise un travail quotidien aux côtés de Jean-Philippe Schneider qui travaille également sur le projet MeDON et plus particulièrement sur la conception et le développement du capteur intelligent *SmartSensor*.

Le projet MeDON dépend actuellement de l'entente cordiale entre les partenaires scientifiques français et anglais. Outre manche, PML dirige l'étude et le management du projet global. A travers des rapports d'avancement transmis à Tim Smith responsable du projet et nos visioconférences, nous discutons des problèmes de conception, techniques ou organisationnels que nous rencontrons. La fréquence de nos communications est un élément clef pour que les travaux soient correctement coordonnés entre tous les acteurs.

D'un point de vue technique lié au système d'information nous maîtrisons les technologies employées. Les câbles immergés relèvent quant à eux, du savoir faire de France Telecom et de l'IFREMER. Les capteurs immergés sont en phase de réalisation et sont basés sur des technologies similaires à d'autres capteurs qui ne sont pas dotés de capacité de calculs ni de stockage.

Ma contribution

Dans le cadre de mon travail comme ingénieur de recherche, j'ai participé à la conception et au développement d'un prototype de système d'informations pour ce projet d'observatoire sous-marin.

Pour cela, j'ai déterminé différents objectifs techniques et fonctionnels, alignés sur les objectifs métiers de nos partenaires afin de pouvoir proposer des infrastructures applicatives répondant aux besoins croissants identifiés. L'objectif est d'aboutir à une solution capable de s'adapter à un grand nombre d'exigences liées à nos besoins scientifiques.

Ma contribution à ce projet s'exprime selon les termes du domaine de la qualité, à travers les trois critères principaux suivant :

- la **portabilité** à travers notre solution de génération de code
- la **validité** à travers la validation de modèle
- la **réutilisabilité** à travers l'urbanisation du système

1.3 LES OBJECTIFS DU MEMOIRE

ENSTA-Bretagne

Mon objectif vis-à-vis de mon employeur est de contribuer à l'étude et à la réalisation d'un démonstrateur permettant l'observation de phénomènes marins en mer d'Iroise. Il s'agit d'apporter une étude de faisabilité via un prototype permettant de traiter des informations en provenance de capteurs immergés, de traiter ces données puis de les présenter au grand public.

Dans ce contexte, mon travail se focalise sur la réalisation du démonstrateur en utilisant la méthodologie d'ingénierie dirigée par les modèles afin de pérenniser les analyses et les développements que nous réalisons en permettant la génération et le déploiement d'autres observatoires reposant sur des concepts identiques. L'ENSTA-Bretagne retrouvera dans ce mémoire les différentes étapes de la conception du système d'information d'un observatoire sous-marin.

Je dois également rédiger la documentation relative aux développements réalisés et justifier de leur alignement par rapport à nos modèles.

Enfin, d'un point de vu personnel, j'ai veillé à réaliser une production dans les règles de l'art à savoir veiller à respecter nos méthodologies de groupe, notre certification ISO9001, et la qualité logicielle des développements en prenant en compte leurs perspectives d'évolutions futures et leur alignement par rapport à nos objectifs Métiers à travers les différentes perspectives de l'urbanisation.

CNAM

Mon objectif vis-à-vis du CNAM est de réaliser une étude présentant mon activité d'ingénieur de ces derniers mois. Durant ce projet, j'ai pu mettre à profit les compétences acquises lors de ma formation CNAM. En effet, il fut nécessaire de mettre en place une véritable démarche projet au sein de notre équipe dans laquelle j'ai pu être force de proposition et d'encadrement de plusieurs équipes d'étudiants. La production de ce mémoire permet de finaliser mon cursus de formation d'ingénieur CNAM pour l'obtention de ce diplôme.

1.4 PRESENTATION DU PLAN

L'objectif principal de ce mémoire est d'expliquer comment l'ingénierie dirigée par les modèles a permis la conception du système d'information d'un observatoire sous-marins.

Ce rapport s'intègre dans la formalisation de mon travail réalisé ces derniers mois. Il s'agit donc pour moi d'un exercice très intéressant consistant dans un premier temps à bien présenter les objectifs du mémoire de façon à faire ressortir le contexte de manière très claire à travers ce cadrage général de l'étude.

Dans un premier temps, j'effectue un état de l'art en matière d'ingénierie dirigée par les modèles afin de voir comment appliquer ce savoir faire à la conception d'observatoires sous-marins distribués. Mon étude conceptuelle permettra de bien comprendre les nombreux concepts pour la compréhension, la reproduction et l'amélioration de l'expérience menée.

Dans un second temps, la partie 2 sur la réalisation permettra de capitaliser une partie significative de la production réalisée et les principaux éléments à mettre en œuvre pour la conception, le développement et l'intégration d'un tel système d'information.

Enfin, la partie 3 sur les perspectives constituera une base de réflexion sur les améliorations que l'on pourrait porter à l'observatoire actuel. Cela devrait permettre aux lecteurs désireux de contribuer à ce type de projet d'avoir des éléments d'améliorations sur lesquels nous pouvons encore travailler et des pistes permettant d'améliorer la conception et la réalisation de systèmes d'observation.

1.5 GESTION DE PROJET

1.5.1 Organisation du projet

Les tâches réalisées

L'équipe s'est mobilisée autour du même objectif sur toute la durée du projet. Cependant, certains d'entre nous ont dû concilier cette activité avec d'autres tâches externe à MeDON comme des activités d'enseignements, de recherche ou de développement sur d'autres projets. Nous pouvons donc décrire notre organisation comme transversale entre plusieurs projets.

Le caractère particulier de MeDON est que nous avons dû identifier les besoins de nos partenaires. Nous n'avions pas de spécifications attendues. Notre première tâche a donc été d'identifier ces attentes en modélisant et en validant les intentions de nos partenaires. De ces premiers modèles nous avons découpé le projet en différentes briques sur lesquels nous pouvions plus facilement apporter nos compétences respectives.

Parmi les différentes tâches auxquelles j'ai participé, nous pouvons citer principalement la réalisation des différents modèles, la mise en place de la base de donnée, la réalisation de l'architecture de service Web, l'administration des différents serveurs (web, gestion de configuration, IRC, liste de diffusion).

D'autres tâches sous la responsabilité de notre équipe ont porté sur les développements matériels et logiciels liés aux capteurs intelligents, bien que ces derniers s'intègrent davantage dans le projet transverse SmartSensor que dans MeDON.

La spécification et la conception

La modélisation est une activité de spécification et de conception des différents éléments de notre système que nous avons conduite durant **toute la durée du projet**. Ces quelques lignes mettent l'accent sur cette approche.

Dans un premier temps, nous avons souhaité réaliser une modélisation UML avec des cas d'utilisation. Pour cela, nous avons utilisé l'outil Rhapsody³. Ma contribution a été de participer à la modélisation des cas d'utilisation, des diagrammes de séquences, de classes et modèles d'objets.

La figure 1.5.1.1 est une vue haut niveau des différents composants qui seront modélisés. La modélisation de chacun de ces composants m'a ensuite permis de générer⁴ une partie du code de l'applicatif de façon à guider et minimiser le temps de développement.

³ Rhapsody est un outil IBM qui permet de réaliser des modélisations UML

⁴ J'ai effectué la génération de code à l'aide de l'outil RulesComposer, un plugin de Rhapsody.

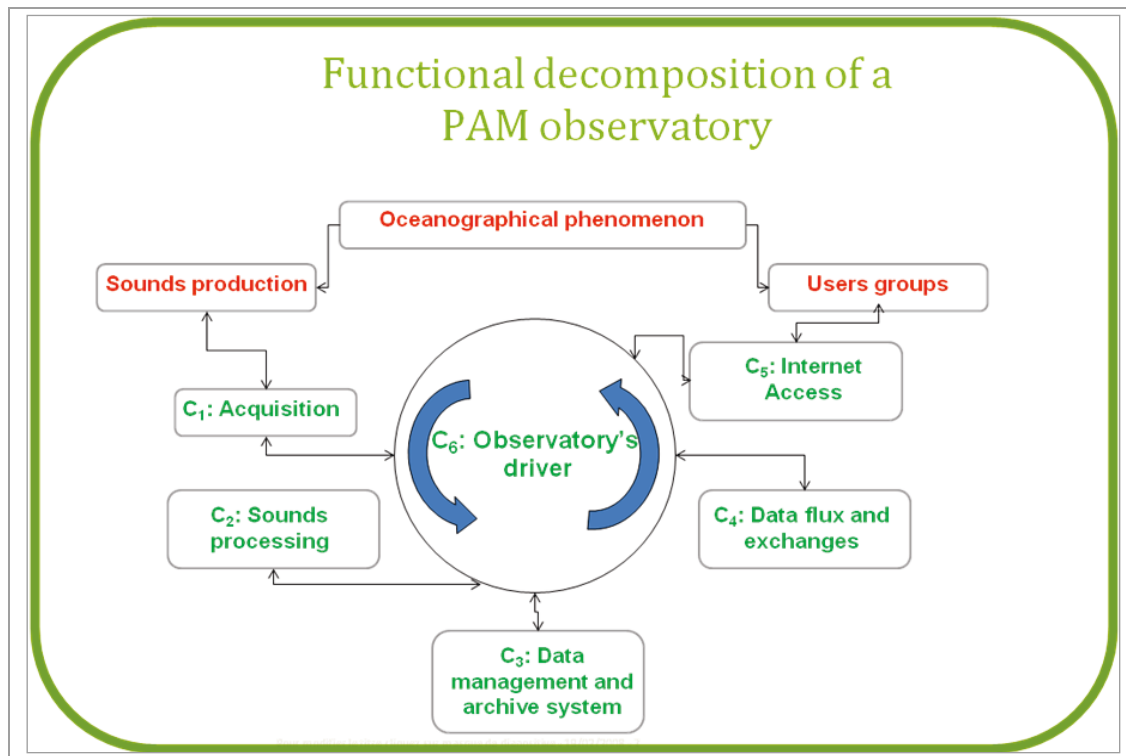


FIG. 1.5.1.1: Composants de l'observatoire MeDON

1.5.2 Les charges et les délais

Une fois les tâches du premier prototype identifiées, l'équipe projet a estimé les charges et les délais de sa conception à sa réalisation. Pour cela nous avons collaboré à la réalisation de spécifications détaillées sur le wiki interne du projet. Nous pouvions ainsi suivre l'avancement de chacune des tâches lors de leur réalisation. Sur la base des composants identifiés en figure 1.5.1.1, nous avons priorisé les tâches pour que l'on puisse aboutir à une structure cohérente des données en base.

Une tâche récurrente concerne la modélisation que nous avons réalisée en groupe de travail à raison d'une fois par semaine. Le raffinement de ces modèles nous a permis de réaliser les différents services d'acquisition, de traitement et d'échange des données. La planification des différentes tâches a été réalisée par notre chef de projet Joël Champeau. Lors de nos réunions mensuelles tout le monde était présent pour l'allocation des ressources sur chacune des tâches.

1.5.3 Mon rôle dans le projet

Ma contribution au sein du projet MeDON est de concevoir et prototyper son système d'information. Pour cela, j'ai suivi une véritable démarche projet durant laquelle j'ai fait émerger des idées en cherchant à :

- Identifier les besoins des utilisateurs finaux susceptibles d'être comblés en fonction de la taille de notre équipe projet et du temps qui nous était alloué
- Identifier les problèmes que nous devons résoudre
- Identifier la qualité ciblée de nos productions en fonction de nos objectifs

De l'analyse de ces situations, j'ai pu identifier des stratégies, planifier mon travail et contribuer à la mise en œuvre du projet global.

Pour veiller à la bonne conduite des tâches projet qui m'étaient assignées, j'ai donc eu à tenir trois rôles dans ce projet :

1. Concepteur architecte
2. Urbaniste
3. Analyste-développeur

Issu d'une formation d'analyste développeur, et ayant reçu une formation CNAM de concepteur architecte, j'ai acquis mes compétences d'urbaniste en auto-formation et par des échanges journaliers avec mes collègues du laboratoire. Ces échanges m'ont permis de faire profiter à ce projet d'un savoir faire diffus au sein de notre laboratoire. En contrepartie, ce projet a permis de profiter d'un cas d'étude concret pour certaines de nos activités de recherche en modélisation.

Tout au long du projet j'ai communiqué avec les différents acteurs de l'équipe projet IDM à laquelle j'appartiens, acoustique passive(AP), ainsi qu'avec nos partenaires scientifiques. Ces réunions nous ont permis de faire des revues d'avancement par rapport à nos objectifs individuels et collectifs. Cela a également permis de nous aligner sur les contraintes identifiées par nos partenaires français et anglais.

En qualité d'ingénieur de recherche, j'ai fait participer plusieurs groupes d'étudiants à nos travaux en leur proposant des sujets liés à nos besoins en développement. Nous avons ainsi étudié plusieurs pistes dans la thématique du laboratoire, du projet et de mes travaux. Les étudiants ont fourni un travail qui pour la plupart a pu être intégré aux différentes versions du démonstrateur, au cours du temps.

Conception et réalisation

La démarche menée visait à organiser de bout en bout le bon déroulement du projet pour répondre aux besoins précis spécifiés par nos homologues anglais et cela dans les délais impartis conformément à l'emploi du temps en annexe figure 3.

1.5.4 Planning prévisionnel

Dans le cadre de mon mémoire d'ingénieur j'ai suivi le planning défini en figure 1.5.4.1. Ce GANTT m'a permis de cadrer l'avancée de mon étude et de visualiser la progression de mes travaux. J'ai pris un peu de temps sur la première partie de façon à bien cadrer mon projet.

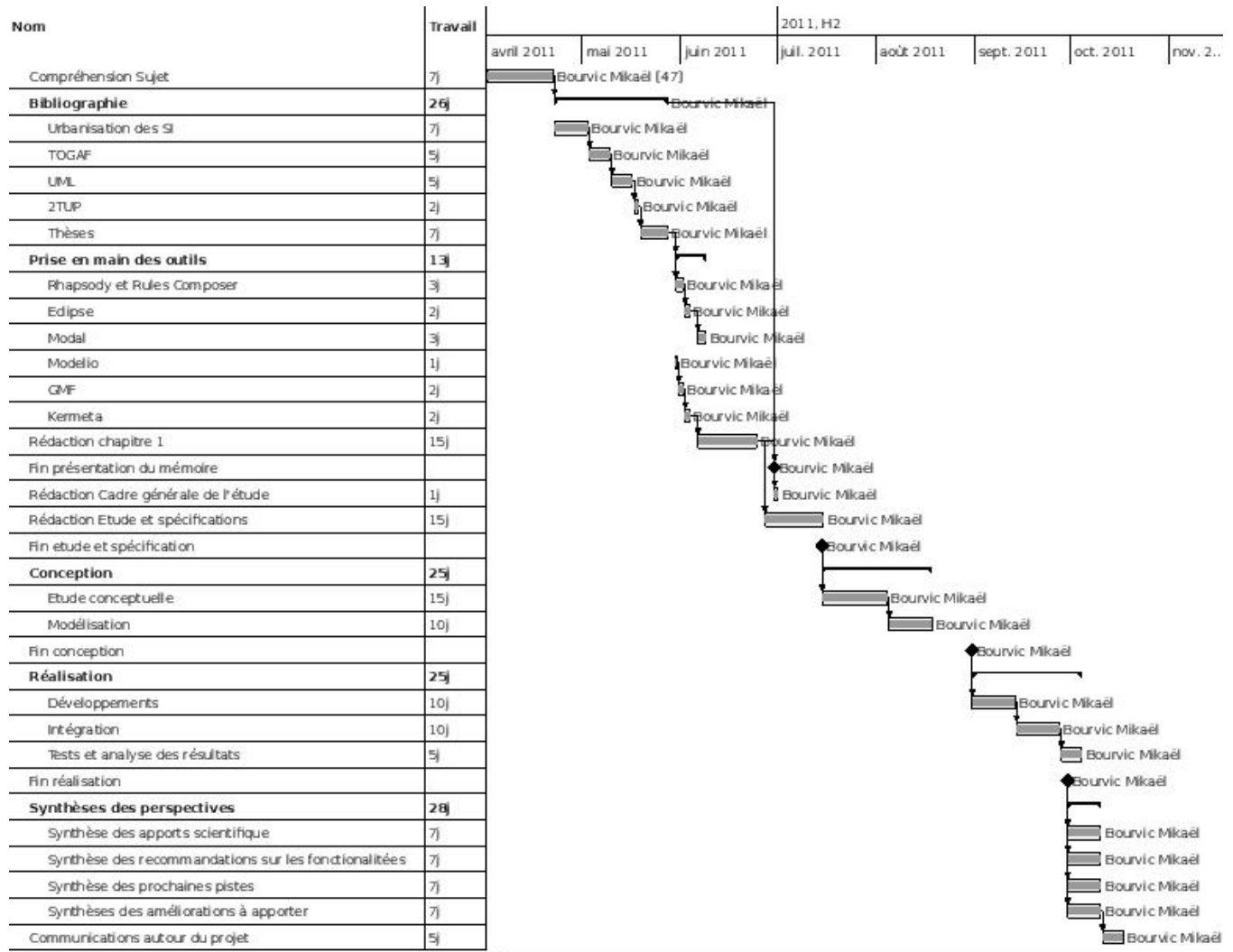


FIG. 1.5.4.1: GANTT du projet, réalisé avec l'outil Planner

Conclusion

Ce chapitre d'introduction a permis de définir le contexte et le sujet de ce mémoire. Les objectifs et l'organisation globale du projet ont été présentés afin de cadrer le travail qui a été mené en entreprise.

Ce chapitre introduit le cadrage général de l'étude qui présente les différentes études liées à ce mémoire et au projet MeDON.

Chapitre 2 - Cadre général de l'étude

Objectif du cadrage général de l'étude

J'ai mené cette étude dans un contexte scientifique à travers le travail de recherche mené tout au long du processus allant de l'identification des besoins à la production d'un prototype. Par ailleurs, ce projet nous a permis de former des étudiants sur de nouvelles technologies en leur proposant une thématique concrète pour des expérimentations.

Dans sa première partie, ce chapitre présente le cadre général de l'étude en synthétisant l'état de l'art au début du projet par l'étude de l'existant.

Par ailleurs, nous avons cherché à apporter une valeur ajoutée à cet existant ce qui m'a permis de mettre à contribution plusieurs étudiants pour que nous puissions étudier davantage de pistes, communiquer sur le projet, et potentiellement gagner du temps sur nos délais de réalisation. La seconde partie s'axe donc sur l'encadrement de mes "projets étudiants" dont est issue une partie de l'architecture du prototype actuel.

2.1 ÉTUDE DE L'EXISTANT

Ma première action sur le projet a été de me documenter sur le concept d'observatoire, de rechercher et d'étudier des implémentations existantes. J'ai établi une documentation et exposé à l'équipe les différentes technologies utilisées. Nous avons alors travaillé ensemble afin de chercher à créer de la valeur ajoutée par rapport aux observatoires existants. Notre approche fut donc de proposer un système d'information capable de communiquer avec des capteurs intelligents dans la mesure où ils embarquent de l'électronique et du logiciel les rendant aptes à interagir les uns avec les autres et avec l'environnement dans lequel ils seront déployés. Le développement de ces capteurs s'intègre dans un autre projet nommé "SmartSensor".

Parmi les observatoires existants qui ont retenu mon attention, il est nécessaire de citer le projet Neptune⁵ Error! Reference source not found. Canada (Fig 2.1.1) dont le responsable, M. Benoit Pirenne est venu nous présenter le concept le 11 février 2010 afin de poser les bases d'une collaboration. A cette occasion j'ai eu la chance de pouvoir m'entretenir avec lui afin d'en apprendre davantage sur les raisons des choix technologiques qu'ils ont effectués pour la mise en place de leur infrastructure. Leur objectif est le même que le nôtre, étudier l'activité des fonds marins. Leur projet dispose de fonds conséquents leur ayant permis de mettre en place une architecture de capteurs couvrant 200 000 kilomètres carrés de fonds marins le long d'un réseau de fibres optiques. Le rendu au grand public est très abouti et il permet de piloter des robots immergés à travers Internet moyennant les droits d'accès adéquats.

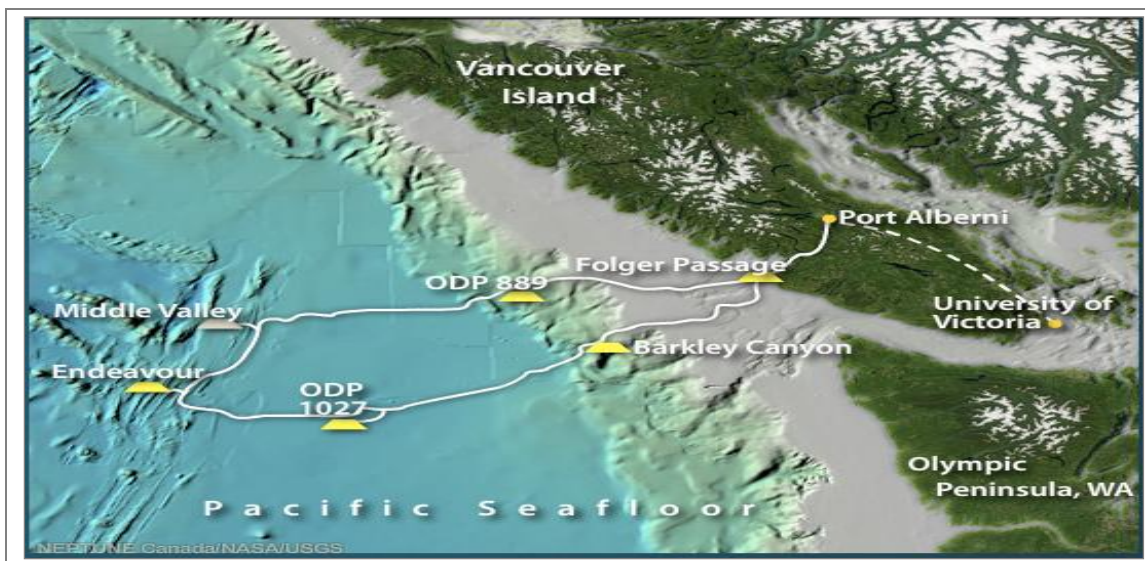


FIG. 2.1.1: Réseau NEPTUNE

Leurs robots immergés sont également capables de se recharger au contact de bornes immergées, et de se rendre sur les lieux d'un événement, par exemple une secousse tellurique ou une éruption sous-marine.

⁵

NEPTUNE est l'acronyme de North-East Pacific Time-series Undersea Networked Experiments

Nous pouvons cependant noter que leurs capteurs ne disposent pas d'intelligence embarquée. Toutes les informations sont remontées sur le continent, et c'est de là que le pilotage est opéré. C'est sur ce point que notre approche diffère et la raison pour laquelle ils ont souhaité nous rencontrer.

Le deuxième point de cette étude à ce stade du projet fut la découverte des standard IOOS⁶.

The Integrated Ocean Observing System (IOOS®) is a federal, regional, and private-sector partnership working to enhance our ability to collect, deliver, and use ocean information. IOOS delivers the data and information needed to increase understanding of our oceans and coasts, so decision makers can take action to improve safety, enhance the economy, and protect the environment. (Source: <http://ioos.gov>)

IOOS(Fig 2.1.2) est une source d'information considérable des standards de fonctionnement possibles pour l'échange d'information entre différents observatoires. Le service d'alerte que j'ai intégré à MeDON est inspiré du service SAS d'IOOS. Le service d'alerte en question permet de prévenir des abonnés lorsque des événements ou des combinaisons d'événements sont détectés par les capteurs. Ces alertes sont alors remontées vers l'utilisateur sur différents canaux de communication (DECT, SIP, Mail, IRC, GoogleMap, GoogleAgenda, SMS,...).

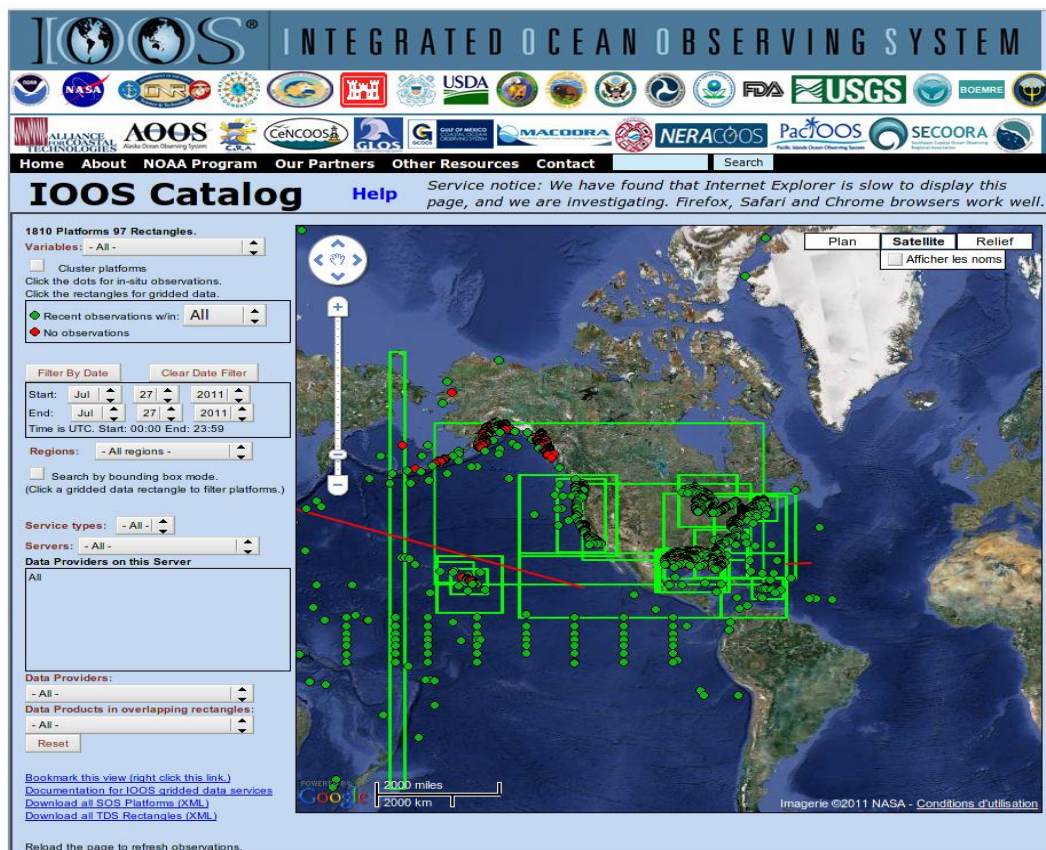


FIG. 2.1.2: Aperçu de l'IHM IOOS

⁶ IOOS est l'acronyme de Integrated Ocean Observing System

2.2 SPECIFICATION TECHNIQUES DE BESOINS

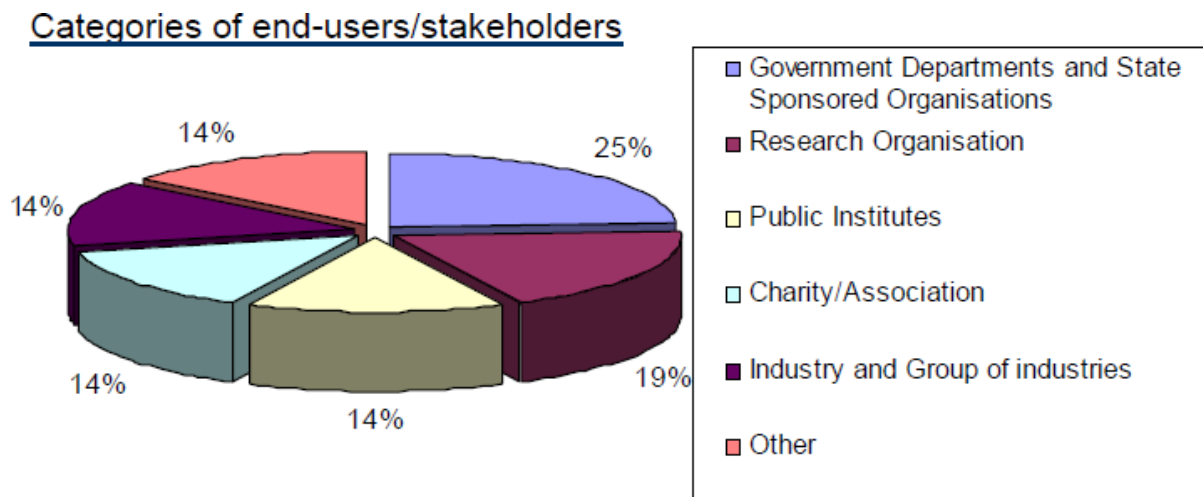
Les principales actions devant être menées lors du projet MeDON concernent les parties prenantes et les utilisateurs finaux.

Voici ces actions :

- Etablissement d'un groupe de parties prenantes et d'utilisateurs finaux concernés par le projet.
- Organisation de groupes de travail interactifs de part et d'autre de la Manche pour mieux comprendre les besoins des utilisateurs finaux.
- Développement d'une stratégie de veille environnementale pour répondre aux besoins des utilisateurs finaux.

Les membres ont donc été identifiés et des groupes de travail ont été créés.

Une étude a été réalisée des deux côtés de la manche pour identifier les utilisateurs finaux qui pourraient être intéressés par le déploiement de capteurs sur l'observatoire MeDON.



Cela nous a permis d'identifier collectivement différents domaines de recherche et une liste de besoins techniques que nous avons priorisés pour la réalisation d'un premier prototype. Dans le cadre de cette première réalisation, il n'était pas possible de répondre à toutes les demandes, notamment de l'acquisition temps réelle de flux vidéo en continu issu de capteurs immergés. Nos contraintes réseau ne nous le permettent pas, et la géo-localisation du site n'est pas adapté à ce type d'observation en raison de l'indice de visibilité.

Parameter to monitor	Nb of selections
Waves	15
Marine mammals	15
Hydrology	14
Tidal resources	13
Sea level rise / storm surges	12
Noise pollution	12
Water quality	12
Pelagic Biodiversity	12
Physical disturbance	11
Benthic Biodiversity	11
Changes in bottom water hydrography	10
Eutrophication – nutrients concentration	10
Ocean Acidification	10
Harmful Algal Blooms (HAB)	9
Microbial activity	8
Benthic bioactivity	8
Fish - Recruitment	8
Fish - Migrations	8
Productivity & Particle Flux	6
Recruitment	6
Oil or chemical spill	6
Particle Dynamics. Organic & Inorganic	7
Chemical Pollution	4
Waste dumping	4
Persistent Organic Pollutants	3

Tableau 2.2.1 : Paramètre à observer

Les besoins identifiés sont décrits dans le tableau 2.2.1. Nous avons construit notre modèle de données en identifiant le type de donnée pour chacun de ces besoins.

2.3 ENCADREMENT DE PROJETS ETUDIANTS

L'encadrement d'étudiants tout au long de ce projet fut une expérience humaine et technique très enrichissante.

Les sujets des différents projets étudiants sont :

-  Génération automatique de code de tests unitaires à partir de modèle UML (Laura Lahaye)
-  Base de données applicative, affichage et services Web (Didier Serre)
-  Annuaire pour la découverte de services Web fournis par les capteurs (Fabrice Galinier)
-  Développement d'une application Android dans le cadre des observatoires marins (Vincent Gadoulet, Thomas Laurentin, Laurent Mattina)
-  Etude des besoins réseau dans le cadre du projet MeDON (Ouchtouban Younes)
-  Génération de code SMS (Smart Modeling for Sensor) vers services Web (François Ngako)
-  Développement d'un client SAS (Quentin Le Doledec)

Ces travaux ont contribué à la construction du système d'information de MeDON autour d'une architecture à base de services Web.

Les objectifs étaient de garder une cohérence d'ensemble, encourager la communication entre les équipes, respecter les délais et au final rendre le projet moins complexe. Il s'agissait principalement de management opérationnel, à savoir organiser les processus en termes d'activité et d'évolution du système d'information. J'ai également participé activement aux différents développements et à leur intégration dans le système en place.

Certaines tâches se sont avérées être plus laborieuses que d'autres où ont permis de soulever des difficultés techniques que nous n'avions pas imaginés. Certains développements se sont également révélés être obsolètes suite aux évolutions des spécifications de nos partenaires. Néanmoins, ils ont tous participé à l'étude et la mise en place de l'observatoire tel qu'il existe aujourd'hui.

Conclusion sur le cadrage de l'étude

Le cadre générale de l'étude permet de faire une réduction de périmètre sur le domaine d'étude couvert par ce mémoire. Il est donc maintenant possible d'effectuer une étude sur l'état de l'art et identifier les contributions que l'on peut apporter.

Première partie

Étude

Chapitre 3 - État de l'art

Objectifs de l'étude

Mon étude bibliographique a consisté en l'étude des différentes technologies à partir de la liste des fonctionnalités offertes par les observatoires marins existants.

A partir de cette étude, j'ai identifié les besoins des communautés scientifiques en termes de système d'informations.

Cette section concerne également l'état de l'art en matière d'ingénierie dirigée par les modèles. L'étude bibliographique synthétise donc une petite partie de la connaissance actuelle de notre communauté sur ce domaine.

Enfin, cette section portera sur mon étude conceptuelle qui a guidé le choix de la méthodologie spécifique retenue. Elle portera également sur l'analyse conceptuelle qui m'a permis de faire un choix d'architecture matérielle pour notre banc de développement (Cf chapitre réalisation).

3.1 L'EXISTANT

En 2009, notre équipe partenaire « acoustique passive » fut à l'initiative de notre participation au projet MeDON. Leur besoin était d'étudier les phénomènes acoustiques sous-marins. Il m'a alors été demandé de réaliser une étude présentant les technologies de l'époque en matière d'observatoire marin. Il s'agissait d'une activité de veille technologique afin d'étudier les méthodes employées par les observatoires existants pour acquérir et diffuser de l'information.

J'ai donc mené cette étude en commençant par répertorier les différents observatoires existant dans le monde puis en listant pour chacun les technologies déployées. Cela m'a permis de réaliser une analyse multicritères permettant d'avoir une vision globale de l'état de l'art en matière d'observatoire. Cette étude nous a alors permis de sélectionner les technologies que nous pourrions à notre tour déployer.

Dans le cadre de mon activité au sein du laboratoire, j'ai proposé et mis en place un wiki afin de nous permettre de réaliser un travail collaboratif. L'étude réalisée autour du projet MeDON fut la première à être déposée sur cette plate-forme et a servi d'exemple aux membres de l'équipe pour la mise en place d'autres projets. Cette étude est maintenant complétée par tous les membres de l'équipe au fur et à mesure de notre veille technologique. Un extrait du tableau de synthèse qui contient des pointeurs vers les différentes études est disponible Figure 3.1.1

FIG. 3.1.1: Extrait du tableau de synthèse sur les Observatoire

Tableau de synthèse 2						
Nom observatoire	Site Web	Type (Observatoire/BDD/...)	Données en ligne	Données en temps réel	Rapport	
Alfred Wegner Institute	http://awi.de/en/research/	Observatoire	N	N		
Advanced Real time Earth Monitoring system in the Area	http://www.jamstec.go.jp/sodc/top_e.html	Observatoire	O	O	arena_obs.doc	
Bonne Bay Observatory	http://bbo.physics.mun.ca/background	Observatoire	O	N	borne_bay_observatory.doc	
Census of Marine Life	http://www.coml.org/	Interface	N	N		
Center for bioacoustic	http://bioacoustics.tamucc.edu/index.html	en construction	O	N		
Chesapeake Bay Observing System	http://cbos.org	Observatoire	O	N		
DODS	http://www.gomoos.org/data/dods.html?b=A01&l=seabird&d=wind_speed&f=Show+datasets	Base de données	O	N		
Field Research Facility	http://frf.usace.army.mil/	Observatoire	O	O		
Gulf of Maine Ocean Observing System	http://www.gomoos.org/	observatoire	x	x		
Integrated System for Holistic Multi-channel Acoustic Exploration and Localization	http://www.pmel.noaa.gov/vents/acoustics/whales/ishmael/	logiciel	N	N		
IOOS	http://ioos.gov/	Base de données	O	N		
Listening network	http://www.listenforwhales.org/netcommunity/Page.aspx?pid=430	Observatoires	N	O	right_whales_boston_obs.doc	
Long-term Ecosystem Observatory	http://marine.ruigers.edu/cool/leo15.htm	Observatoires	N	N		
Neptune Canada	http://www.neptunecanada.ca/	Observatoires	O	O	neptune.doc	
OBSEA	http://www.upc.edu/cdsarti/OBSEA/info/intro/intro_eng.html	Observatoires	O	O		
Observations Registry	http://www.obsregistry.org/map2009/map.php	Base de données	O	N		
Office of Naval Research	http://www.onr.navy.mil/	Interface	N	N		
Openloos	http://www.openloos.org/	Interface	N	N		
OOSTelthys	http://www.oostelthys.org/	ServicesWeb	N	N		
Martha's Vineyard Coastal observatory	http://mvcodata.whoi.edu/cgi-bin/mvco/mvco.cgi	observatoire	O	O	martha.doc	
Monterey Accelerated Research System (USA)	http://www.mbari.org/mars/	observatoire	O	O	monterey_accelerated_research_system.doc	
Neptune Canada	http://www.neptunecanada.ca/	observatoire	O	O		
New Millennium Observatory	http://www.pmel.noaa.gov/vents/nemo/	observatoire	O	O	new_millennium_observatory.doc	
Real-Time Forecasts for Central and Northern California	http://www.nrimry.navy.mil/coamps-web/web/cenclal	Base données	O	O		

Au cours de cette étude préliminaire j'ai acquis une expertise du domaine, qui a inspiré et guidé ce chapitre. L'état de l'art va donc concerner plus spécifiquement mon analyse sur le monde de l'IDM, les systèmes d'information et comment nous avons conçu notre système de composants communicants pour les observatoires marins en étant guidé par les modèles et en ayant en tête une volonté de prendre en compte les préceptes d'urbanisation des systèmes d'information.

3.1.1 Le monde de l'IDM

Les récents développements dans les techniques de transformation de modèles et de métamodèles ont contribué à susciter l'intérêt d'un plus grand nombre de développeurs/concepteurs vers des approches guidées par les modèles. Il est vrai que l'utilisation de telles approches augmente les temps de conception des applications mais cet investissement est compensé par le temps gagné dans le développement global de l'application, la qualité des architectures déployées et le prix de la réutilisabilité (et support collaboratif) des logiciels créés dans le cadre de gros projets (distribués).

Déplacer l'accent du code vers le modèle permet aux entreprises de se concentrer sur leur domaine métier, leurs processus, services et les interfaces de collaboration entre ces services sans avoir à se préoccuper des choix techniques des infrastructures physiques ou logicielles sous-jacentes.

Cette frontière entre MOA⁷ et MOE⁸ est ainsi mieux tracée et tous deux peuvent utiliser le même support pour leurs communications.

L'OMG définit quatre niveaux de modélisation.

Au niveau M0, le modèle définit le monde réel ou le système modélisé (ex : un logiciel).

Modéliser un système d'information c'est cartographier au sein de différents niveaux d'abstraction la diversité des éléments du monde réel ou abstrait. La sémantique du terme 'abstraction' est importante ici. La modélisation d'une abstraction nous donne aussi un modèle. Le spectre de la modélisation ne se limite donc pas au monde réel. Nous parlons ici du niveau M1.

La métamodélisation consiste à fournir la définition du langage de modélisation sous forme de modèle également. Cette définition abstraite et rigoureuse représente une spécification pour les différentes implantations du langage défini. Ce modèle de définition représente un métamodèle de niveau M2.

Le niveau de métamodélisation repose également sur la définition d'un langage qui permet l'expression des métamodèles, c'est le niveau M3. Ce niveau est suffisant concis et expressif pour être réflexif, c'est-à-dire la capacité de s'autodéfinir.

⁷ MOA : Le maître d'ouvrage, désigne le donneur d'ordre au profit duquel l'ouvrage est réalisé

⁸ MOE : Le maître d'œuvre désigne une personne ou entité chargée de la conduite opérationnelle de travaux.

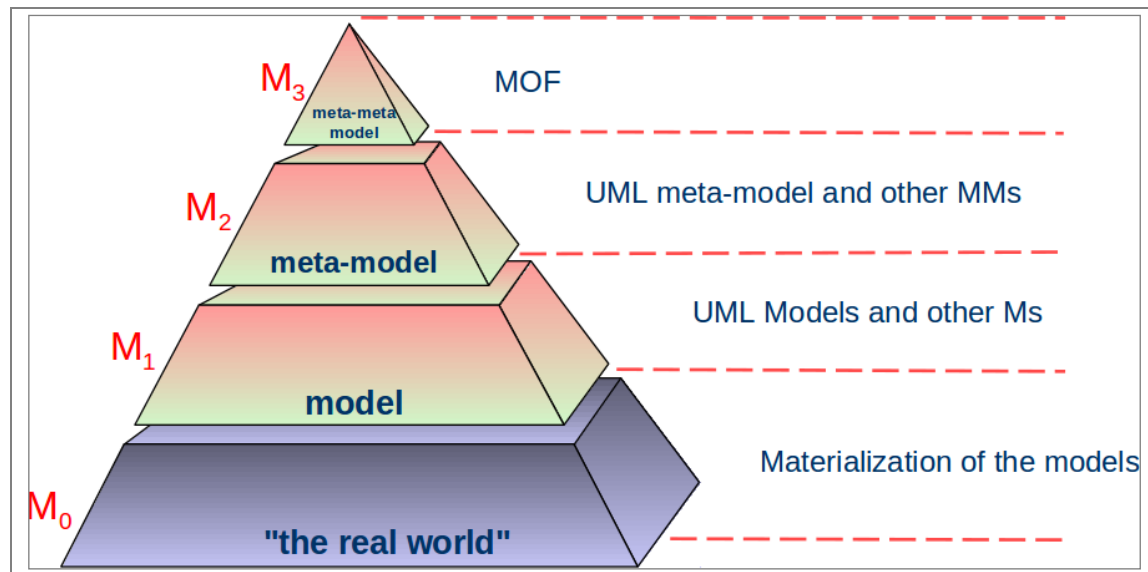


FIG. 3.1.1.1: The 4 Level Paradigm

Les concepts peuvent être exprimés par différents modèles à travers les différents niveaux d'abstraction du MOF au monde réel (Cf Figure 3.1.1.1 : *The 4 Level Paradigm*).

