



N° d'ordre : 2010-ISAL-0044

Thèse en cotutelle

Modélisation conceptuelle pour les applications géomatiques temps réel

Présentée devant
l'Institut National des Sciences Appliquées de Lyon
et l'Ecole Supérieure de Communications de Tunis

Pour obtenir
Le grade de docteur

Spécialité : Informatique
École doctorale française : Infomaths
École doctorale tunisienne : TIC

Par
Malek Ben Youssef épouse Ben Jebara

Soutenue le 07 Juillet 2010 devant la Commission d'examen

Jury MM.

Professeur Ziad BELHADJ, SUP'COM (Tunisie), directeur de thèse,
Professeur Mohamed Rached BOUSSEMA, ENIT (Tunisie), examinateur,
Docteur Sami FAIZ, Université de Jendouba (Tunisie), rapporteur,
Professeur Robert LAURINI, INSA de Lyon (France), Directeur de thèse,
Professeur Ahmed LBATH, Université Joseph Fourier de Grenoble (France), co-directeur,
Professeur Thérèse LIBOUREL, Université de Montpellier II (France), rapporteur,
Docteur Mohamed Saber NACEUR, INSAT de Tunis (Tunisie), examinateur,
Professeur Christine PARENT, Université de Lausanne (Suisse), examinatrice.

Laboratoires de recherche :

INSA-Lyon : Laboratoire d'Informatique en Image et Systèmes d'Information
SUP'Com Tunis : Unité de Recherche en Imagerie Satellitaire et ses Applications.

Modélisation conceptuelle pour les applications géomatiques temps réel

Résumé

Le domaine des applications géomatiques est en plein essor et le nombre de logiciels spécialisés dans la manipulation des données spatio-temporelles a connu une croissance considérable cette dernière décennie. Cette évolution a été accompagnée, dans la littérature, par la proposition de plusieurs méthodes de conception dédiées aux applications géomatiques. Les applications temps réel connaissent également une grande évolution et font référence à la localisation spatiale.

L'avènement des télécommunications a suscité l'intérêt de prendre en compte la distribution et la répartition de l'information pour les applications géomatiques. La discipline de télégéomatique est née d'une fusion de la géomatique et des télécommunications en se caractérisant par l'usage des systèmes modernes de positionnement de type GPS (Global Positioning System).

Actuellement, l'émergence des systèmes d'information pervasifs ouvre plusieurs axes de recherche. Ainsi les applications géomatiques temps réel soulèvent plusieurs problématiques : l'architecture logicielle de telles applications doit supporter des modèles mathématiques et des algorithmes qui ont des capacités de calculs importantes, la prise en compte de la granularité du temps réel et la conception et la modélisation de telles applications doit permettre de formaliser les spécifications et d'apporter des outils logiciels cohérents par rapport aux besoins des utilisateurs et des contraintes liées à la réactivité et éventuellement la mobilité des composants du système.

Les différents formalismes qui existent dans la littérature et qui sont supportés par des méthodes de conception des applications géomatiques et des applications temps réel sont insuffisants et ne répondent pas au besoin du concepteur.

Un travail de recherche dans la prise en compte au niveau conceptuel des contraintes liées à la dimension temps réelle de la nouvelle génération des applications géomatiques temps réel est certainement prometteuse.

Les recherches que nous avons menées nous ont permis de proposer des concepts de base s'appuyant sur des extensions d'UML (Unified Modeling Language) afin de prendre en compte au niveau conceptuel les problématiques de communications et de localisation des données et des applications.

Nous proposons un profil UML pour la conception des applications géomatiques temps réel. C'est le profil GRT (Geographic Real Time) dédié à la conception des applications géomatiques temps réel. Ce travail débouche sur un prototype doté d'un atelier de génie logiciel supportant le formalisme proposé. Une validation à travers des exemples concrets nous permettra d'enrichir ce prototype.

Mots-Clés : Système d'Information Géographique – modélisation conceptuelle – base de données spatiales – base de données temporelles – profil UML – temps réel.

Conceptual modeling for real time geoprocessing applications

Abstract

Nowadays, the field of the geoprocessing applications is more and more important and the number of software-tools specialized in the handling of the space-time data witnessed a considerable growth during this last decade. Several methods dedicated to the design of geoprocessing applications have been proposed in the literature.