Sur cette base, l'ingénierie dirigée par les modèles offre en théorie des avantages non négligeables en termes de maintenance et d'évolution des systèmes d'information des entreprises et plus particulièrement pour les infrastructures orientées services. En pratique, de nombreuses difficultés doivent encore être résolues pour permettre une application directe de l'approche.

Dans le but de résoudre ces problèmes, différentes avancées ont été nécessaires sur la définition des modèles :

-  Objectifs métiers
-  Interopérabilité fonctionnelle
-  Interopérabilité sémantique⁹

et dans les méthodes d'ingénierie :

-  Concepts des designs dans les systèmes distribués¹⁰
-  Langage
-  Métamodèles
-  Profil
-  Cadriciel de spécification

⁹ L'interopérabilité entre les composants d'un système d'information distribué peut se définir comme la capacité de ces composants à échanger des services et des données. Comme le montre le rapport Fieschi[Fie09], la gouvernance de l'interopérabilité sémantique est au cœur du développement des systèmes d'information. Ce rapport met cependant l'accent sur les systèmes d'information liés à la santé.

¹⁰ Les technologies liées aux systèmes distribués ont évolué de façon significative ces dernières années. Le livre [DS01] présente ces évolutions et met l'accent sur l'Internet, les intranets, la mobilité et le middleware.

Dans le cadre de MeDON et des systèmes embarqués en général, le développement peut nécessiter de prendre en compte des contraintes de temps réel ou d'architecture spécifique ce qui peut soulever divers types de problèmes. Ceci est particulièrement vrai lorsque les équipes de développements ont à prendre des décisions touchant à des architectures très spécifiques ou à des éléments non fonctionnels. Dans le cadre de MeDON cela pourrait être les paramètres liés à la consommation énergétique ou l'empreinte mémoire de certains algorithmes.

Ces dernières années ont vu un regain d'activité sur ces problématiques car des modèles de domaines permettent désormais de capturer des architectures spécifiques et les informations non fonctionnelles de manière formelle. Ces modèles peuvent être structurés en différents niveaux d'abstraction et présenter des systèmes construits en couches dans lesquels les aspects spécifiques aux plateformes sont séparés des architectures et aspects non fonctionnels. Le système final, global, est obtenu ultérieurement par transformation des modèles. Les architectures hétérogènes peuvent alors être décrites dans des modèles organisés (Fig 3.1.1.2).

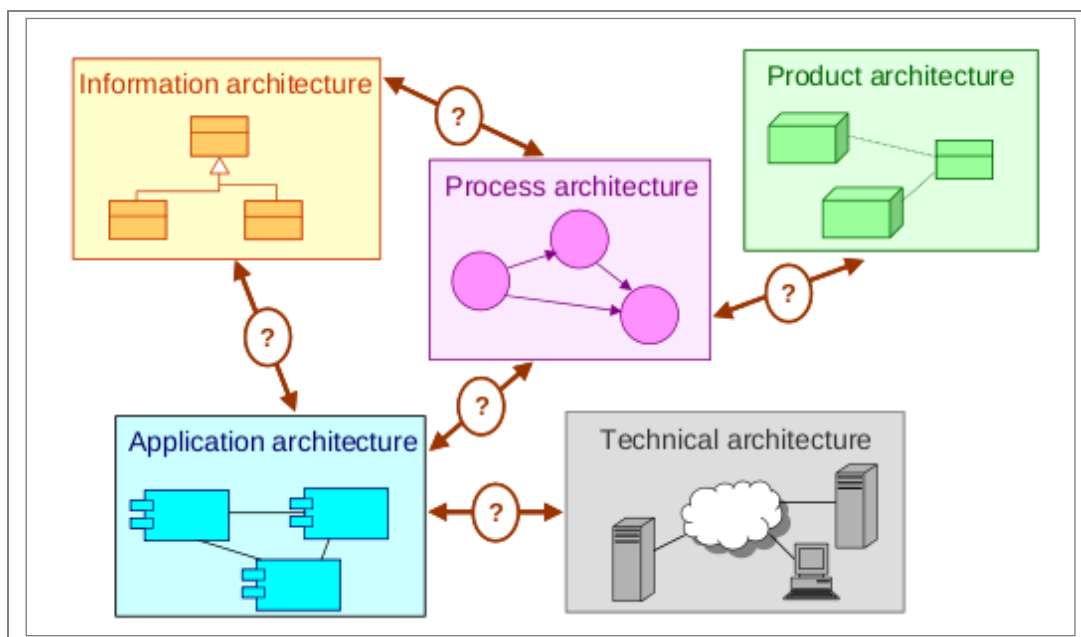


FIG. 3.1.1.2: Domaine d'architecture hétérogène

Le projet ArchiMate[LtAt04] apporte des éléments de réponse à ces questions en proposant une approche d'architecture intégrée qui décrit et permet de visualiser les différents domaines d'activité d'un système et leurs relations. Cela permet donc aux acteurs de prendre conscience de l'impact et des effets de leurs choix.

SOA

Les architectures orientées services (SOA) [JH08] font référence au paradigme qui se concentre sur le fait de maximiser le partage, la réutilisabilité et l'interopérabilité des données d'entreprise à travers le réseau. Pour ce faire, il est nécessaire d'utiliser un middleware¹¹, des standards et l'intégration de fabriques¹².

¹¹ En architecture informatique, un middleware (anglicisme) est un logiciel tiers qui crée un réseau d'échange d'informations entre différentes applications informatiques. Le réseau est mis en œuvre par

Les concepts, terminologies et sémantiques des SOA permettent une meilleure communication entre la MOA et la MOE sur des sujets techniques. Dans le cadre de mon travail, l'étude de ces ontologies m'a permis de mieux architecturer mes modèles et mieux choisir mes terminologies.

Le document [Ope10] m'a notamment été très utile pour la réalisation d'un métamodèle de système d'information. A l'aide de ce métamodèle j'ai pu réaliser un modèle conforme du système d'information de MeDON. Ces deux modèles m'ont permis de générer une partie de l'architecture des classes PHP que j'ai ensuite utilisées dans le cadre de mes services Web côté client. Il existe bien évidemment des librairies et des plugins Eclipse qui permettent de générer ce code de manière automatique, mais le fait de le faire à partir des modèles rend selon moi l'approche plus homogène. L'idée est également d'acquérir une chaîne complète permettant d'aboutir à une solution fonctionnelle à partir des modèles. L'utilisation d'outils externes ad hoc est donc à éviter. Par ailleurs, partir des modèles est pour moi une preuve que ceux-ci contiennent toute l'information nécessaire.

J'ai également développé un outil pour réaliser une animation Web basée sur l'invocation des services décrits dans les modèles conformes. Visuellement, il était alors possible d'apprécier l'état du système et donc de ses différents composants.

BPM ou la gestion des processus métiers

Derek Miers[LtAt08] affirme que de nos jours, 20% du travail d'un employé est structuré. Les 80% restants du travail répondent à des processus complètement dynamiques. BPM[KG08] est là pour apporter une solution permettant de structurer les processus. De manière pragmatique, cela consiste à spécifier de manière très concrète l'orchestration de différentes tâches pour en faire de vrai processus optimisé.

Le site du Ceisar est une mine d'informations sur le sujet. Celles-ci sont disponibles librement et l'on peut notamment obtenir le livre blanc BPM08[CEI08]. L'objectif du Ceisar est de communiquer avec des termes simples et de produire des livres blancs visant à formaliser et à développer les meilleures pratiques en Architecture d'Entreprise.

Les Master Data Management, MDM

Les *Master Data* sont les informations critiques échangées entre les différents acteurs d'une organisation. Cela représente toutes les informations qui peuvent se présenter sous forme de catalogues, graphiques, listes. Bien souvent chaque entreprise manipule ces informations avec des outils particuliers comme ERP ou CRM et les stockent dans des formats propriétaires. Des problèmes se posent lorsqu'il s'agit de partager ces informations avec des partenaires.

l'utilisation d'une même technique d'échange d'informations dans toutes les applications impliquées à l'aide de composants logiciels.

¹² La fabrique est un patron de conception créationnel utilisé en programmation orientée objet. Elle permet d'instancier des objets dont le type est dérivé d'un type abstrait. La classe de l'objet n'est donc pas connue par l'appelant. (Source : Wikipédia)

La méthodologie 2TUP

2TUP est une autre méthodologie de développement découplant les aspects logiciels des aspects matériels. Cela répond à nos besoins dans la mesure où l'observatoire devra être déployable par définition sur tous types d'architectures.

2TUP (2 track unified process, prononcez "toutiyoupi") est un processus de développement logiciel qui implémente le Processus Unifié.

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à :

- Chapitre 1 - identifier : les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système

- Chapitre 2 - produire le cahier des charges

- Chapitre 3 - modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui. Sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent).

Le processus s'articule ensuite autour de 3 phases essentielles :

- ℞ une branche technique

- ℞ une branche fonctionnelle

- ℞ une phase de réalisation

Le cycle de développement suit donc ce Y tel que représenté par le diagramme figure 3.1.1.3.

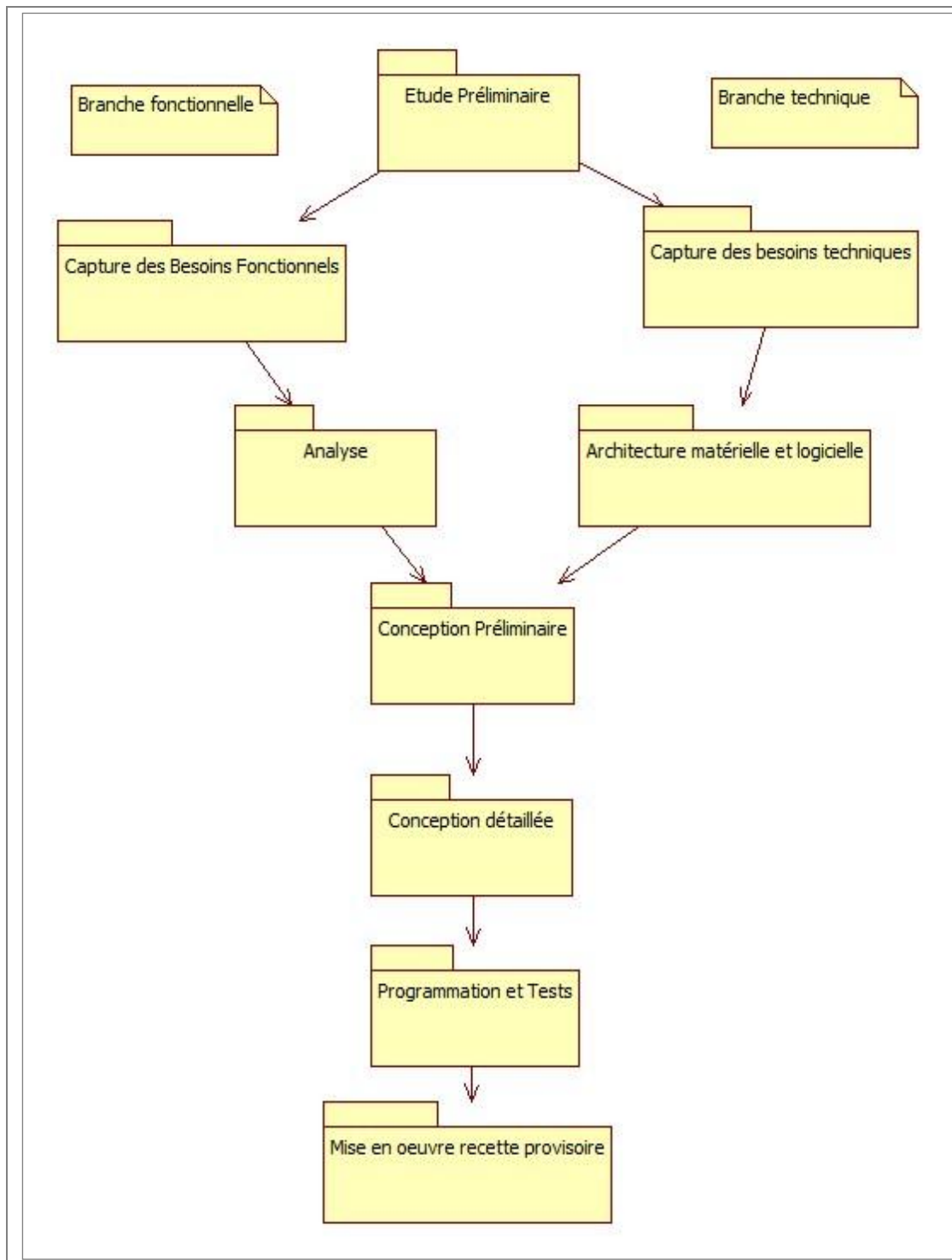


FIG. 3.1.1.3: Le cycle en Y

La norme Spem

SPEM¹³ est un langage flexible permettant de modéliser des processus logiciels et de faciliter le partage des informations. Les développements logiciels devenant de plus en plus complexes en termes de technologies à maîtriser il permet aux équipes de s'appuyer sur une modélisation du processus de production répondant à leur problématique. C'est autant un métamodèle de conception de processus qu'un cadre conceptuel fournissant les outils et concepts pour :

- R modéliser
- R documenter
- R présenter
- R gérer
- R rendre concret le développement

L'ingénieur processus, le responsable du projet ou le gestionnaire de programmes dispose alors d'un langage permettant de modéliser un référentiel processus au sens large.

Cette norme de l'OMG est définie dans [OMG08]. Ali Koudri, qui a réalisé une thèse au laboratoire STIC/IDM, s'est également intéressé à cette norme pour fournir une méthodologie de conception qui capitalise un savoir faire dans la maîtrise des processus. Se rendant compte de certains manques lorsqu'il a tenté de faire converger l'ingénierie des processus et ceux de l'IDM, il a alors réifié les concepts d'objectifs et de plans sous les termes respectifs d'intention et de stratégie à travers l'outil Modal décrit dans le prochain paragraphe.

La méthodologie Modal

Modal est un outil développé par l'équipe IDM du pôle STIC. Cet outil permet de modéliser des processus et des méthodologies afin de réaliser des modèles exécutables. Il s'agit d'une extension de SPEM afin de pouvoir définir des objectifs, des plans de processus de développement ainsi que des activités de conception et d'analyse pour la conception d'un système.

Deux concepts sont très importants, les intentions et les stratégies. Selon les besoins nous effectuons des choix raisonnés. Ce sont ces raisons qui sont désignées par le concept d'intention. Les intentions définissent donc les objectifs méthodologiques. Une stratégie vise à satisfaire une intention. C'est un ensemble de ressources ou d'activités interconnectées dont la collaboration permettra l'atteinte de l'objectif.

Dans le domaine des systèmes d'information et selon [ABCR99] une intention est un objectif à un instant donné. Un processus est donc un ensemble d'intentions reliées par des stratégies.

¹³ SPEM Software Process Engineering Metamodel - que l'on peut traduire par métamodèle d'ingénierie des procédés logiciels

MDA

Le MDA est un leitmotiv de l'équipe IDM. La définition officielle est la suivante :

L'architecture dirigée par les modèles ou MDA (de l'anglais Model Driven Architecture) est une démarche de réalisation de logiciels, proposée et soutenue par l'OMG. C'est une variante particulière de l'ingénierie dirigée par les modèles (IDM, ou MDE pour l'Anglais Model Driven Engineering). D'autres variantes de l'IDM ont été développées, par exemple par Microsoft (DSL Tools). (Source : Wikipédia)

Le MDA définit trois niveaux d'abstraction dans les modèles. Ces niveaux peuvent être raffinés par transformation (3.1.1.4). Si l'on considère le concept de modèle de système d'information comme une abstraction d'un système physique il est alors possible de prendre en compte quatre mécanismes, le CIM (computation independant model=processus métier), PIM (platform independant model=analyse fonctionnelle détaillée), PDM (platform description model=description de la plateforme) et PSM (platform specific model=conception technique) qui sont définis dans le tableau 5.

Mécanisme	Modèle du MDA
Prise en compte de la plate-forme, dans notre cas l'architecture réseau déployée	PSM
Description du comportement et des éléments applicatifs indépendamment de la plate-forme	PIM
Description d'une infrastructure technique	PDM
Modèle du système de type boîte noire représentant l'environnement du projet	CIM

TAB. 5: Mécanismes du cadre d'architecture du MDA

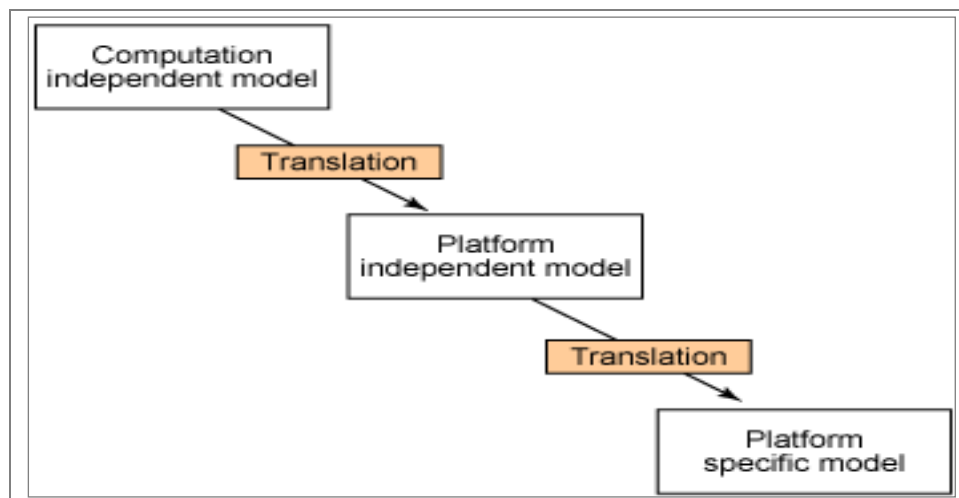


FIG. 3.1.1.4: Transformations du MDA

Les translations entre les modèles sont détaillées dans la figure 3.1.1.5.

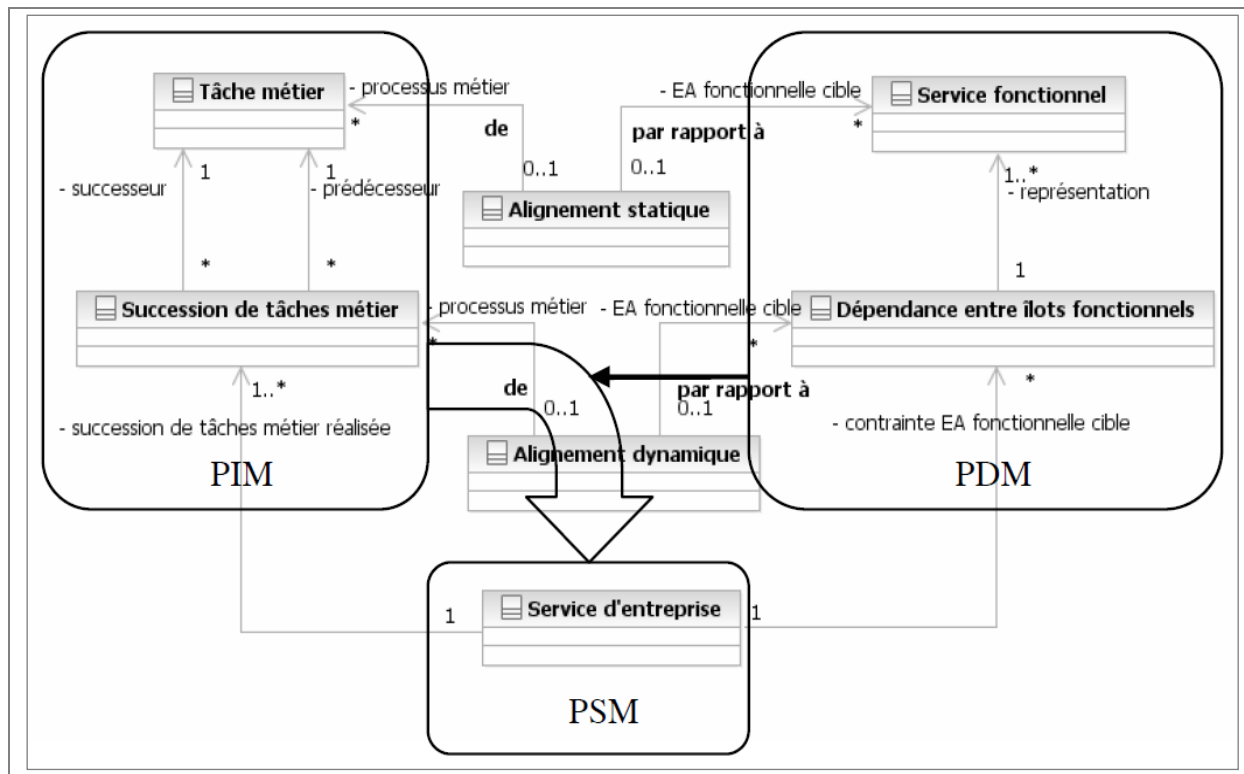


FIG. 3.1.1.5: Relations entre PIM, PDM et PSM

De nombreux travaux ont été réalisés en ce sens dont la thèse de Ali Koudri[KOU10] laquelle m'a inspiré par la description de Modal et des différents niveaux d'abstractions identifiés à savoir AML (Abstract Modeling Level), EML (Execution Modeling Level) et DML (Detailed Modeling Level) du processus MoPCoM dont je vais décrire en trop peu de mots l'idée générale.

La capacité croissante des circuits intégrés programmables permet aujourd'hui l'intégration de systèmes complets. Cette intégration incombe à des ingénieurs d'expertise variée comme des analystes système, des architectes de circuits, des ingénieurs hardware et logiciel, des ingénieurs de tests & validation et d'industrialisation.

Le projet MoPCoM¹⁴ est une plate-forme servant à la conception de composants électroniques programmables à performance très élevée, adaptée au codage vidéo et au traitement d'images. Cette conception ne repose plus sur le partitionnement classique hardware/logiciel mais sur des transformations de modèles approuvées, répétables et semi-automatiques.

Le concept d'abstraction permet de regrouper au sein d'un même niveau des caractéristiques et concepts communs afin d'en simplifier la manipulation. Le découpage en plusieurs niveaux permet de décrire des concepts selon différentes perspectives.

¹⁴ MoPCoM est un projet qui vise à exploiter l'ingénierie dirigée par les modèles pour la spécification et la conception des systèmes par la fiabilisation des processus basés sur les modèles. (Source : <http://mopcom-i.gforge.inria.fr>)

Le niveau AML permet de spécifier des choix d'implantation en termes de concurrence ou de modèle de communication. Il s'agit d'implantation sur des modèles de plate-forme correspondant à la notion de modèle de calcul (MoC¹⁵). Ce niveau permet donc de faire apparaître les notions de ressources concurrentes, partagées ainsi que les liens de communication entre ces blocs et les différentes allocations réalisées sans se préoccuper des problèmes de communication/synchronisation dont pourrait souffrir la plate-forme d'exécution.

Le niveau EML permet de définir une plate-forme d'exécution pour le niveau AML sans contraintes chiffrées en termes de ressources matérielles telle que la quantité de mémoire ou la puissance processeur par exemple. Seule la topologie des services nous intéresse à ce niveau.

Le niveau DML donne une spécification suffisamment détaillée de la plate-forme permettant la génération de code VHDL pour la partie matérielle ou C pour la partie logicielle par exemple.

Dans le cadre de mon travail, je ne peux pas utiliser ces notions directement. Je peux cependant porter ces concepts aux services de mon système d'informations et le décomposer en couches abstraites similaires. La connaissance que j'ai acquise par cette étude m'a donné une vision plus concrète de MDWorkBench¹⁶ et du MDA. L'exemple de MopCom-soc est donc pour moi un exemple concret de ce que je peux réaliser, de la méthodologie que je dois suivre, et je peux maintenant essayer de m'appuyer comme Ali Koudri sur 2TUP, SPEM et MDA, pour concevoir, réaliser et manipuler mes modèles et contribuer à cet état de l'art.

Méthodologie TOGAF

TOGAF (The Open Group Architecture Framework) est une méthode d'architecture (figure 3.1.1.6) permettant de favoriser le développement global des architectures d'entreprise. Il prône la collaboration entre les différentes MOA et MOE afin de rendre plus agile les développements.

¹⁵ MoC pour Model of Computation. Un MoC c'est le modèle d'un système tel qu'on le connaît. Un modèle d'exécution est le modèle qu'a le système quand il s'exécute, c'est-à-dire composé du MoC en plus du support d'exécution de la plate-forme. Wikipédia donne la définition suivante : « En ingénierie dirigée par les modèles, un MoC définit le comportement d'un systèmes en fonction du comportement de chacun de ses composants. »

¹⁶ *MDWorkBench est un environnement de développement intégré basé sur Eclipse. Il permet la conception et la manipulation de modèles, leur transformation ainsi que la génération de code et de documentation (Source : société Sodius).*

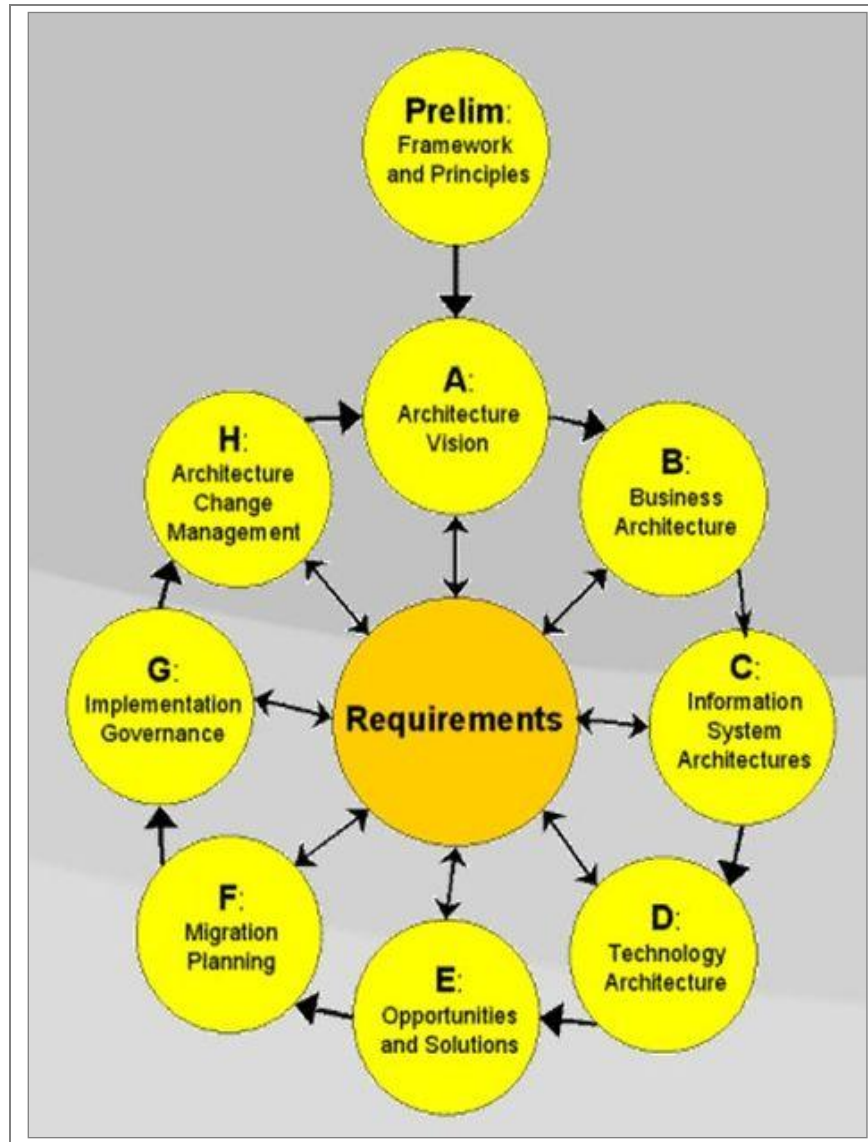


FIG. 3.1.1.6: TOGAF Architecture Development Method

Cette méthodologie de développement recentre le travail de l'urbaniste au sein de l'entreprise.

3.1.2 Les Systèmes d'informations

L'état de l'art des systèmes d'informations porte sur de nombreuses perspectives. Nous ne nous intéresserons ici qu'aux nouvelles technologies et méthodologies que l'on pourra combiner pour proposer une solution méthodologique et un prototype. Le monde de l'IDM nous apporte l'ingénierie pour transformer des modèles abstraits en artefacts liés au monde réel.

De nos jours, les systèmes d'informations sont au cœur de toutes les entreprises car ils sont censés apporter des solutions de communication, de partage et donc de travaux collaboratifs au sein de grandes structures potentiellement distribuées.

Un système d'information peut être vu comme un artefact capitalisant les faits passés et présents d'une organisation. Différentes catégories de méthodes se différencient pour en réaliser des modèles et prendre en compte une plus large gamme de demandes utilisateur :

- ↳ Les méthodes cartésiennes
- ↳ Les méthodes systémiques
- ↳ Les méthodes assistées par des outils

Ces méthodologies (état de l'art en 1990) apportent des perspectives [CR90] permettant de construire une nouvelle gestion de ces systèmes.

Pour répondre à ces problématiques, j'ai axé ce chapitre sur deux éléments principaux :

- La thèse de Mounir Lallali [LAL09], *Modélisation et test fonctionnel de l'orchestration de Service Web*
- L'activité d'urbanisation de systèmes d'information (Figure 3.1.2.1)

L'urbanisation de système d'information apporte un savoir-faire relatif aux CASE¹⁷ et intègrent la solution à ce dont souffraient les méthodes systémiques en termes de gestion des processus dynamiques. D'un point de vue système d'information, cela consiste à prendre en compte le fait que les interactions entre ses composants vont évoluer dans le temps.

Modélisation et test fonctionnel de l'orchestration de service Web

L'arrivée du premier ordinateur date de 1946 avec L'ENIAC. En 1949 le premier réseau existait, nommé SAGE. En 1955 le premier réseau informatique à but commercial a été créé. Dès lors, l'outil informatique a commencé à répondre à des demandes d'échange de données. La notion de services est apparue bien plus tard et ce n'est que récemment que l'on commence à architecturer nos systèmes autour de modèles exécutables de haut niveau.

Mounir Lallali expose BPEL¹⁸ afin d'établir la conformité de l'implémentation d'un service composé par rapport à sa spécification. Dans l'évolution de l'informatique, il s'agit maintenant de permettre aux machines de s'échanger les informations directement. Cette notion d'IMM (interface machine machine) permet de concevoir des services composés.

Une interface qui s'auto-décrit est une solution pour ce type de besoins. Le langage XML est aujourd'hui un standard largement utilisé dans ce domaine.

Dans le cadre d'infrastructures immergées et d'un système d'information distribué, la modélisation des cas d'utilisation, des diagrammes de séquences et des modèles de données en XML ne suffisent pas à la gestion des processus. Le test et l'orchestration sont, selon moi, deux activités primordiales dans la gestion concrète de telles infrastructures.

Le prototype MeDON peut donc s'appuyer sur ce concept à savoir une architecture SOA reposant sur la notion de services/composants déployables, composables, modulaires et testables.

L'objet du prochain chapitre sur l'urbanisation des systèmes d'information permet de structurer ce besoin pour des systèmes complexes.

¹⁷ CASE pour Computer Assisted Software Engineering (Atelier logiciel)

¹⁸ BPEL est un langage standard d'orchestration de services Web

L'urbanisation des systèmes d'informations

L'urbanisation des systèmes d'information (figure 3.1.2.1) consiste à déterminer les objectifs métiers des acteurs d'une entreprise pour fournir différentes vues permettant de cartographier un système.

Le terme d'urbanisation remonte à 1871 suite au grand incendie de Chicago et la volonté de reconstruire une ville harmonieuse. L'urbanisation au sens informatique consiste à permettre de garantir un alignement entre les objectifs, le métier et l'infrastructure dynamique d'un système dans un environnement contraint. Le maintien de ce dynamisme est la problématique dont souffrent les systèmes complexes et à fortiori les systèmes de systèmes. Il faut donc penser le système à travers différentes vues transverses, dans une logique de réification progressive des modèles.

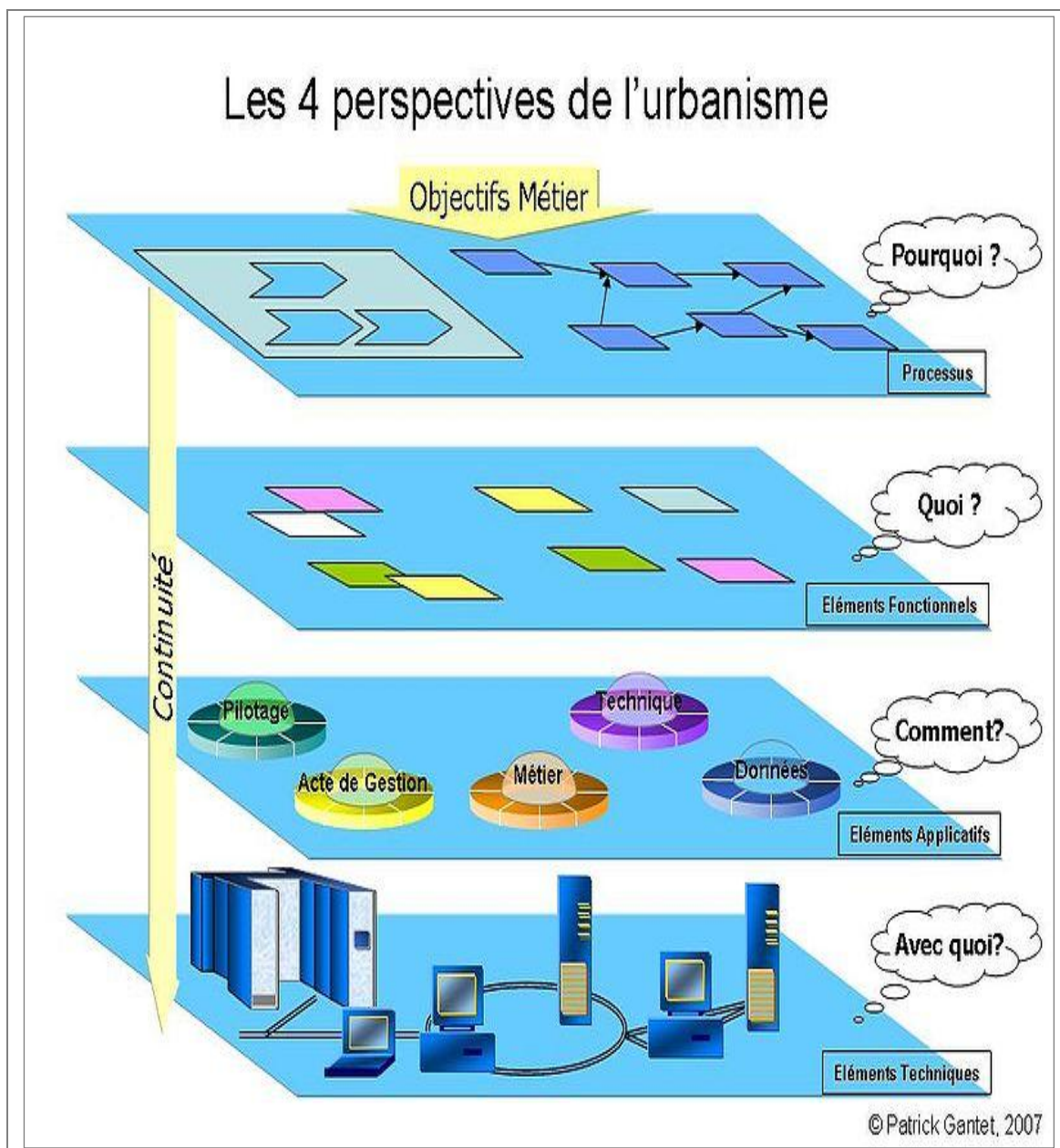


FIG. 3.1.2.1: Les 4 perspectives de l'urbanisme

Le livre orange de Valtech [val07] sur *l'Urbanisation et l'intégration de systèmes* est une compilation de savoir-faire dans le domaine de l'urbanisation. Plusieurs éléments clefs sont à retenir dans la réalisation des modèles des différentes architectures métiers, fonctionnelles, applicatives et techniques telles que représentées en détail dans la figure 3.1.2.2.

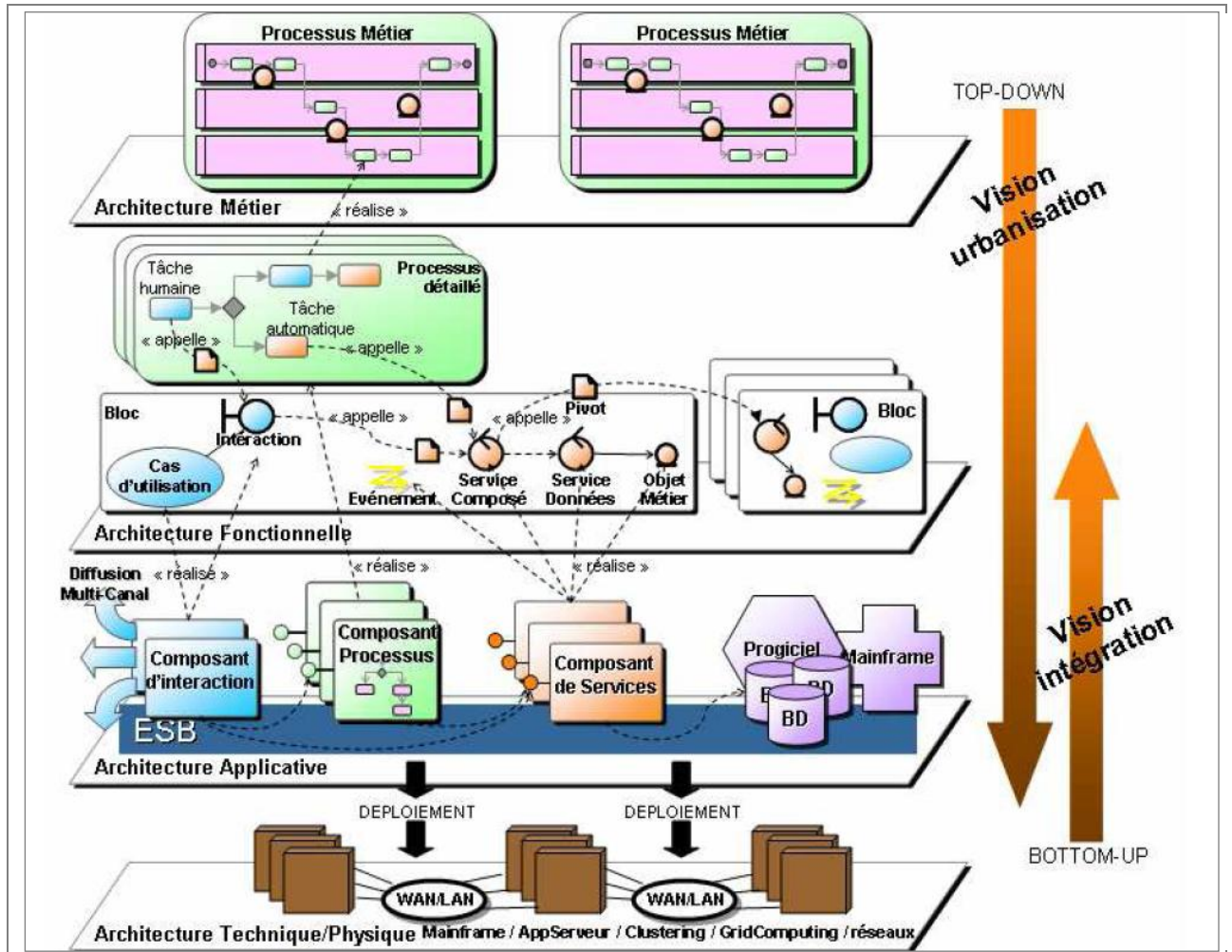


FIG. 3.1.2.2: Les 4 niveaux d'architecture du SI pour une SOA

Le niveau d'architecture métier que je dois modéliser concerne les processus clefs de l'activité d'observation sous-marine. Cela permet de se concentrer sur les acteurs qui jouent un rôle dans les processus considérés. Techniquement, la réalisation du modèle se fait en identifiant les rôles des acteurs. Pour ce faire, il est possible d'utiliser le standard BPMN et de distinguer au sein du modèle des couloirs d'exécution pour les processus. Chaque acteur remplissant une tâche. Différentes stratégies reliant ces tâches.

3.1.3 Les systèmes d'information dans un contexte d'observatoires sous-marins distribués

Le mariage entre MDA et électronique embarquée reconfigurable est en passe de se réaliser. La difficulté réside selon moi dans la collaboration d'agents intelligents. Le fait est que la particularité des observatoires marins est que l'électronique immergée est difficilement remplaçable. A moyen terme, il s'agit donc de pouvoir répondre à de nouveaux objectifs en faisant évoluer les stratégies tel que le décrivait Ali Koudri. Le système d'information est alors un système classique agrémenté de capteurs considérés comme des agents et l'information emprunte des flux à destination de systèmes de calcul. Plusieurs notions apparaissent comme l'acquisition, le traitement, le stockage et la diffusion de ce que l'on considère encore comme le centre du système, l'information et le service.

La notion de système de systèmes est ici à prendre en compte. Un exemple pour illustrer ce propos serait (GEOSS¹⁹), fournit par les Etats-Unis, l'Union européenne, le Japon et l'Afrique du Sud. C'est donc un système de systèmes mondial. Son rôle est l'observation de la terre. Ce projet vise à englober la gamme entière des observations effectuées in situ et par télédétection²⁰.

Si l'on considère le concept de modèle de système d'information comme une abstraction d'un système physique il est alors possible de prendre en compte les trois mécanismes définis dans le tableau 5 : Mécanismes du cadre d'architecture du MDA.

La figure 3.1.3.1 extraite de l'introduction à la sémiotique²¹ de Guy Spielmann propose une schématisation résumée d'un système de systèmes comme plusieurs éléments ayant des interfaces.

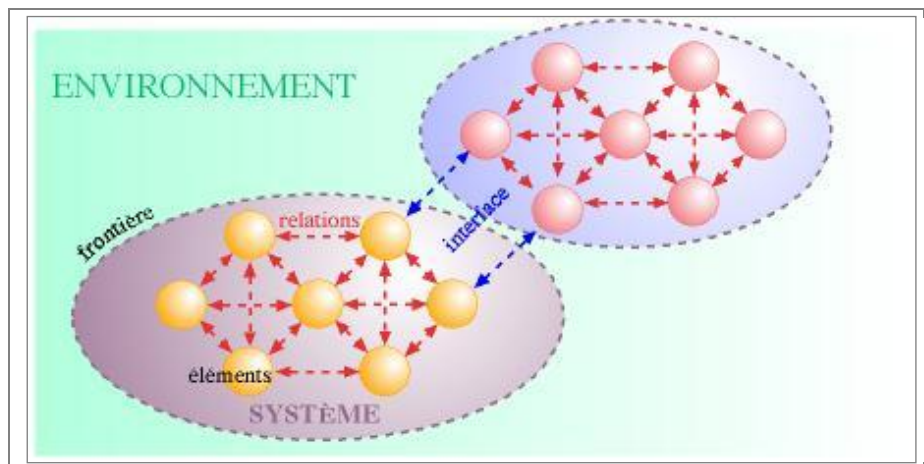


FIG. 3.1.3.1: Traits indispensables pour définir un système de systèmes

¹⁹ GEOSS : Global Earth Observation System of Systems

²⁰ La télédétection capitalise les observations effectuées depuis l'espace et au moyen d'aéronefs

²¹ La sémiotique est l'étude des signes et de leur signification et ses trois dimensions qui sont la sémantique, la syntaxe et la pragmatique nous fournissent des outils nécessaires à l'examen critique des symboles et des informations dans notre domaine

Cette vision simplifiée nous permet de décrire notre architecture sous la forme de sous-systèmes, lesquels peuvent former des hiérarchies par emboîtement (super-système, supra-système et méta-système). Lorsqu'ils ne sont pas hiérarchisés on parle de macro-systèmes, ses éléments étant des micro-systèmes.

Je préfère décomposer ici notre « solution » selon la notion de PSM, de PIM et de CIM.

Le PSM décrit les spécifications concernant une plate-forme spécifique. Ces données devront être utilisées pour implémenter le code exécutable. Dans le cadre de mon étude, j'ai réalisé un PSM sur papier puis un PDM sous Visio (Outil pour la mise au point de schéma liés aux infrastructures informatiques). J'ai mis à jour régulièrement ce modèle en fonction des évolutions que prenait le projet tel que décrit dans le premier chapitre de ce mémoire. Ce modèle m'a permis de communiquer avec les étudiants, mes collègues ainsi qu'avec nos partenaires de l'IFREMER avec qui nous avons travaillé sur l'architecture physique du réseau.

Conclusion sur l'état de l'art

Cette étude a permis d'identifier quelques acteurs publics dans le domaine de l'observation sous-marine. Cette étude nous a confirmé qu'il n'existait pas encore d'observatoire « intelligent » ni de méthodologies permettant de déployer facilement ces systèmes.

Cette étude a également présenté différents concepts qui seront manipulés dans ce mémoire.

Chapitre 4 - Étude conceptuelle

Objectifs de l'étude conceptuelle

Dans ce chapitre je réalise des choix basés sur l'étude du besoin, l'état de l'art en conception de systèmes d'information et l'analyse des objectifs fixés par notre équipe de travail. L'objectif de ce chapitre est d'appliquer les techniques de l'urbanisation à notre concept de système d'observatoires sous-marin.

Je réalise donc une étude conceptuelle basée sur l'état de l'art en matière d'ingénierie par les modèles pour urbaniser l'architecture de notre système d'information d'observatoire sous-marin.

L'étude porte dans un premier temps sur la modélisation UML du système. J'ai alors eu les éléments pour réaliser les modèles des différents niveaux d'architectures.

4.1 ÉTUDE ET CONCEPTION DES MODELES UML

Lors du projet nous avons dû intégrer dans notre conception de nouvelles fonctionnalités ("scope creep") dû à divers événements :

- Ř Le choix du site où seront immergés les capteurs a changé ce qui a remis en question notre diagramme de déploiement
- Ř Nos partenaires scientifiques ont souhaité que dans le cadre de l'observatoire, des flux vidéo puissent être remontés ce qui a induit de nouvelles contraintes en termes de réseaux
- Ř L'architecture des cartes électroniques des capteurs a évolué et certaines briques de code sont devenues inutilisables

J'ai mis à jour nos modèles UML en fonction de ces changements. Cette section présente nos principaux modèles à savoir :

- Ř L'identification des acteurs
- Ř Les diagrammes de cas d'utilisation qui donnent une vision globale du comportement fonctionnel du système.
- Ř Les diagrammes de séquences d'un de ces cas d'utilisation

Notre système d'information doit permettre l'enregistrement et la gestion de données. Il doit également permettre l'accès à ces données, en assurer la traçabilité et l'échange entre les producteurs et consommateurs (Figure 4.1.4.1.1).

Trois types d'acteurs sont amenés à communiquer avec notre système. Ce sont leurs objectifs métiers qui vont guider notre conception. Nous retrouvons cette première étape d'identification des acteurs dans diverses méthodologies comme 2TUP, Merise et l'IDM en général.

- Ř Les administrateurs dont le travail sera de *maintenir l'architecture et l'infrastructure logicielle*.
- Ř Les scientifiques souhaitant *réaliser des traitements*.
- Ř Les utilisateurs publics souhaitant *consulter des données scientifiques*.

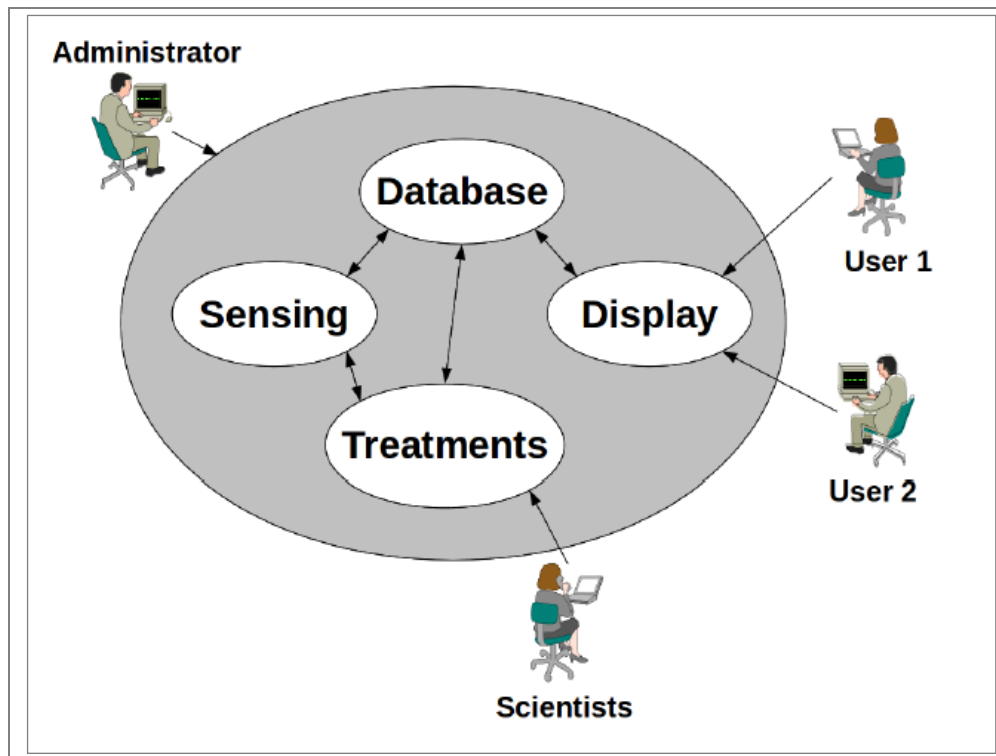


FIG. 4.1.1: Besoins du système d'information MeDON

Sur ce schéma j'ai identifié les liens particuliers entre les composants qui répondent à nos spécifications. Le tableau 4.1.1 en donne une description.

Tableau 4.1.1 : Composants du système

Sensing	Représente un composant réalisant de l'acquisition de données
Treatments	Représente un composant qui réalise des calculs sur les données
Database	La base de donnée répond à la spécification d'archivage des données
Display	Ce composant est chargé respectivement de la présentation et de l'affichage de nos services Web et des données auprès de nos visiteurs et de nos partenaires.

Le composant *Display* intègre donc un quatrième acteur, les machines avec lesquelles nos services vont communiquer.

Plusieurs de nos objectifs métiers peuvent alors être traités. Nous pouvons considérer ces composants comme des boîtes noires et analyser plus en détails certaines interactions.

Une interaction peut être détaillée dans un diagramme que nous appelons *cas d'utilisation*. Le composant *C2-Treatment* par exemple peut être décrit par le diagramme 4.1.4.1.2. Ce diagramme nous présente les interactions de ce composant communicant. Ces deux objectifs sont d'informer

lorsque de nouvelles données ont été traitées et de récupérer de nouvelles données pour pouvoir les traiter. Les algorithmes des scientifiques peuvent alors accéder à ces données.

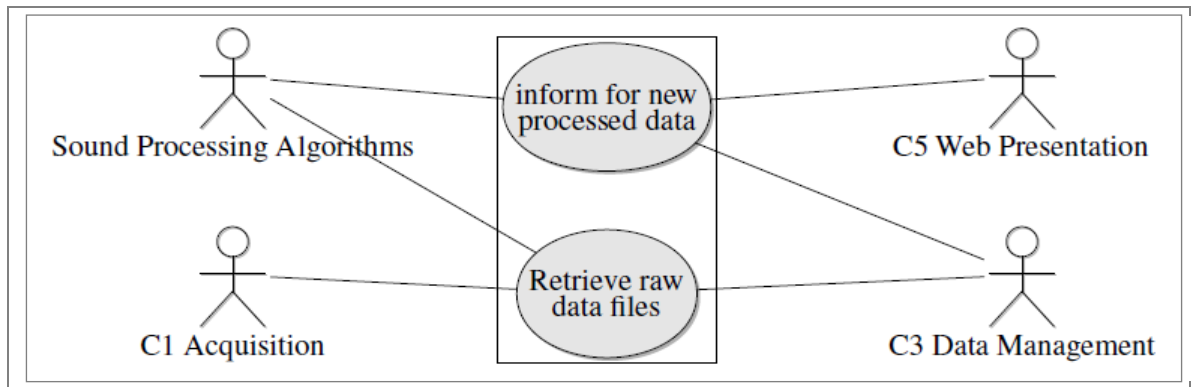


FIG. 4.1.2: Diagramme de cas d'utilisation du composant C2

Les diagrammes de séquence nous permettent de présenter l'orchestration interne de ces processus. Le processus de récupération des données est illustré par le diagramme de séquence de la figure 4.1.4.1.3.

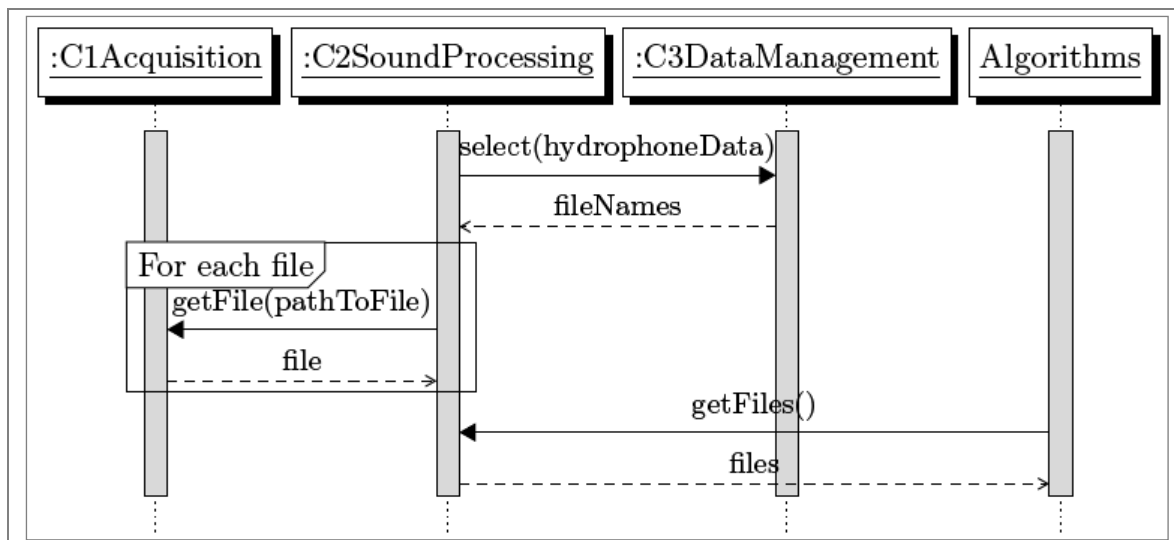


FIG. 4.1.3: Diagramme de séquence du processus de récupération de données du composant C2

4.2 ARCHITECTURE METIERS

L'architecture métier de MeDON telle que représentée figure 4.2.1 reprends nos 3 acteurs principaux :

- R Scientifique, chargé de surveiller, d'analyser et de partager ses résultats.
- R Partenaire, qui accédera aux interfaces de notre système pour interagir avec certains services
- R Architecte, chargé de concevoir, déployer, contrôler puis détruire des services.

Le déploiement d'un équipement²² sous-marin permet d'acquérir de nouvelles données. Ces données brutes sont mises à disposition d'un scientifique via le déploiement d'un service de la part de l'architecte. Cela lui permet de surveiller dans ces données certaines métriques relatives aux phénomènes acquis.

Les capteurs intelligents peuvent alors être recalibrés ou reprogrammés pour collaborer sur de nouvelles tâches.

Les données traitées resteront accessibles pour que :

- R les partenaires puissent effectuer leurs propres traitements
- R le grand public puisse suivre l'activité de l'observatoire

²² Le réseau MeDON sera équipé d'équipements de mesure du son, de la température ainsi qu'une vingtaine d'autres équipements.

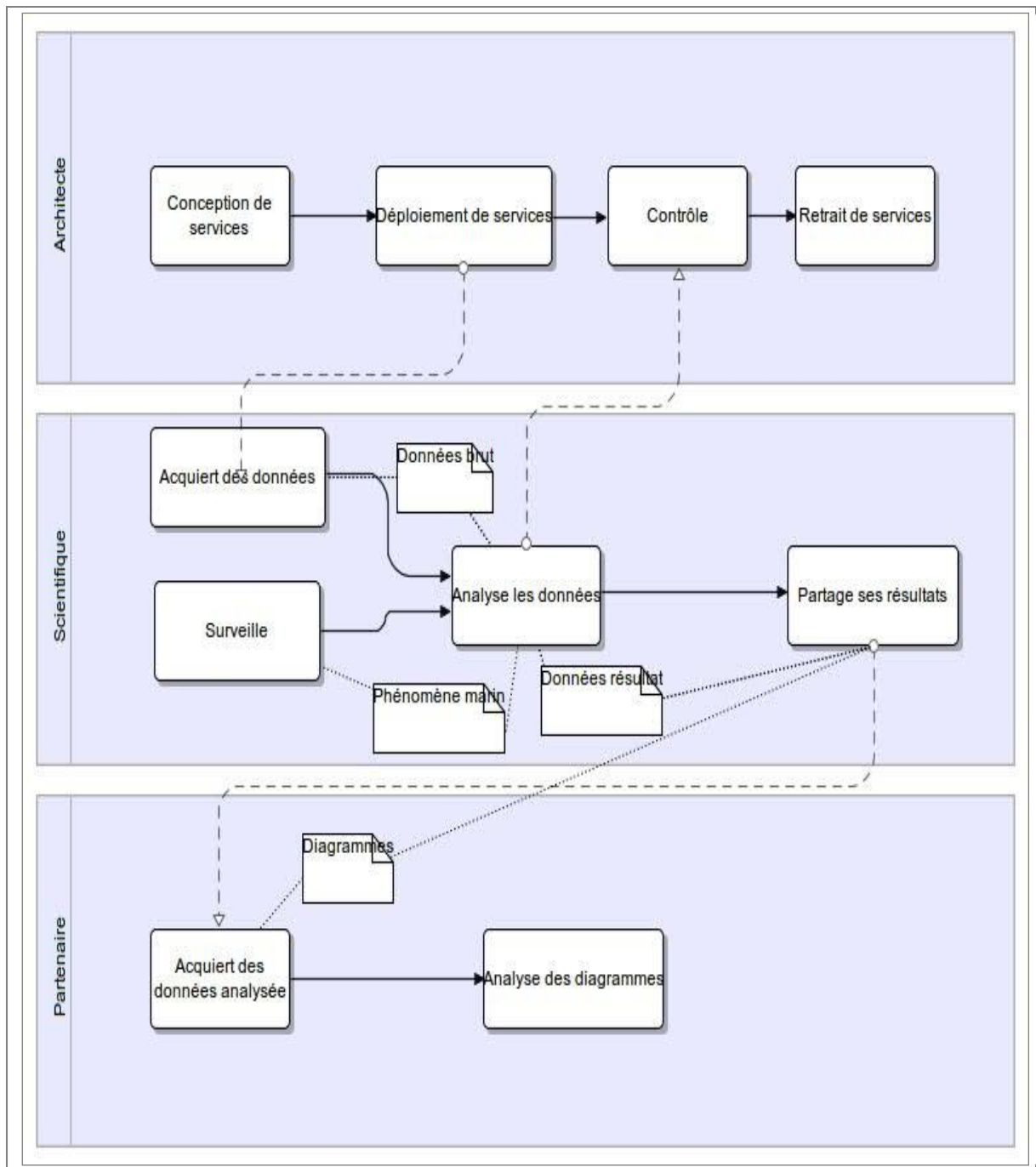


FIG. 4.2.1: Architecture Metier MeDON

Dans le cadre de notre prototype, cette perspective représente les principaux processus et les objectifs que les acteurs doivent pouvoir atteindre. Il s'agit à ce niveau d'abstraction d'un outil d'échange et de communication entre les MOA²³ et MOE²⁴.

²³ Le maître d'ouvrage (abrégié en MO ou MOA) est le donneur d'ordre au profit de qui l'ouvrage est réalisé.

²⁴ Le terme maîtrise d'œuvre (souvent abrégé MOE ou MCE) désigne une personne ou entité chargée de la conduite opérationnelle de travaux.

Au fur et à mesure de la modélisation des processus clefs différents objets métiers émergent. Ils représentent les concepts manipulés par notre système. Cette première architecture fait apparaître quatre objets métiers manipulés : le phénomène scientifique sous surveillance, la donnée brute acquise, la donnée brute traitée et les diagrammes de présentation des résultats pour le grand public. Ce processus peut donc être décrit comme :

Un phénomène sous-marin sous surveillance enregistré sous forme de données brutes pour être analysé.

Les données résultats peuvent alors être partagées.

Le modèle synthétique UML2 de ces objets métiers MeDON est représenté figure 4.2.2.

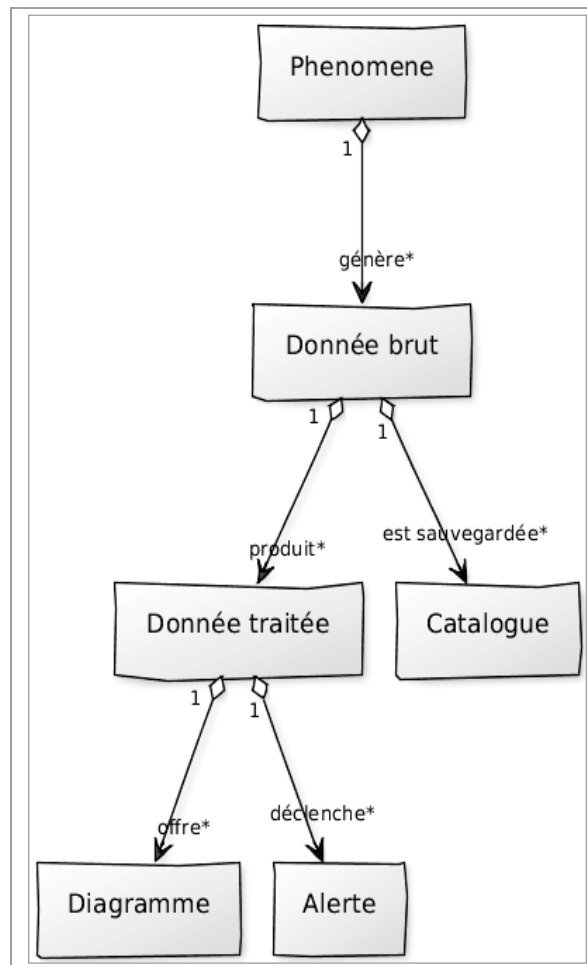


FIG. 4.2.2: Modèle Synthétique des objets métier pour MeDON

C'est sur ce processus clef que nous allons détailler et construire notre architecture fonctionnelle sous forme de blocs. J'ai choisi de détailler cette partie du système de MeDON car elle permet de faire apparaître les 6 blocs fonctionnels de notre système. La démarche est similaire pour les autres processus.

La notion d'alerte de la figure 4.2.2 fait référence au besoin utilisateur d'être averti du résultat de traitements de certaines données traitées.

4.3 L'ARCHITECTURE FONCTIONNELLE

Une architecture fonctionnelle est une vue composée des services de notre système. Ces services vont réaliser un ensemble de tâches métiers conformes à nos objectifs métiers. Les objets métiers que nous avons identifiés peuvent être classés par blocs de gestion fonctionnelle comme l'illustre la figure 4.3.1.

Cette illustration est intéressante car elle nous permet de représenter les 6 blocs fonctionnels de notre système actuel à savoir :

- R la gestion des acquisitions de données brutes, « *Bloc* » *Gestion acquisition*
- R La gestion des traitements pour générer des données résultats, « *Bloc* » *Gestion traitement*
- R La gestion des données pour archivage et management, « *Bloc* » *Gestion données*
- R La gestion de l'extraction de certaines données de l'observatoire, « *Bloc* » *Gestion échange*
- R La gestion de la présentation des données archivées, « *Bloc* » *Gestion présentation*
- R La gestion du pilotage du système « *Bloc* » *Gestion pilotage*

Ces blocs de gestion fonctionnelle peuvent être regroupés en un îlot fonctionnel : « *ilot* » *gestion des composants*.

Chacun des processus Métiers doit être modélisé de la sorte. Et pour chacun des *îlots fonctionnels* identifiés, nous devons réaliser un diagramme de cas d'utilisation. La figure 4.3.1 montre les interactions entre ces cas d'utilisations. Le cas d'utilisation du composant C2 (figure 4.1.2) du chapitre précédent correspond au cas d'utilisation *Traiter données*.

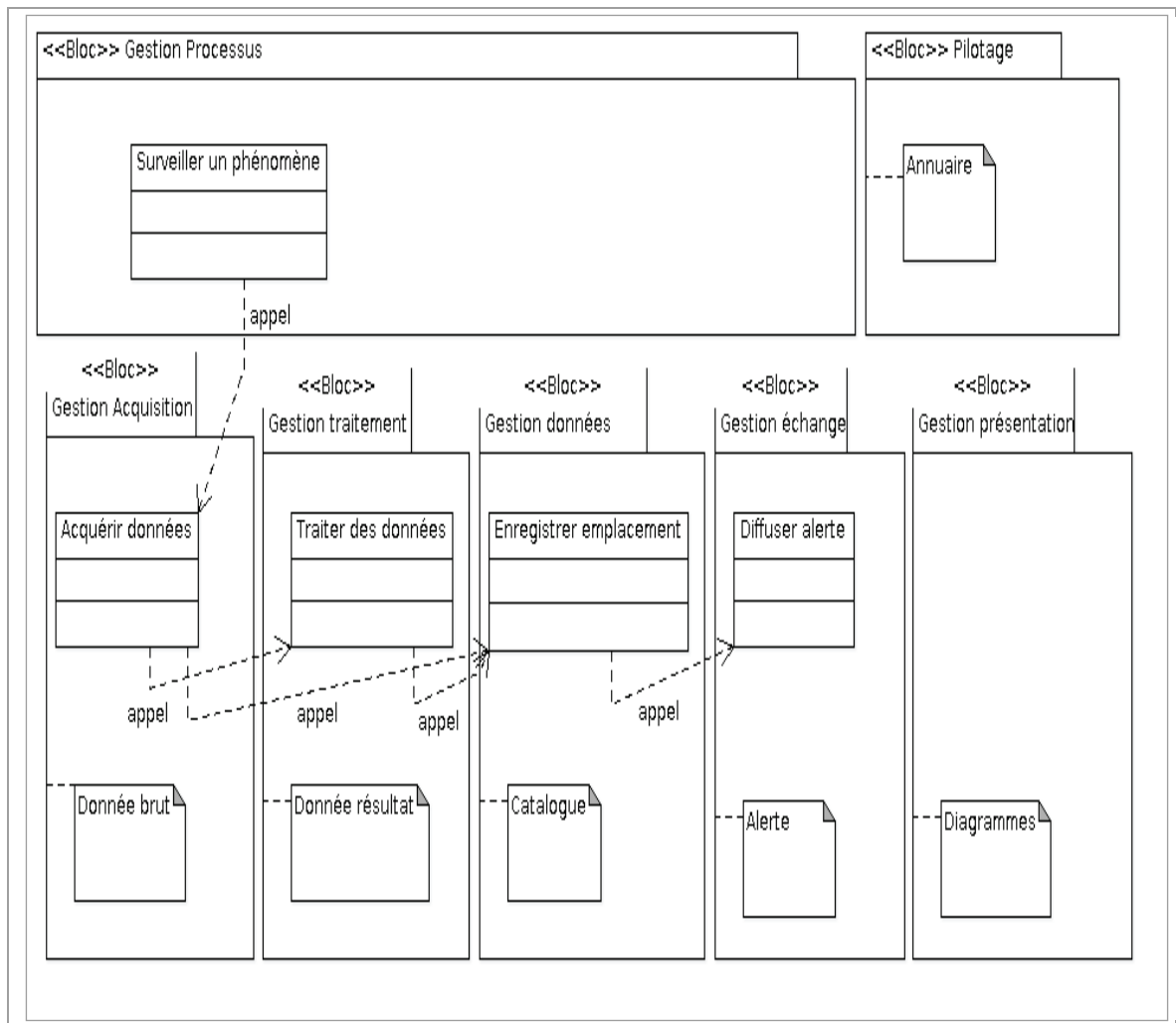


FIG. 4.3.1: Architecture Fonctionnelle MeDON du processus de surveillance de phénomènes

4.4 L'ARCHITECTURE APPLICATIVE

L'architecture applicative figure 4.4.1, cible les différents composants logiciels du SI. Nous devons nous assurer que chacun des concepts de la couche fonctionnelle repose sur une solution applicative. C'est une étape délicate qui peut cependant être prise en main plus facilement en se basant sur des patrons d'architecture existants. L'idée n'est pas de dupliquer une architecture mais au contraire d'améliorer un existant et de proposer des solutions pour améliorer le SI.

Les blocs *gestion échange* et *gestion présentation* reposent tous deux sur le concept d'interaction entre les données de l'observatoire et son monde extérieur. Le premier communique avec des IMM et le second avec des internautes. Nous pouvons donc regrouper deux composants chargés de ces fonctionnalités au sein d'une couche de composants d'interactions. Ce seront deux points d'entrée sous forme de portail, l'un présentant une interface Web, l'autre un annuaire de services Web auto-décrits en WSDL.

L'ensemble des processus peut également être traduit dans le modèle de cette architecture applicative. Nos objectifs métiers traduits en processus et manipulant des objets métier, peuvent s'intégrer dans une couche de *composants Processus*. Ainsi, via le portail Web, il est possible de suivre l'activité de l'observatoire qui s'appuie sur un catalogue de données. L'objet sera appelé « *catalogue de donnée brutes et de données traitées* ». Un processus est chargé d'acquérir ces données et un autre de les stocker. Un autre processus peut se servir lui aussi de ce catalogue pour traiter les données et produire des données de résultats qui pourront être échangées avec l'extérieur. Ces composants de processus s'appuient sur les composants de service, la base de ce patron applicatif orienté : « *composants et services* ».

Les composants de services sont la traduction de nos îlots fonctionnels en termes d'éléments applicatifs fournissant des familles²⁵ de services. Ces différents composants permettent la gestion des données, des traitements et des services. Le composant de gestion des services est particulier car c'est lui qui rend possible le pilotage des autres composants via un mécanisme d'échange d'annuaire. Le diagramme de séquence correspondant à l'enregistrement de C1 auprès de C6 est donné en figure 5.2.1.1 pour illustrer ce mécanisme.

Les ressources applicatives sont composées d'un ensemble de serveurs rendant possible le stockage et le traitement de l'information. Les échanges entre ces ressources doit se faire dans la mesure du possible en utilisant les standards SOAP ou le style d'architecture REST. Nous retrouvons donc les artefacts suivant :

- Un **SGBD** structurant les différentes données et les divers événements du SI au sein d'une base conforme à notre modèle de données
- Des **serveurs Web** communicant via SOAP ou REST avec les capteurs intelligents et rendant possible l'échanges entre les différents services
- Un **serveur de calcul** réalisant les traitements sur les données brutes
- Un composant assimilable à un **EJB** public exportant les données vers nos partenaires
- Un **serveur Web** public rendant possible la présentation des données au grand public

²⁵ Le terme famille représente ici un ensemble de services articulés autour d'une même gestion d'objectifs métiers.

Les différentes ressources applicatives (couloir du bas) ne devraient pas apparaître dans la vue 4.4.1 mais permettent de montrer l'alignement entre les architectures applicative et fonctionnelle. Ces ressources seront détaillées dans l'architecture physique.

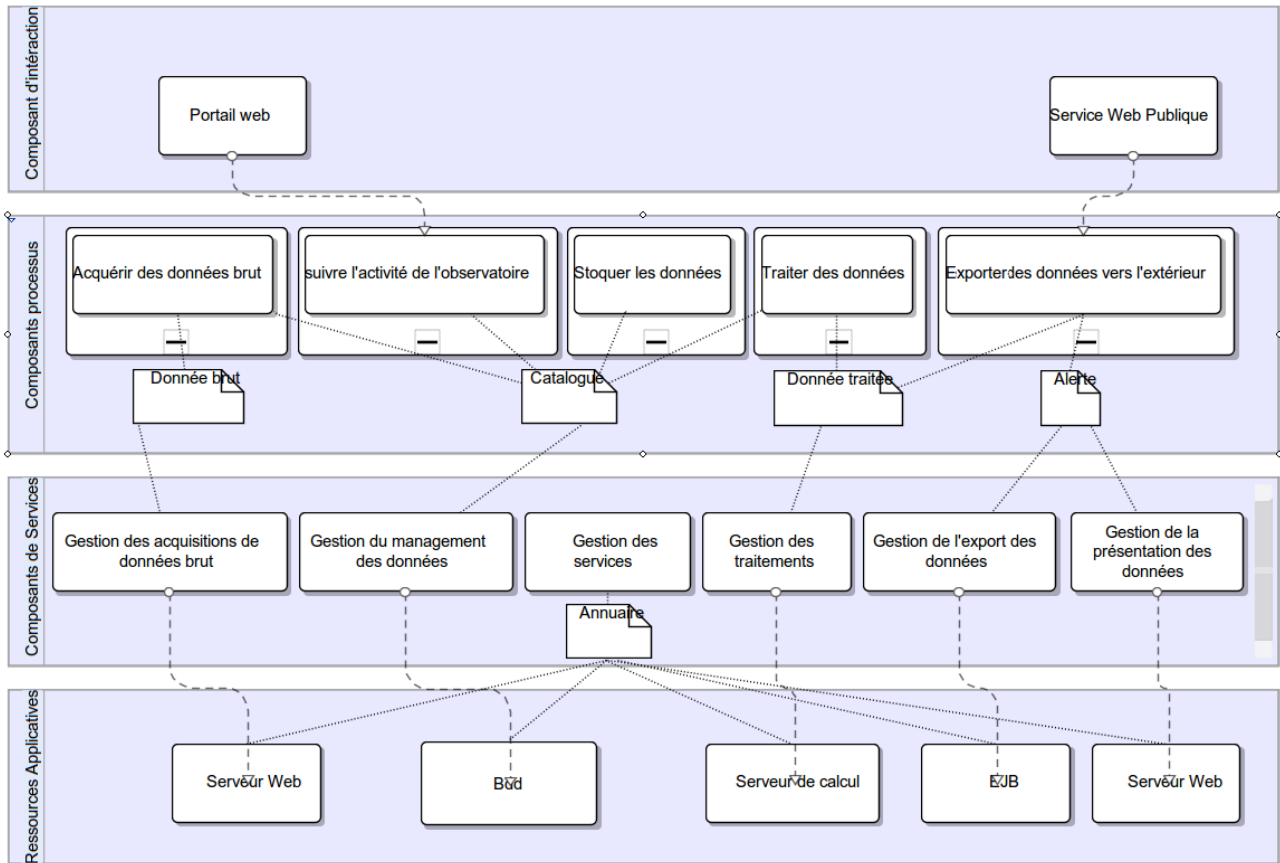


FIG. 4.4.1: Architecture Applicative MeDON

4.5 L'ARCHITECTURE PHYSIQUE ET TECHNIQUE

L'architecture physique²⁶ détaillée figure 4.5.1 est une perspective qui reprend les différents éléments de mon diagramme de déploiement UML. La personne en charge de la conception d'une telle architecture est un architecte matériel.

Cette architecture doit répondre à différentes exigences d'architecture technique tel que décrit dans le tableau 8.

TAB. 8: Exigences d'architecture technique

Exigence	Description
Fonctionnelle	L'architecture doit être alignée avec les fonctionnalités de l'application
Disponibilité / Fiabilité / Plage d'ouverture	Métrique spécifiant les périodes durant lesquelles l'application doit être active
RTO ²⁷ et RPO ²⁸	En cas d'incident ou de désastre, des plans de reprise peuvent ou doivent être mis en place.
Sécurité	Plusieurs domaines sont couverts par cette exigence, la disponibilité, l'intégrité, la confidentialité et la traçabilité.
Performances	Différents éléments sont pris en compte en termes de nombres d'utilisateurs, temps et fréquences de traitements et temps de réponse.
Scalabilité	La scalabilité ²⁹ est la capacité de l'architecture à s'adapter en cas de montée en charge.
Conservation des données	Les capacités de stockage n'étant jamais illimitée, il faut prévoir des mécanismes de purge ou d'archivage.
Modifiabilité	Il peut être nécessaire de devoir effectuer certaines modifications sur le système à chaud. Cette question fut posée dans le cadre du reparamétrage des capteurs immergés. Ce n'est pas une exigence de notre infrastructure terrestre.
Utilisabilité	L'utilisabilité concerne l'ergonomie des fonctionnalités de notre système.

²⁶ Dans le domaine des systèmes d'information, l'architecture physique (ou technique) décrit l'ensemble des composants matériels supportant l'architecture applicative.

²⁷ RTO pour Recovery Time Objective, c'est le temps maximal pour la reprise des activités.

²⁸ RPO pour Recovery Point Objective, c'est la perte maximale de données acceptable après redémarrage.

²⁹ On distingue deux types de scalabilité, horizontale (on rajoute des serveurs) et verticale (on améliore la puissance de traitement)

Nous devons donc aligner nos composants matériels sur nos composants logiciels. Les principaux composants logiciels que je vais décrire sont les virtualiseurs, les systèmes d'exploitation, le SGBD et les serveurs Web. Les autres serveurs applicatifs de notre système d'information (tels que SSH, DNS, SMTP par exemple) ne seront pas détaillé ici.

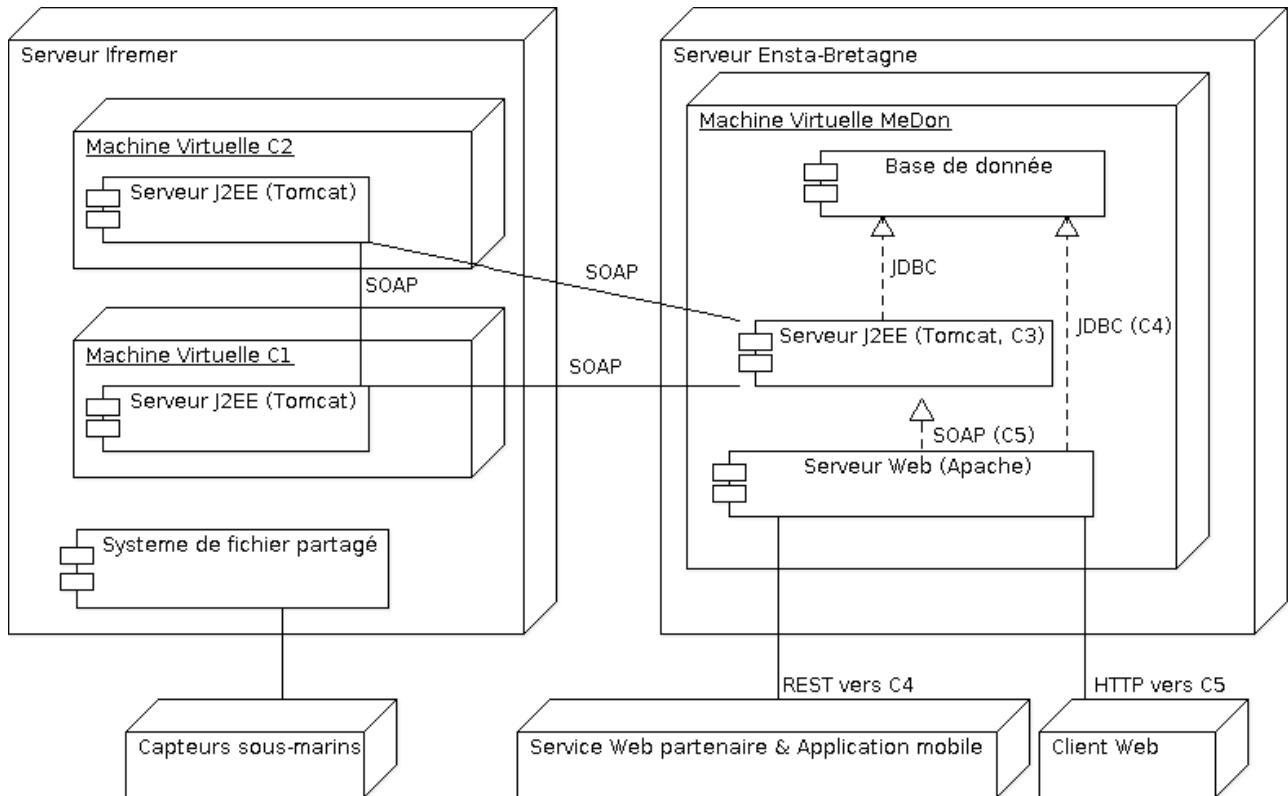


FIG. 4.5.1: Architecture Technique MeDON

4.5.1 La virtualisation

Dans le cadre de nos prototypes, j'ai mis en place cinq machines virtuelles pour isoler nos cinq ressources applicatives.

Différents virtualiseurs étaient éligibles pour ce déploiement. J'ai donc effectué quelques tests, tiré parti de l'expérience d'administrateurs système expérimentés et retenu quatre virtualiseurs principaux :

-  User Mode Linux
-  VirtualBox
-  VMWare
-  Xen

A titre personnel, j'ai beaucoup utilisé *User Mode Linux* pour virtualiser des systèmes RedHat et Debian. Le déploiement est facile, les performances sont acceptables et l'administration repose sur des scripts développés par la communauté. Cette solution ne me paraissait toutefois pas être acceptable car elle nécessite de dédier le système virtualisé à User Mode Linux.

Dans un second temps, j'ai testé VirtualBox, une solution que nous utilisons dans le cadre de nos enseignements pour les élèves. Les différents membres de l'équipe projet disposent donc de ce virtualiseur et nous pouvons collaborer ensemble à la résolution des problèmes qui peuvent arriver. Ce choix m'a permis de collaborer à différentes documentations internes afin d'expliquer comment mettre en place des configurations particulières pour le réseau et l'optimisation de certains services. J'ai testé VMWare, solution utilisée actuellement par notre service informatique et sur laquelle j'ai pu m'entretenir avec nos administrateurs système et réseau afin de mieux faire mon choix. J'ai pu la tester en temps qu'utilisateur et j'y ai déployé deux ressources applicatives, notre SGBD et un serveur Web.

Enfin, j'utilise pour mes propres besoins le virtualiseur Xen qui répond à nos besoins en termes de performance, de sécurité et de fonctionnalités. C'est également l'avis de plusieurs collègues ainsi que de nos partenaires de l'IFREMER. C'est la raison pour laquelle j'ai donné mon aval à ce que cette solution qu'ils maîtrisent, soit utilisée sur leur serveur physique.

Dans le cadre de la mise en place de nos prototypes, j'ai utilisé trois serveurs physiques dédiés à cette tâche pour effectuer la virtualisation avec VirtualBox. Maintenant que notre architecture est stable, j'ai fait héberger une machine virtuelle par notre service informatique qui se charge de sa maintenance. Sur cette machine sont déployés nos composants de partage de l'information vers l'extérieur à destination d'autres services et du grand public ainsi que le composant chargé du management des données en base et enfin le composant chargé de l'affichage des données à savoir C3, C4 et C5 (Cf figure 1.5.1.1). Sur le serveur de l'IFREMER se trouve les composants C1 et C2 pour l'acquisition et le traitement des données. Notre composant pilote C6 est lui aussi déployé sur un serveur à l'ENSTA-Bretagne.

4.5.2 Système d'opérations

Notre système d'information est orienté services. Le système d'exploitation sur lequel s'appuient ces services est un support qui peut être remplacé sans incidences sur la description des services. C'est d'ailleurs ce qui s'est passé au fil de l'évolution de notre conception. Nous avons dans un premier temps utilisé un système d'exploitation Windows sur différentes machines virtuelles.

Pour des raisons de performances, de facilité d'administration et d'intégration, j'ai ensuite migré ces machines vers la distribution linux Debian Squeeze 6.

4.5.3 Le SGBD

Le système de gestion de base de données (SGBD), est un ensemble de logiciels informatiques qui sert à la manipulation des bases de données. Il effectue des opérations ordinaires telles que consulter, modifier, construire, organiser, transformer, copier, sauvegarder ou restaurer des bases de données. Il est souvent utilisé par d'autres logiciels ainsi que les administrateurs ou les développeurs (Référence : Wikipédia).