Real time applications are also experiencing great changes and referring to the spatial location. The advent of telecommunications has drawn the attention of taking into account the distribution and allocation information for geoprocessing applications. Telegeoprocessing discipline was born from the alliance of telecommunications and geomatics, characterized by the use of modern positioning systems like GPS (Global Positioning System).

Currently, the emergence of pervasive information systems opens several research areas. Thus real time geoprocessing applications raise several problems. In fact, their architecture must support mathematical models and algorithms with important capacities of calculations; besides there is the difficulty to take in account of the real time granularity. They also present major problems at the design stage because of the absence of a specific standardized formalism.

The different formalisms that exist in the literature which are supported by design methods of geomatic applications and real time applications are inefficient to answer the problems of design of the real time geoprocessing applications.

The research we conducted allowed us to offer basic concepts based on UML (Unified Modeling Language) extensions to take into account the conceptual issues of communication, location data and applications.

Thus we propose a UML profile that addresses real time constraints in the field of real time geoprocessing applications. The GRT (Geographic Real Time) profile targets the design of real-time geoprocessing applications.

This work has led to a prototype with a CASE (Computer Aided Software Engineering) tool supporting the proposed formalism. A validation through concrete examples enabled us to enrich this prototype.

Keywords: Geographic Information System – conceptual modeling – spatial databases – temporal databases – UML profile – Real Time.

تمائل تصورات خاص بالتطبيقات الجيوماتيكا المتعلقة بالوقت الحقيقي

ملخص

يعيش ميدان التطبيقات الجيوماتيكا ازدهارا كبيرا ويعرف عدد البرمجيات المتخصصة في التعامل مع المعطيات الفضائية المحدودة في الوقت نموا معتبرا خلال العشرية الأخيرة. وهذا التطور رافعة في الميدان الأدبي اقترح للعديد من مناهج التصوير المخصصة للتطبيقات الجيوماتيكا.

وتعرف التطبيقات الخاصة بالوقت الحقيقي تطورا كبيرا أيضا وتعود بالمرجعية إلي التحديد الفضائي.

وقد دفع ظهوره الاتصالات إلي الاهتمام بضرورة أن يؤخذ بعين الاعتبار توزيع المعلومات بالنسبة إلي التطبيقات الجيوماتيكا. وقد نشأت مادة الجيوماتيكا عن بعد من دمج الجيوماتيكا والاتصالات مع الاختصاص باستعمال أنظمة عصرية للمركز من صنف GPS (نظام التمركز الشامل). ويفتح ظهور أنظمة معلومات انتشارية مجالات عديدة أمام البحث.

و بذلك تطرح التطبيقا للجيوماتيكا من صنف الوقت الحقيقي إشكاليات عديدة :

- الهندسة البرمجية لهذه التطبيقات يجب أن تتحمل أمثلة رياضية وخوارزمية ذات قدرات حساب هامة.

- الأخذ في الاعتبار مسألة تحديد الوقت الحقيقي.

- ان تصميم ونمذجة مثل هذه التطبيقات يجب أن يسمحا باقرار هذه التخصصات وأعطاء أدوات برمجية واضحة بالمقارنة مع احتياجات المستعملين والشروط المتعلقة برود فعل مكونات النظام وإمكانية تحوله.

إن الإقرارات الموجودة في الأدب والتي تتحملها مناهج تصور التطبيقات الجيوماتيكا وتطبيقات الوقت الفعلي غير كافية ولا تستجيب لحاجيات المستعمل.

ويبقى البحث في مستوي تصوير الضغوطات المرتبطة بالوقت الحقيقي والجيل الجديد من التطبيقا للجيوماتيكا واعداد بصفة متأكدة.

وقد مكنتنا البحوث التي قمنا بها من اقتراح مفاهيم قاعدية تستند على التوسع في المثال اللغوي الموحد وذلك من اجل أن تؤخذ بعين الاعتبار على مستوى المفاهيم إشكاليات التواصل وتحديد المعطيات و التطبيقات.

ونقترح مثلا جامعا من اجل تصور تطبيقات جيوماتيكا خاصة بالوقت الحقيقي. ان مثال GRT (الوقت الفعلي الجغرافي) المخصص لتصوير التطبيقات الجيوماتيكا الخاصة بالوقت الحقيقي.

وهذا العمل يحتوي على ورشة برمجيات تحتل الإقرار المقترح والموافقة عبر امثلة فعلية تمكننا من إثراء هذا المثال المشترك.

الكلمات المفاتيح : نظام معلومات جغرافي - تماثل للتصورات -قاعدة معطيات فضائية -قاعدة معطيات خاصة بالوقت -

مثال UML (مثال حقيقي لغوي موحد) - وقت حقيقي.

INSA Direction de la Recherche - Ecoles Doctorales – Quadriennal 2007-2010

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	<u>CHIMIE DE LYON</u> http://sakura.cpe.fr/ED206 M. Jean Marc LANCELIN Insa : R. GOURDON	M. Jean Marc LANCELIN Université Claude Bernard Lyon 1 Bât CPE 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72.43 13 95 Fax : lancelin@hikari.cpe.fr
E.E.A.	<u>ELECTRONIQUE,</u> <u>ELECTROTECHNIQUE,</u> <u>AUTOMATIQUE</u> http://www.insa-lyon.fr/eea M. Alain NICOLAS Insa : C. PLOSSU ede2a@insa-lyon.fr Secrétariat : M. LABOUNE AM. 64.43 – Fax : 64.54	M. Alain NICOLAS Ecole Centrale de Lyon Bâtiment H9 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60 97 Fax : 04 78 43 37 17 eea@ec-lyon.fr Secrétariat : M.C. HAVGOUDOUKIAN
E2M2	<u>EVOLUTION, ECOSYSTEME,</u> <u>MICROBIOLOGIE, MODELISATION</u> http://biomserv.univ-lyon1.fr/E2M2 M. Jean-Pierre FLANDROIS Insa : H. CHARLES	M. Jean-Pierre FLANDROIS CNRS UMR 5558 Université Claude Bernard Lyon 1 Bât G. Mendel 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cédex Tél : 04.26 23 59 50 Fax 04 26 23 59 49 06 07 53 89 13 e2m2@biomserv.univ-lyon1.fr
EDISS	<u>INTERDISCIPLINAIRE SCIENCES-</u> <u>SANTE</u> Sec : Safia Boudjema M. Didier REVEL Insa : M. LAGARDE	M. Didier REVEL Hôpital Cardiologique de Lyon Bâtiment Central 28 Avenue Doyen Lépine 69500 BRON Tél : 04.72.68 49 09 Fax :04 72 35 49 16 Didier.revel@creatis.uni-lyon1.fr
INFOMATHS	<u>INFORMATIQUE ET</u> <u>MATHEMATIQUES</u> http://infomaths.univ-lyon1.fr M. Alain MILLE	M. Alain MILLE Université Claude Bernard Lyon 1 LIRIS - INFOMATHS Bâtiment Nautibus 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72. 44 82 94 Fax 04 72 43 13 10 infomaths@bat710.univ-lyon1.fr - alain.mille@liris.cnrs.fr
Matériaux	<u>MATERIAUX DE LYON</u> M. Jean Marc PELLETIER Secrétariat : C. BERNAVON 83.85	M. Jean Marc PELLETIER INSA de Lyon MATEIS Bâtiment Blaise Pascal 7 avenue Jean Capelle 69621 VILLEURBANNE Cédex Tél : 04.72.43 83 18 Fax 04 72 43 85 28 Jean-marc.Pelletier@insa-lyon.fr
MEGA	<u>MECANIQUE, ENERGETIQUE,</u> <u>GENIE CIVIL, ACOUSTIQUE</u> M. Jean Louis GUYADER Secrétariat : M. LABOUNE PM : 71.70 –Fax : 87.12	M. Jean Louis GUYADER INSA de Lyon Laboratoire de Vibrations et Acoustique Bâtiment Antoine de Saint Exupéry 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél :04.72.18.71.70 Fax : 04 72 43 72 37 mega@lva.insa-lyon.fr
ScSo	<u>ScSo*</u> M. OBADIA Lionel Insa : J.Y. TOUSSAINT	M. OBADIA Lionel Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Tél : 04.78.77.23.88 Fax : 04.37.28.04.48 Lionel.Obadia@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

REMERCIEMENTS

Je tiens tout d'abord à remercier mes directeurs de thèse Monsieur le Professeur Robert Laurini, Monsieur le Professeur Ahmed Lbath et Monsieur le Professeur Ziad Belhadj pour leur encadrement et leur soutien tout au long de ces années de doctorat.