Notre équipe dispose de compétences sur différents SGBD à savoir Oracle, MySQL, Sqlite et Postgres. Nous utilisons et administrons ces systèmes quotidiennement. Je donne moi-même annuellement quelques heures d'enseignements aux étudiants dans le cadre d'un stage de réalisation d'un site Web dynamique pour lesquels l'une des contraintes est l'utilisation de MySQL.

Notre architecture applicative nous rappelle que pour la gestion du management des données, nous devons manipuler un catalogue de données brutes, traitées et de suivi de l'activité de l'observatoire. Ce suivi de l'activité peut être exprimé de différentes manières. Celle qui me paraît offrir des perspectives intéressantes est la géolocalisation des événements. Parmi ces 4 SGBD, le module Postgis de Postgres offre la possibilité de manipuler à l'heure actuelle des bases de données géographiques. J'ai mis en place ce module pour le tester sur l'une de nos bases, et avec Pascale Gautron, enseignante à l'ENSTA-Bretagne et membre de l'équipe IDM, nous avons commencé à obtenir des résultats intéressants. N'ayant pas d'algorithmes ni de véritables données à l'heure actuelle, nous ne pouvons mettre un tel module en exploitation. Nous pouvons cependant continuer nos expérimentations et présenter nos propres résultats aux intéressés qui pourraient ignorer certaines possibilités offertes par ces outils. D'un point de vue conceptuel, l'intégration de cette solution dans les prochaines évolutions du système est envisagée.

Un système d'information géographique (SIG) est un système d'information capable d'organiser et de présenter des données alphanumériques spatialement référencées, ainsi que de produire des plans et des cartes. Ses usages couvrent les activités géomatiques de traitement et diffusion de l'information géographique. La représentation est généralement en deux dimensions, mais un rendu 3D ou une animation présentant des variations temporelles sur un territoire sont possibles. (Source : <http://afigeo.asso.fr/les-sig.html>)

4.5.4 Les serveurs Web

Nos différents blocs applicatifs communiquent via l'échange de messages et d'objets reposant sur SOAP et REST. Nos services sont donc des interfaces et les flux reposent sur des protocoles Web entre les différents composants. Notre modèle physique a évolué durant le projet. Nous avons prévu de déployer un serveur physique par composant et seulement deux désormais. L'idée est de concevoir un système pouvant s'adapter à différents besoins.

Désormais, les composants de présentations, d'échanges et de gestion de la base de données seront déployés sur une seule machine virtuelle d'un serveur de l'ENSTA-Bretagne.

Les composants de traitement et d'acquisitions des données seront déployés sur 2 machines virtuelles hébergées sur le serveur de l'IFREMER. En résumé, pour la machine virtuelle de l'ENSTA-Bretagne, nous pouvons déployer un seul serveur apache et tomcat ainsi qu'une BDD postgres. Les flux échangés entre ces composants sont détaillés dans le diagramme de déploiement figure 4.5.1.

4.6 CONCEPTION DU METAMODELE DE L'OBSERVATOIRE

L'étude des différentes architectures liées à l'urbanisation du système m'a permis de faire émerger plusieurs concepts fondamentaux que l'on peut représenter dans le métamodèle figure 4.6.1.1. Les différentes perspectives peuvent être associées afin de métamodéliser le système suivant l'approche MDA en respectant les règles de l'urbanisme.

4.6.1 Meta-modèle du SI MeDON

J'ai conçu le métamodèle de tel sorte que les différents concepts de l'urbanisme dans un cadre d'architecture orientée services puissent être exploités et que les modèles que nous utiliserons soient conformes à cette approche.

Concernant la perspective Métier, le métamodèle prend en compte le fait qu'une organisation puisse employer différents acteurs ayant une responsabilité et donc un rôle dans différents processus de travail. Cela peut être par exemple le laboratoire dans lequel je travaille qui emploie mon collègue Cédric Gervaise, ayant pour rôle d'être acousticien et travaillant à la surveillance d'une zone sous-marine.

Ces processus de travail sont composés de tâches recevant et produisant différents événements métiers déclenchés par des données métier. Par exemple, le processus de surveillance sous-marine est composé des tâches *écoute dauphin* et *traitements* lorsque l'événement *acquisition* est reçu.

Le deuxième point de vue du métamodèle fait apparaître les concepts de la vue fonctionnelle. Les différentes fonctionnalités pouvant avoir des liens de dépendance forment les îlots fonctionnels et peuvent donc avoir des types. Ces îlots fonctionnels sont regroupés sous forme de service fonctionnel qui exploite des données fonctionnelles. Ces services composés sont alignés sur nos tâches métiers. L'exemple de la tâche métier *écoute dauphin* met en jeu différents types de blocs fonctionnels à savoir la gestion des acquisitions, des traitements et des données. Pour chacun de ces blocs, différents services fonctionnels permettront d'exploiter nos données fonctionnelles (Référence figure 2.1.2).

Le troisième point de vue du métamodèle représente l'architecture applicative de notre SI orienté services sur laquelle se basent nos services fonctionnels. Ces services reçoivent en entrée différents paramètres à traiter et communiquent via des protocoles qui manipulent des données applicatives. La notion de composant est ici liée au type de service fonctionnel dont il est question. Parmi nos composants de services, nous pourrions représenter au niveau M1 (cf figure 1.2.3.1) les composants C1 à C6 (Fig 1.5.1.1).

Enfin, le point de vue le plus bas du métamodèle représente les concepts physiques de l'observatoire. Celui-ci ayant un nom et se matérialisant par différents serveurs physiques (*Hoster*) pouvant héberger des serveurs virtuels/applicatifs (*Server*). Les serveurs sont singularisés par leur type et exploitent des données physiques.

Si l'on reprend le diagramme de déploiement, dans le cas de MeDON et du prototype *survIrvoise* (Figure 4.5.1) de la mer d'Iroise, nous aurons un observatoire dont le nom sera

SurvIroise, deux *Hosters* qui seront l'IFREMER et l'ENSTA-Bretagne hébergeant différents serveurs applicatifs pouvant offrir des espaces de stockages.

J'ai introduit trois liens d'agrégation entre les rôles que peuvent jouer les acteurs dans les processus et l'infrastructure physique. Ces liens apparaissent dans le métamodèle car ils peuvent être vus comme des services associés à des fonctionnalités répondant à des tâches métier.

Ces trois liens sont :

- \hat{R} isOwner
- \hat{R} isAdministrate
- \hat{R} isModerate

Ces notions permettent si nécessaire de spécifier clairement le rôle de certains acteurs ce qui permettra de rendre plus clair les modèles conformes et d'embarquer ces notions dans une partie du code qui pourra être généré.

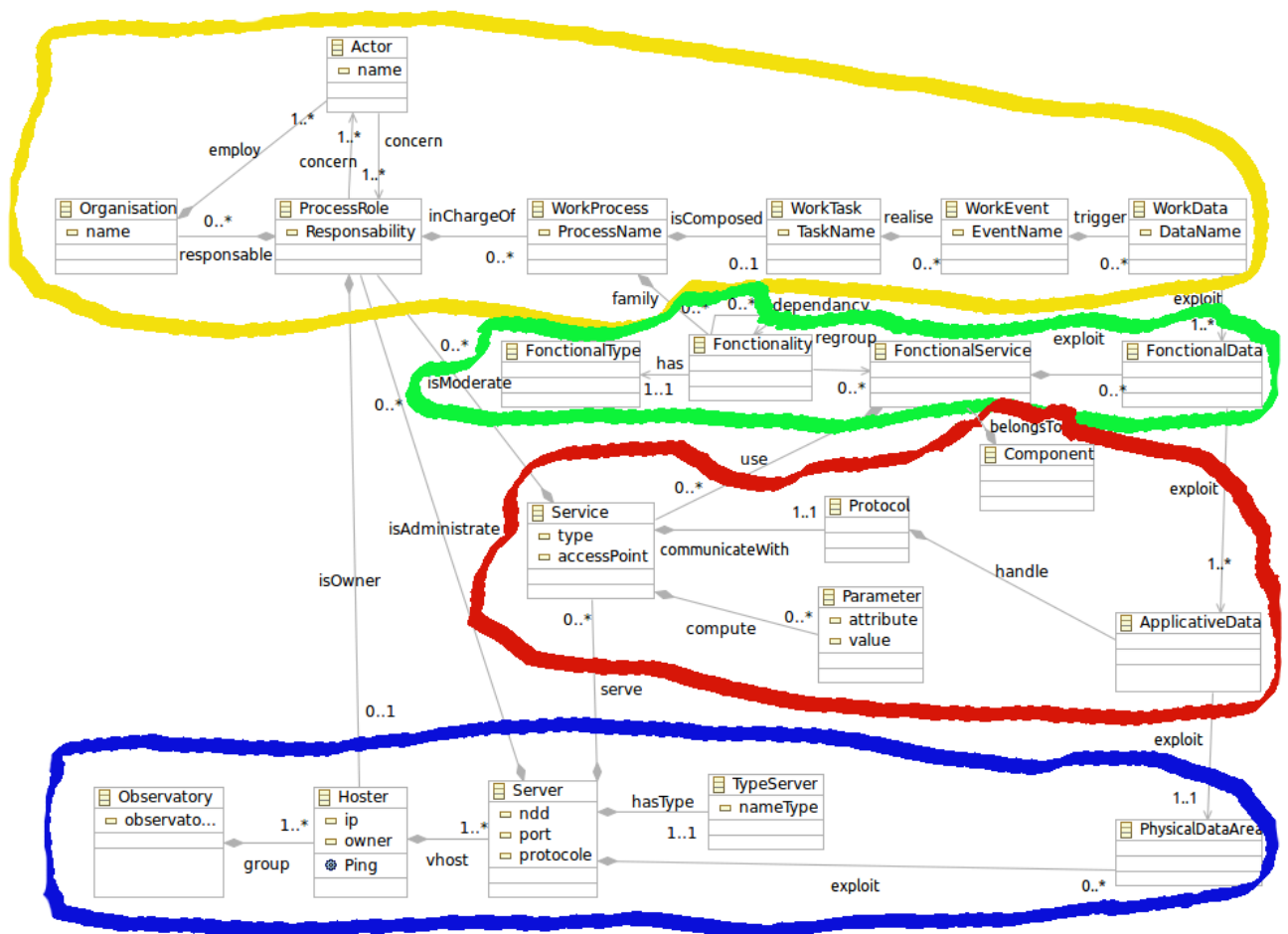


FIG. 4.6.1.1: Meta-modèle du système d'information de MeDON

Conclusion sur l'étude conceptuelle

Cette étude conceptuelle a permis d'appliquer les techniques de l'urbanisation à notre concept d'observatoires sous-marin. Elle a permis d'identifier différents modèles qui cartographient notre système à travers les perspectives : métier, fonctionnelle, applicative et physique.

L'étude a également permis l'ébauche de la conception d'un métamodèle servant par la suite à exprimer les concepts communs à l'ensemble des modèles de notre observatoire.

Deuxième partie

Réalisation

Chapitre 5 - Modélisation

Les objectifs de la modélisation

Ma contribution au projet MeDON a également consisté à réaliser différents prototypes. En se basant sur l'état de l'art et nos études conceptuelles, j'ai cherché à modéliser, développer ou générer des solutions techniques pouvant être déployées sur notre architecture. La contrainte étant que cette architecture pouvait et allait évoluer au cours du temps. Les notions de test et de vérification de non-régressions sont donc très importantes ici afin de nous permettre de surveiller l'état de nos services et globalement de notre système. Ce sont tous ces différents aspects qui seront présentés dans cette partie.

L'objectif principal de cette partie est de dresser les principaux cas d'utilisation et diagramme de séquences de MeDON. Cela afin de permettre grâce à eux de réaliser des transformations de modèle et de la génération de code.

La réalisation de la modélisation a été effectuée en UML à l'aide de l'outil Rhapsody.

Deux plugins eclipse m'ont permis de travailler mes modèles et métamodèles :

- Le plugin EMF (eclipse modeling framework)

- Le plugin GMF (graphical modeling framework)

Rhapsody est un logiciel de modélisation développé par IBM suite au rachat de la société suédoise Telelogic.

La volonté de l'équipe était de capitaliser au sein de ce même outil les différents modèles du projet. Les raisons qui ont guidé ce choix porte sur l'homogénéisation de nos productions, une communication plus facile auprès de nos partenaires et enfin les fonctionnalités interne du logiciel qui intègre l'import/export des productions dans d'autres formats (XMI, TOPCASED uml et sysml, ecore) ou encore les possibilités de génération de code.

J'ai cependant eu recours à d'autres outils comme les plugins d'eclipse afin de profiter du savoir faire de mes collègues en matière de conception et de transformation de modèles. Cela m'a permis de monter en compétence dans ce domaine par la production de solutions "temporaires" en utilisant les mêmes outils que mes collègues. Ainsi, mes productions étaient directement « critiquables » et pouvaient s'intégrer plus facilement si ce n'est dans la continuité, dans la même ligné que certains de leurs propres travaux (ex : modal, dodaf, kermeta).

5.1 CAS D'UTILISATIONS

Comme nous l'avons vu lors de l'étude conceptuelle, chacun des modèles est exprimé dans un langage supposé apporter les concepts adéquats pour un niveau d'abstraction donné. La première étape dans l'approche IDM est de modéliser le domaine et donc les entités et relations à l'aide de cas d'utilisation.

5.1.1 Cas d'utilisation global

Le cas d'utilisation principal donné en figure 5.1.1.1 représente une vue globale du système d'information SurvIroise pour la surveillance de la mer d'Iroise. Sur ce cas d'utilisation nous retrouvons nos principaux acteurs à savoir les managers et les scientifiques :

- R Biologiste
- R Acousticien
- R Internaute

Les différents acteurs scientifiques interagissent avec le système via des services publics. Les managers peuvent quant-à eux organiser le système c'est-à-dire les données, les algorithmes et les capteurs.

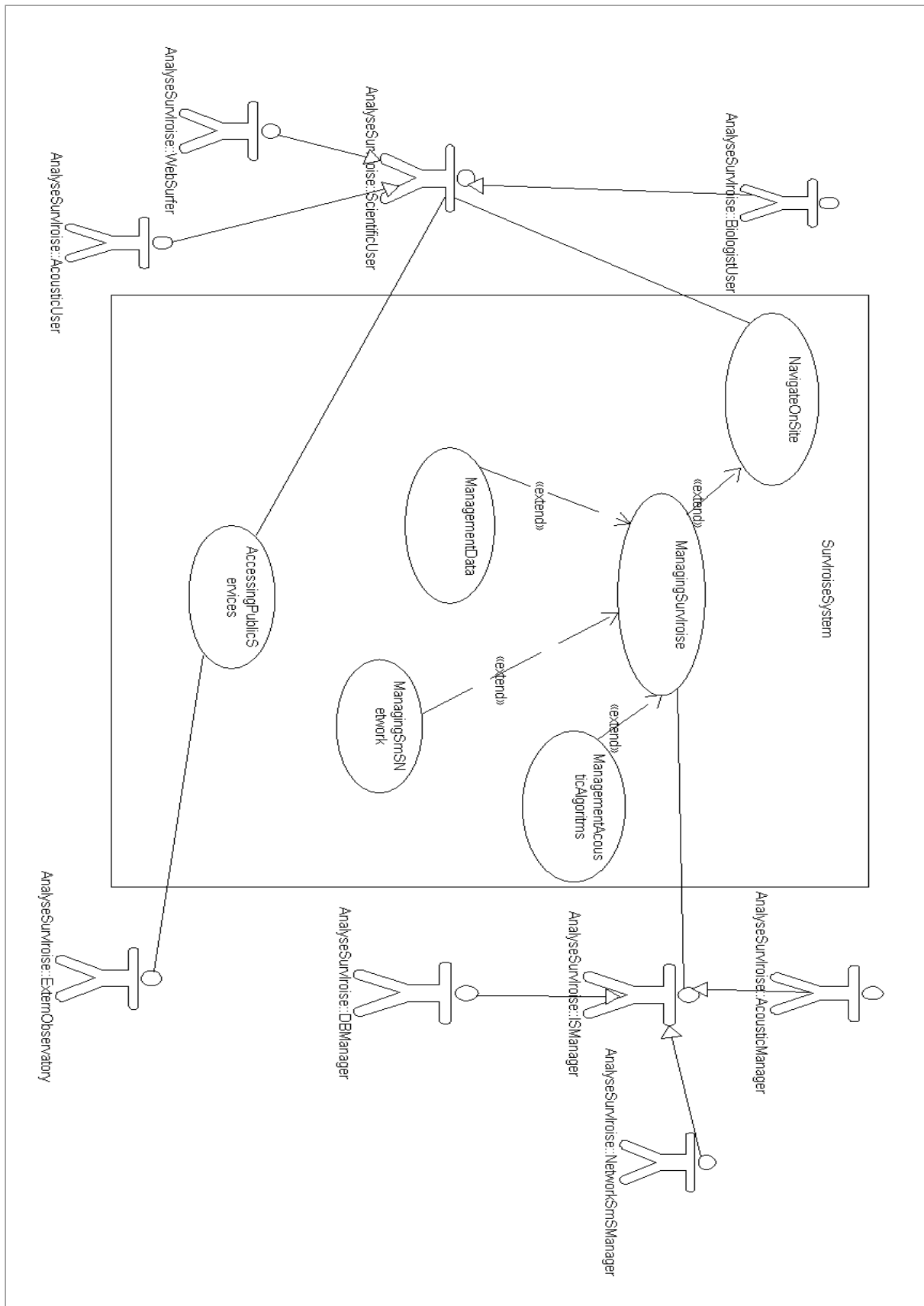


FIG. 5.1.1.1: Cas d'utilisation du système d'information SurvIroise

Chaque cas d'utilisation doit ainsi être détaillé. L'accès au service public qui apparaît dans le diagramme de cas d'utilisation figure 5.1.1.1 est ainsi détaillé dans le diagramme 5.1.1.2.

Le cas d'utilisation 5.1.1.2 met en présence les acteurs qui sont des scientifiques et les acteurs qui sont externes à l'observatoire. L'idée est que l'acteur externe représente ici un autre observatoire MeDON.

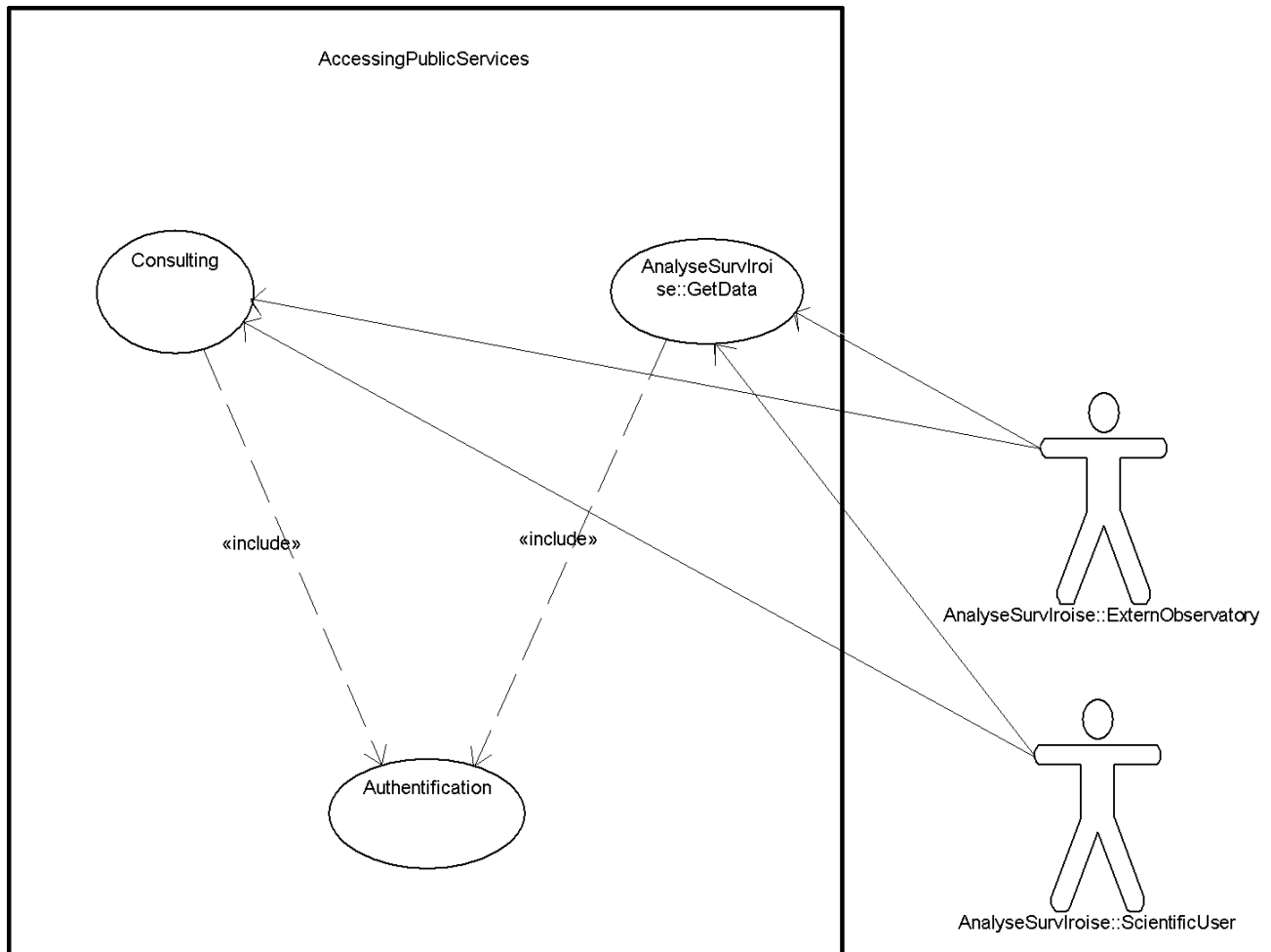


FIG. 5.1.1.2: Cas d'utilisation de l'accès aux services publics

Les observatoires externes accèdent à nos services Web via l'interface que nous appellerons *GetData*. Les utilisateurs scientifiques accèdent aux données via une IHM Web représentée par *Consulting*.

Je n'ai pas implémenté de module d'authentification à part entière. Dans le cadre de la réalisation du prototype, et pour répondre aux cas d'utilisations présentés ici, le mécanisme d'authentification repose sur notre base de données. Pour accéder aux données, Consulting et GetData doivent effectuer des requêtes sur la base, et donc s'y connecter. Pour s'y connecter, il est nécessaire de fournir les accréditations adéquates. La gestion des autorisations d'accès est donc déportée sur le SGBD sur lequel nous pouvons créer différents rôles. Chaque rôle pouvant être autorisé ou non à accéder à certaines vues ou tables.

5.1.2 Cas d'utilisation Composant 1

Le composant C1 est responsable de l'acquisition des fichiers de données brutes en provenance des capteurs. Nous retrouvons donc cet acteur dans le cas d'utilisation 5.1.2.1 relié à cette activité.

Les données acquises doivent alors être *publiées* sur le composant C3. Le composant C2 est également notifié des nouvelles acquisitions afin qu'il puisse le cas échéant réaliser ses traitements.

Un manager peut communiquer au composant C1 via le composant C5 l'existence de nouveaux senseurs et donc de nouvelles capacités d'acquisition. Nous n'avons pas implémenté de mécanismes de découverte automatique du parc de capteurs. L'enregistrement ou la suppression des capteurs doit se faire via une interface Web sur le composant C5.

Enfin, le composant C1 interagit avec le composant C6 lors de son initialisation afin de notifier C6 de son existence. Lorsqu'un autre composant sera initialisé, C6 pourra alors prévenir C1 pour lui permettre de communiquer avec le nouveau composant.

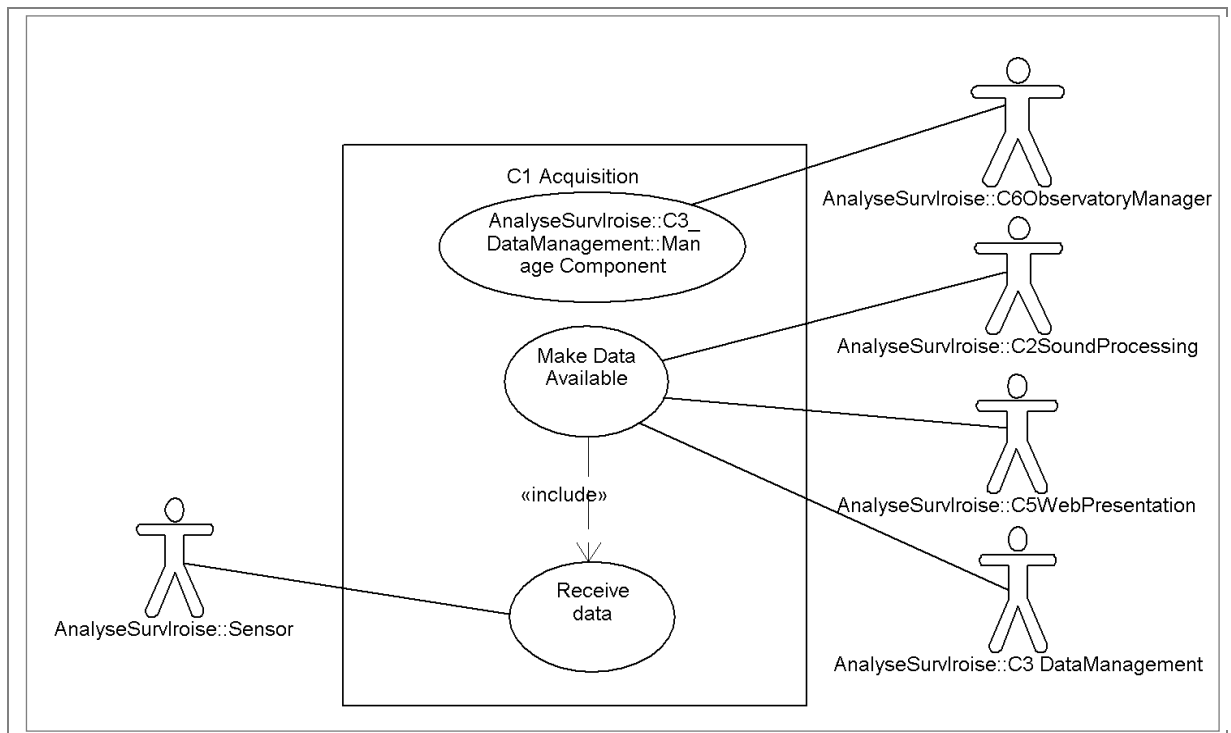


FIG. 5.1.2.1: Cas d'utilisation du composant C1 Acquisition

5.1.3 Cas d'utilisation Composant 2

Le composant C2 a la responsabilité de permettre aux algorithmes d'accéder aux données, d'interagir avec elles et de rendre public leurs résultats. Pour atteindre ces objectifs, quatre activités se démarquent.

La première est la communication entre C1 et C2 afin que les données brutes soient mises à disposition de C2. Lorsque de nouvelles données brutes sont disponibles, C1 doit en notifier C3 car l'une des spécifications de notre système est que tous les événements de ce type puissent être *tracés* en base de donnée.

Lorsque les algorithmes ont finis leur traitement, le composant C3 est notifié que des données traitées sont disponibles.

Un manager peut gérer les algorithmes via l'interface Web du composant C5. C'est-à-dire ajouter, retirer, activer ou désactiver des algorithmes. Il n'est pas prévu que les données brutes soient extraites de C2 dans le cadre de notre prototype, car le volume de données n'est pas supporté par nos contraintes réseaux.

Enfin, de même que pour le composant C1, le composant C2 interagit avec C6 lors de son initialisation et lorsque de nouveaux composants sont rajoutés dans le système afin que chacun garde une carte P2P³⁰ du système. C'est-à-dire que les composants récupèrent une « liste » d'abonnés auprès d'un serveur central, puis sont capables d'aller communiquer directement avec ces abonnés et continuer leurs activités même si le serveur central venait à tomber.

La figure 5.1.3.1 synthétise ces spécifications à travers le cas d'utilisation du composant C2.

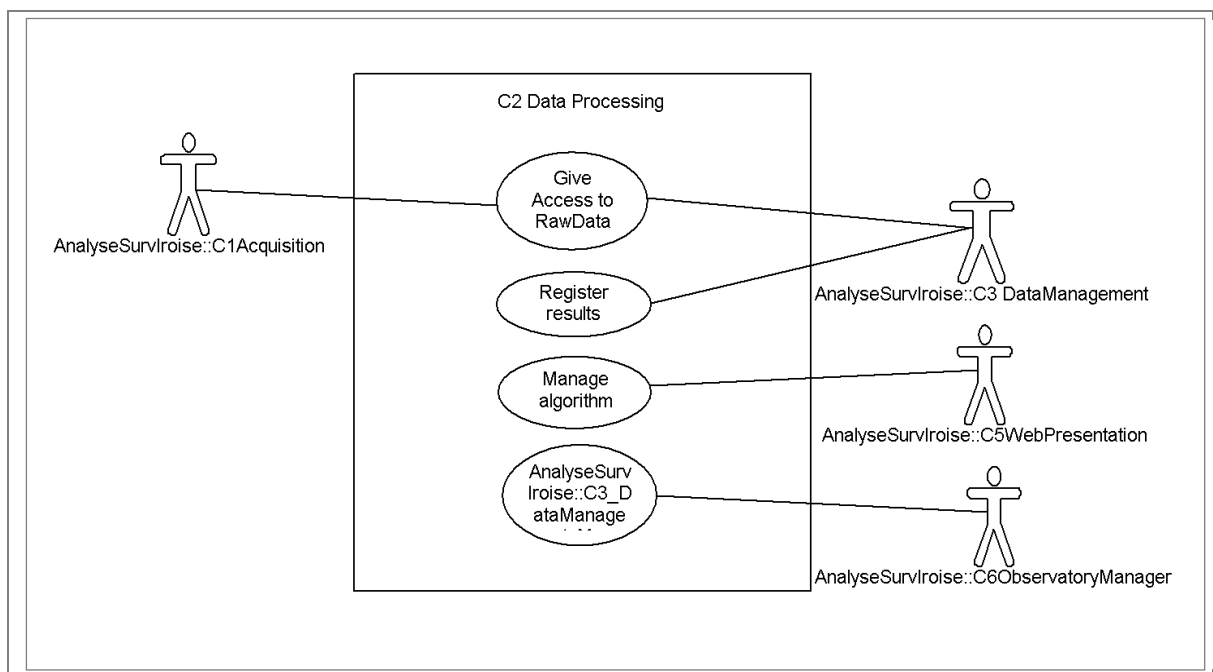


FIG. 5.1.3.1: Cas d'utilisation du composant C2 Data Processing

5.1.4 Cas d'utilisation Composant 3

Le cas d'utilisation 5.1.4.1 tient un rôle central dans notre système d'information. Le composant C3 communique avec les cinq autres composants de notre système pour permettre différentes activités.

℞ La gestion des capteurs est réalisée via le composant C5, l'interface Web.

℞ La gestion des fichiers de données est effectuée à travers le composant C1 qui effectue les acquisitions de données brutes et le composant C2 qui effectue les traitements. Cette mécanique sera détaillée dans le diagramme de séquence du prochain chapitre.

³⁰

La notion de P2P pour peer to peer permet à notre système d'être plus robuste face aux pannes. J'ai emprunté cette idée des réseaux torrent et l'idée est de décentraliser ce type d'informations.

-  Les fichiers résultats sont manipulés par le composant C2 qui effectue les traitements, et le composant C5 qui demande à y accéder.
-  Des événements sont identifiés et déclenchés par le composant C2 qui en notifie C3.

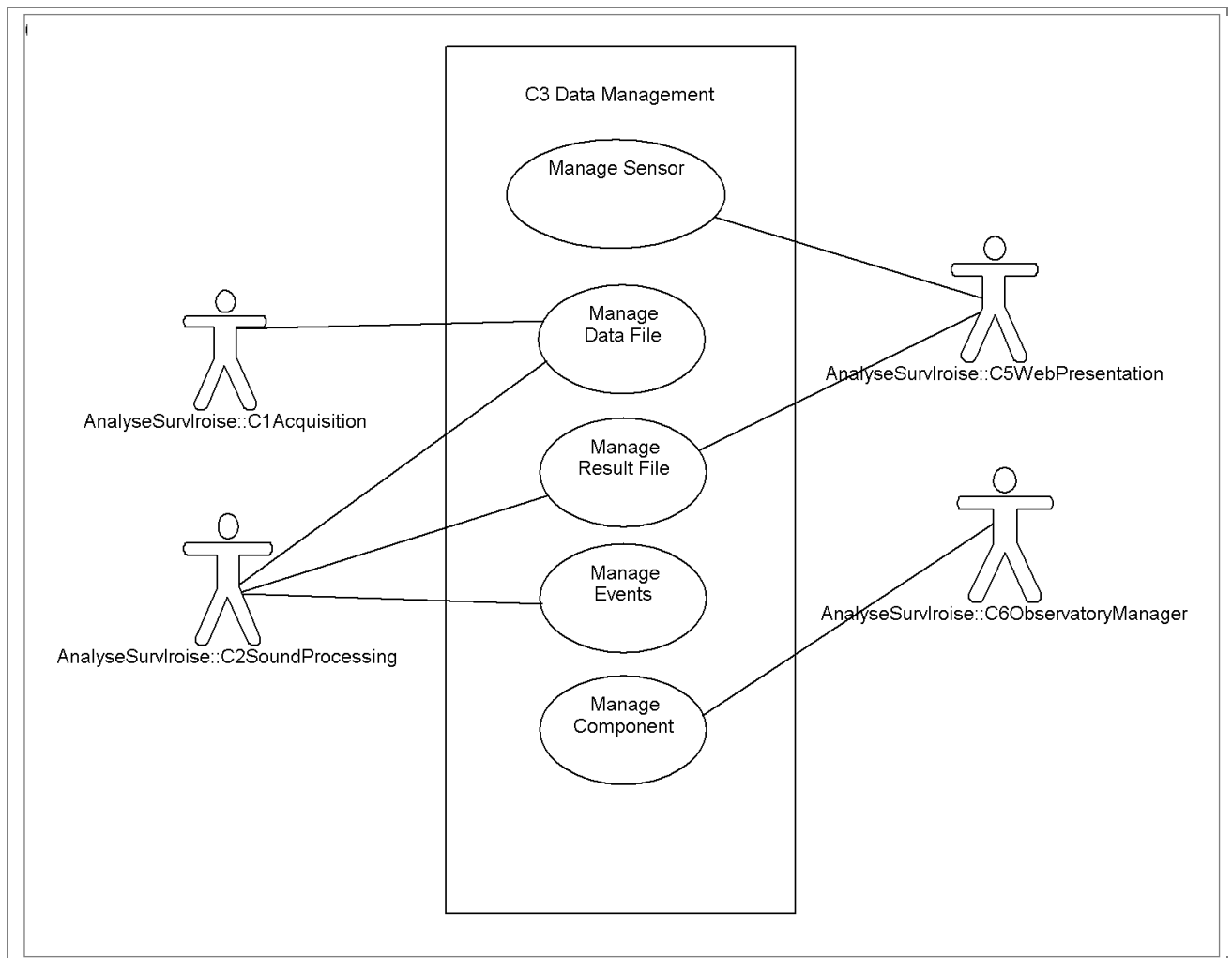


FIG. 51.4.1: Cas d'utilisation du composant C3 Data Management

5.1.5 Cas d'utilisation Composant 4

Un autre cas d'utilisation qu'il me paraît intéressant de détailler ici est relatif au service d'alertes *SAS* figure 5.1.5.1.

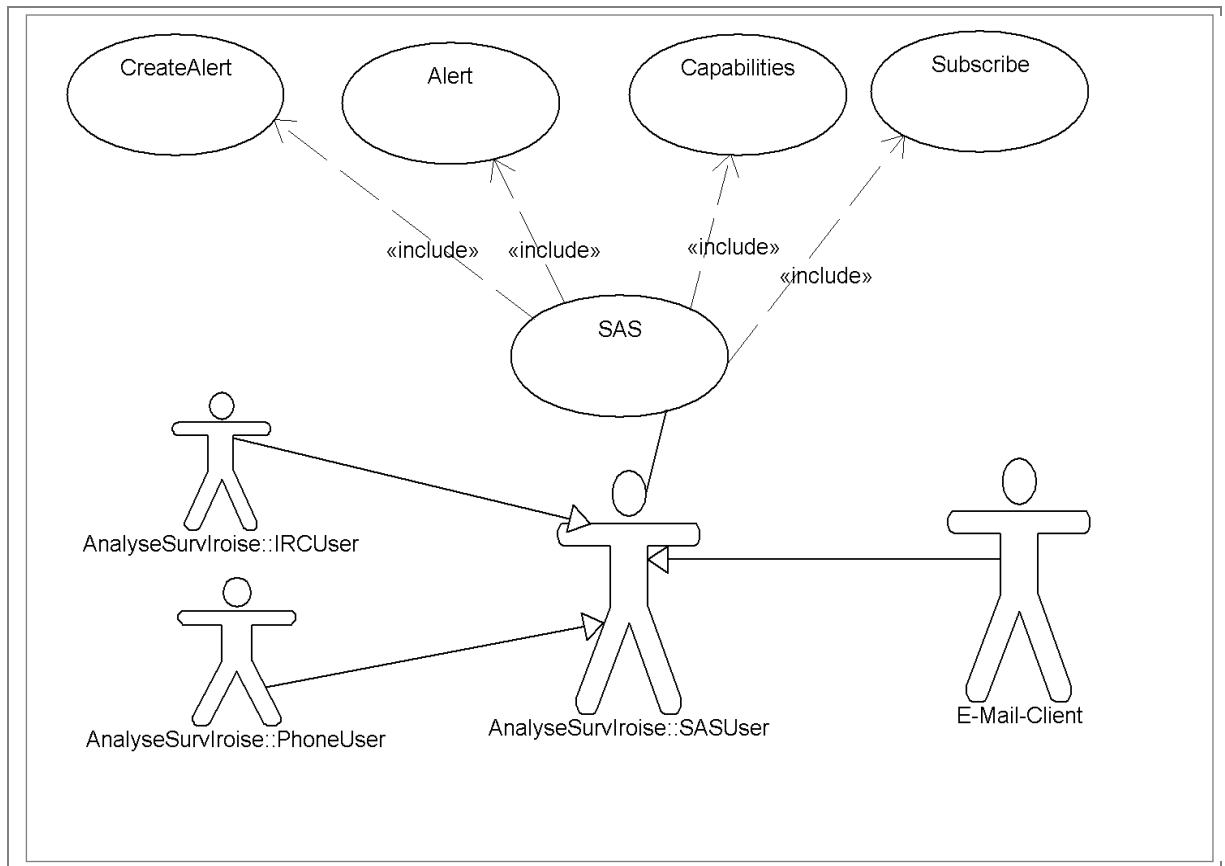


FIG. 5.1.5.1: Cas d'utilisation du système d'alerte SAS

Ce cas d'utilisation fait apparaître les différents types d'utilisateurs pouvant interagir avec des alertes à savoir des internautes utilisant :

- le système de messagerie IRC
- de la VoIP
- une messagerie

Ces utilisateurs peuvent en fonction des événements de l'observatoire :

- s'inscrire (*Subscribe*)
- récupérer une liste d'alertes (*Capabilities*)
- créer des alertes (*CreateAlert*)
- recevoir des alertes (*Alert*)

Ce service était important pour la validation du prototype car il met en jeu les mécanismes allant de l'acquisition des données brutes jusqu'au transport de données traitées vers les utilisateurs finaux en passant par la mise en place de filtres au niveau de notre composant de gestion des données.

Ce service m'a également permis de former des étudiants aux technologies Android, aux services Web ou encore aux problématiques réseau via différents projets reposant sur lui.

5.1.6 Cas d'utilisation Composant 5

Le composant C5 est un composant de présentation des données aux scientifiques et au grand public via une IHM Web. Son rôle est de présenter des données humainement compréhensibles. Le composant C5 interagit lui aussi avec C6 pour notifier sa présence ou acquérir la liste des autres composants via l'activité *Manage component map*.

Un utilisateur peut accéder aux données que C5 acquiert de C3. Enfin, un scientifique peut gérer des alertes ou des inscriptions à des alertes via une interface Web. Les alertes disponibles et les souscriptions étant stockées en base de données.

Le cas d'utilisation du composant C5 est présenté à la figure 5.1.6.1.

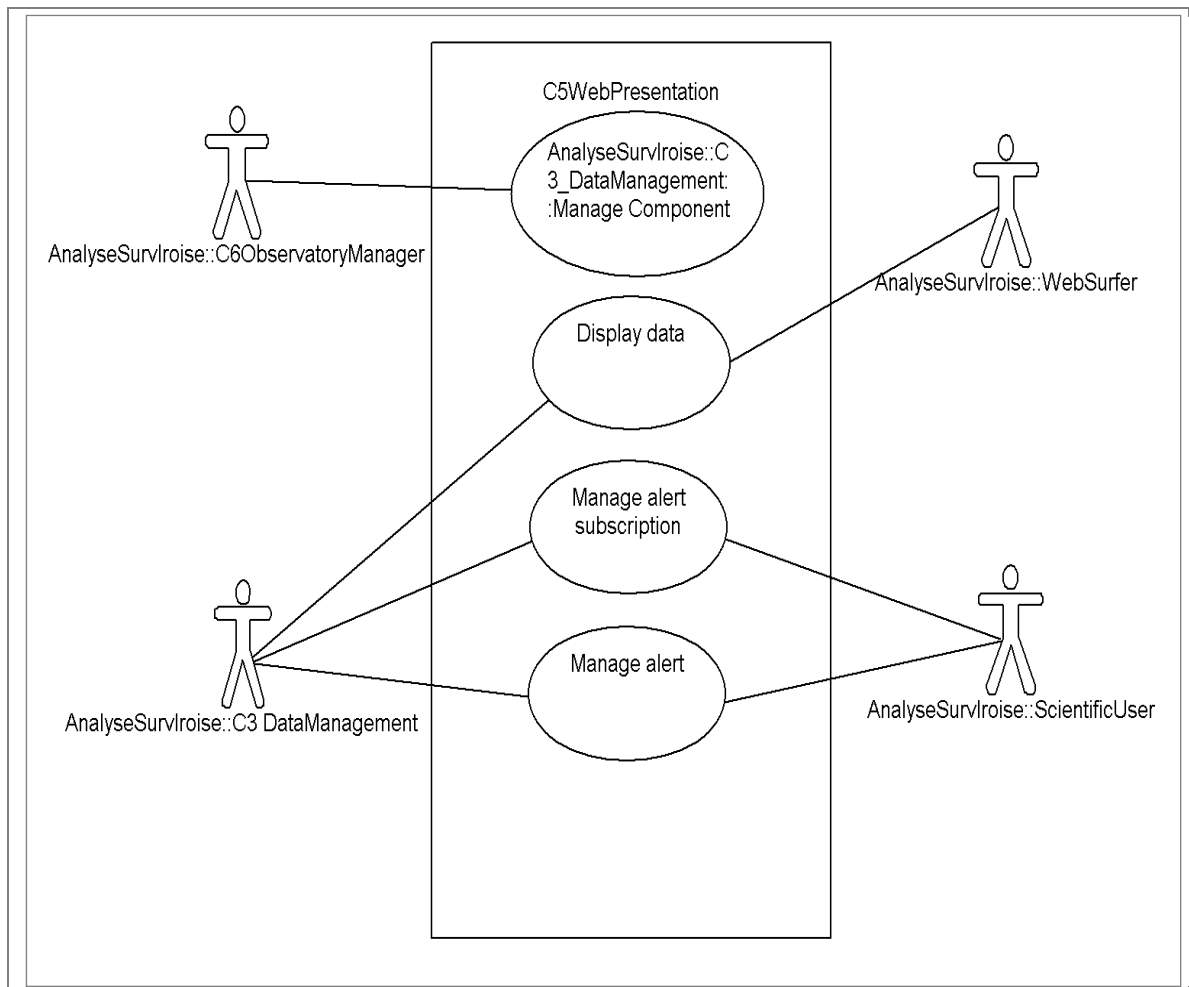


FIG. 5.1.6.1: Cas d'utilisation du composant C5 Web Présentation

5.1.7 Cas d'utilisation Composant 6

Comme nous l'avons vu dans les cas d'utilisation précédents, le composant C6 est chargé de communiquer avec les autres composants pour maintenir à jour une *cartographie* du système (Figure 5.1.7.1).

Les différents composants s'enregistrent lors de leur initialisation et reçoivent une liste des différents services disponibles sur l'observatoire.
Ce composant peut également vérifier que les composants sont toujours actifs et nous pouvons ainsi avoir une vue instantanée des services disponibles.

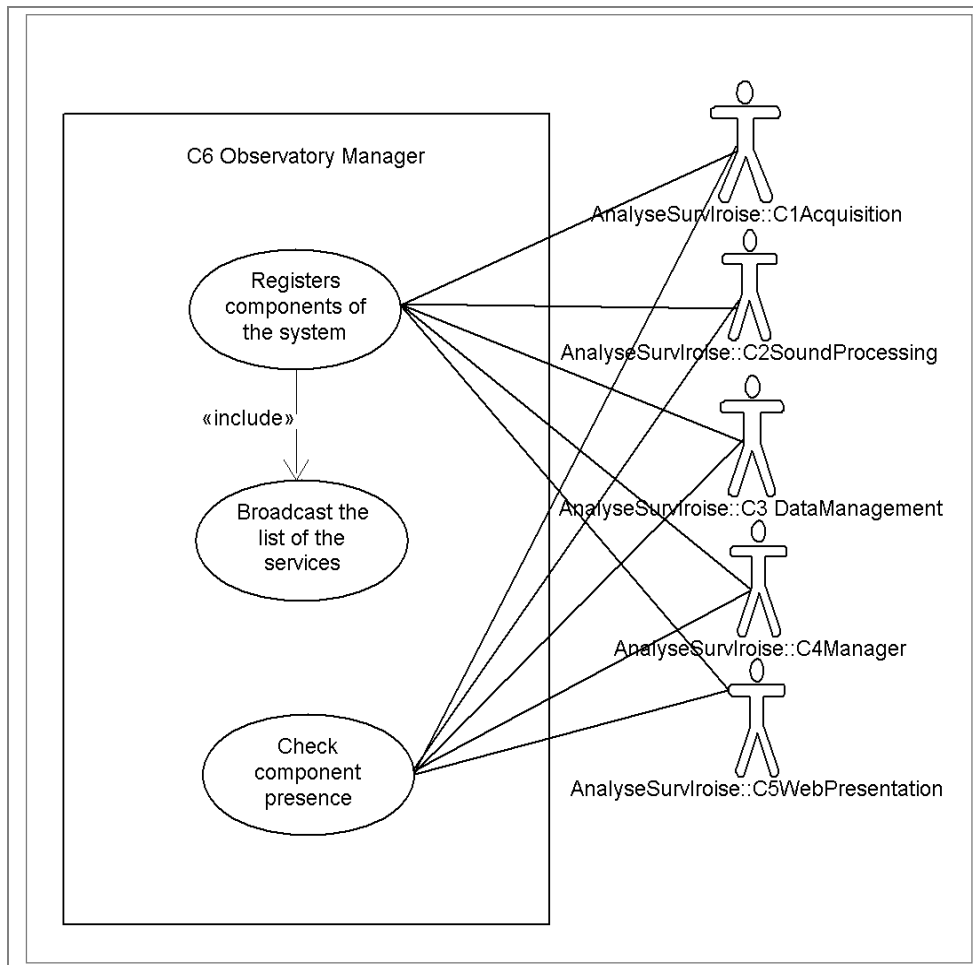


FIG. 5.1.7.1: Cas d'utilisation du composant 6 Observatory Manager

5.2 DIAGRAMMES DE SEQUENCE

Les diagrammes de séquences de ce chapitre représentent les interactions entre les acteurs et le système des principaux scénarios des cas d'utilisation de la partie précédente.

Nous allons nous intéresser au déploiement de nouveaux composants, à l'acquisition et au traitement de données ainsi qu'au déclenchement d'une alerte afin de considérer les réalisations relatives aux principaux objectifs métiers identifiés dans la partie 1 de ce travail.

5.2.1 Déploiement de la carte/annuaire du système

La figure 5.2.1.1 illustre l'enregistrement de C1 auprès de C6.

Dès lors que celui-ci est initialisé, il communique à C6 la liste des services qui forment ce nouveau composant. C6 diffuse alors à tous les composants la nouvelle liste de services disponibles. Lorsqu'un composant est retiré du système, C6 en informe également les autres composants.

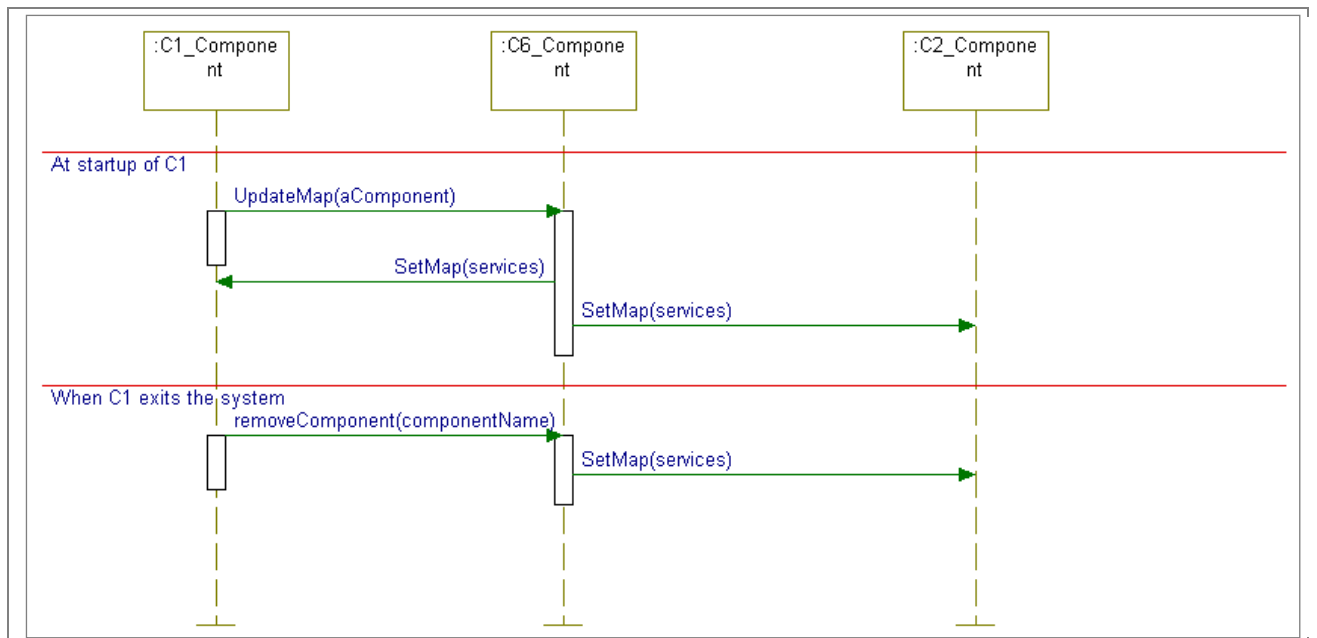


FIG. 5.2.1.1: Diagramme de séquence du déploiement de l'annuaire

Ce mode de diffusion de la carte permet à nos composants de continuer à communiquer entre eux-même si C6 devient indisponible temporairement. Le nombre de composants n'est pas voué à être très volatile et ce mode de fonctionnement entraînera moins de requêtes sur le réseau.

La disponibilité des composants est une information importante pour la maintenance du système ou le déclenchement d'alertes en cas de défaillance. Afin de répondre à ce problème, un mécanisme de *Ping* (Figure 5.2.1.2) permet de s'assurer que les composants enregistrés répondent toujours.

Nous pourrions imaginer que lorsqu'un composant ne répond plus, une nouvelle carte soit diffusée à l'ensemble du réseau. Cependant il peut arriver qu'un composant ne réponde pas suite à une

surcharge ou pour diverses autres raisons. Nous avons donc choisi de ne pas désactiver un composant lorsque celui-ci ne répond pas laissant cette possibilité au bon vouloir de l'administrateur système qui traitera l'information.

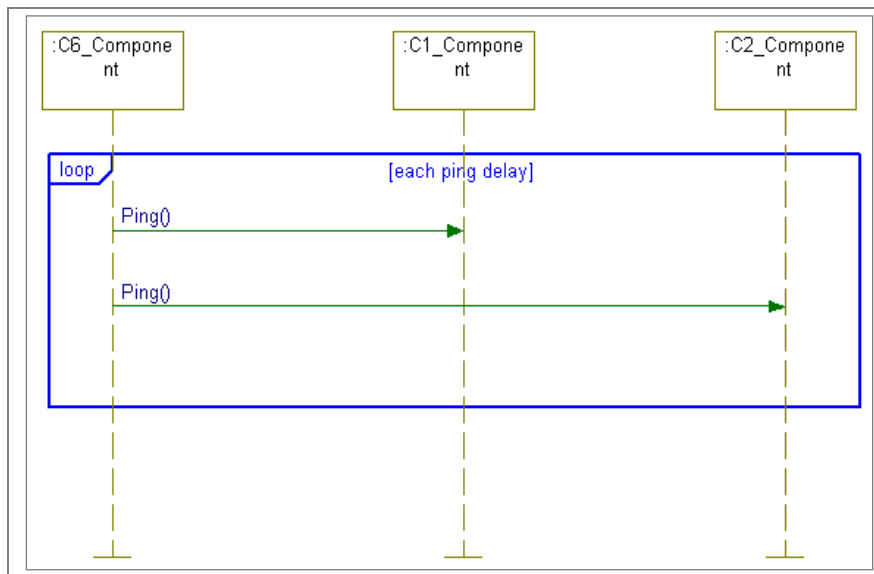


FIG. 5.2.1.2: Diagramme de séquence du test des composant déployé

5.2.2 Acquisition de données brutes

Les données brutes des capteurs sous-marins sont acquises périodiquement et stockées à disposition de C2 (Illustration figure 5.2.2.1).

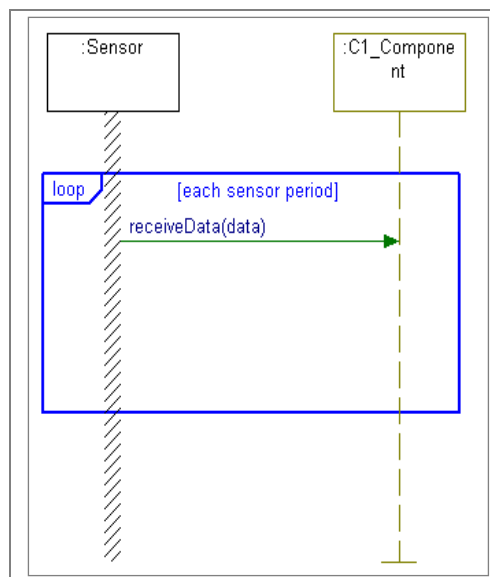


FIG. 5.2.2.1: Diagramme de séquence de l'acquisition de données brutes

Les données reçues sous forme de fichiers pourront alors faire l'objet d'un traitement par C2. Dans cette idée, le composant C1 notifie à C3 la mise à disposition d'un nouveau fichier. Lorsque C3 aura communiqué l'information à C2 et que le fichier aura été copié, C3 pourra informer C1 que le fichier peut être supprimé pour libérer de l'espace pour de nouvelles données.

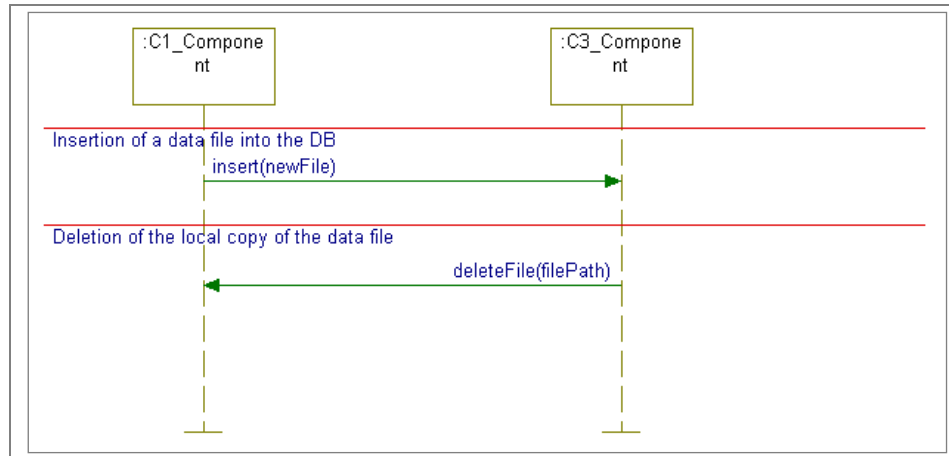


FIG. 5.2.2.2: Diagramme de séquence des échanges lors de l'arrivée d'un nouveau fichier brut

Il n'est pas nécessaire de détailler ici le processus d'ajout de nouveaux capteurs. La démarche est similaire.

Le composant C1 acquiert les données brutes des capteurs actifs. Cette liste est acquise auprès de C3.

Périodiquement, le composant C1 parcourt une arborescence de dossiers à la recherche des nouveaux fichiers. La figure 5.2.2.3 détaille donc la figure 5.2.2.2 en faisant apparaître le mécanisme permettant de parcourir l'arborescence de données acquises.

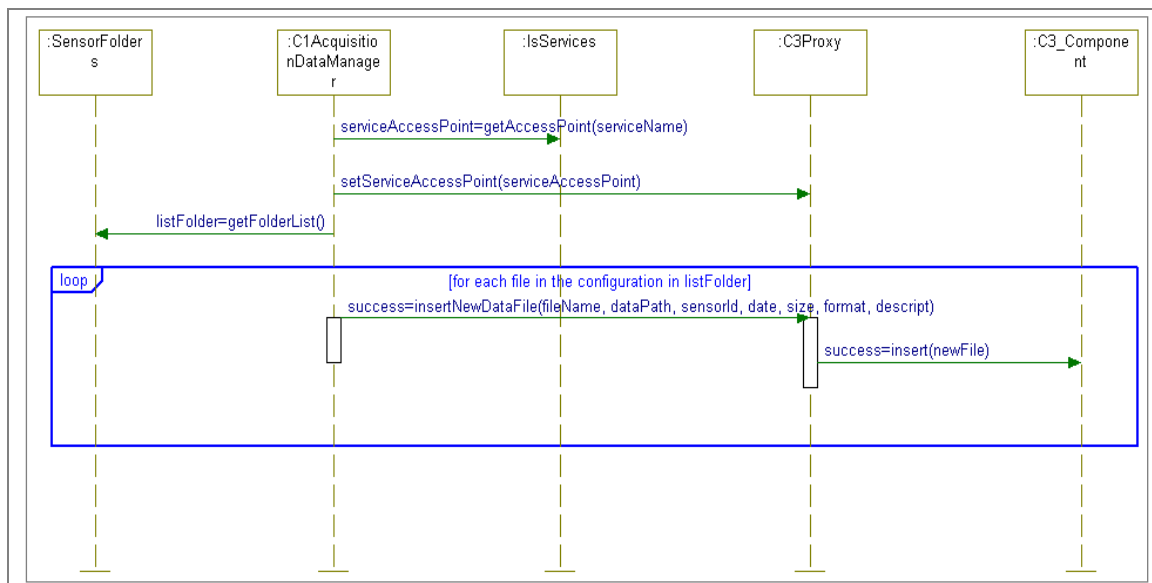


FIG. 5.2.2.3: Diagramme de séquence de l'échange permettant l'insertion du chemin d'un nouveau fichier en base de données

5.2.3 Traitements des données

Le composant C2 doit acquérir les fichiers bruts du composant C1 pour exécuter les différents algorithmes scientifiques. Le composant C2 envoie donc une requête au composant C3 pour connaître la liste des fichiers qu'il devra récupérer sur C1 tel que représenté sur le diagramme de séquence 5.2.3.1.

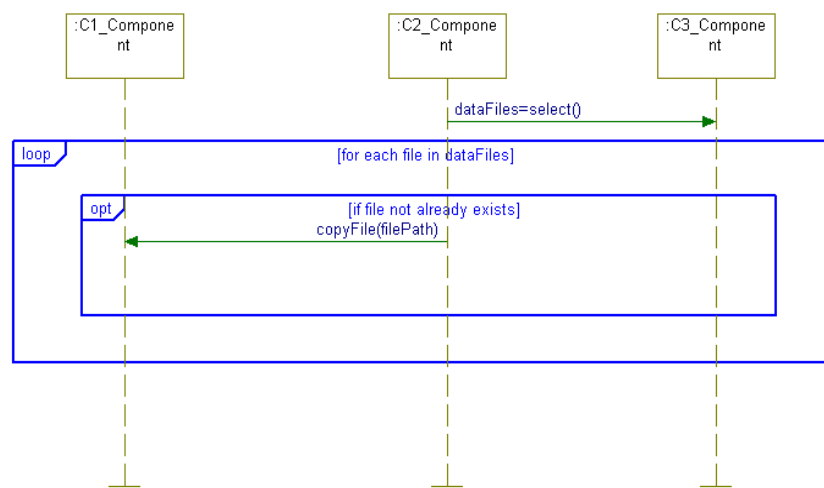


FIG. 5.2.3.1: Diagramme de séquence de l'échange permettant la copie sur C2 des fichiers de données brutes

Les spécifications relatives aux algorithmes réalisant les traitements ont évolué au cours du projet. Au lancement du projet, l'équipe acoustique passive à l'initiative du projet souhaitait déployer des exécutables développés en C. Leurs algorithmes, relativement complexe, étant développé en Matlab, il s'est avéré que leurs traductions en C n'était pas une tâche qu'ils pouvaient mener dans le temps impartis.

Dans le cadre de l'un des prototypes, j'ai embarqué sur une carte l'un de ces codes. Oussama Kassen Sein a également porté du code en JSP³¹ sur le composant C2 pour en faire un service Web. Désormais, pour des raisons de licence et donc d'homogénéisation avec les choix de nos partenaires, il est question de les traduire en Scilab³², un équivalent de Matlab, libre de droit. Un second choix à l'étude est de les traduire en Python. Ce choix n'est cependant pas de mon ressort. Je considère toutefois ces différentes solutions comme autant de cas de figure qu'il faut que l'on puisse traiter.

³¹ Les JSP (Java Server Pages) sont un standard permettant de développer des applications Web interactives.

³² Scilab 5 est régi par la licence CeCILL (compatible GPL) respectant les principes de diffusion des logiciels libres.

Actuellement, pour chacune de ces solutions, nous sommes en état de déclencher leur exécution via nos services Web. En effet, nous pouvons déclencher l'exécution via des appels à des scripts CGI³³ par exemple, pour déclencher l'exécution des scripts spécifiques. C'est l'un des avantages de cette technologie que de pouvoir déléguer au bon composant la gestion du rôle qu'il a à jouer dans le système, indépendamment des moyens déployés.

J'ai réalisé différents tests afin que nos services web puissent invoquer des traitements nécessitant l'exécution de codes Python, Perl, PHP, Ruby, C et Java mais aussi Matlab et Scilab. J'ai réalisé ces tests sur les plate-formes Linux et Windows selon les besoins. J'ai noté quelques variations liées aux plate-formes cibles. Cela m'a permis d'identifier les problèmes qui allaient se poser lors de la génération du code, et donc de modéliser des PSM intéressants (approche bottom-up). Cependant je n'ai pas eu le temps d'aboutir dans cette démarche par manque de temps, la priorité étant donnée à la finalisation d'un prototype fonctionnel. N'ayant pas de réelles spécifications sur la nature des algorithmes, il m'a semblé qu'effectuer une veille technologique sur ce domaine n'était pas l'objectif prioritaire.

5.2.4 Déclenchement d'une alerte

Comme vu dans le cas d'utilisation 5.1.5.1, le déclenchement d'une alerte n'est qu'une fonctionnalité du service d'alertes mis en place. Les diagrammes de séquence 5.2.4.1 et 5.2.4.2 illustrent la réalisation de ce mécanisme.

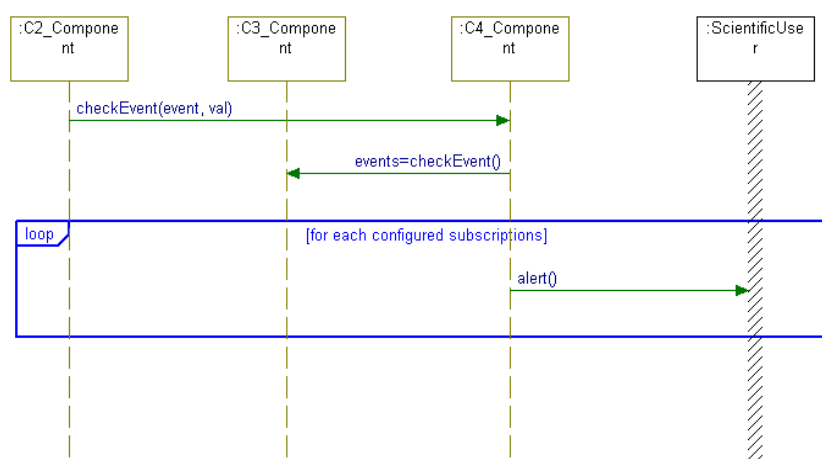


FIG. 5.2.4.1: Diagramme de séquence de l'échange permettant le déclenchement d'une alerte

Lorsqu'un algorithme détecte un événement tel qu'un *clic*³⁴ de dauphin par exemple, il peut invoquer le service d'alerte sur C4 pour analyser l'événement reçu. En fonction des alertes paramétrées, de la nature et de la valeur de l'événement, les abonnés éventuels seront notifiés via leur canal d'enregistrement.

³³ CGI pour Common Gateway Interface. C'est une interface utilisée par les serveurs HTTP pour déclencher l'exécution d'un programme et retourner le résultat comme s'il s'agissait d'une page Web.

³⁴ Les clics de dauphin sont des sons particuliers émis par ces mammifères. L'étude des métriques issues de l'analyse de fichiers audio permet à nos scientifiques d'étudier différents phénomènes liés à leur évolution dans le milieu marin.

Les canaux définis actuellement sont :

 Ré IRC, système de messagerie instantané

 Ré Mail

 Ré SMS via l'API de GoogleAgenda : Le mécanisme de notification de google est exploité afin de nous permettre d'envoyer gratuitement des SMS vers les abonnés ayant un compte google et un numéro de téléphone portable enregistré.

 Ré Google Agenda : L'api de google est utilisée comme preuve de concept. Nous pourrions utiliser d'autres solutions

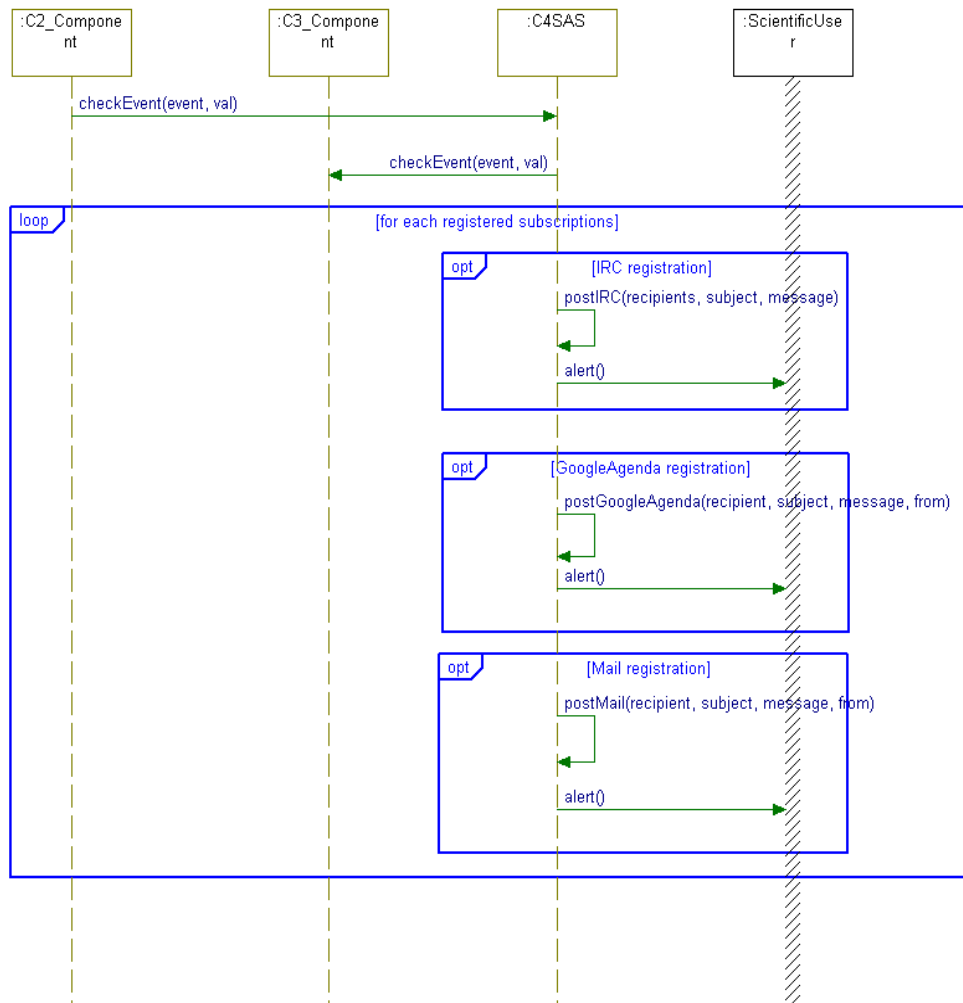


FIG. 5.2.4.2: Diagramme de séquence inter-composant de l'échange permettant le déclenchement d'une alerte

La particularité de ce mécanisme réside dans le fait que les alertes sont "poussées" vers l'utilisateur. Cela induit moins de charges réseaux dues à ce service. L'idée est d'éviter aux usagers ou au composant C4 d'effectuer des requêtes récurrentes alors que de nouvelles données n'ont pas été acquises.

L'ajout de nouveaux canaux de communication comme vecteur d'alertes se fera par ajout de nouveaux modules au sein du service approprié sur C4. L'architecture du code et les patrons de

conceptions³⁵ utilisés sont détaillés dans la documentation relative à ce composant. L'un des patrons au coeur du processus de sélection des canaux est le patron stratégie³⁶.

Dans le cadre du démonstrateur, les différents modules cités ci-dessus sont fonctionnels. Le module IRC doit toutefois être revu suite aux modifications majeures engendrées par la restructuration de nos codes autour de nos modèles. Celui-ci ayant été développé dans le cadre du tout premier prototype.

³⁵ En informatique, un patron de conception, motif de conception ou modèle de conception est un concept de génie logiciel destiné à résoudre les problèmes récurrents suivant le paradigme objet.

³⁶ En génie logiciel, le patron stratégie est un patron de conception (design pattern) de type comportemental grâce auquel des algorithmes peuvent être sélectionnés à la volée au cours du temps d'exécution selon certaines conditions, comme les stratégies utilisées en temps de guerre.

5.3 TRANSFORMATIONS

5.3.1 Modèle Objet vers Modèle relationnel

Le système d'information de notre observatoire manipule des données générées par des capteurs. Ces capteurs doivent pouvoir communiquer au sein du système. J'ai suivi la mise au point d'un langage particulier nommé SMS (pour Smart Modeling for Sensors) qui permet la description de ces capteurs et des concepts qu'ils manipulent. Ce langage est décrit dans notre métamodèle SMS.

Le métamodèle *SMS* sensor.ecore (Listing1 : Sensor.ecore) (disponible en annexe) est conforme à son méta-métamodèle.ecore. Mon idée était de transformer un modèle objet³⁷ en un modèle relationnel³⁸ fonctionnel à partir d'une feuille de style XSLT. Cela nous permet d'utiliser un modèle de base de données à même de recevoir la structure et les données de cette partie du système.

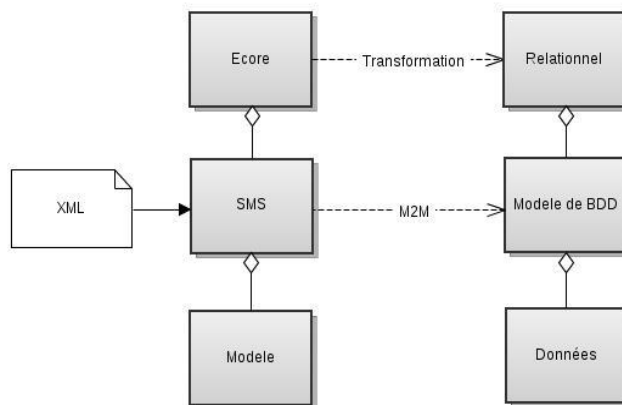


FIG. 5.3.1.1: Transformation objet vers relationnel de SMS

Pour réaliser cette transformation, j'ai dans un premier temps développé un script PHP qu'il faut exécuter pour générer le schéma au format SQL. Cette première version m'a permis de réfléchir avec un collègue sur cette problématique, et ensemble, nous avons mis au point une transformation à base de feuilles de style XSLT (Listing 5.1 : SMS2sql). Nos données en entrée sont au format

³⁷ La conception de logiciel mettant en œuvre le modèle objet repose sur une idée élémentaire : un système informatique réalise certaines actions sur certaines données ; pour obtenir des systèmes souples, réutilisables et évolutifs, il vaut mieux structurer le logiciel autour des données plutôt qu'autour des actions.

³⁸ Les concepts mis en œuvre dans le modèle relationnel sont fondés sur une théorie mathématique directement issue de l'algèbre relationnelle, de la théorie des ensembles et de la logique formelle. Cette technologie a vu le jour dans les années 70 avec les travaux de E. Codd.

XML, l'utilisation de feuilles de style permet de mettre en forme ces données tout en respectant la sémantique et la syntaxe du langage SQL.

La feuille de style est donnée Listing 5.1. Les *eClassifier*³⁹ de *Ecore* sont transformé en table SQL. Les *eReference* sont vues comme des attributs de table et leur type est ici considéré soit comme chaîne de caractère, soit comme un entier. Si le type est une *eReference*, alors nous pouvons considérer qu'il s'agit d'une clef étrangère pour SQL. Nous avons également considéré que la clef primaire serait le nom de la table précédée de *id_* et que son type serait *SERIAL*, ce qui correspond au type auto-incrémenté de Postgres.

Cette transformation nous a permis de faire un premier pont entre le monde des modèles et le modèle physique de données attendu. Cependant la description SMS ne prend pas en compte tous les concepts à stocker en base de données. Dans le cadre de la mise en place du prototype il a donc été nécessaire d'affiner le schéma final de la base, notamment pour prendre en compte les besoins du service d'alerte. Le schéma de base de données du service d'alerte peut être déployé sur un SGBD indépendant. La fusion des bases pourra être envisagée dans les perspectives d'évolution du composant C3.

³⁹ *Un eClassifier est une classe eCore (Cf JavaDoc).*

Listing 5.1: SMS2sql

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0"
3  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4  xmlns:xmi="http://www.omg.org/XMI"
5  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore">
7  <xsl:output method="text" encoding="utf-8"/>
8  <xsl:template match="/">
9    <xsl:apply-templates select="ecore:EPackage"/>
10 </xsl:template>
11 <xsl:template match="ecore:EPackage">
12   <xsl:apply-templates select="eClassifiers" mode="table"/>
13   <xsl:apply-templates select="eClassifiers" mode="alter"/>
14 </xsl:template>
15 <xsl:template match="eClassifiers" mode="table">
16   <xsl:text>DROP TABLE IF EXISTS </xsl:text>
17 <xsl:value-of select="@name"/> CASCADE;
18 CREATE TABLE <xsl:value-of select="@name"/>(
19   id_<xsl:value-of select="@name"/> SERIAL NOT NULL,
20 <xsl:apply-templates select="eStructuralFeatures" mode="table"/>
21 CONSTRAINT pk_<xsl:value-of select="@name"/>
22 PRIMARY KEY (id_<xsl:value-of select="@name"/>));
23 </xsl:template>
24 <xsl:template match="eClassifiers" mode="alter">
25   <xsl:apply-templates select="eStructuralFeatures" mode="alter"/>
26 </xsl:template>
27 <xsl:template match="eStructuralFeatures" mode="table">
28 <!-- a choose would be better -->
29 <xsl:text> </xsl:text><xsl:value-of select="@name"/>
30 <xsl:if test="@xsi:type = 'ecore:EReference'">
31   <xsl:text> integer NOT NULL</xsl:text>
32 </xsl:if>
33 <xsl:if test="@eType = 'ecore:EDatatype http://www.eclipse.org/emf/
34 2002/Ecore#//EString'">
35   <xsl:text> text</xsl:text></xsl:if>
36 <xsl:if test="@eType = 'ecore:EDatatype http://www.eclipse.org/emf/
37 2002/Ecore#//EInt'">
38   <xsl:text> integer</xsl:text></xsl:if></xsl:template>
39 <!-- I suppose the name of eRelations are unique between two classes,
40 i.e. they does not share a common name for different links -->
41 <!-- I suppose also that the foreign key are always the primary key,
42 but a manual schema could use other columns as foreign key -->
43 <xsl:template match="eStructuralFeatures" mode="alter">
44   <xsl:if test="@xsi:type = 'ecore:EReference'">
45     ALTER TABLE <xsl:value-of select="../@name"/>
46     ADD CONSTRAINT <xsl:value-of select="concat(@name,'_',../@name,'_',
47 substring-after(@eType,'#//'))"/> FOREIGN KEY (
48 <xsl:value-of select="@name"/>) REFERENCES <xsl:value-of
49 select="substring-after(@eType,'#//')"/>
50 (id_<xsl:value-of select="substring-after(@eType,'#//')"/>);</xsl:if>
51 </xsl:template>
52 </xsl:stylesheet>

```


Conclusion sur la modélisation

Ce chapitre a présenté un échantillon des modèles que l'on retrouve dans la méthodologie IDM que nous avons suivi.

Les illustrations de ce chapitre couvrent les fonctionnalités principales des composants de l'observatoire.

Enfin, les cas d'utilisation, diagrammes de séquences et l'outillage de transformation permettent de générer du code pour les développements du chapitre suivant.

Chapitre 6 - Développements

Objectifs des développements

L'objectif de ce chapitre est de présenter le cycle de développement suivi et la réalisation des trois premiers prototypes. Ce chapitre présente également le générateur de code et les services d'annuaire et de gestion des alertes.

Cette démarche illustre un échantillon des développements réalisés pour aboutir à une solution intégrable et livrable.

6.1 LE CYCLE DE DEVELOPPEMENT

Ma contribution au projet MeDON a en parti consisté à mettre en place les services Web de divers composants en respectant un cycle itératif de développement. MeDON est un projet innovant et nous l'avons basé sur des prototypes pour identifier de manière incrémentale les difficultés technologiques à résoudre.

Afin de mieux comprendre l'utilité ainsi que le fonctionnement du prototype final, il est important de s'intéresser au processus de développement logiciel qui a rythmé les différents prototypes du projet. La méthodologie a été de suivre des cycles de type roue de Demming sur la production, la documentation, les modèles, les codes, les tests et de façon transverse sur l'architecture globale.

La spécification

C'est l'étape de définition des exigences fonctionnelles, c'est-à-dire quels sont les objectifs du logiciel, les modes de fonctionnements, les aspects liés à la logistique (interfaces), les contraintes majeures. Les performances à atteindre ne font pas parties des exigences fonctionnelles.

Nous définissons nos spécifications lors de nos réunions de travail en collaboration avec notre équipe acoustique passive et le laboratoire de Plymouth qui pilote l'exécution des différentes phases du projet.

J'ai participé activement à la rédaction des spécifications techniques pour chacun des composants mais principalement ceux lié à la gestion des données en base, leur transport et leur partage. J'ai également proposé différentes solutions pour leur orchestration et le design de l'infrastructure physique.

Pour que cela soit possible, il fût nécessaire d'identifier les processus métiers, proposer une architecture fonctionnelle, applicative et technique. L'idée est d'aligner les services suivant ces différents niveaux d'abstraction. D'un point de vue personnel, la thèse de Jacques Simonin[JS10] et la méthodologie d'urbanisation de système d'information du systèmes d'information orienté services du livre Orange de Valtec[val07] me furent très utiles dans l'acquisition de ces nouveaux concepts.

La conception

Cette phase correspond à la création de l'architecture du logiciel, afin d'obtenir une solution répondant aux besoins exprimés. Cela se concrétise dans le projet MeDON par la modélisation UML avec l'outil Rhapsody. J'ai également utilisé MDWorkbench, un environnement de développement intégré basé sur Eclipse. Il permet notamment la conception et la manipulation de modèles, leur transformation ainsi que la génération de code et de documentation. Un certain nombre d'outils et de langages sont supportés dont notamment: Rational Rhapsody, DOORS (IBM), Simulink (MathWorks), Excel, Visio, Word (Microsoft), UML 2.1 (Source <http://sodius.com/mdworkbench>).

Durant cette phase et sur la base des études conceptuelles, j'ai fait les choix technologiques correspondant aux différentes perspectives issues de l'urbanisation du système. Ces différents choix seront analysés dans ce rapport.

En quelques mots, si l'on suit une approche bottom-up, nous retrouvons notre architecture physique et celle-ci rend possible l'exécution d'applications JAVA et PHP. D'un point de vue fonctionnel ce sont des services alignés à nos objectifs métiers et donc définis en étroite collaboration avec nos utilisateurs finaux.

La réalisation

Sur la base de communications fréquentes avec l'équipe projet, j'ai pris en main la réalisation des prototypes et nous avons mis en place des tableaux de bords afin de présenter graphiquement les résultats. Le chef de projet pouvait ainsi prendre des décisions d'arbitrage en cas de déviations et apprécier l'évolution des développements. En interne, des rapports d'avancement ont permis à tous les acteurs d'être informés des actions en cours et achevées.

J'ai réalisé une transformation de notre modèle de données pour générer une première version du script SQL pour notre base de données Postgres. Concernant la base de données, j'ai fait un second test pour utiliser cette fois une feuille de style XSLT pour effectuer la transformation d'une version du modèle exportée en XML.

J'ai également utilisé l'outil Kermeta pour effectuer une génération de code à partir d'un métamodèle et d'un modèle. Rapidement j'ai développé mon propre outil pour générer du code PHP car je souhaitais comprendre et manipuler les mécanismes de cette transformation.

Le spectre des technologies et langages utilisés est assez large car je devais expérimenter différentes manières de manipuler les informations. J'ai ainsi développé différents clients/serveurs en Python, Perl, Ruby, PHP, Java et C afin de simuler des capteurs et réaliser quelques expérimentations. J'ai développé des services Web SOAP et REST dans différents langages (PHP, JSP, Java) et sur les plate-formes Linux et Windows. J'ai également réalisé du code pour une carte embarquée afin d'exporter vers le réseau des fichiers son. J'ai également fait acheter un téléphone Android et réalisé des développements pour cette plate-forme.

J'ai eu à réaliser de l'administration réseau afin de surveiller les différents flux de données, configurer les tables de routages et diagnostiquer les problèmes. J'ai également administré notre base de données Postgres. Au lancement du projet j'ai également mis en place postGIS un module de postgres pour les bases de données géographiques. Suite à cette expérience, j'ai pu déployer ce module sur un autre serveur ce qui a permis à l'un de mes collègues de s'en servir pour donner quelques heures d'enseignements.

Dans le cadre de ce projet et de mes expérimentations, j'ai également administré six machines virtuelles et réalisé différents tests en utilisant VirtualBox, VMware et enfin Xen. La solution User Mode Linux n'avait pas d'intérêts particuliers étant donné que son point fort réside d'avantage dans les possibilités de portage sur des architectures embarquées que sur des critères de performance ou de sécurité.

J'ai également administré un serveur Apache lors de la mise en place de notre plate-forme intranet de travail collaboratif et mis en place quelques scripts pour la communication avec notre annuaire LDAP.

J'ai mis en place un serveur IRC dans le cadre du test du service d'alertes de l'observatoire. J'ai ainsi pu faire travailler un étudiant sur ce serveur afin qu'il réalise un programme destiné à interagir avec les autres utilisateurs présents sur un canal IRC (Bot IRC) pour y diffuser les alertes que je transmettais via le service Web approprié.

J'ai également réalisé quelques prototypes HTML/PHP/Ajax et aussi manipulé GWT⁴⁰ pour générer du code Web à partir de classes Java.

Ces différentes expérimentations respectaient les modèles définis durant la phase de conception.

La validation

Cette phase de description et de conduite de tests permet de s'assurer que le logiciel répond à ses spécifications (conformité à toutes les exigences). Chaque test doit être décrit de manière complète afin qu'une personne extérieure au projet puisse les rejouer sans problèmes.

Dans le cadre des services Web, les tests sont décrits sous la forme de documents XML contenant les différents points d'entrée vers les services. Les points d'entrée sont une description au format WSDL. Ces tests sont alors exécutables avec soapUI⁴¹ qui vérifie que les services en questions réalisent les tâches demandées.

Je n'ai utilisé que très peu JUnit qui permet de tester nos classes Java. Cette tâche était assignée à l'un de mes collègues. Cependant j'ai régulièrement participé au déroulement des tests afin de vérifier que mes nouvelles fonctionnalités ne provoquaient pas d'effet de bords.

Déploiement/Maintenance

C'est l'étape intervenant une fois le logiciel validé. Il s'agit de déployer sur notre infrastructure de production notre architecture logicielle et de fournir un jeu de documentations techniques et utilisateur.

⁴⁰ GWT pour GoogleWebToolkit est le cadriciel de google pour générer des pages HTML dynamiques. Ce "framework" repose sur une librairie Java et un compilateur google.

⁴¹ SoapUI pour Simple Object Access Protocol User Interface, permet de faire des tests unitaires ainsi que des tests de charge sur les web services, il permet notamment de tester des Web Services exposés, mais aussi de simuler la présence de Web Services

6.2 LE PROTOTYPAGE

J'ai participé à la conception et au développement de trois prototypes depuis le début du projet. La raison à cela est que notre connaissance de l'état de l'art a évolué, nos spécifications se sont affinées et la frontière et l'interface entre notre production et celles de nos partenaires se sont progressivement construits.

6.2.1 La première version

Lors de la conception de la première version du prototype, notre objectif était de mettre en place une infrastructure logique répondant aux besoins exprimés par l'équipe acoustique passive. Il s'agissait donc de répondre à six spécifications à savoir être capable de :

- ℞ réaliser des acquisitions de sons provenant d'hydrophones
- ℞ exécuter des traitements sur ces données brutes et générer des fichiers résultats
- ℞ enregistrer en base de données l'activité de l'observatoire
- ℞ partager les résultats obtenus avec nos partenaires scientifiques
- ℞ valoriser via une interface Web certains résultats spécifiques
- ℞ être capable de piloter l'observatoire

Chacune de ces spécifications peut être vue comme un composant ayant une activité autonome. Ces composants doivent s'échanger des messages et des données. J'ai donc proposé l'utilisation de services Web comme solution fonctionnelle pour la communication de ces composants.

Cette première version nous a permis de poser les premières briques logicielles, de prendre du recul sur le type de messages échangés entre les composants et avoir un aperçu des différentes orchestrations nécessaires aux processus de l'observatoire.

6.2.2 La deuxième version

La deuxième version de l'observatoire s'intègre dans une démarche d'amélioration continue. Suite à l'affinement de nos modèles, nous avons précisé notre infrastructure logique et fait évoluer notre code vers une version orientée objet.

Nous avons également architecturé le code des composants en utilisant un modèle en couches en le décrivant comme la composition d'un noyau métier et d'une couche technique. Le noyau métier a des fonctionnalités comportementales et structurelles permettant au composant de réaliser ses tâches. La couche technique permet au composant d'être déployé sur un système ciblé (Cf figure 6.2.2.1).

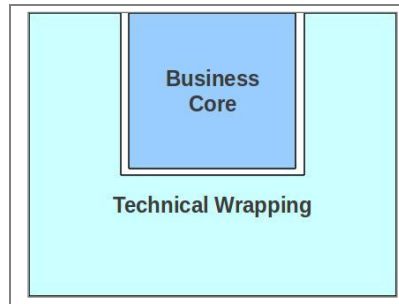


FIG. 6.2.2.1: BusinessCore / TechnicalWrapping

Nous avons également décidé de construire l'application autour du modèle d'architecture Entity-Controller-Boundary (Cf figure 6.2.2.2). [GH09]

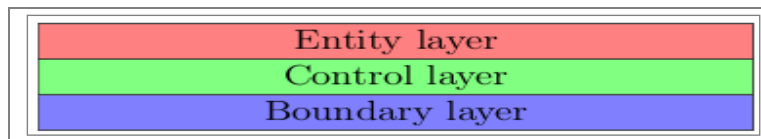


FIG. 6.2.2.2: Architecture en couche

Le paradigme EBC est une approche qui permet de concevoir des systèmes modulaires. Bien que généralement cette approche soit largement utilisée dans les applications à base d'interfaces graphiques orientées objets, ses principes sont applicables sur une grande gamme d'applications. Elle permet de réduire la quantité de code "artificiel" lié à la simple approche orientée objets et de s'assurer de conserver un couplage faible entre les composants de l'application à travers des flux de données maîtrisés.

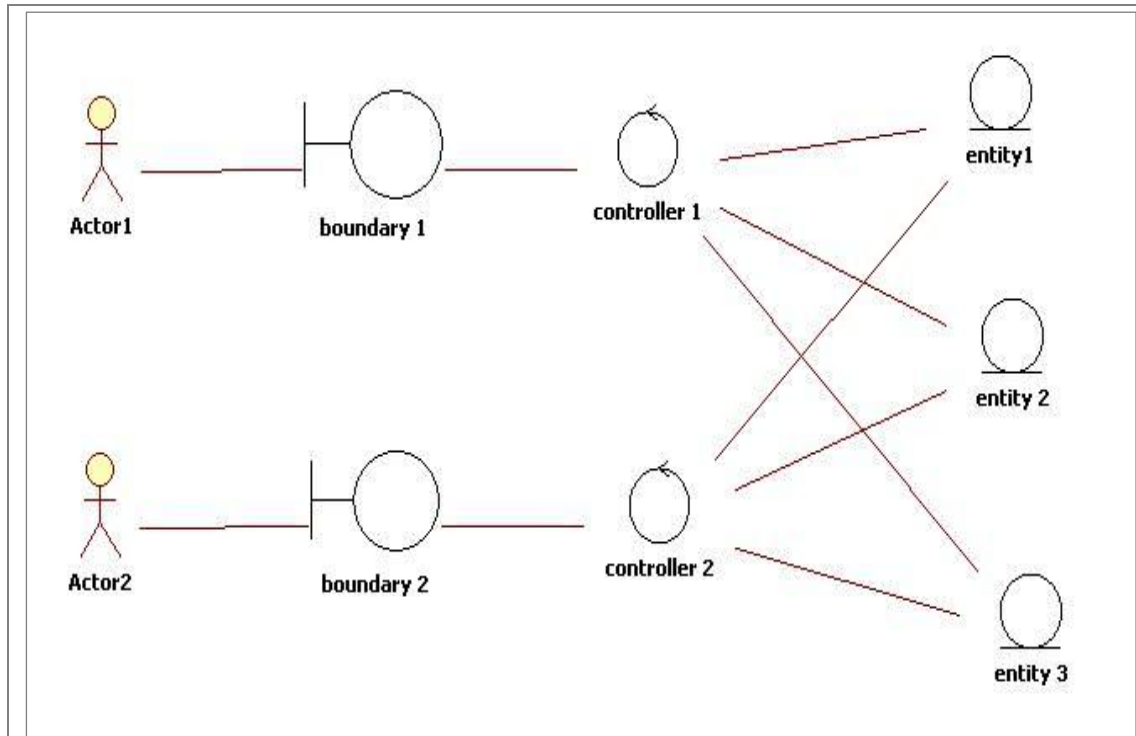


FIG. 6.2.2.3: Pattern Entity-Control-Boundary

L'idée principale dans la conception d'un grand système complexe est de le décomposer en plusieurs sous-systèmes plus simples qui se composent entre eux. Il existe de nombreuses façons de décomposer un système. Le patron EBC (figure 6.2.2.3) permet à chaque composant d'avoir des éléments métiers, orchestrateur et d'accès.

- Ř Les éléments *Entity* sont persistants et passifs. Ils contiennent les données persistantes et une partie de la logique métier du composant.
- Ř Le Control est un élément orchestrateur qui implémente un scénario fonctionnel particulier.
- Ř Boundary s'occupe de la communication depuis (front-end) et vers l'extérieur (back-end).

6.2.3 La troisième version

La troisième version du système d'information de l'observatoire est toujours en cours de conception. Il s'agit de proposer une version urbanisée de notre système et de générer du code à partir de nos modèles qui sont maintenant validés syntaxiquement par EMF.

6.3 LA GENERATION DE CODE

6.3.1 Outils personnels

La modélisation du métamodèle et d'un modèle de l'observatoire MeDON conforme à ce métamodèle m'a permis développer un petit générateur de classes PHP utilisables par nos services Web. Un exemple de code généré est donné en figure 6.1.

Listing 6.1: Server.php