Je voudrais exprimer mes reconnaissances à Monsieur Robert Laurini pour les longues discussions et les conseils qu'il m'a donnés. Ses observations judicieuses et ses remarques ont été des plus déterminantes dans la conduite de ces travaux et à la forme finale qu'ils ont pris.

Je tiens spécialement à adresser mes reconnaissances à Monsieur Ahmed Lbath qui m'a suivie de très près dans mes travaux (d'abord à Lyon puis à Grenoble) et y a consacré son temps, son énergie et m'a permis d'avancer au mieux dans ma recherche. Son encadrement restera pour moi un très bon souvenir et m'aura grandement appris. Il a su me faire découvrir la visualisation scientifique, me donner la passion de la recherche, aiguïser mon esprit critique, motiver ma curiosité....

Je tiens à adresser mes reconnaissances à Monsieur Ziad Belhadj, pour la confiance qu'il m'a laissée par une large autonomie dans mes orientations en veillant et en m'aidant à la progression de cette thèse. Sa bonne humeur comme son professionnalisme ont été des véritables moteurs et ce fut une bien belle thèse.

J'ai été honoré que Monsieur Rached Boussema, Professeur à l'Ecole Nationale d'Ingénieurs de Tunis, ait accepté de présider mon jury de thèse. J'en profite pour remercier également :

- Madame Christine Parent, Professeur à l'Université de Lausanne,
- Monsieur Mohamed Saber Naceur, Professeur à l'Institut National des Sciences Appliquées et de Technologie de Tunis,

pour avoir bien voulu consacrer une partie de leur temps à ma thèse et pour l'honneur qu'ils m'ont fait en acceptant de participer à mon jury de thèse.

Je remercie plus particulièrement Madame le Professeur Thérèse Libourel, Professeur à l'Université Montpellier II et Monsieur Sami Faiz, Maître de conférences à l'Université de Jendouba d'avoir accepté d'être mes rapporteurs. Leurs remarques et suggestions ont été très pertinentes et ont permis d'améliorer mon travail.

Je remercie chaleureusement tous les membres des laboratoires LIRIS et URISA et plus particulièrement Madame Amel Ben Azza directrice de l'URISA pour son aide et son soutien durant ces années. Je tiens spécialement à adresser toute ma reconnaissance et mon amitié à Monsieur Riadh Abdelfatteh qui n'a jamais cessé de m'encourager.

Je vais laisser place à mes sentiments pour exprimer toute mon amitié à toutes les personnes que j'ai connues et qui m'ont soutenue lors de cette thèse. Plus particulièrement, tous mes amis et mes cousins, plus particulièrement Kais.

Je n'oublie pas de remercier également l'ensemble de mes collègues à Sup'COM, plus particulièrement Madame Ouachani, Madame Ayachi, Madame Fouzai et Monsieur Bouhari.

Je remercie ma tante Raoudha et mon oncle Bechir avec qui j'ai passé des moments agréables. Par leur gentillesse et leur convivialité, ils ont rendu mes séjours à Lyon très agréables parmi eux. Qu'ils trouvent ici l'expression de toute ma gratitude.

On dit qu'on garde le meilleur pour la fin et c'est ma famille que je remercie très particulièrement. Toute ma reconnaissance à ma famille pour m'avoir soutenue et pour avoir eu beaucoup de patience pendant ces longues années de thèse. Sans ma famille, cette thèse n'aurait pas eu lieu.

Merci donc à mes parents qui m'ont toujours poussée à aller de l'avant et qui ont toujours cru en moi.

Je remercie mon mari qui m'a encouragée, a fait preuve de beaucoup de patience et je ne peux lui dire qu'un grand merci.

Je remercie ma sœur, mon frère, mes belles sœurs et mon beau frère de m'avoir soutenue.

Finalement, je remercie mes enfants Achraf et Elyes qui sont des sources de joie pour moi chaque jour, de m'avoir supportée durant les durs moments de la thèse et avec qui je souhaite « combler le temps perdu ».

Je souhaite bonne lecture à celles et ceux qui s'intéresseront à ce mémoire.

Avertissement au lecteur

Dans de nombreux diagrammes de ce mémoire la couleur a de l'importance. Si le lecteur dispose d'une copie en noir et blanc malheureusement certains schémas peuvent être incompréhensibles prière de m'excuser.