```

1 <?
2 include 'ndd.php';
3 include 'port.php';
4 include 'serve.php';
5 include 'service.php';
6 class server {
7
8     public $ndd;
9     public $port;
10    public $service=array();
11    public function __construct($ndd,$port){
12        $this->ndd=$ndd;
13        $this->port=$port;
14    }
15
16    public function set($ndd,$port){
17        $this->__construct($ndd,$port);
18    }
19
20    public function get($attr){
21        return $this[0];
22    }
23
24    public function json(){
25 $msg="{id:\"".uniqid()."\",name:\"ndd\", data:{},{id:\"".uniqid()."
26 \",name:\"port\", data:{}}"; $msg.=""; return $msg;
27 }
28     public function __toString(){
29         return $this->ndd;
30     }
31
32 function set_service($service){ array_push($this->service,$service);}
33 function get_ndd(){ return $this->ndd;}
34 function get_port(){ return $this->port;}
35
36 protected static $single_connection = null; static public function
37 getInstance($params = array()){if (!is_null(self::$
    $single_connection)){return self::$single_connection;} if (!isset
    ($params['adapter'])){throw new Exception('You must define an
    adapter (MySQL, Oracle...)}; $adapter = ucfirst(strtolower(
    $params['adapter']));if (!class_exists($adapter)){throw new
    Exception(sprintf('The given adapter "%s" (%s) is not installed
    or does not exist', $params['adapter'], $adapter));}self::$
    $single_connection = new $adapter($params); return self::$
    $single_connection;}
38 }?>

```

Ce outil m'a permis de mieux comprendre une partie des mécanismes de génération de code guidée par les modèles. Je n'ai pas réalisé tout le code de l'application mais l'architecture des classes et intégré des accesseurs.

Sur cette base de code, j'ai développé un second outil permettant de visualiser graphiquement les différents composants de notre système d'information : services, serveurs virtuels et serveurs physiques. La visualisation graphique dans une IHM Web a permis de retrouver la perspective physique de notre modèle d'urbanisation. Par ailleurs, du code javascript permet d'invoquer les accesseurs de nos services web et donc d'exécuter les différentes opérations liées aux artefacts. Il

est possible via cette implémentation, d'interagir sur le système réel. J'ai donc commencé à développer une librairie d'opérations dont le *Ping* est un exemple.

Ce second outil m'a permis de valider le fonctionnement du code généré et de contribuer au démonstrateur en offrant un outil de visualisation et d'interaction avec le système d'un point de vue administrateur.

6.3.2 RulesComposer

Rules Composer est un plugin du modelleur UML Rhapsody. Il permet d'effectuer de la génération de code à partir des modèles Rhapsody.

La génération de code avec cet outil est relativement simple. Il s'agit de manipuler deux types de fichiers. Le premier, les « patrons », servent à décrire l'architecture de code à générer. Le second permet de *nourrir* les patrons avec les modèles UML.

J'ai à ce titre réalisé un patron au format *tgt* (figure 6.2) des classes PHP attendues. J'ai également développé un fichier *mqr* (figure 6.3) chargé de parcourir les modèles et d'utiliser les bons patrons de code.

Listing 6.2: PSMClassPHP.tgt

```
[#package GeneC5php]

[#template public PSM_ClassPHP(cls : rhapsody.Class, chaineAttr:
    String)]
[#file]Generated/php/lib/${cls.name}.php[/#file]
/*generated class*/
class ${cls.name}{
    [#foreach att : rhapsody.Attribute in cls.attributes]
        ${att.visibility} ${att.name};
    [/#foreach]
    public function ${cls.name}(${chaineAttr}){
        [#foreach att : rhapsody.Attribute in cls.attributes]
            $this->${att.name} = ${att.name};
        [/#foreach]
    }
    //Getter
    [#foreach att : rhapsody.Attribute in cls.attributes]
        function get${att.name}(){
            return $this->${att.name};
        }
    [/#foreach]
    //Setter
    [#foreach att : rhapsody.Attribute in cls.attributes]
        function set${att.name}(${att.name}){
            $this->${att.name}=${att.name};
        }
    [/#foreach]
}
[/#template]
```

Listing 6.3: phpGen.mqr

```
package GeneC5php;
public ruleset phpGen(in model : rhapsody) {
    public rule main() {
        var chaineAttr: String;
        var virg : String;
        foreach (cls: rhapsody.Class in model.getInstances("
            Class")) {
            chaineAttr="";
            virg="";
            foreach(attributes in cls.attributes){
                chaineAttr=chaineAttr+virg+"$"+
                    attributes;
                virg=",";
            }
            $PSM_ClassPHP(cls,chaineAttr);
        }
    }
}
```

Le code figure 6.4 est un exemple de deux classes que j'ai générées via l'outil Rules Composer de Rhapsody.

Listing 6.4: Alert.php

```

1  /*generated class*/
2
3  class Alert{
4      private $alert_id;
5      private $alert_name;
6      private $alert_guard;
7      private $alert_end_date;
8      public function Alert($alert_id,$alert_name,$alert_guard,
9          $alert_end_date){
10         $this->alert_id = $alert_id;
11         $this->alert_name = $alert_name;
12         $this->alert_guard = $alert_guard;
13         $this->alert_end_date = $alert_end_date;
14     }
15     //Getter
16     function getalert_id(){
17         return $this->alert_id;
18     }
19     function getalert_name(){
20         return $this->alert_name;
21     }
22     function getalert_guard(){
23         return $this->alert_guard;
24     }
25     function getalert_end_date(){
26         return $this->alert_end_date;
27     }
28     //Setter
29     function setalert_id(alert_id){
30         $this->alert_id=$alert_id;
31     }
32     function setalert_name(alert_name){
33         $this->alert_name=$alert_name;
34     }
35     function setalert_guard(alert_guard){
36         $this->alert_guard=$alert_guard;
37     }
38     function setalert_end_date(alert_end_date){
39         $this->alert_end_date=$alert_end_date;
40     }
41 }

```

Ces deux classes s'intègrent dans la librairie que j'ai nommé CallSOAP. Cette librairie d'objets et de méthodes permet à nos partenaires d'invoquer nos services Web plus facilement.

Il existe des outils commerciaux ou libres permettant de générer du code à partir de fichiers WSDL. L'inverse est également possible, c'est-à-dire générer un fichier WSDL à partir du code. Nous pouvons citer à ce titre Java2WSDL et WSDL2Java qui sont deux plugins d'Eclipse pour Axis⁴² et qui réalisent ces tâches.

⁴² *Axis est un serveur de déploiement associé au serveur applicatif Tomcat pour offrir des services web.*

6.4 SERVICE D'ANNUAIRE

Je me suis intéressé à la notion d'annuaire de services Web afin de disposer d'un outil permettant d'accéder à une liste complète des services déployés dans un environnement dynamique.

Dans un premier temps je me suis tourné vers UDDI⁴³, une spécification de OASIS⁴⁴ mais cette solution ne me paraissait pas adaptée pour une communication optimale entre nos services à l'intérieur de notre système d'information. La notion d'annuaire ne suffisait pas à la prise en compte de la notion d'orchestration de services composés. J'ai cependant développé un annuaire basé sur UDDI pour la première version du prototype. L'architecture cible était encore imparfaite, l'annuaire est maintenant distribué dans chaque composants qui viennent se déclarer au *composant pilote* lors de leur initialisation. (Cf les différents cas d'utilisation vu dans le chapitre *Etude conceptuelle* de ce mémoire).

Cette approche est inspirée du mécanisme de référencement des données de certains réseaux décentralisés de type P2P à la différence que notre composant pilote ne sélectionne pas à ce jour de stratégie particulière répondant aux besoins des composants qui l'interrogent. Par ailleurs, notre composant pilote bien que remplaçable et n'étant plus indispensable une fois le système d'information stabilisé, pourrait être intégré dans chacun des composants actuels. Ainsi, il ne serait plus nécessaire de connaître le pilote mais simplement l'un des composants pour intégrer le système. Bien que l'intérêt intellectuel et technique soit discutable, cette dernière fonctionnalité ne m'a pas semblé être prioritaire. Elle n'apporte pas d'intérêts particuliers pour l'atteinte de nos objectifs actuels.

⁴³ Universal Description Discovery and Integration, connu aussi sous l'acronyme UDDI, est un annuaire de services fondé sur XML et plus particulièrement destiné aux services Web.

⁴⁴ L'OASIS (acronyme de Organisation for the Advancement of Structured Information Standards) est un consortium mondial qui travaille pour la standardisation de formats de fichiers ouverts basés notamment sur XML.

6.5 SERVICE D'ALERTES

Le service d'alertes SAS est l'un des services ayant permis de faire évoluer nos prototypes en nous aidant à identifier les différents points conceptuels ou techniques posant problèmes. Comme nous l'avons vu précédemment, il permet de mettre en œuvre toute la chaîne de services pour remonter l'information sous différentes formes des capteurs jusqu'aux utilisateurs finaux et en passant par différents canaux.

Le terme SAS a été donné par OGC⁴⁵ dans le cadre du projet SWE⁴⁶. SAS est plus précisément la spécification d'interface pour la publication et la souscription d'alertes relatives aux données acquises par des capteurs.

Conclusion sur les développements

Les développements réalisés ont concrétisé une longue phase de conception. Ils ont été réalisés dans le but d'obtenir des prototypes fonctionnels mais ont également permis de produire des solutions de générations de code. Il est donc désormais possible de régénérer nos solutions pour **d'autres cibles** moyennant un effort moindre en termes d'étude de faisabilité et conceptuelle.

Le chapitre suivant porte sur l'intégration de ces développements en mettant l'accent sur les aspects méthodologie, architecture d'intégration, gestion de version ou encore habilitations pour l'accès aux services.

⁴⁵ OGC de l'anglais Open Geospatial Consortium, est un consortium international pour développer et promouvoir des standards ouverts, les spécifications OpenGIS®, afin de garantir l'interopérabilité des contenus, des services et des échanges dans les domaines de la géomatique et de l'information géographique.

⁴⁶ SWE pour Sensor Web Enablement est un groupe de travail souhaitant développer des standards d'interface et de méta-données dans le cadre de services interopérables. Ces standards sont destinés aux développeurs qui créent des applications, des plate-formes ou des produits destinés à communiquer via des services Web des données issues de capteurs physiques.

Chapitre 7 - Intégration

Objectifs de l'intégration

L'objectif principal de ce chapitre est de présenter la méthodologie et les bonnes pratiques suivies pour l'intégration de notre solution.

Pour atteindre cet objectif, les premiers points sont consacrés à la méthodologie d'intégration et l'architecture de déploiement mise en place.

Des aspects plus techniques seront abordés à travers la description de la technique d'habilitation utilisée.

7.1 METHODOLOGIE D'INTEGRATION SUIVIE

Pour l'intégration de notre architecture, des composants ainsi que des différents éléments du système, nous avons suivis la méthodologie du PDCA (figure 7.7.1.1 Planifier Déployer Contrôler Agir). En effet, chaque membre de l'équipe a réalisé suivant sa planification les tâches qui lui étaient attribuées. Nos revues de projet hebdomadaires nous ont permis de vérifier l'avancée du projet global et en fonction des problèmes rencontrés nous pouvions corriger ou adapter notre travail.

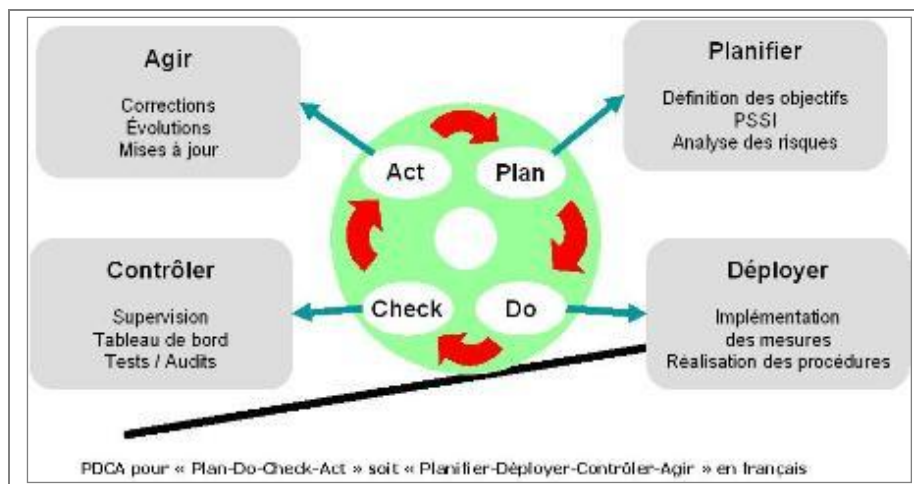


FIG. 7.1.1: Planifier-Déployer-Contrôler-Agir

7.2 L'ARCHITECTURE DE DEVELOPPEMENT

Dans le cadre de l'intégration de nos développements, j'ai mis en place une architecture de serveurs virtuels permettant de tester nos codes lors des développements de nos composants.

Cette architecture peut être vue comme une plate-forme d'intégration continue utilisant les dernières versions de notre système de gestion de version facilitant ainsi le travail collaboratif et la validation de nos avancées en termes de développement logiciel. Le déroulement de nos jeux de tests sur cette plate-forme unique nous permet de contrôler les non-régressions de l'application en cours de développement.

La figure 7.2.1 issue du blog de Joel Touchet illustre cette architecture et les contraintes à respecter relatives à la livraison des produits.

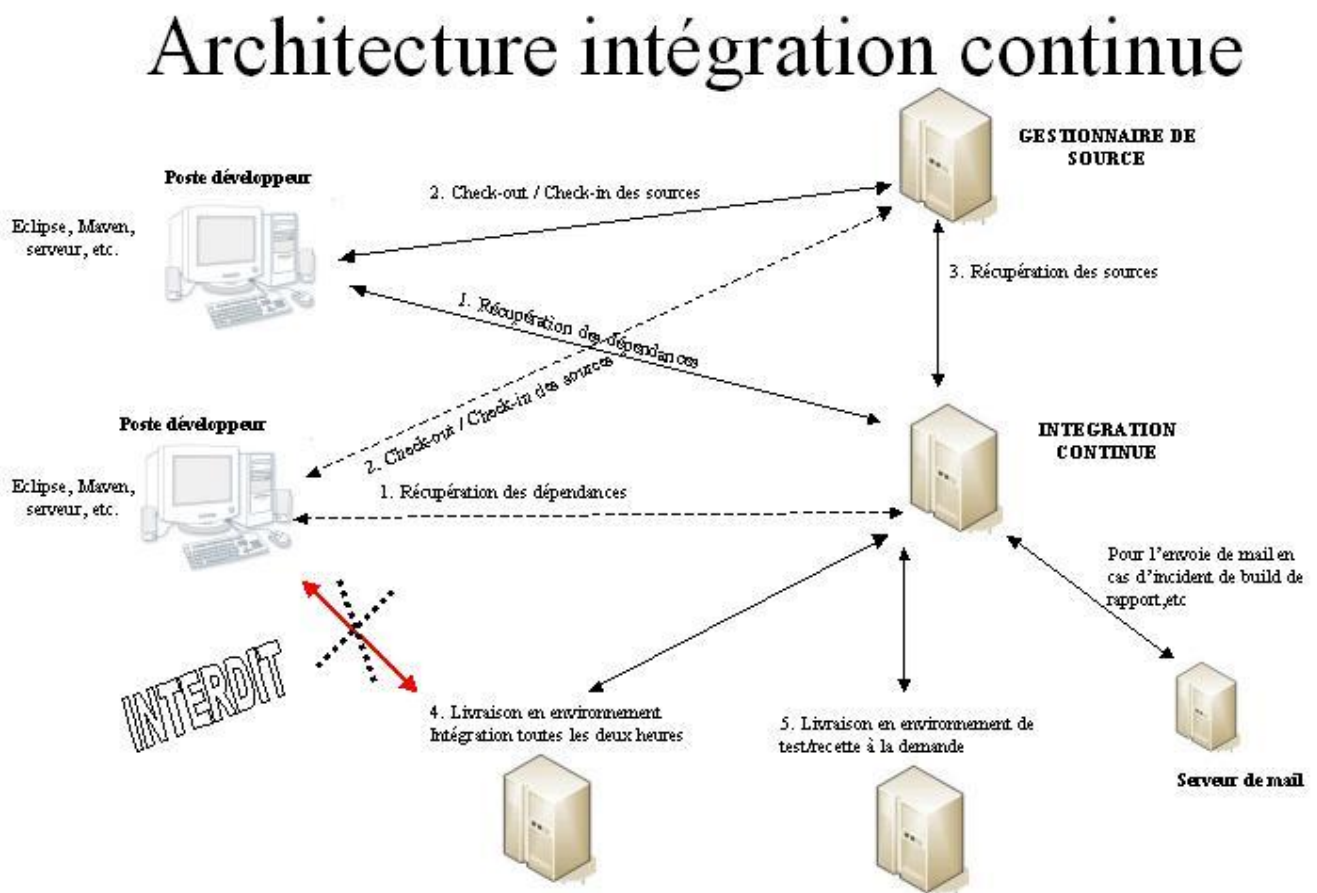


Figure 7.2.1: Intégration continue

7.3 LA GESTION DES VERSIONS

7.3.1 Le code

Les développements ont été soumis à un contrôle de version dont l'hébergement était à la charge de notre service informatique. Nous n'avons de ce fait subi aucune perte de données. Par ailleurs, notre équipe de travail étant relativement peu nombreuse, nous avons collaboré ensemble avec un partage des tâches équilibré et intelligent permettant d'éviter les redondances et les incompréhensions de certains développements ou tâches qui nous étaient affectés.

La gestion de version a été assurée par un serveur SVN. J'ai pour ma part utilisé différents clients en ligne de commande ou graphiques, notamment pour Linux et Windows. De même que pour la documentation relative à VirtualBox, j'ai contribué, au sein du laboratoire, en complétant des tutoriaux concernant l'installation et l'utilisation de ce système.

7.3.2 La base de donnée

Conformément à la solution proposée par Jan Plosky, Wilhelm Hasselbring et Jochen Rehwinkel[JP07], dans le cadre des évolutions de notre base de données qui est un élément central de notre architecture, j'ai proposé un système de contrôle de version de notre schéma relationnel de données (SRD) (Figure 7.3.2.1). Ce contrôle de version nous a permis de travailler collectivement et de pouvoir tracer les évolutions de nos schémas (point 2 et 7 de la figure 7.3.2.1).

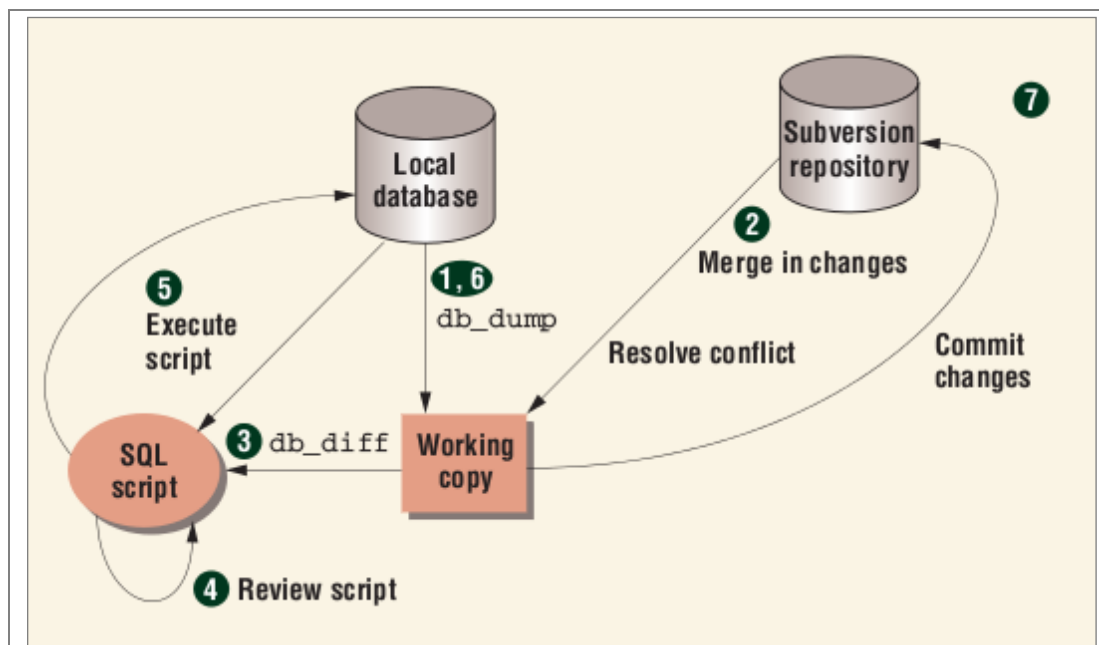


FIG. 7.3.2.1: Flux de données, opérations de récupération et de mise à jour avec un contrôle de version de la BDD

7.4 HABILITATIONS

7.4.1 Les services

Les services métiers composites et unitaires n'embarquent pas de règles d'habilitations. Il ne serait pas très utile de devoir se ré-authentifier sur chacun des services. Les informations d'authentification sont propagées vers notre composant de management des données qui se charge de fournir des données correspondantes à un acteur identifié et habilité pour les données demandées. Les habilitations sont donc gérées par le SGBD ce qui permet d'affiner différents niveaux de granularité de permissions d'accès et de traitement des données.

7.4.2 Habilitation orientée donné

Les privilèges touchent aux opérations du CRUD⁴⁷ mais également aux règles *RULE*. Les règles permettent de définir des commandes spéciales à exécuter lors de certaines opérations sur une table ou une vue.

Les privilèges sur les opérations peuvent porter sur différents objets :

- R table
- R view
- R sequence
- index

Ces privilèges sont attribués sur certains objets en fonction du rôle que jouent les acteurs dans le système. Les acteurs identifiés dans la perspective métier de cette étude correspondent à ces rôles.

7.4.3 Autorisation d'accès à distance

Concernant l'accès à distance au SGBD, un seul composant autre que C3 est autorisé à s'y connecter. Il s'agit de C4 dans le cadre de l'export des données vers nos partenaires ou clients mobiles.

La configuration de cet accès se fait à travers la configuration de postgres en spécifiant l'IP de l'hôte accrédité.

7.4.4 Format d'échange des données

Une intégration harmonieuse de services hétérogènes doit passer par le respect et l'utilisation de standards. Dans le domaine des services Web et plus généralement de la structuration de

⁴⁷ L'acronyme informatique anglais CRUD désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données. Soit : Create, Read, Update, Delete

l'information, JSON et XML apportent chacun certaines particularités dont les principales sont résumées dans le tableau 7.1.

TAB. 7.1: Tableau comparatif des standards JSON et XML

	JSON	XML
Transformation	Accomplie via (dé)sérialisation (Ex Json Jackson, Gson, ...)	Via XSLT ou extraction=>recomposition
Avantages	>Manipulation naturelle pour les technologies Web >Plus léger que XML en termes de ressources	>Intégrité de la structure plus facile à vérifier
Inconvénients	>Pré-suppose de connaître la structure des données	>Format verbeux >DOM et SAX peuvent être gourmands en ressources

Conclusion sur l'intégration

Ce chapitre a donc permis de présenter les mécanismes mis en place dans le projet pour faciliter l'intégration de nos développements. Les développements versionnés peuvent alors être déployés via différents scripts dont l'un sera analysé dans le chapitre suivant.

Chapitre 8 - Déploiement de l'architecture

Le déploiement de nos composants sur l'architecture physique ciblée peut-être automatisé. Cela touche l'installation, la configuration mais également l'administration des serveurs.

Nous allons nous intéresser ici à l'installation de nos codes en suivant pour cela notre diagramme de déploiement.

L'objectif est de fournir un script permettant de déployer nos composants sur une architecture physique conforme à notre modèle de déploiement.

8.1 SCRIPT DE DEPLOIEMENT

Afin de déterminer ce à quoi le script allait ressembler, je l'ai dans un premier temps développé à la main. Un extrait se trouve figure 8.1.1.

```

1  !/usr/bin/expect -f
2
3  set login "root"
4  set HOME "/home/mik"
5  set C1 "172.20.10.217"
6  set C2 "172.20.26.21"
7  set C3 "192.168.1.36"
8  set C4 "192.168.1.36"
9  set C5 "192.168.1.36"
10 set C6 "172.20.10.47"
11
12 spawn scp -r $HOME/workspace/C1Acquisition/bin/enstabretagne/
13          c1acquisition admin@c1.ensieta.ecole:"//var//lib//tomcat6//
14          webapps//axis//WEB-INF//classes/enstabretagne/"
15 expect "password:*"
16 send -- "xxxxxxxx\r"
17 interact
18
19 spawn scp -r $HOME/workspace/C4DataFlux/bin/enstabretagne/
20          c4dataflux \${login}@\${C4}:"/var/lib/tomcat6/webapps/axis/WEB-
21          INF/
22          classes/enstabretagne/"
23 expect "password:*"
24 send -- "xxxxxxxx\r"
25 interact
26
27 spawn scp -r \${HOME}/workspace/C3DataManagement/bin/enstabretagne/
28          c3datamanagement/\${login}@\${C3}:"/var/lib/tomcat6/webapps/axis/
29          WEB-INF/classes/enstabretagne/"
30 expect "password:*"
31 send -- "xxxxxxxx\r"
32 interact

```

Figure 8.1.1 : Deploy.sh

La particularité de ce script est qu'il contient les informations nécessaires à l'authentification pour la copie des fichiers, et interagit de lui-même avec les serveurs cibles.

Pour cela, la première partie du script permet de fixer les variables contenant les adresses des serveurs, le pseudonyme et le mot de passe. Pour chacun des serveurs, les classes des composants Java sont copiées à travers une connexion ssh.

Ce petit script pourra être généré automatiquement à partir de notre diagramme de déploiement.

Dans notre premier prototype, nous avons un serveur virtuel dédié pour chaque composant logique. Nous avons donc optimisé les ressources nécessaires à notre déploiement en prenant en compte la réalité de l'architecture physique et des différentes contraintes réseaux. C'est donc dans le respect de cette logique que nous avons capitalisé au sein de deux machines virtuelles l'utilisation des serveurs Web Apache et Tomcat. L'analyse théorique et simulé des flux réseaux ainsi que la charge

CPU moyenne sont des paramètres qui nous ont servi à choisir le serveur physique. La prise en compte des problématiques de sécurité en est également simplifiée pour plusieurs raisons comme une meilleure maîtrise des entrées/sorties du système via et vers internet ou encore des possibilités de monitoring des instances virtualisées.

L'utilisation de techniques de virtualisation peut être discutée. Dans le cadre de notre observatoire nous pensons que cela nous permettra de mieux "calibrer" les ressources réelles consommées. L'idée est de se donner les moyens d'affiner les ressources allouées. Le prototype doit en théorie pouvoir être déployé sur des architectures pouvant être très différentes du démonstrateur actuel. La gestion des flux d'informations doit être prise en compte en fonction des contraintes physiques et des fourchettes de sollicitations prévisibles. N'ayant pas encore de chiffres réels, ce sont nos simulations et notre expérience qui ont guidé le déploiement actuel. Nous avons toutefois pris les mesures nécessaires avec notre système de *carte du SI* pour être capable de réagir et re-déployer différemment nos serveurs virtuels si nécessaire. Je pense notamment à notre composant C3. Actuellement nous l'avons placé à la frontière de notre système, c'est-à-dire plus près des utilisateurs finaux et de nos partenaires. Ce choix nous permet d'extraire plus rapidement nos données et de le soulager lors des requêtes notre liaison entre Molène et le continent. Les conséquences de ce choix ne sont pas négligeables. Cela va provoquer un flux mieux maîtrisé entre l'île et le continent lors de l'acquisition de nouvelles données ou du déclenchement des algorithmes (Cf diagramme de séquence C2).

Nous pensons qu'il est préférable de prévoir un flux d'informations constant et maîtrisé plutôt que de risquer la congestion du réseau si un trop grand nombre de requêtes devaient être traitées. L'utilisation de systèmes de caches a ses limites car nous préférons fournir des données réelles en petit nombre plutôt que des données potentiellement périmées. Par ailleurs le déploiement actuel devrait permettre de satisfaire nos clients.

Notre espace de stockage repose sur la technologie Raid5 et nous avons prévu des mécanismes d'extraction des données traitées. Les données brutes devraient constituer un trop gros volume, il n'est pas envisageable de les extraire par le réseau.

L'utilisation d'un seul serveur physique sur l'île de Molène devrait nous permettre de réduire la consommation électrique, les coûts d'acquisition et de maintenance tout en permettant une augmentation rapide de la puissance de calcul sur les composants ciblés. Le TCO⁴⁸ sera donc réduit.

Le déploiement logiciel sur chacun des composants est lui aussi décrit dans notre diagramme de déploiement.

Les services Web sont fournis par un serveur Tomcat axis dans le cas de SOAP et par Apache dans le cas du style d'architecture REST.

Sur chacune des machines virtuelles, un serveur SSH offre un accès sécurisé et pour certaines sur un port non standard. La solution Fail2ban permet lors de 3 tentatives de connexions erronées, de

⁴⁸Définition Wikipédia : Le coût du cycle de vie (CCV) (en anglais : life cycle cost ou LCC), dénommé également coût total de possession (en anglais : total cost of ownership ou TCO), coût global de possession ou encore coût global, est le « coût cumulé d'un produit tout au long de son cycle de vie », depuis le début de sa conception jusqu'à son démantèlement.

bannir l'ip à l'aide d'iptables et netfilter. Cette mesure de précaution nous permettra d'être alertés en cas de tentatives d'accès non autorisées en ssh.

Conclusion sur le déploiement de l'architecture

Ce chapitre a permis de présenter l'automatisation du déploiement de nos développements sur notre cible en fonction de notre modélisation.

Cette automatisation a induit un gain de temps non négligeable facilitant l'exécution des tests par exemple que nous analyserons dans le chapitre suivant.

Nous pouvons noter qu'il serait intéressant de développer un outil pour générer le script de déploiement à partir de notre diagramme de déploiement.

Chapitre 9 - Tests et analyse des résultats

Au moment d'écrire ce rapport, le logiciel développé est livrable. En effet, tous les composants sont conçus, codés, testés et déployés sur notre banc de développement et partiellement déployés sur notre infrastructure de production. Le nombre de problèmes remontés se stabilise et l'architecture semble maîtrisée. Elle est aussi documentée grâce notamment au travail de suivi qui fût mené tout au long du projet.

L'objectif principal de ce chapitre est de montrer comment les tests ont été prévus, réalisés et analysés dans ce projet. Ils ont été intégrés dans les développements et analysé jusqu'à ce que l'intégration du système soit complète.

J'ai assisté à la soutenance de thèse de M. Mounir Lallali sur *la modélisation et les tests fonctionnels de l'orchestration de services Web*. Durant cette thèse M. Mounir Lallali a attisé mon intérêt pour BPEL et BPMN et les différentes possibilités de ces langages. Je m'étais, pour ma part, intéressé quelques mois auparavant aux mêmes problématiques en prenant un chemin différent. Alors que je ne m'attachais à ne tester que la disponibilité et quelques cas unitaires, M Mounir Lallali a vraiment modélisé les contextes, l'architecture de tests, ainsi que les services et leur comportement en termes d'attentes et d'interactions. Il a utilisé des automates en IF⁴⁹ et son objectif était de produire un système formel.

La problématique de l'exhaustivité des tests reste bien sûr un problème difficilement soluble et c'est la raison pour laquelle chercher à s'orienter vers une solution à base de tests semi-générés permet d'apporter selon moi un gage de confiance mesurable.

Nous pouvons distinguer plusieurs types de tests :

- ✦ Les tests unitaires, qui ont consisté à tester nos services durant la phase de développement
- ✦ Les tests d'intégration, qui nous ont permis d'intégrer nos différents composants les uns aux autres
- ✦ Les tests systèmes, qui ont consisté à tester nos services dans un environnement proche de notre cible
- ✦ Les tests Alpha, menés par nos "clients" sur notre environnement de test d'intégration. Pour ma part, les étudiants ont été mes testeurs à travers les projets que je leur ai proposés.
- ✦ Les tests bêta, réalisés sur l'environnement de production, pourrons commencer d'ici quelques semaines
 - Les tests de régression, rendus possibles à travers les différentes métriques que l'on peut tirer des performances et services du système.

⁴⁹ IF est un langage à automates temporisés.

Un article de GNU Linux Magazine d'avril 2009[BOU09] de Marion Boulay reprend ces différents cas et présente la théorie sur différentes méthodes de tests logiciel et exprime d'une façon très claire la nécessité de ceux-ci.

9.1 PHASE DE DEVELOPPEMENT

Les tests unitaires sont réalisés en parallèles des développements. L'utilisation de JUnit (Figure 9.1.1) permet de valider les développements des services Web SOAP réalisés en Java pour le serveur d'application Apache Tomcat.

Les tests unitaires sont là pour tester chacune des méthodes afin de vérifier qu'elles répondent vraiment à leurs objectifs. Cela permet de vérifier également la non-régression dans le code lors de nouveaux développements.

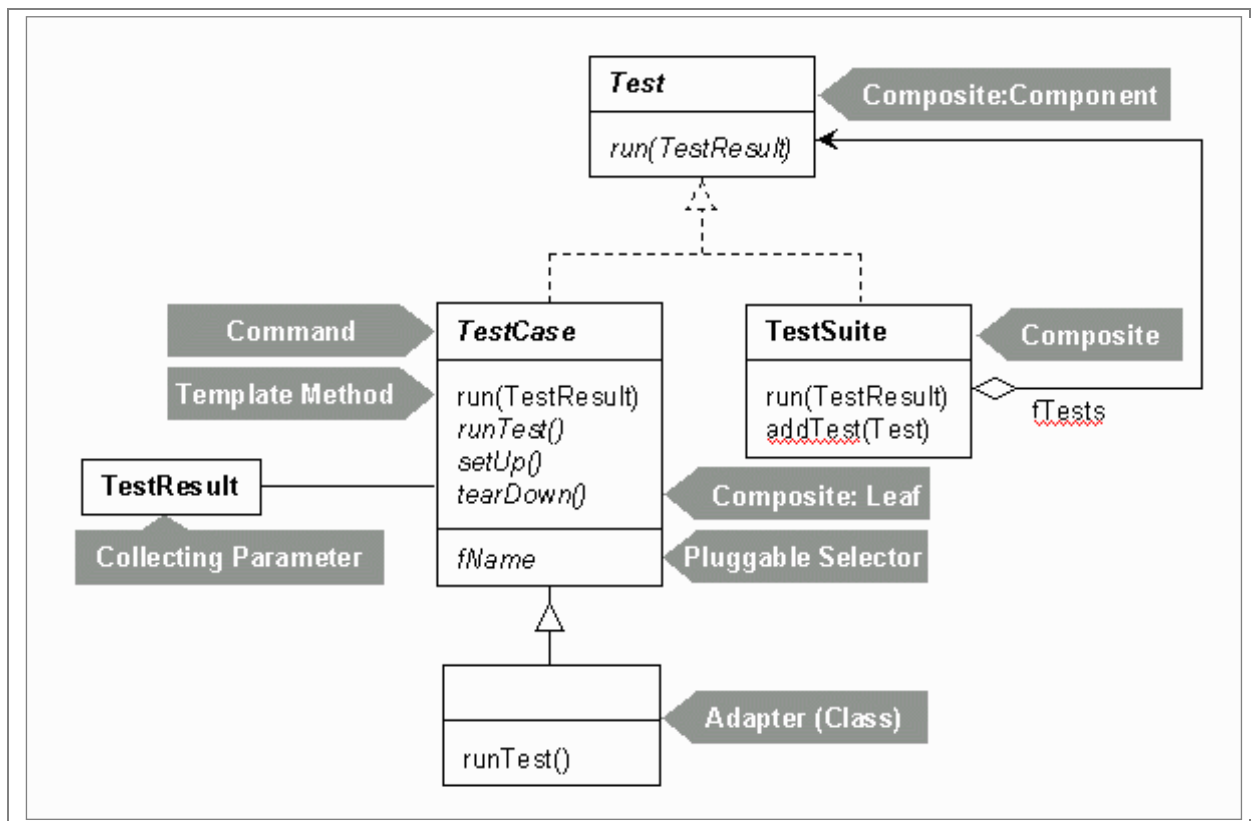


FIG. 9.1.1: JUnit Patterns (Source : <http://gfx.developpez.com/tutoriel/java/junit>)

L'utilisation de SoapUI (Figure 9.1.2) permet de réaliser des tests unitaires ainsi que des tests de charge sur les services Web exposés mais aussi de simuler la présence de services Web. SoapUI peut s'utiliser depuis son interface graphique, depuis la ligne de commande ou être automatisé avec ANT ou Maven. Dans le cadre de cette intégration, seule l'interface graphique a été utilisée.

Il permet d'élaborer une suite de tests à la fois côté client ou serveur pour différentes technologies de services Web. Dans notre projet il nous a permis de tester nos services SOAP.

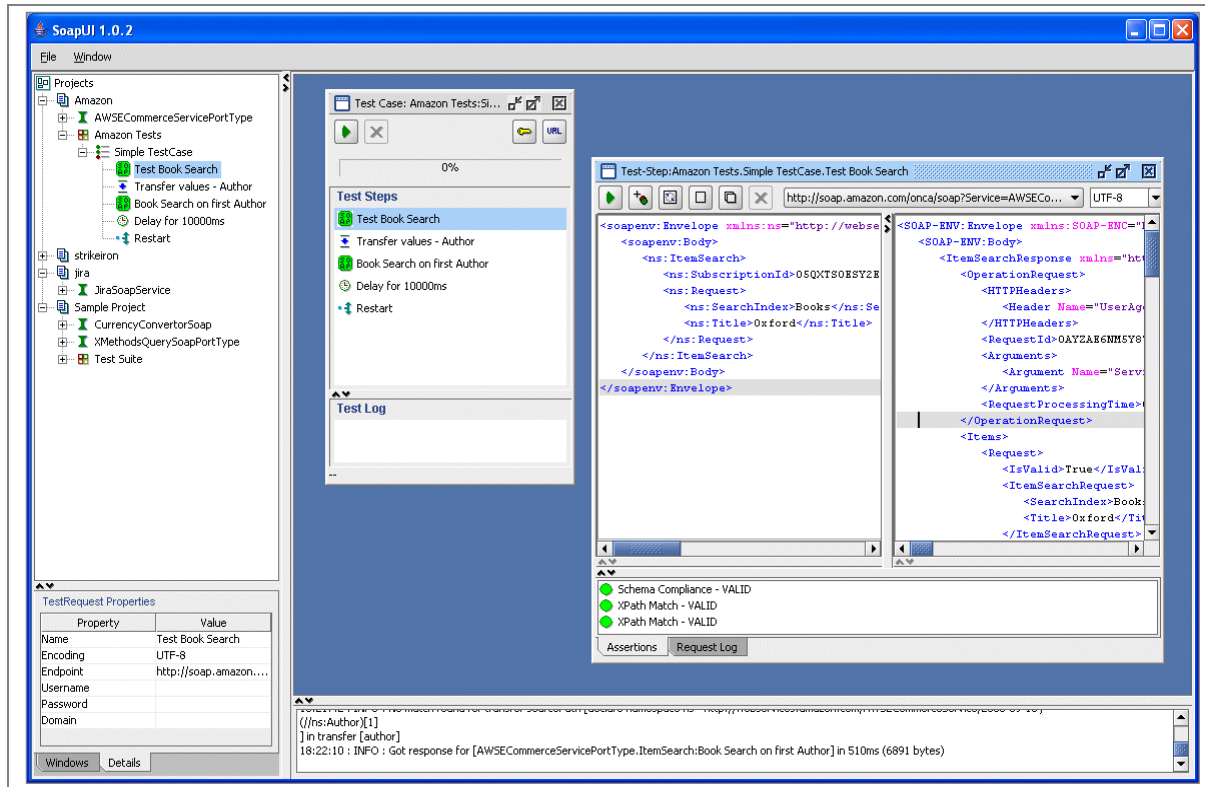


FIG. 9.1.2: Interface graphique de SoapUI

SoapUI a donc été un outil très utile dans le cadre de nos tests. J'ai également eu recours à l'utilisation de scripts PHP qui m'ont permis de jouer des processus complets d'invocations de services Web en suivant différentes orchestrations et en utilisant les classes obtenues par génération automatique de code. J'ai suivi certaines orchestrations de processus qui ont été modélisées, et d'autres qui m'ont permis de corriger certains problèmes durant cette phase développement.

9.2 PHASE D'INTEGRATION

Mes scripts PHP de tests ont porté pour l'essentiel sur le processus d'alerte mais également sur chacun des services *atomiques* de notre SI.

J'ai ainsi développé des tests unitaires en PHP afin de valider le comportement de nos services suivant les contextes définis par des processus bien déterminés. Il ne sera pas possible de rejouer ces processus une fois que le système sera en activité pour plusieurs raisons :

- Le contexte testé va être perturbé par les différents processus rythmant la vie du système en activité
- Les processus testés ont permis de démontrer les réactions du système dans des états particuliers d'échec ou de succès. Le système pouvant prendre un nombre très élevé d'états, il serait maintenant préférable de tester la non-régression et la disponibilité des services.

En résumé, les tests atomiques sur les services pourront continuer à être utilisés. Seuls certains contextes que j'ai cherchés à construire pour tester le système deviendront caducs. Le listing figure 9.2.1 est un extrait des tests réalisés en PHP.


```

1  <? include("lib/callSoap.php");
2  $host=array("hungaroring"=>"hungaroring.ensieta.ecole",
3            "C1:Acquisition"=>"c1.ensieta.ecole",
4            "C2:SoundProcessing"=>"c2.ensieta.ecole",
5            "C3:DataManagement"=>"c3.ensieta.ecole",
6            "C4:DataFlux&Exchange"=>"c4.ensieta.ecole",
7            "C5:InternetAccess"=>"c5.ensieta.ecole",
8            "C6:ObservatoryManager"=>"c6.ensieta.ecole"
9            );
10 //Creation des objets
11 $service=new Service(0,"metric");
12 $NewService=new Service("", "NewMetric");
13 $NewAlert=new Alert("", "newAlert", "<10", date("U"), null);
14 $TrueAlert=new Alert(1, "AlertTest", "<10", date("U"), null);
15 $typeSubscriptionToAdd=new TypeSubscription("", "googleAgenda2");
16 $typeSubscription=new TypeSubscription("3", "googleAgenda");
17 //allUser correspond a un utilisateur lambda et sasuser contient les
18 //donnees relative au service SAS
19 $allUser=new AllUser("userTestiaaa", "userTestLogin", "userTestPasswor
20 );
21 $sasuser=new SasUser("bourvimi@enstabretagne.fr", "0", "#SasIrc");
22 //Deroulement du processus
23 $msg="Liste des senseurs dispo"; (sizeof(file_get_contents("http://
24 anonymous:anonymous@c4.ensieta.ecole/sensor.xml"))>0)?vrai($msg, "
25 Ok"):faux($msg, "Nok");
26 $msg="Liste des evenements pass&#233;"; (sizeof(file_get_contents("
27 http://anonymous:anonymous@c4.ensieta.ecole/event.xml"))>0)?vrai(
28 $msg, "Ok"):faux($msg, "Nok");
29 $msg="Insertion service"; $objNewService=insertService($NewService);
30 $NewService->setId($objNewService->service_id); ($idNewService>=0)
31 ?vrai($msg, "Ok"):faux($msg, "Nok");
32 $msg="Selection des services"; (sizeof(selectService())>0)?vrai($msg, "
33 Ok"):faux($msg, "Nok");
34 $msg="Insertion alerte"; $ins=insertAlert($NewAlert, $NewService); ($ins
35 >=0)? vrai($msg, "Ok"):faux($msg, "Nok"); $NewAlert->setId($ins->
36 alertId);
37 //On peut creer la nouvelle alerte
38 $subscription=new Subscriber($sasuser, $NewAlert, date("U"),
39 $typeSubscription);
40 $msg="Selection des alertes"; (sizeof(selectAlert())>0)?vrai($msg, "Ok"
41 ):faux($msg, "Nok");
42 $msg="Inscription d'un utilisateur"; (registerUser($allUser, $sasuser)
43 ==True)?vrai($msg, "Ok"):faux($msg, "Nok");
44 $msg="Selection des utilisateurs: allUser et sasUser"; (sizeof(
45 selectAllUser())>0)?vrai($msg, "Ok"):faux($msg, "Nok");
46 $msg="Insertion d'un nouveau type de notification"; (
47 insertTypeSubscription($typeSubscriptionToAdd)==1)?vrai($msg, "Ok"
48 ):faux($msg, "Nok");
49 $msg="Inscription d'un utilisateur &#224; une Alerte"; (
50 insertSubscriber($subscription)==true)?vrai($msg, "Ok"):faux($msg,
51 "Nok");
52 echo "<fieldset><legend>Liste des inscriptions</legend>";
53 var_dump(selectSubscriber());
54 echo "</fieldset>";
55 echo "<fieldset><legend>Simulation declenchement evaluation par Algo
56 </legend>";
57 // $res=file_get_contents('http://c4.ensieta.ecole:8080/axis/WS_SAS.
58 jws?method=CheckEvent&event=NewMetric&val=1');
59 $event="NewMetric";
60 $val="1";
61 $msg="On check un nouvel evenement de type '$event' de valeur '$val'";
62 ($val!=""); (sizeof(checkEvent($event, $val))>0)?vrai($msg, "Ok"):faux
63 ($msg, "Nok");
64 echo "</fieldset>";
65 $msg="Annulation inscriptions"; (cancelSubscription($subscription)==
66 True)?vrai($msg, "Ok"):faux($msg, "Nok");
67 $msg="Suppression d'un type de notification"; (deleteTypeSubscription(
68 $typeSubscriptionToAdd)==1)? vrai($msg, "Ok") : faux ($msg, "Nok");
69 $msg="Suppression d'une alerte"; (deleteAlert($NewAlert)==True)? vrai(
70 $msg, "Ok") : faux ($msg, "Nok");
71 $msg="Suppression compte utilisateur"; (removeUser($allUser)==True)?
72 vrai($msg, "Ok") : faux ($msg, "Nok");
73 $msg="Suppression service"; (deleteService($NewService)==True)? vrai(
74 $msg, "Ok") : faux ($msg, "Nok");
75 $msg="Ping C1"; (ping ("c1.ensieta.ecole/") ==True)? vrai($msg, "Ok") :
76 faux($msg, "Nok");
77 ?>

```

Figure 9.2.1 : Extrait du script de tests unitaire en php

9.3 TESTS SYSTEME

J'ai dirigé quelques projets autour de MeDON qui m'ont permis de mettre au point nos services et de les tester. Notamment :

- le projet d'application Android qui m'a permis de mettre au point la DTD⁵⁰ finale pour l'extraction XML des données de l'observatoire sur notre composant C4.
- le projet de simulation et de test des capacités réseau de notre système qui m'a permis de mettre au point les services de test de disponibilité de nos services Web
- le projet de prototypage d'une IHM Web sur C5

Le projet Android m'a permis d'affiner la DTD finale dans le but d'optimiser le volume de données nécessaires pour que nos clients puissent interpréter les données qu'ils reçoivent. Notre client était ici une application mobile pour smartphone qui venait invoquer notre service Web sur C4.

Le projet de simulation et de test des capacités réseau m'a poussé à finaliser les premiers services de tests de disponibilité de nos composants afin d'obtenir des métriques relatives aux temps d'accès aux différents composants.

Le projet de prototypage d'une IHM Web sur C5 m'a poussé à finaliser les services de bases de nos composants de manière à ce que mon stagiaire puisse travailler avec des données réelles et puisse à son tour manipuler ou composer nos services.

⁵⁰ Source Wikipedia : La Document Type Definition (DTD), ou Définition de Type de Document, est un document permettant de décrire un modèle de document SGML ou XML. Le modèle est décrit comme une grammaire de classe de document[1] : grammaire parce qu'il décrit la position des termes les uns par rapport aux autres, classe parce qu'il forme une généralisation d'un domaine particulier, et document parce que on peut former avec un texte complet.

9.4 METRIQUES SUR LES DEVELOPPEMENTS

Le tableau 9.4.1 synthétise le volume des développements réalisés depuis février 2011 pour chaque composant. Nous étions deux développeurs et l'utilisation de l'IDM a permis de réaliser des développements propres dans la mesure où l'architecture était claire et efficace car nous savions exactement ce qu'il y avait à faire, et comment le faire.

TAB. 9.4.1: Métriques sur les développements depuis février 2011

Composant	Nombre de classes	Nombre de lignes de code
C1 Acquisition	23	1658
C2 Sound Processing	25	1830
C3 Data Management	29	2787
C4 Data Flux	24	2712
C5 Web Presentation	16	4109
C6 Observatory Manager	16	1370
Common Types	21	1704
Total	189	16170

Conclusion sur les tests et de l'analyse des résultats

Les tests menés sur le système nous ont permis de contrôler les développements car ils ont été prévus en amont dans les phases de spécification du système. Ce chapitre a permis de faire le point sur les technologies utilisées pour réaliser ces tests et les raisons qui ont guidé ces choix.

L'analyse des résultats et la cartographie de l'état de nos services nous ont permis de concentrer nos développements, révisions de code et corrections de bugs de façon à optimiser notre productivité.

Les tests et l'analyse des résultats nous ont permis de valider notre travail et la bonne prise en compte des exigences du système. Les délais ont donc été respectés et l'analyse des résultats nous a permis de communiquer lors de la revue finale du projet auprès de nos partenaires français et anglais.

L'analyse des résultats permet de faire le point sur les apports scientifique de notre production et d'un point de vue macroscopique, réfléchir aux perspectives du projet.

Troisième partie

Perspectives

Chapitre 10 - Apports scientifique

Ce chapitre présente de manière concise les résultats obtenus.

Dans le contexte scientifique actuel du projet, l'axe développé permet d'émettre des hypothèses relatives aux perspectives de la méthodologie IDM et de ce que l'on peut principalement attendre de ce type d'observatoires.

L'objectif de ce chapitre est donc de faire le point sur les perspectives en termes d'objectifs atteignables concernant les observatoires, puis l'IDM.

Dans un second temps l'accent est porté sur la communication qui doit être faite autour du projet par ses acteurs et sur l'aspect collaboratif, nécessaire pour l'avancée de ces travaux.

10.1 DE NOUVEAUX OBJECTIFS

10.1.1 Les observatoires

Outillage scientifique

Le projet MeDON a ouvert de nouvelles perspectives pour l'acquisition, le traitement et le partage de données scientifiques. Il est maintenant envisageable de le déployer à grande échelle pour la recherche et la compréhension de certains phénomènes naturels qui restent encore aujourd'hui des mystères non résolus.

La surveillance des phénomènes physiques marins en temps réel et l'utilisation de réseaux interconnectés nous offrent enfin l'équipement nécessaire à la compréhension, l'anticipation et l'application de mesures correctives des incidences de l'activité humaine.

Les récents développements des technologies embarquées et surtout la mise à disposition au grand public de plate-formes disposant de capacités de calcul conséquentes ouvrent de nombreuses perspectives d'évolutions à notre réseau. Nous avons imaginé un observatoire constitué de composants communicants ayant chacun un ensemble de fonctionnalités destinées à des buts bien particuliers. Une famille de composants identifie alors un observatoire. Il nous reste à travailler sur le pilotage de plusieurs observatoires, le déploiement de stratégies d'observation à grande échelle, le traitement de ces stratégies et enfin l'utilisation et l'exploitabilité de ces résultats.

Outillage de veille environnementale

D'un observatoire destiné à l'étude des phénomènes marins, il serait possible, pourquoi pas, de contribuer à une surveillance globale : marine, terrestre et aérienne. La lutte contre le réchauffement climatique, la détection et l'anticipation de catastrophes naturelles ou encore l'information sur une ré-habilitation potentielle d'une zone contaminée, sont autant d'éléments qui motivent nos institutions à persévérer dans le développement de ce type de projets.

Pérennités des observations

Un second point qui me semble intéressant dans notre approche concerne l'automatisation de l'acquisition des résultats sur le long terme. En effet, les expéditions scientifiques telles que celles réalisées au pôle nord, dans des volcans ou autres régions difficiles d'accès induisent des coûts, des délais et une qualité de mesures qui peuvent varier d'une expédition à l'autre. Le déploiement d'observatoires pérennes nous permettrait d'économiser sur ces trois plans, et réduirait le nombre de risques pris par les scientifiques lors de certaines expéditions.

Un observatoire libre et des capteurs intelligents

Là où notre approche diffère des observatoires existants est que nous immergeons des capteurs dits "intelligents". Nous pouvons les reprogrammer ou les faire communiquer entre eux afin de réaliser des services collaboratifs. A court terme nous devons déployer des stratégies d'observations et communiquer sur ces résultats. Les scientifiques pourront alors apprécier les capacités de l'observatoire et pourrons librement déployer leurs propres algorithmes.

10.1.2 L'ingénierie dirigée par les modèles

L'IDM apporte des réponses à différentes problématiques de conception, de développement, et d'intégration du logiciel. Elle permet notamment une meilleure traçabilité sur tout le cycle de développement face à des systèmes de plus en plus complexes.

Ce projet nous a permis de faire un petit pas supplémentaire vers la spécification et le développement d'un cadre d'intégration pour la conception et le maintien d'une chaîne d'outillage permettant le développement de systèmes embarqués.

De nos jours la complexité des applications informatiques devient un problème qu'il faut savoir maîtriser. Cette complexité vient du fait qu'elles peuvent être distribuées, mobiles, communicantes ou composées de plusieurs sous-systèmes. Leur modélisation en UML devient alors une activité qui peut sembler être "réservée" à des spécialistes. D'autant plus que la modélisation du logiciel perd alors de son sens si le système lui-même n'est pas modélisé. SysML (Figure 10.1.2.10.1.2.1 est un langage⁵¹ récent qui promet de combler ce manque dans les années avenir.

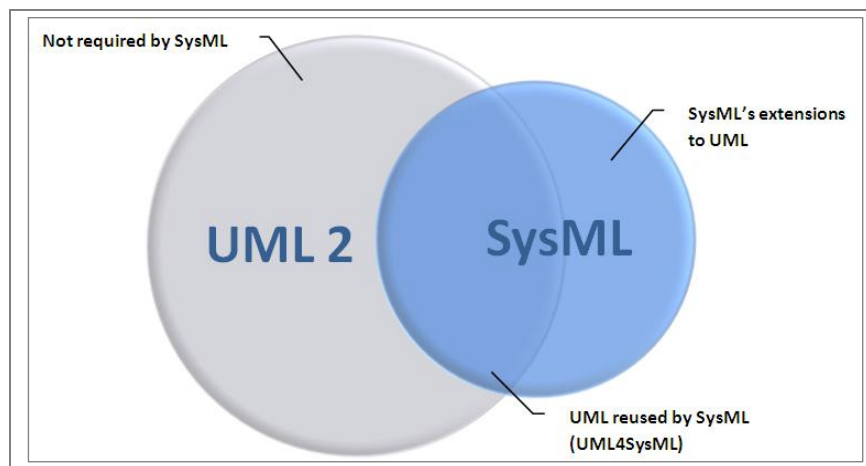


FIG. 10.1.2.1: Positionnement SysML et UML2

La modélisation des exigences et des tests sont également deux domaines qui peuvent être considérés comme étant très jeunes ou non aboutis. Les travaux sont en cours, mais il n'existe pas encore de réelles chaînes d'outillages complètes, unifiées, permettant de prendre en compte ces aspects dans l'ensemble du cycle de développement.

L'impossibilité d'exécuter des modèles limite actuellement les profits de la modélisation. L'objectif est de pouvoir valider en amont les exigences en termes de comportement logiciel. La mise au point de couplages forts entre des outils de simulation et de modélisation est une perspective intéressante pour les industriels. UML seul ne prend pas en compte ces aspects de co-simulation et de communication entre les outils ainsi que l'intégration dans les modèles eux-mêmes, de règles de calculs ou de savoir-faire métiers.

⁵¹ SysML (Systems Modeling Language) est basé sur UML et remplace la modélisation de classes et d'objets par la modélisation de blocs pour un vocabulaire plus adapté à l'Ingénierie Système. Un bloc englobe tout concept logiciel, matériel, données, processus, et même la gestion des personnes.

Plusieurs grands centres de recherches s'intéressent à l'IDM (notamment VERIMAG, IRIT, LIP6, INRIA) et les grandes entreprises (Thales, Airbus, CEA, CNES,...) ont véritablement besoin de ce genre de méthodologies dans différents pôles de compétitivités (AerospaceValley, Minalogic, System@tic, ...).

Les chercheurs se concentrent actuellement sur différentes thématiques comme :

- le model checking⁵²,
- la définition de processus de développement orientés modèles (Domino, Topcased,...)
- la génération de code à partir de modèles (Scade, Gene-auto),
- la traçabilité des exigences et des transformations,
 - le développement d'outillages (MDPapyrus, TOPCASED, ATL, Kermeta, MDWorkbench, Aceleo).

Ces différentes thématiques correspondent aux besoins des industriels avec qui il faut donc coopérer pour une intégration plus poussée de ces outils, une meilleure ergonomie des solutions et donc selon moi une vulgarisation de l'approche.

⁵² Le model checking permet de transformer des modèles semi-formels en modèles formels (SPaCIFY, VerifME). VerifME est l'un des projets de notre équipe de travail.

10.2 COLLABORATION

Fin juin a eu lieu la dernière réunion d'avancement de MeDON à laquelle j'ai assisté. Il s'agissait de faire un point sur l'avancement du projet avec nos partenaires anglais. Ce premier projet collaboratif sur la thématique des observatoires nous a permis de nous rapprocher avec nos collaborateurs anglais sur la scène Européenne. Cette expérience nous a permis de travailler ensemble vers un objectif commun et de partager des savoir-faire.

Cette réunion nous a permis de faire le point sur le travail réalisé et les perspectives à court, moyen et long terme pour ce type d'observatoire et ce type de collaboration. Nous avons continuellement échangé et cette collaboration nous a permis de mutualiser nos moyens pour atteindre des objectifs ambitieux.

Cette collaboration a permis la communication entre diverses parties prenantes et utilisateurs finaux concernés par le projet. Ces partenaires vont maintenant pouvoir s'organiser en groupes de travail interactifs afin de faire évoluer notre compréhension des besoins et la qualité des solutions qu'ensemble nous pouvons construire.

Ce travail va permettre la mise en place de stratégies de veille environnementale pour répondre aux besoins que les groupes de travail ont identifiés.

10.2.1 Visibilité du laboratoire

Notre participation à ce projet devrait avoir un effet positif sur la visibilité du laboratoire auprès du grand public et de nos partenaires scientifiques. Nous avons prouvé un savoir-faire à travers notre méthodologie et notre communication. D'autres partenariats de ce type sur la thématique des observatoires sont à l'étude.

10.2.2 Échange de données

Nous sommes désormais capables d'échanger des données avec les observatoires existants. Cela devrait nous permettre de réaliser de nouvelles collaborations à court terme.

Durant la phase 2 du projet qui commence là où mon travail s'achève, un second observatoire sera conçu et déployé outre manche en se basant sur l'expérience acquise lors de nos développements. Différentes études de sites ont déjà été réalisées pour optimiser la position du second observatoire et le tracé de son câble immergé.

Conclusion sur les apports scientifique

Ce chapitre a permis de présenter les nouveaux objectifs de l'observatoire et quelques perspectives de l'IDM. La collaboration entre les différentes parties prenantes a permis un rapprochement des équipes de recherches. Les fonctionnalités de l'observatoire vont permettre à d'autres chercheurs de contribuer à la compréhension de phénomènes sous-marins et sensibiliser le grand public à ces aspects.

Chapitre 11 - Fonctionnalités de l'observatoire

L'axe fonctionnel se focalise sur l'activité d'observation sous-marine et le concept d'observatoire et met en valeurs les perspectives d'évolutions et de communications de telles infrastructures.

L'élaboration des spécifications techniques des instruments, des capteurs et du système d'information que nous avons réalisée peut maintenant permettre le déploiement de nouveaux observatoires.

L'outil de veille par monitoring passif que nous avons développé permet quant-à lui d'effectuer des analyses sur des données acquises en temps réel. Par ailleurs, le système d'alerte est personnalisable et les « abonnements » couvrent déjà plusieurs réseau comme l'IRC, la messagerie instantanée et les systèmes de mails (abonnés, newsletters, liste de diffusion).

L'objectif principal de ce chapitre est de prendre en compte certaines recommandations, étudier les prochaines pistes et perspectives d'amélioration de la solution que nous proposons, et de celle développée en parallèle par nos partenaires anglais dans le cadre de la suite du projet.

11.1 RECOMMANDATIONS

Les phases de conception et de réalisation de ce système ont fait apparaître certains dysfonctionnements. Il est intéressant d'en comprendre les raisons afin d'améliorer l'évolution des futurs projets similaires. A ce titre, différentes recommandations me semblent primordiales.

Avant tout, il est nécessaire dans ce type de travail collaboratif, que chaque partenaire (scientifique, administrateur système, utilisateur final, ...) se mettent d'accord dès le début du projet sur les fonctionnalités **de base** de l'application. La compréhension des fonctionnalités secondaires pouvant évoluer, mais tout le monde doit construire le système dans la même direction et faire des compromis. Il me semble qu'il est plus aisé de réaliser cela dès lors que les tâches sont clairement définies, non concurrentes et que chacun puisse avoir une vision globale des résultats attendus.

Pour ce faire, les exigences doivent spécifier différents concepts :

- R Quels sont les objectifs métiers des acteurs ?
- R Qui va interagir avec le système et comment ?
- R A quoi sert le système ?
- R Quelles sont les données que l'on manipule et que doit-on en faire ?
- R ...

Il est alors absolument nécessaire de bien définir le rôle de chacun des partenaires et leur périmètre d'actions sur le système. Il est également nécessaire de définir une politique d'accès aux différentes parties du SI.

L'utilisation du langage UML est à mon sens l'un des meilleurs moyen pour communiquer et créer de tels systèmes. Nous prôtons son utilisation de la mise en place des spécifications jusqu'au déploiement sur les cibles.

11.2 LES PROCHAINES PISTES

11.2.1 Méthodologique

Une méthodologie pour être opérationnelle doit avoir trois composantes :

- Une démarche
- Des formalismes
- Une organisation

D'autres méthodologies tel que Merise2 par exemple nous aurait également permis d'aboutir à de bons résultats. La démarche, les formalismes et l'organisation auraient alors été différents.

Ces travaux ont permis d'identifier de manière incrémentale les besoins d'un système d'information pour ce type de projet d'innovation.

11.2.2 Le concept de Cloud Servicing

Parmi les perspectives technologiques que nous offrent de telles architectures, nous pourrions intégrer le concept de cloud computing (Figure 11.2.2.11.2.2.1). Le cloud computing est le mariage entre le concept de virtualisation et de load balancing. Nous pouvons voir notre SI actuel comme un nuage de ressources intégrant ces deux concepts. La perspective est ici de permettre à notre architecture actuelle d'évoluer en ce sens, et permettre une distribution intelligente des calculs. Par distribution intelligente il faut entendre que plusieurs composants peuvent réaliser les mêmes types de calcul. Il faut alors que notre pilote puisse distribuer les traitements en prenant en compte les différentes charges.

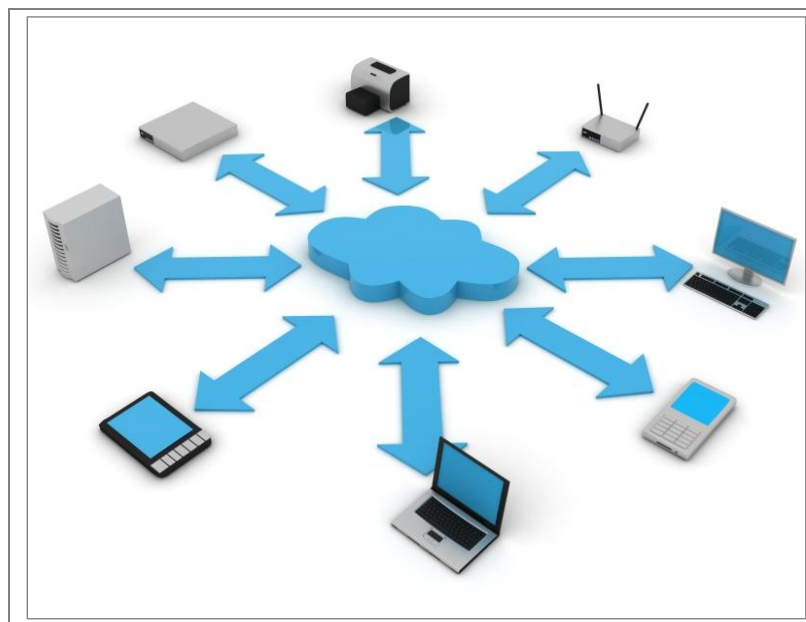


FIG. 11.2.2.1: Le cloud computing

Le load balancing est donc la capacité de distribuer les charges, et la virtualisation permet d'émuler des environnements hétérogènes au sur une architecture de ressources physiques. La spécification que l'on a donnée à notre observatoire peut donc être vue comme ce que j'appellerais le concept de *cloud servicing*, ou *nuée d'entretien*. Notre système est destiné à offrir des services reposants sur un cloud à la différence du cloud computing ou la vocation est simplement d'offrir une capacité de calculs pérenne. Nous proposons ici un accès à des ressources de calcul intelligentes et distribuées. Par ailleurs, le caractère distribué de nos observatoires, et les cibles technologiques choisies telles que l'embarqué pour le matériel, et des standards ouverts pour le logiciel, font que nous ne reposons plus sur une architecture de calcul monolithique et donc plus robuste aux pannes. Ce type de services s'intègre parfaitement dans l'évolution de l'informatisation des systèmes actuels. A l'heure où notre domotique commence à rejoindre le réseau, nous pouvons proposer ici une méthodologie pour permettre aux chercheurs de déployer des stratégies de communication entre des systèmes intelligents.

11.2.3 Les interfaces

La problématique de la génération automatique d'IHM reste encore un défi à relever. La raison réside dans le fait qu'il est difficile de concevoir un système souple de générateur de générateur (2 fois) de code prenant en compte toutes les spécifications ergonomiques désirées au vu du nombre de possibilités. Il existe bien-sûr les simples générateurs d'IHM qui par définition visent une cible spécifique.

Nous avons contourné cette problématique en fournissant une IHM simple et en consacrant notre temps à réfléchir au format d'échange des données. Nos interfaces permettent de *transporter* l'information dans un format d'échange standard et transformable. Ces interfaces se présentent comme des services ouverts. Sur la base fonctionnelle des services Web, l'esprit est ici de fournir des outils interagissant les uns avec les autres. Bien que notre équipe soit en dehors de la communauté OSLC (Figure 11.2.11.2.2)⁵³, nous en respectons ici les concepts. C'est-à-dire fournir des services ouverts et faciliter la communication entre les artefacts.

Source OSLC :

The intention is to make life easier for software and product developers and tools vendors, by making it easier for tools to work together.

⁵³ L'initiative « Open Services for Lifecycle Collaboration » (OSLC) crée une famille de spécifications de services Web pour des produits, services et autres outils qui soutiennent toutes les phases du cycle de vie des logiciels et produits. L'objectif de ces spécifications est de permettre l'intégration entre des produits qui opèrent la « gestion du cycle de vie des applications » (Application Life-cycle Management RALM) ou la « gestion du cycle de vie du produit » (Product Life-cycle Management RPLM).

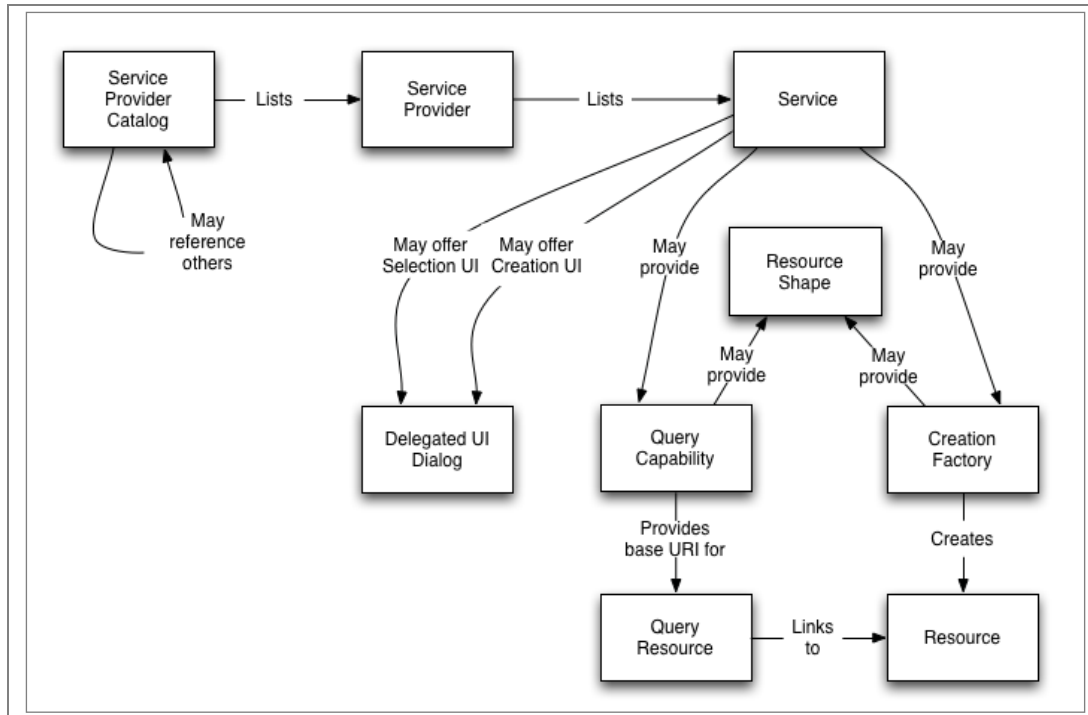


FIG. 11.2.2: Concepts et relations dans la spécification de base OSLC

A court terme, nous devons maintenant concevoir et diffuser les premiers services clients pour que des applications puissent utiliser nos résultats et que nous puissions intégrer les leurs. Le projet utilisant la technologie Android lié à ces travaux, s'intègre dans cet esprit (Cf chapitre sur le service d'alerte SAS du cas d'utilisation du composant 4, flux de données et échange, figure 5.1.5.1).

Nous devons donc communiquer sur le projet auprès du grand public et inviter nos homologues à utiliser notre système et à nous faire part de leurs critiques tel que prévu pour la prochaine phase du projet.

11.3 LES PERSPECTIVES D'AMÉLIORATIONS

11.3.1 La validation par les utilisateurs finaux

A ce jour nous ne disposons pas encore des résultats des bêta-tests. L'avenir du projet est encore incertain. Des financements sont encore nécessaires avant que le prototype concret ne puisse être éprouvé. La dynamique autour du projet a tout de même permis de renforcer nos échanges avec nos partenaires européens et canadiens.

De ce fait, certaines interrogations pour lesquelles nous n'avons pu tester les solutions que de manière théorique (notamment le système d'échange de carte entre les composants communicants ainsi que les métriques relatives à l'architecture physique) pourraient si nécessaire faire l'objet d'une seconde étude.

11.3.2 Matériel

L'observatoire MeDON est matériellement conçu pour que ses capteurs puissent rester immergés une quinzaine d'années. Les premières cartes électroniques permettent de démontrer le concept et n'ont pas vocation à être réutilisées. Elles devraient faire l'objet de diverses améliorations une fois le concept validé. Cela afin de répondre aux requêtes des scientifiques.

Le câble reliant les capteurs devrait lui pouvoir être réutilisé. Il a été conçu pour résister à de fortes contraintes de courants marins. Par ailleurs les pêcheurs de la zone sont alertés de la présence du câble, et équipés pour ne pas l'endommager (Source : France telecom).

11.3.3 Logiciel

La qualité

Si l'on se rapporte à la roue de Demming (figure 11.3.3.1), nous avons planifié et réalisé le premier prototype d'observatoire. Il reste maintenant à vérifier que le prototype réponde bien aux attentes de nos clients. En fonction de leurs rapports, il sera alors nécessaire de réagir en conséquence et de progresser dans ce processus d'amélioration permanente.

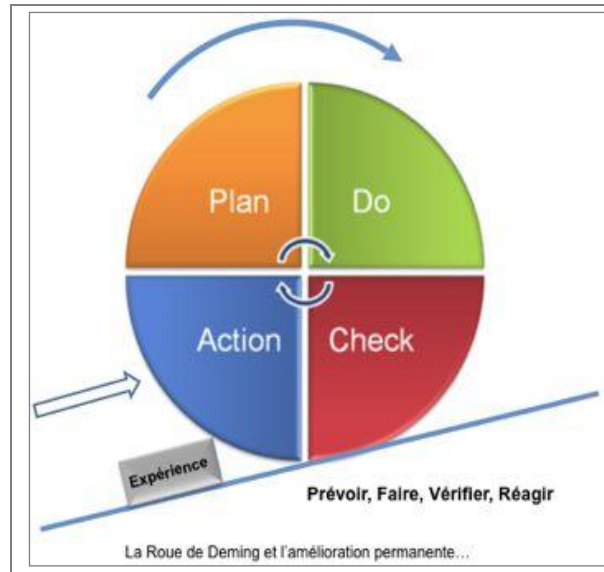


FIG. 11.3.3.1: Plan / Do / Check / Act

En interne, cette approche de la qualité s'exprime par le fait que la roue a déjà réalisé les tours des premiers prototypes. Nous avons donc réalisé trois tours et notre expérience a donc déjà progressé. Notre documentation constitue une base assez solide pour que d'autres équipes puissent continuer nos travaux. A court terme, nous avons planifié l'approfondissement de la modélisation de tels systèmes.

Une reconfiguration dynamique du réseau de capteur

Dans le domaine du génie logiciel, l'accent est porté sur le fait de développer des applications modulaires et faiblement couplées. Cela permet d'envisager de nombreuses fonctionnalités plus facilement interchangeables et remplaçables en cas de défaillances. D'un point de vue logiciel, la reconfiguration dynamique de ces composants devient une perspective qui nous intéresse particulièrement. Dans le cadre de nos développements, nous avons déjà imaginé une configuration terrestre du réseau, rendant possible une mise au point plus aisée de notre architecture réseau. Les capteurs peuvent facilement être calibrés en fonction du milieu dans lequel ils opèrent.

Une reconfiguration dynamique va plus loin qu'un simple recalibrage des capteurs. Nous souhaitons ici pouvoir proposer à terme des mécanismes permettant le remplacement total ou partiel de composants logiciels embarqués. Dans le cadre de notre prototype, cette technologie aurait été trop risquée car coûteuse en temps, en délais de réalisation, et en argent.

Désormais cette perspective va devenir possible, et c'est un concept que nous devrions embarquer dans les prochains capteurs intelligents. Ce type de fonctionnalité peut donc consister à paramétrer un capteur en fonction de son environnement, activer ou désactiver certaines fonctionnalités suivant la stratégie de monitoring de l'environnement ou encore du niveau de batteries d'un capteur par exemple. Là encore, ces stratégies ne peuvent être totalement modélisées en UML pur et d'autres langages comme SysML nous permettent d'exprimer ces comportements "système".

Une nouvelle stratégie d'acquisition de données

La stratégie de surveillance des phénomènes marins est-elle aussi à l'étude. Dans le cadre de MeDON, dans le souci du respect des contraintes environnementales, il a été choisi d'utiliser

uniquement des technologies passives et donc non intrusives pour l'environnement. Il existe des techniques de surveillance active c'est-à-dire basées sur l'émission de signaux. Il serait intéressant pour nos scientifiques de pouvoir croiser les deux approches.

Notre équipe *acoustique passive* est déjà en relation avec le laboratoire E3I2 (Figure 11.3.3.2) (Extraction et Exploitation de l'Information en Environnements Incertains) de l'ENSTA-Bretagne qui travaille sur des technologies visant à la mise en place d'un système d'information et d'assistance pour l'aide à la décision dans un environnement perturbé et évolutif. Ainsi, les travaux de ce laboratoire sont orientés vers la modélisation électromagnétique et acoustique, l'extraction et l'exploitation de l'information issue de signaux sonar et radar.

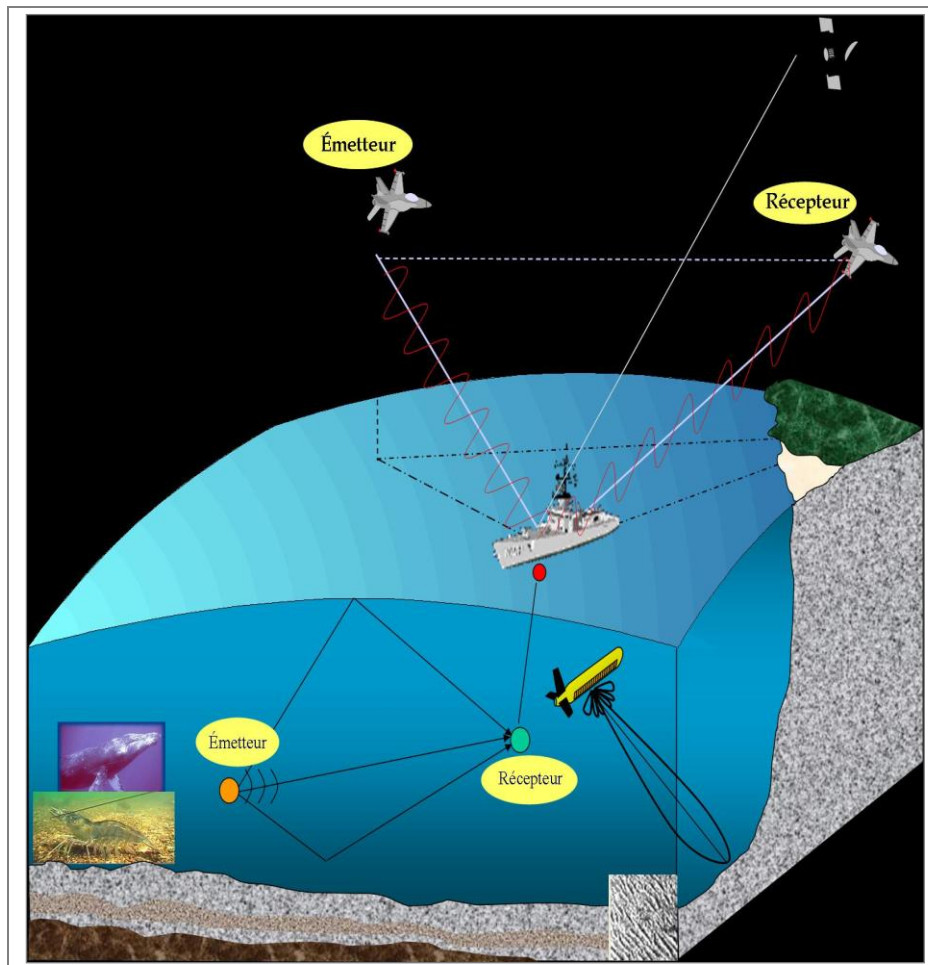


FIG. 11.3.3.2: Activités du laboratoire E3I2

Conclusion sur les fonctionnalités de l'observatoire

Les fonctionnalités de l'observatoire analysées dans ce chapitre présentent les points forts et les points d'amélioration de notre solution. Cette analyse est une étape importante sur les bases de laquelle il sera possible d'améliorer l'existant ou étudier d'autres pistes dans les projets futurs.

Chapitre 12 - Communications autour du projet

Objectifs

D'un point de vue communication, à court terme, nous devons participer à la constitution de ressources multimédias pour la réalisation d'une campagne de promotion du projet. A ce titre j'ai accepté de participer aux séances vidéo et contribuer à cette campagne en expliquant nos recherches à travers divers documents qui seront rendus public prochainement.

Afin de sensibiliser le public, différentes enquêtes ont été effectuées auprès des visiteurs des aquariums de Brest et de Plymouth sur la thématique des observatoires. L'idée était d'identifier les médias par lesquels nous pourrions par la suite communiquer de façon optimale. Différentes solutions ont été proposées du site Internet, aux campagnes radiophoniques, télévisuelles ou de street marketing. Nous avons donc identifié selon différents critères nos cibles et nos médias.

Les enquêtes menées auprès de 237 visiteurs des aquariums d'Océanopolis (135) et National Marine Aquarium of Plymouth (102) ont portées sur les thèmes de :

- R Les interrogations du public sur le thème des observatoires marins.
- R L'environnement marin.
- R Les observatoires sous-marins.
- R Le niveau de confiance que l'on porte aux sources d'informations

Le tableau 12.1 synthétise quelques résultats qui vont guider les perspectives à court terme.

L'objectif de ce chapitre est de proposer des solutions pour contribuer à la communication autour du projet à court, moyen et long terme.

FIG. 12.1: Synthèse des résultats de l'analyse d'impacts autour de la thématique des observatoires auprès du grand public

Underwater Observatories	
How informed do you feel about acoustics	Very low (88,9% not or very poorly informed)
How informed do you feel about sea currents	Very low (82% not or very poorly informed)
How informed do you feel about temperature	Very low (85,9% not or very poorly informed)
Information sources	
Where, if at all, have you seen or heard of about marine environment?	Specialized Magazines 14,8% Press 54,1% Internet 25,2% Radio 38,5% Television 63%
In your opinion, what is the best support to inform the public about marine environment?	Exhibitions 19,3% Films 39,3% Conferences 18,5% Documents (books, flyers...) 14,8% Internet web sites 8,1%

12.1A COURT TERME

Mise en place d'exposition du démonstrateur

A court terme, différentes expositions itinérantes seront mises en place ainsi que diverses activités éducatives auprès du public scolaire et bénévole. Exposer ce démonstrateur au grand public nous permettra de communiquer sur nos travaux, d'améliorer notre visibilité et potentiellement de nouer de nouveaux partenariats.

Déploiement d'un second observatoire (PML)

Le déploiement du second observatoire sous-marin est entamé. L'étude du sol est déjà réalisée, les premiers prototypes matériels ont déjà été immergés et les données commencent à être remontées à la surface.



FIG. : Déploiement du second observatoire à PML

Connexion entre MeDON Iroise et MeDON PML

Nous commençons à exploiter et mettre en valeur les données issues de leurs premiers tests d'acquisition de données. Nous avons déjà identifié différentes divergences dans la manière dont les données sont représentées. Les données acquises à l'heure actuelle sont formalisées en CSV et non en XML par exemple. Bien que dans cet exemple nous puissions facilement développer des outils pour transformer les données, nous devons continuer à communiquer, entendre les arguments de nos partenaires, et leur faire profiter de notre expérience. L'idée dans ce deuxième déploiement est aussi de profiter d'une nouvelle perspective et d'encourager de nouvelles solutions dans la mesure où nous parvenons à éviter des écarts technologiques rendant trop difficile la communication entre nos architectures. Nous devons dans la mesure du possible respecter des standards dans la formalisation des données de façon à optimiser la communication entre des architectures qui pourront être hétérogènes.

Amélioration des stratégies de communications entre les capteurs

A court terme, et dans le cadre d'un projet parallèle à MeDON , nous avons planifié de pouvoir rendre possible la communication entre les capteurs. C'est leur capacité à interagir les uns avec les autres en fonction de leur environnement qui rend le réseau *intelligent*. Le projet en question est nommé "SmartSensor". Maintenant que nous disposons d'un système d'information capable de gérer l'acquisition, le traitement, le stockage et le transport de l'information, nous souhaitons permettre aux scientifiques d'utiliser nos services pour interagir et déployer des stratégies directement sur les capteurs. C'est un point que je n'ai pas détaillé dans le cadre de ce mémoire qui se focalise sur le système d'information, mais nous avons déjà développé les services nécessaires à ces mécanismes au niveau du composant C1-Acquisition chargé d'acquérir les données.

Trouver de nouvelles sources de financements

Le financement du projet doit couvrir plusieurs postes de dépenses matériels et humains à savoir le déploiement des infrastructures matérielles, la rémunération du personnel, et la promotion du projet dans l'événementiel. Bien que nous ayons eu une visibilité globale sur la phase I, nous pouvons en interne étudier les possibilités de financement supplémentaires basées sur nos résultats actuels afin d'étudier la possibilité pour notre laboratoire de réaliser de nouvelles recherches dans la continuité de ce projet, qui pourrait alors nous profiter pleinement.

Le financement actuel est assuré à 50% par Interreg et l'autre moitié est sur fonds propres.

Continuer la communication autour du projet

La communication autour du projet est un processus continu. A court terme il est nécessaire de continuer à participer aux différents événements français et internationaux qui sont organisés autour de la thématique des observatoires, de l'environnement et plus généralement de la recherche.

Concernant la médiation scientifique, c'est-à-dire la diffusion pédagogique des connaissances scientifiques issues des analyses des données de l'observatoire, différentes activités sont prévues dont notamment :

- la dissémination des résultats du projet au niveau de l'espace Manche dans son ensemble et au niveau européen à travers un rapport final
- Une conférence sur la thématique « observatoire sous-marin ».

La phase II du projet qui va maintenant commencer va donc permettre la mise en place d'un réseau à long terme utilisant les infrastructures de Brest et de Plymouth renforçant ainsi le jumelage entre ces deux villes.

Dé-mobiliser l'équipement immergé

Dans une année, nous avons prévu de retirer nos équipements immergés en mer d'Iroise. Nous comptons restituer la zone de déploiement en l'état.

Ce démantèlement est dû au type du projet que nous mettons en place. Il existe 5 types de prototypes :

- R Le prototype virtuel (3D, logiciel, modélisation)
- R La maquette d'aspect
- R Le prototype fonctionnel ou démonstrateur
- R Le prototype d'industrialisation
- Le prototype de présérie

Les maquettes n'embarquent pas de code fonctionnel. Les prototypes n'ont pas pour vocation à être directement industrialisés. MeDON s'inscrit dans le cadre du démonstrateur et bien que la zone choisie pour le déploiement ait un réel intérêt pour les scientifiques et que cette zone soit potentiellement réutilisée pour un projet similaire par la suite, nous cherchons juste à montrer la preuve du concept.

Nous avons donc en perspective, à court terme, de montrer la faisabilité de notre approche afin de permettre le déploiement d'autres observatoires sous-marins similaires.

12.2A MOYEN TERME

Le déploiement d'autres observatoires en Europe pourra se faire à moyen terme. Dans un premier temps, PML doit déployer l'observatoire de Plymouth. L'observatoire d'Iroise et celui de Plymouth devront alors être à même d'échanger des données. Les scientifiques pourront alors valider avec succès de nouveaux algorithmes déployés.

Le déploiement de nouveaux observatoires sous-marins en Europe devrait permettre à nos scientifiques de travailler sur de grandes quantités de données temps réel et archivées de qualité.

Le déploiement d'observatoires terrestres pourrait se faire sur le même modèle. L'interconnexion des deux types de réseaux serait naturelle.

Enfin, à moyen terme, il serait envisageable d'étendre le réseau de capteurs au grand public en utilisant les capacités de leurs terminaux mobiles.

Une modélisation SysML de ce système de systèmes permettrait par exemple, d'avoir un instantané de l'ampleur d'une telle architecture.

12.3A LONG TERME

A long terme nous pouvons imaginer l'interconnexion d'observatoires hétérogènes par leurs services, leurs localisations physiques et donc par le type de données qu'ils auront à manipuler.

Nous pouvons également imaginer que les services de ces observatoires déclenchent des alertes « officielles » comme c'est déjà presque le cas aux Etats-Unis dans la détection des tremblements de terre.

Par interconnexions nous pouvons aussi entendre coupler un observatoire de type « MeDON » avec un autre observatoire, utilisant une toute autre technologie mais s'appuyant sur des standards communs.

L'idée des fondateurs de ce projet, est qu'à long terme, nous puissions disposer d'outillages fonctionnels, applicatifs et physiques pour une surveillance globale terrestre.

Conclusion

Bilan de mon travail

Au cours de mon activité, j'ai mené différentes tâches de l'identification des objectifs projets, à la rédaction des documentations pour les utilisateurs finaux. Tout au long de ce processus, j'ai eu à prendre des décisions tant sur le fond que sur la forme de manière à anticiper différentes problématiques et produire un travail de qualité, dans le cadre de ma formation et vis à vis de nos partenaires côté entreprise.

La présentation des résultats est une tâche que j'ai menée avec beaucoup d'attention. Je pense qu'un ingénieur doit aussi savoir vulgariser des sujets complexes en étant capable de prendre du recul par rapport aux tâches qui lui sont confiées.

J'ai organisé mon temps afin de respecter nos délais. J'ai proposé des solutions techniques et fonctionnelles en fonction de nos moyens et j'ai participé à la réalisation d'un démonstrateur répondant aux objectifs fixés par les partenaires du projet.

Dans le cadre de ce mémoire, je me suis attaché à bien identifier mes contributions personnelles, mes objectifs et ceux du projets ainsi que de présenter mon travail en suivant une progression logique qui permette à un lecteur externe de comprendre ce travail en lui donnant les clefs pour aller plus loin.

Il me semble que le choix de l'ingénierie dirigée par les modèles pour ce projet est un choix qui s'est avéré être judicieux pour plusieurs raisons. Nous avons tenu nos engagements et contribué à fournir une méthodologie réutilisable dans d'autres contextes.

Lors de mon travail j'ai pu présenter à mon équipe différentes pistes technologiques à creuser comme les interfaces OSLC où encore les services d'alertes par exemple.

D'un point de vue « enseignement » j'ai accompagné plusieurs étudiants qui, je l'espère ont apprécié de travailler sur la thématique des observatoires.

Bilan du projet

Concernant le bilan du projet, au delà des développements techniques, mon travail a permis d'avancer sur plusieurs axes. Deux autres devront encore susciter quelques recherches, notamment autour des aspects liés à la simulation et à la re-configuration logicielle des cibles embarquées.

Les simulations que j'ai menées ont permis d'identifier les exigences en termes de métriques réseaux. Ces simulations ne pourront toutefois pas être réutilisées en l'état car elles sont liées aux choix d'architectures que nous avons réalisés. C'est un point d'amélioration que l'on peut noter pour les prochaines versions. Il aurait par exemple été intéressant de pouvoir générer un support pour l'architecture qui puisse être interprété directement par un simulateur. Cela ne pouvait pas être fait dans cette phase du projet par manque de temps, et ne faisait pas partis des objectifs prioritaires. Nous avons utilisé un simulateur propriétaire (Cisco). Le format des fichiers étant fermé, nous ne pouvions pas facilement y intégrer un diagramme de déploiement. L'objectif de cette

expérimentation était simplement d'obtenir des métriques afin de pouvoir étayer notre argumentation autour du déploiement des composants logiques sur l'infrastructure physique.

J'ai quitté le projet suite au déploiement réussi de notre prototype. Je n'ai cependant pas pu assister à l'intégration de notre carte SmartSensor en mère d'Iroise. Bien qu'ayant participé (de loin) à sa conception à travers différents prototypage rapide et des réunions avec les réalisateurs, sa reconfiguration dynamique était encore en phase d'implémentation. Sur des aspects techniques, une reconfiguration à chaud du code bas niveau d'une cible embarquée et de surcroît immergée, est une tâche risquée et implique donc davantage de contrôle sur le cycle de développement avant une mise en production, d'où les délais plus longs.

J'ai accompli ma tâche d'urbanisation du système d'information. Celui-ci est maintenant documenté, harmonieux et nous disposons de modèles et d'outillages qui permettent un redéploiement aisé de notre solution selon d'autres objectifs métier, de nouvelles logiques fonctionnelles, applicatives et/ou techniques.

Enfin, nous disposons d'IHM Web permettant nos premières communications auprès du grand public des données de l'observatoire. Ce dernier point nous permet d'afficher nos premiers résultats et donc de rendre visible auprès de nos partenaires une partie du travail accompli.

Bilan personnel

Je pense avoir joué un rôle intéressant dans la conception et la réalisation de ce système dans la mesure où j'ai trouvé ma place au sein de l'équipe en facilitant la communication entre les experts des différents domaines, qui évoluent parfois dans des *mondes* différents. J'ai ainsi apprécié les défis méthodologiques et techniques qui m'étaient présentés, même si parfois certains assemblages restent perturbants. En effet, la production d'une solution au sein d'une équipe de projet doit répondre à des impératifs contractuels et toutes les pistes ne peuvent ou ne doivent pas être prises. Je pense avoir respecté les différentes contraintes qui m'étaient proposées afin d'apporter une solution déployable dans une grande variété d'environnements. Notre système d'information est fonctionnel, et l'approche suivie y a certainement contribué pour le respect des délais et de la qualité de la production actuelle, et de ce que l'on peut attendre si d'autres devaient reprendre nos travaux.

Le travail d'un ingénieur ne se résume pas à la partie technique de la conception et du codage, j'ai ainsi pu prendre connaissance d'un processus de recherche dans son ensemble et me familiariser avec les exigences des entreprises travaillant sur des technologies de pointe. J'ai appris à m'intégrer dans des équipes pluridisciplinaires, à chercher à intégrer des problématiques transverses à ces domaines d'activité et proposer des solutions argumentées pour qu'ensemble nous puissions atteindre nos objectifs.

De plus, cette expérience autour du projet MeDON, un projet porteur offrant des solutions liées à des problématiques très graves de dérèglement climatique terrestre, autour de technologies innovantes, s'avère être un bon démonstrateur des nouvelles compétences que j'ai acquises depuis ces dernières années.

Annexes

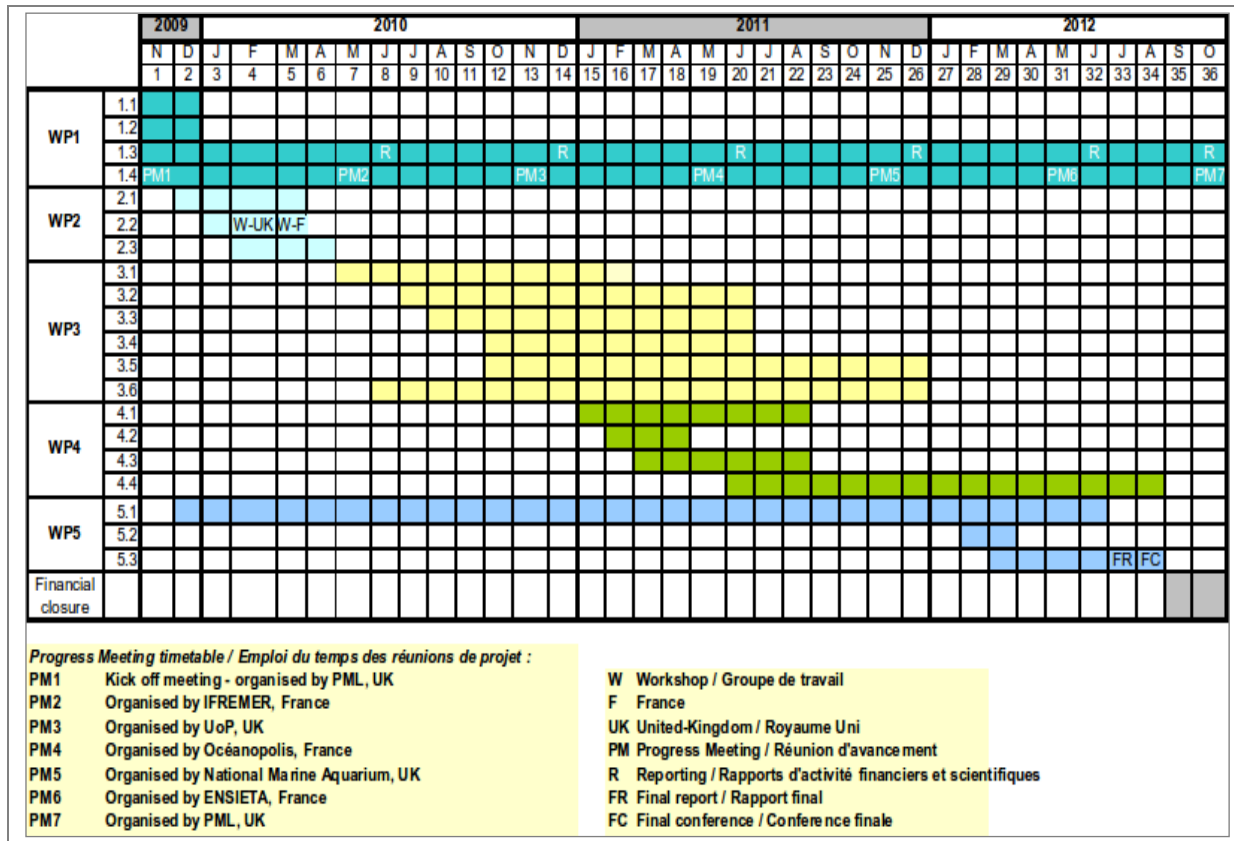


FIG. 3: Emploi du temps des réunions de projet

FIG. 4: Liste des actions Work Package 1. Coordination et gestion de projet

Action	Description	Livrable
Action 1.1 PML	<i>Consolidation du partenariat et mise en place du comité de pilotage.</i> L'action 1.1 s'assure que la structure de partenariat et que le management du projet sont en accord avec la convention de partenariat. La gestion du projet s'appuiera sur un Comité de Pilotage qui sera créé au cours de cette action. Ce Comité de Pilotage veillera au respect du budget et du planning ainsi qu'à la réalisation des objectifs. Il sera constitué d'un représentant de chaque partenaire, et sera présidé par le PML. Il interviendra pour valider les évolutions et les modifications des objectifs du projet, après consultation et accord de l'Autorité de Gestion. L'action 1.1 établira également un Groupe Consultatif d'Experts dont les compétences sont reconnues pour le sujet concerné, pour traiter de certains aspects techniques du projet.	1 Rapport décrira précisément les objectifs du projet et la structure de partenariat mise en œuvre. 1 comité de pilotage (composition, mode de fonctionnement, dates de réunions). 1 groupe consultatif d'experts (composition, mode de fonctionnement, dates de réunions).
Action 1.2 PML	<i>Règles et priorités pour les échanges d'expertise et de savoir avec les partenaires, les instituts de recherche et les entreprises privées..</i> Des aspects importants de ce projet sont le renforcement des capacités (par exemple expertise acquise sur le pilote) et le potentiel de commercialisation par les sociétés privées qui seront associées au projet. Action 1.2 garantira que le transfert de connaissances entre les partenaires, centres de recherche et le secteur privé se déroulent au sein d'un cadre approuvé par tous les partenaires.	1 accord R protocoles de transfert de connaissance
Action 1.3 PML	<i>Gestion administrative et financière.</i> Action 1.3 garantira une bonne gestion du projet et des financements alloués, en concordance avec la proposition du projet et la demande de subvention : Collecte de toutes les informations concernant le projet et rédaction des rapports d'avancement Collecte des rapports intermédiaires et finaux d'information financière des partenaires et dépôt pour chaque demande de versement au STC. Redistribution des subventions FEDER aux partenaires. Vérification des dépenses de chaque partenaire par un auditeur externe, et organisation d'un audit des comptes du projet par un auditeur externe	6 audits et 6 rapports d'avancement
Action 1.4 PML	<i>Communication interne et externe.</i> Action 1.4 s'assurera des échanges réguliers entre les partenaires afin que les activités du projet soient réalisées en concordance avec la proposition de projet et la demande de subvention. Ceci sera effectué par le biais d'appels téléphoniques, visioconférences et réunions d'avancement régulières. Un extranet sera également	1 Site Web avec extranet 7 réunions d'avancement (1 tous les 6 mois, la première étant le „kick-off meeting) 7 newsletters Matériel

	mise en place sur le site Web du projet. Action 1.4 s'attellera également à la communication externe avec le STC (e-mails réguliers, réunions le cas échéant) et communication auprès du public concerné (site Web, dépliants, autre matériel promotionnel) L'information du public et la dissémination des résultats seront effectuées au sein du WP5.	promotionnel (dépliants)
--	---	--------------------------

FIG. 5: Liste des actions Work Package 2. Parties-prenantes et utilisateurs finaux

Action	Description	Livrable
Action 2.1 PML	<p><i>Création d'un groupe de parties-prenantes et d'utilisateurs finaux.</i> L'identification des utilisateurs finaux est une priorité. Cette action réalisera une analyse des parties-prenantes et utilisateurs finaux ayant un intérêt dans :</p> <ul style="list-style-type: none"> R observatoires côtiers R gestion côtière R veille marine et environnementale R données météo R sensibilisation du public R impacts des activités maritimes sur les écosystèmes marins <p>A cette fin, nous contacterons d'abord les parties-prenantes et utilisateurs déjà identifiés en phase de préparation. Nous leur demanderons ensuite s'ils pensent que nous avons omis d'importants acteurs, afin d'étendre notre base de contacts. Nous contacterons aussi d'autres projets Interreg (achevés ou en cours) pour être sûrs que notre base est exhaustive.</p>	1 groupe d'acteurs et utilisateurs finaux. Coordonnées et courte description
Action 2.2 PML	<p><i>Identification des besoins des utilisateurs finaux et parties prenantes/gestionnaires côtiers.</i> Nous enverrons tout d'abord des questionnaires aux utilisateurs et acteurs sélectionnés par l'action 2.1. Il s'agira probablement d'un questionnaire en ligne. Nous organiserons ensuite des groupes de travail interactifs de part et d'autre de la Manche pour mieux comprendre les besoins des utilisateurs précédemment identifiés. Pour ceux qui ne peuvent assister aux réunions, nous assurerons le suivi par e-mails, téléphone et entretiens directs.</p>	Groupes de travail utilisateurs 2 ateliers d'une demi journée (1 à Plymouth, 1 à Brest) pour identifier les besoins et attentes des participants
Action 2.3 PML	<p><i>Développement d'une stratégie de veille pour les sites pilotes</i> Cette action fera suite aux résultats</p>	Spécifications : R que mesurer

de 2.2. Elle ciblera en particulier les acteurs impliqués dans la gestion du Parc Naturel Marin d'Iroise (Agence des Aires Marines Protégées) et de la zone de conservation spéciale de Plymouth (Tamar Estuary Consultative Forum), car ces zones ont été sélectionnées pour l'installation des deux observatoires permanents de la phase II. Les besoins des parties prenantes et utilisateurs finaux en termes de suivi marin seront pris en compte au sein d'une stratégie de veille : choix de paramètres quantifiables à mesurer et suivre, fréquence des mesures pour optimiser les coûts, sites qui sont les plus adaptés pour accueillir les observatoires à Plymouth et Brest.	<p>À comment</p> <p>À où</p> <p>À quand et avec quelle fréquence</p>
--	--

FIG. 6: Liste des actions Work Package 3. Développements scientifiques et techniques

Action	Description	Livrable
Action 3.1 IFREMER	Spécifications fonctionnelles du réseau Cette action s'appuiera sur la stratégie d'observation et de mesure établie dans l'action 2.3. Les spécifications techniques des instruments et des capteurs mis en œuvre seront élaborées. Les utilisateurs finaux seront consultés pour valider les choix. On reprendra si possible des instruments ayant atteints un stade de développement assurant un fonctionnement opérationnel en milieu marin. Trois grandes catégories d'applications sont envisagées : l'observation et l'étude des grands mammifères marins (ce point sera plus détaillé dans l'action 3.6) la mise au point technologique de la surveillance vidéo sous-marine, le monitoring des paramètres hydrologiques (notamment courants, vagues, hauteur d'eau, température, biogéochimie).	Visite transfrontalière Echange de scientifiques de 2 à 3 jours Cahier des charges techniques liste des instruments et leur mode d'utilisation. Cette documentation exposera le mode de choix des matériels.
Action 3.2 IFREMER	<i>Développement technologique</i> Partant du cahier des charges, cette action sélectionnera la technologie constituant la colonne vertébrale du réseau. Il s'agit d'assurer la communication bi-directionnelle avec tous les composants. On veillera en particulier à l'aspect modulaire et évolutif des solutions retenues. Il faut réaliser dans un environnement très difficile les liaisons avec : boîte de jonction primaire et des boîtes de jonction secondaires sur lesquelles seront connectés tous les équipements, la terre (résoudre	Cahier des charges : Technologie, Echelle, Cout, Portabilité

Action 3.3 ENSTA-Bretagne	<p>l'atterrissage du câble), l'alimentation en énergie. La participation de l'IFREMER au projet Vénus (Canada) et l'expérience acquise seront pleinement valorisées.</p> <p><i>Gestion des données et archivage</i></p> <p>La gestion des données est un élément fondamental pour le succès du projet. Les besoins des utilisateurs (WP2) seront analysés et synthétisés ; la technologie s'adaptera aux exigences exprimées. Pour le traitement, l'archivage (sécurité et pérennité) l'IFREMER utilisera la base de données OCO (Océanographie Côtière Opérationnelle) et les outils du programme Prévimer pour assurer ce service. Cette chaîne d'outils permet la diffusion après un contrôle qualité automatique. Pour les données vidéos et audios, l'IFREMER assurera l'archivage en garantissant la sécurité. Le système s'ajustera au besoin d'un public élargi en permettant la diffusion en temps quasi réel des données et des observations. Les standards européens seront pris en compte.</p>	<p>Rapport: structure du centre de données et une notice d'utilisation. outils Web (logiciel) : l'accès et la mise à disposition de séries temporelles.</p>
---------------------------	--	---

FIG. 6: Liste des actions Work Package 3. Développements scientifiques et techniques

Listing 1 : Sensor.ecore