Table des matières

CHAPITRE 1. INTRODUCTION GENERALE	16
1.1 INTRODUCTION	16
1.2 PROBLEMATIQUE	16
1.3 CONTRIBUTIONS.....	18
1.4 PLAN DE LA THESE	18
PARTIE 1 CONTEXTE DE LA THESE	21
CHAPITRE 2. ETAT DE L'ART	22
2.1 INTRODUCTION	22
2.2 LES METHODES DE CONCEPTION POUR LES APPLICATIONS GEOMATIQUES.....	22
2.2.1 <i>La méthode MODUL-R</i>	23
2.2.2 <i>La méthode Géo2</i>	24
2.2.3 <i>La méthode MECOSIG</i>	26
2.2.4 <i>La méthode Pollen</i>	28
2.2.5 <i>La méthode GeoOOA</i>	30
2.2.6 <i>La méthode OMEGA</i>	31
2.2.7 <i>La méthode MADS</i>	33
2.2.8 <i>La méthode T-OMEGA</i>	37
2.2.9 <i>La méthode OMT-G</i>	40
2.2.10 <i>La méthode GeoFrame</i>	41
2.2.11 <i>La méthode Geo-UML</i>	42
2.2.12 <i>Le formalisme STER</i>	43
2.3 LES METHODES DE CONCEPTION POUR LES APPLICATIONS TEMPS REEL.....	43
2.3.1 <i>Système temps réel</i>	43
2.3.2 <i>Caractéristiques des applications temps réel</i>	45
2.3.3 <i>Conception des applications temps réel</i>	46
2.4 LES PRINCIPAUX LANGAGES DE SPECIFICATIONS FORMELLES	52
2.4.1 <i>Le langage Z</i>	52
2.4.2 <i>Le langage B</i>	53
2.5 COMPARAISON DES METHODES EXISTANTES	54
2.6 CONCLUSION	56

PARTIE 2 CONTRIBUTIONS	57
CHAPITRE 3. UNE METHODE DE CONCEPTION POUR LES APPLICATIONS GEOMATIQUES TEMPS REEL	58
3.1 INTRODUCTION : PROFIL UML POUR LES APPLICATIONS GEOMATIQUES TEMPS REEL (GRT)	58
3.2 DEMARCHE METHODOLOGIQUE	59
3.3 TYPES D'OBJETS A MODELISER	60
3.4 PRESENTATION DES DIFFERENTS MODELES DU FORMALISME ETENDU	60
3.4.1 <i>Le diagramme de classes de GRT</i>	61
3.4.2 <i>Le diagramme de priorités</i>	75
3.4.3 <i>Le diagramme d'état transition de GRT</i>	77
3.4.4 <i>Le diagramme d'activités de GRT</i>	78
3.4.5 <i>Le diagramme d'interventions</i>	81
3.5 CONCLUSION	85
CHAPITRE 4. REGLES DE PASSAGE DU MODELE CONCEPTUEL AU MODELE LOGIQUE DANS GRT	86
4.1 INTRODUCTION	86
4.2 LES REGLES DE TRANSFORMATION DANS GRT	87
4.2.1 <i>Transformations structurelles</i>	87
4.2.2 <i>Transformations d'attributs</i>	88
4.2.3 <i>Transformation d'une association spatiale</i>	91
4.2.4 <i>Transformation d'un attribut spatial d'une classe spatiale en une nouvelle classe</i>	92
4.2.5 <i>Transformation d'une classe spatiale</i>	93
4.2.6 <i>Transformation d'une classe à géométries combinées</i>	93
4.3 IMPLEMENTATION DES DIFFERENTES REGLES DE TRANSFORMATION DANS GRT	95
4.3.1 <i>Exemple 1</i>	95
4.3.2 <i>Exemple 2</i>	97
4.3.3 <i>Exemple 3</i>	99
4.4 CONCLUSION	101
CHAPITRE 5. EXPERIMENTATION : IMPLEMENTATION D'UN AGL (AGRT)	102
5.1 INTRODUCTION	102
5.2 UN OUTIL POUR LA CONCEPTION DES APPLICATIONS GEOMATIQUES TEMPS REEL	102
5.2.1 <i>Description de l'AGL</i>	103
5.2.2 <i>Architecture de l'AGL</i>	103
5.2.3 <i>Modèle conceptuel d'AGRT</i>	104

5.3	ETUDES DE CAS	117
5.3.1	<i>Quelques exemples d'applications géomatiques temps réel</i>	118
5.3.2	<i>Cahier des charges et spécification pour une application géomatique temps réel : « gestion des alarmes dans un réseau GSM »</i>	121
5.4	CONCLUSION	149
CHAPITRE 6. CONCLUSION ET PERSPECTIVES		150
6.1	CONCLUSION	150
6.2	PERSPECTIVES	151
6.2.1	<i>Perspectives d'applications</i>	151
6.2.2	<i>Perspectives du profil</i>	151
6.2.3	<i>Perspectives de l'AGL</i>	152
REFERENCES BIBLIOGRAPHIQUES		153
ANNEXES		168
ANNEXE A. LES ALARMES SELON LE SYSTEME BSS ALCATEL B6.2		169
ANNEXE B. LES REGLES DE PASSAGE DU MODELE CONCEPTUEL MADS A UN MODELE LOGIQUE		172
ANNEXE C. CAHIER DES CHARGES ET SPECIFICATION D'UNE APPLICATION GEOMATIQUE TEMPS REEL : « UN SYSTEME DE DETECTION D'ALARME DANS LES WSN»		179
ANNEXE D. CONVENTION DE THESE EN COTUTELLE		194

Liste des tableaux

Tableau 1. Récapitulatif de quelques méthodes de conception pour les applications géomatiques temps réel.....	56
Tableau 2. Présentation des types géométriques simples.....	64
Tableau 3. Présentation des types géométriques complexes.....	64
Tableau 4. Pictogrammes spécifiques aux associations spatiales (1).....	66
Tableau 5. Pictogrammes spécifiques aux associations spatiales (2).....	66
Tableau 6. Pictogrammes spécifiques aux associations temporelles.....	68
Tableau 7. Représentation de la notion du temps au niveau du diagramme d'activités GRT. ..	79
Tableau 8. Représentation des opérateurs du temps au niveau du diagramme d'interventions.	82
Tableau 9. Liste des types géométriques suivant SDO_GTYPE.....	89

Liste des figures

Figure 1. Architecture d'un système temps réel [IDO 06].	44
Figure 2. SADT: datagramme et actigramme.	50
Figure 3. Présentation générale du profil GRT.	59
Figure 4. Présentation d'un exemple de classe définie par le diagramme de classes de GRT.	61
Figure 5. Présentation d'une classe importée au niveau du diagramme de classes de GRT.	62
Figure 6. Exemple de spécification de géométrie de stockage et d'affichage.	63
Figure 7. Représentation d'une classe spatiale.	64
Figure 8. Représentation d'une classe spatiale avec un attribut spatial.	65
Figure 9. Représentation d'une classe spatiale en utilisant la combinaison de types géographiques.	65
Figure 10. Représentation d'une classe spatiale en utilisant une géométrie d'affichage et une géométrie de stockage.	65
Figure 11. Représentation d'une association spatiale de type inclusion.	66
Figure 12. Représentation d'une classe temporelle ayant une spatialité temporelle.	68
Figure 13. Représentation d'une classe spatiale possédant un attribut spatio-temporel.	68
Figure 14. Présentation des attributs spéciaux.	69
Figure 15. Exemple de représentation d'une classe avec des attributs.	69
Figure 16. Méta-Modèle UML : Paquetages Principaux.	71
Figure 17. Méta-Modèle UML : Paquetage « Fondement ».	71
Figure 18. Méta-Modèle UML : Paquetage « Eléments du Comportement ».	72
Figure 19. Extension du Méta-Modèle UML : Diagramme de classes.	73
Figure 20. Extension du Méta-Modèle UML : Paquetage Géomatique.	74
Figure 21. Présentation d'un diagramme de priorités selon GRT.	76
Figure 22. Extension du Méta-Modèle UML : Diagramme de priorités.	77
Figure 23. Extension du Méta-Modèle UML : Diagramme d'état transition.	78
Figure 24. Présentation d'un diagramme d'activités selon GRT.	80
Figure 25. Extension du Méta-Modèle UML : Diagramme d'activités.	81
Figure 26. Présentation d'un diagramme d'interventions selon GRT.	83
Figure 27. Extension du Méta-Modèle UML : Diagramme d'interventions.	84
Figure 28. Définition d'un trigger permettant l'élimination d'un attribut énuméré.	88
Figure 29. Création du type SDO_GEOMETRY.	89
Figure 30. Création du type SDO_POINT_TYPE.	90
Figure 31. Trigger1 vérifiant la cohérence des dates.	91
Figure 32. Trigger2 vérifiant la cohérence des dates.	91
Figure 33. Exemple d'une association spatiale de type égalité.	92
Figure 34. Définition d'un trigger assurant le respect d'une association spatiale de type égalité.	92