```

<?xml version="1.0" encoding="UTF-8"?>
<ecore:EPackage xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" name="sensor"
  nsURI="http://sensor/1.0" nsPrefix="sensor">
  <eClassifiers xsi:type="ecore:EClass" name="Sensor" eSuperTypes="#//SensorInterface">
    <eStructuralFeatures xsi:type="ecore:EReference" name="hasBehavior" eType="#//BehaviorSensor"
      containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="hasProprieteStatique" eType="#//ProprieteStatique"
      containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="Name" eType="ecore:EDataType
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="HasType" eType="#//SensorType"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="SourceChannel" upperBound="-1"
      eType="#//SensorChannel" eOpposite="#//SensorChannel/Target"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="TargetChannel" upperBound="-1"
      eType="#//SensorChannel" eOpposite="#//SensorChannel/Source"/>
  </eClassifiers>
  <eClassifiers xsi:type="ecore:EClass" name="ProprieteStatique">
    <eStructuralFeatures xsi:type="ecore:EReference" name="HasIdentification" eType="#//Identification"
      containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="HasLocation" eType="#//Location"
      containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EReference" name="HasCapacity" eType="#//Capacity"
      containment="true"/>
    <eStructuralFeatures xsi:type="ecore:EAttribute" name="StaticPropName" eType="ecore:EDataType
      http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  </eClassifiers>

```

```

<eClassifiers xsi:type="ecore:EClass" name="Interface" eSuperTypes="#//EventConcept">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="InterfaceName" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="HasServices" upperBound="-1"
    eType="#//Service" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="HasBehavior" eType="#//BehaviorSensor"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="BehaviorSensor">
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasStates" upperBound="-1"
    eType="#//State" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasTransitions" upperBound="-1"
    eType="#//Transition" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasStart" eType="#//Start"
    containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="BehaviorName" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Identification">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="shortname" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="longName" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="id" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EInt"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Location">
  <eStructuralFeatures xsi:type="ecore:EReference" name="Longitude" eType="#//Longitude"
    containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="Latitude" eType="#//Latitude"
    containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="Altitude" eType="#//Altitude"
    containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="Position" eType="#//Position"
    containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Title" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Event">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="description" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="nature" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="type" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Service" eSuperTypes="#//EventConcept">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="name" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasBehavior" eType="#//BehaviorService"
    containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasOperations" upperBound="-1"
    eType="#//Operation" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasAttributes" upperBound="-1"
    eType="#//Attribute" containment="true"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="State">
  <eStructuralFeatures xsi:type="ecore:EReference" name="OutComingTransitions" upperBound="-1"

```



```

    eType="#//Transition" eOpposite="#//Transition/source"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="InComingTransitions" upperBound="-1"
    eType="#//Transition" eOpposite="#//Transition/target"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="name"      eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Attribute">
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="name"      eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="Type"      eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="Description" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Transition">
  <eStructuralFeatures xsi:type="ecore:EReference" name="source" eType="#//State"
    eOpposite="#//State/OutComingTransitions"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="target" eType="#//State"
    eOpposite="#//State/InComingTransitions"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="name"      eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="ServiceTrans" eType="#//Service"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="eventTrans" eType="#//Event"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="guard"      eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="BehaviorService">
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasStateServices" upperBound="-1"
    eType="#//StateService" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasTransitionServices"
    upperBound="-1" eType="#//TransitionService" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasStartService" eType="#//StartService"
    containment="true"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="ServiceBehavior" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="StateService">
  <eStructuralFeatures xsi:type="ecore:EReference" name="OutComingTransitionServices"
    upperBound="-1" eType="#//TransitionService" eOpposite="#//TransitionService/sourceService"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="InComingTransitionServices"
    upperBound="-1" eType="#//TransitionService" eOpposite="#//TransitionService/targetService"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="name"      eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="TransitionService">
  <eStructuralFeatures xsi:type="ecore:EReference" name="sourceService" eType="#//StateService"
    eOpposite="#//StateService/OutComingTransitionServices"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="targetService" eType="#//StateService"
    eOpposite="#//StateService/InComingTransitionServices"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="name"      eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="Operation" eType="#//Operation"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="event" eType="#//Event"/>
  <eStructuralFeatures      xsi:type="ecore:EAttribute"      name="guardService" eType="ecore:EDataType
http://www.eclipse.org/emf/2002/Ecore##/EString"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="EventSubmitted" upperBound="-1"
    eType="#//Event"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Start" eSuperTypes="#//State"/>

```



```

<eClassifiers xsi:type="ecore:EClass" name="StartService" eSuperTypes="#//StateService"/>
<eClassifiers xsi:type="ecore:EClass" name="Operation">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="name" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="returnType" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="nbrParameters" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EInt"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="hasParameters" upperBound="-1"
eType="#//Parameter" containment="true" eOpposite="#//Parameter/operation"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Description" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Parameter">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Name" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Type" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Description" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Mode" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Position" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EInt"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="operation" eType="#//Operation"
eOpposite="#//Operation/hasParameters"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="EventConcept">
  <eStructuralFeatures xsi:type="ecore:EReference" name="HasEvents" upperBound="-1"
eType="#//Event" containment="true"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="SensorSystem">
  <eStructuralFeatures xsi:type="ecore:EReference" name="HasSensors" upperBound="-1"
eType="#//Sensor" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="HasSensorTypes" upperBound="-1"
eType="#//SensorType" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="SensorChannel" upperBound="-1"
eType="#//SensorChannel" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="ConfigurationSystem" upperBound="-1"
eType="#//Configuration" containment="true"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="SensorType" eSuperTypes="#//SensorInterface">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="TypeName" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="SensorChannel">
  <eStructuralFeatures xsi:type="ecore:EReference" name="Source" lowerBound="1"
eType="#//Sensor" eOpposite="#//Sensor/TargetChannel"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="Target" lowerBound="1"
eType="#//Sensor" eOpposite="#//Sensor/SourceChannel"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Name" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EReference" name="Event" eType="#//Event"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Altitude">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="UnitOfMeasure" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="ValueOfAttitude" eType="ecore:EDDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>

```

```

</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Latitude">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="UnitOfMeasure" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="ValueOfLatitude" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Longitude">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="UnitOfMeasure" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="ValueOfLongitude" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Position">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="UnitOfMeausre" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="ValueOfPosition" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Capacity">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="BandeFrequence" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="LatencyTime" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="BatteryPower" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Sensibility" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="ProfondeurMax" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EInt"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="SensorInterface">
  <eStructuralFeatures xsi:type="ecore:EReference" name="HasSensorInterface" eType="#//Interface"
containment="true"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="SensorConfiguration">
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="NbrBits" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="FrequenceEchan" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="Resolution" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="GainAmplification" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="SensorName" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
<eClassifiers xsi:type="ecore:EClass" name="Configuration">
  <eStructuralFeatures xsi:type="ecore:EReference" name="SensorConfiguration" upperBound="-1"
eType="#//SensorConfiguration" containment="true"/>
  <eStructuralFeatures xsi:type="ecore:EAttribute" name="ConfigName" eType="ecore:EDataType"
http://www.eclipse.org/emf/2002/Ecore#//EString"/>
</eClassifiers>
</ecore:EPackage>

</xml>

```

Références

- [ABCR99] N. Prakash A. Benjamin C. Rolland. A multi-model of process modelling. Requirement Engineering. Springer-Verlag London Ltd, 1999.
- [BOU09] Marion BOULAY. Le test logiciel. GNU Linux magazine France, avril 2009.
- [Bui11] Information Builder. Architecture orientée services (soa). <http://www.informationbuilders.fr/soa-architecture-orientee-service.html>, 1996-2011.
- [CEI08] CEISAR. Business Process Modeling. CEISAR, Avril 2008.
- [CR90] A. Flory C. Rolland. La conception des systèmes d'information : Etat de l'art et nouvelles perspectives. Nouvelles perspectives des systèmes d'information, 1990.
- [Fie09] Professeur Marius Fieschi. La gouvernance de l'interopérabilité sémantique est au cœur du développement des systèmes d'information en santé. Technical report, Ministère de la Santé et des Sports, 9 Juin 2009.
- [GH09] Jeremy Denham George Heineman. Entity, boundary, control as modularity force multiplier. 3rd Workshop on Assessment of Contemporary Modularization Techniques, Oct 2009.
- [JH08] Marcia Kaufman Dr. Fern Halper Judith Hurwitz, Robin Bloor. Service Oriented Architecture for DUMMIES. Wiley Publishing, Inc, 2008.
- [JP07] Jochen Rehwinkel Jan Plosky, Wilhelm Hasselbring. Ieee software. IEEE Computer Society, 2007.
- [JS10] Jean-Marc Jézéquel Jacques Simonin, Philippe Picouet. Conception fonctionnelle de services d'entreprise fondée sur l'alignement entre cœur de métier et système d'information. DOI :10.3166/ISI.15.4.37-61 © 2010 Lavoisier, Paris, Decembre 2010.
- [KG08] Bruce Williams Kiran Garimella, Michael Lees. Les Bases du BPM POUR LES NULS. Wiley Publishing, Inc, 2008.
- [KOU10] Ali KOUDRI. Processus de Conception Conjointe Logiciel Matériel Dirigées par les Modèles. PhD thesis, L'Université des Sciences et Technologies de Lille, 2010.
- [LAL09] Mounir LALLALI. Modélisation et Test Fonctionnel de l'Orchestration de Services Web. PhD thesis, l'INSTITUT NATIONAL DES TELECOMMUNICATIONS, 2009.
- [LtAt04] Marc Lankhorst and the ArchiMate team. Archimate language primer. Technical report, Telematica Instituut, 26-08-2004.
- [LtAt08] Marc Lankhorst and the ArchiMate team. Dynamic bpm R the value of embedding process into dynamic work. Technical report, HANDY Soft, 01-04-2008.
- [OMG08] OMG. Software and systems process engineering meta-model, v2.0. Technical report, OMG, Avril 2008.
- [Ope10] OpenGroup. Service-oriented architecture ontology. Technical Standard, 2010.
- [RF99] J. Mogul H. Frystyk L. Masinter P. Leach T. Berners-Lee R. Fielding, J. Gettys. "6.1.1 status code and reason phrase". <http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html>, June 1999.
- [Tro09] Jean-Pierre Troll. Le test logiciel. GNU Linux Magazine France n115, Avril 2009.
- [val07] valtech. Livre Orange, Urbanisation Intégration de Systèmes. Valtech, Septembre 2007.
- [VAU10] Jean-François VAUTIER. Eléments de systémique illustrés par les arts. Res-Systemica, Decembre 2010.
- [DS01] George Coulouris, Jean Dollimore et Tim Kindberg, Distributed Systems : Concepts and Design Edition 3

Table des figures

Liste des tableaux

ADSL Asymmetric Digital Subscriber Line
API Application programming interface
CPU Central Processing Unit
ENSTA Ecole National Supérieur en Technique Avancée
IDM Ingénierie Dirigée par les Modèles
IHM Interface Homme Machine
IP Internet Protocol
IRC Internet Relay Chat
JPG Joint Photographic Experts Group
MeDON Marine edata Observatory Network
MVC Modèle Vu Contrôleur
OSLC Open Service for Lyfecicle Collaboration
PDCA Plan - Do - Check - Act
PDM Plate-forme Dependant Modèle
PHP Hypertext Preprocessor
PIM Plate-forme Independant Modèle
PNG Portable Network Graphics
PSM Plate-forme Spécifique Modèle
SdS Système de Système
SGBD Système de gestion de base de données
S.I. Système d'information
SMS (Langage) Smart Modeling for Sensor
SOA Service Oriented Architecture
STIC Sciences et Technologies de l'Information et de la Communication
XML Extensible Markup Language

Chapitre A

Webographie

- Ř <http://www.urba-ea.org/urbanisme.php>
- Ř <http://blogs.msdn.com/b/mikewalker/archive/2008/08/19/zachman-updates.aspx>
- Ř <http://www.ceisar.fr/a-propos-du-ceisar.html>
- Ř http://fr.wikipedia.org/wiki/Urbanisation_%28informatique%29#Le_syst.C3.A8me_d.27information
- Ř http://fr.wikipedia.org/wiki/Model_driven_architecture
- Ř http://www.ENSTA-bretagne.fr/stic/index.php?option=com_content&task=view&id=73&Itemid=185
- Ř <http://fr.wikipedia.org/wiki/Syst%C3%A9mique>
- Ř http://en.wikipedia.org/wiki/Service-oriented_architecture
- Ř <http://www.informationbuilders.fr/soa-architecture-orientee-service.html>
- Ř http://www-igm.univ-mlv.fr/~dr/XPOSE2003/JUnit_tour/
- Ř <http://www.soapui.org>