Figure 35. Présentation d'une classe spatiale ayant des géométries combinées par le connecteur « ET ».	93
Figure 36. Présentation d'une classe spatiale ayant des géométries combinées par le connecteur « OU ».	93
Figure 37. Définition d'un trigger permettant la vérification pour une classe ayant des géométries combinées de stockage liées par le connecteur « OU ».	94
Figure 38. Définition d'un trigger permettant la vérification pour une classe ayant des géométries combinées d'affichage liées par le connecteur « OU ».	94
Figure 39. Schéma conceptuel présentant deux classes spatiales liées par une association simple.	95
Figure 40. Le code source généré à partir du modèle conceptuel.	96
Figure 41. Schéma conceptuel présentant une association spatiale de type égalité.	97
Figure 42. Le code source généré à partir du modèle conceptuel.	98
Figure 43. Schéma conceptuel présentant des entités à géométries combinées.	99
Figure 44. Le code source généré à partir du modèle conceptuel.	100
Figure 45. Diagramme de cas d'utilisation d'AGRT : création et sauvegarde du projet et des diagrammes (ce diagramme a été crée à l'aide d'AGRT)	105
Figure 46. Diagramme de cas d'utilisation d'AGRT : édition des diagrammes et génération du code.	106
Figure 47. Diagramme de classes présentant les composants graphiques d'AGRT (méta-modèle d'AGRT).	108
Figure 48. Diagramme de classes d'AGRT : affichage, manipulation et stockage des diagrammes.	110
Figure 49. Diagramme de classes d'AGRT : génération du code source.	112
Figure 50. Exemple de diagramme de classes réalisé par AGRT.	113
Figure 51. Code source de l'interface « interface1 ».	113
Figure 52. Code source de l'interface « interface2 ».	113
Figure 53. Code source de la classe « Classe1 ».	114
Figure 54. Code source de la classe « Classe2 ».	115
Figure 55. Code source de la classe « ClasseMere ».	116
Figure 56. Code source de la classe « GPoint ».	116
Figure 57. Code source de la classe « GSurface ».	117
Figure 58. Le concept cellulaire.	123
Figure 59. L'architecture du Réseau GSM.	125
Figure 60. Diagramme de priorités.	131
Figure 61. Diagramme de classes d'applications.	133
Figure 62. La classe d'application SGBD-Géographique.	134
Figure 63. La classe d'application Site-de-supervision.	135
Figure 64. La classe d'application SGBD-Alarmes.	136
Figure 65. Les sous classes de la classe « Alarme_TR_BTS ».	137
Figure 66. Les sous classes de la classe « Alarme_TD_BTS ».	138

Figure 67. Les sous classes de la classe « Alarme_TM_BTS ».	138
Figure 68. Les sous classes de la classe « Alarme_TR_BSC ».	139
Figure 69. Les sous classes de la classe « Alarme_TD_BSC ».	140
Figure 70. Les sous classes de la classe « Alarme_TM_BSC ».	140
Figure 71. Scénario 1 : réaction en cas de déclenchement d'une alarme critique.	141
Figure 72. Scénario 2 : réaction en cas de déclenchement d'une alarme majeure.	142
Figure 73. Scénario 3 : réaction en cas de déclenchement simultané d'une alarme critique et d'une alarme mineure.	143
Figure 74. Présentation du thème « El-Ghazela ».	144
Figure 75. Présentation du thème BTS.	145
Figure 76. Présentation du thème BSC.	145
Figure 77. Présentation du thème lien Abis.	146
Figure 78. Présentation de l'ensemble des thèmes.	146
Figure 79. Exemple de manifestation d'une alarme critique.	147
Figure 80. Différents boutons pour visualiser l'historique des alarmes.	148
Figure 81. Exemple d'historique d'alarmes pour un BSC.	148
Figure 82. Exemple 1 d'une association n -aire avec un rôle monovalué.	173
Figure 83. Exemple 2 d'une association n -aire avec un rôle monovalué.	173
Figure 84. Définition d'un trigger permettant l'élimination d'une association n -aire avec un rôle monovalué.	174
Figure 85. Présentation d'un lien d'héritage.	175
Figure 86. Architecture du WSN.	180
Figure 87. Architecture d'un nœud de réseau de capteurs [CAR 06].	182
Figure 88. Diagramme de priorités.	185
Figure 89. Diagramme de classes.	186
Figure 90. Scénario 1 : déclenchement d'une alarme du « Sink ».	187
Figure 91. Scénario 2 : déclenchement de deux alarmes de priorités différentes.	188
Figure 92. Scénario 3 : déclenchement d'une alarme spécifiant une panne d'un capteur.	189
Figure 93. Scénario 4 : déclenchement d'une alarme spécifiant une destruction d'un capteur.	190
Figure 94. Cartographie du réseau de capteurs sans fils.	191
Figure 95. Exemple d'une table relative à un capteur.	192
Figure 96. Détection des pannes de capteurs.	193

CHAPITRE 1.

INTRODUCTION GENERALE

1.1 Introduction

Le travail que nous présentons dans ce mémoire est le fruit d'une thèse en cotutelle entre l'Institut National des Sciences Appliquées (INSA) de Lyon et l'Ecole Supérieure des Communications de Tunis (SUP'COM). Ce travail a fait l'objet d'une collaboration entre le laboratoire LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information), le Laboratoire d'Informatique de Grenoble (LIG) de l'Université de Grenoble et le laboratoire URISA (Unité de Recherche en Imagerie Satellitaire et ses Applications) de SUP'COM. Comme les applications géomatiques temps réel ne disposent pas de méthodes de conception spécifiques, l'objectif de ce travail est de fournir une méthode de conception permettant la conception des applications géomatiques temps réel.

1.2 Problématique

Le domaine des applications géomatiques est en plein essor et le nombre de logiciels spécialisés dans la manipulation des données spatio-temporelles a connu une croissance considérable cette dernière décennie. Cette évolution a été accompagnée, dans la littérature, par la proposition de plusieurs méthodes de conception dédiées à la modélisation des informations à référence spatio-temporelle. Il existe des méthodes de conceptions pour les applications géomatiques.

Sachant que la géomatique est l'étude de l'acquisition, la production, la gestion et la diffusion d'informations rattachées à des coordonnées géographiques [PIN02]. Une application géomatique est une application s'adressant à une classe de problèmes de gestion de données spatiales orientées utilisateur final non expert [LBA97]. Alors qu'une application télégéomatique est une application géomatique intégrant les nouvelles technologies en matière de communication. Les applications télégéomatiques peuvent être considérées comme des applications géomatiques communiquant et interagissant entre elles.

Les applications temps réel évoluent aussi énormément ces dernières années et ils manipulent de plus en plus des données à références spatiales.

Une application temps réel est une application dans laquelle l'exactitude ne dépend pas seulement du résultat mais aussi du temps auquel ce résultat est

produit. Il existe aussi plusieurs méthodes et profils pour la modélisation des applications temps réel mais celles-ci ne tiennent pas compte de la référence spatiale au niveau conceptuel.

Actuellement, l'émergence des systèmes d'information pervasifs (systèmes offrant à l'utilisateur des informations voire des services indépendamment du lieu géographique et du positionnement dans le temps) ouvre plusieurs axes de recherche. Ce sont des systèmes qui traitent des applications géomatiques temps réel. Ainsi, plusieurs problématiques sont soulevées dans de nombreuses applications telles que l'application de gestion de flottes de véhicules terrestres par GPS (Global Positioning System), l'application de gestion d'alarmes en temps réel dans un réseau GSM (Global System of Mobile Communications), l'application de gestion de l'épuration des eaux usées, l'application de gestion d'alarmes dans les réseaux de capteurs sans fil ou des applications d'aide à la décision temps réelle alimentant une base de données spatio-temporelle.

Parmi ces dernières, on cite l'architecture logicielle de telles applications qui doit supporter des modèles mathématiques et des algorithmes qui ont des capacités de calculs importants ; de plus la prise en compte de la granularité du temps réel (par exemple la latitude du temps de réactivité des systèmes) et la conception et la modélisation de telles applications doivent permettre de formaliser les spécifications et d'apporter des outils logiciels cohérents par rapport aux besoins des utilisateurs et des contraintes liées à la réactivité et éventuellement la mobilité des composants du système.

Les différents formalismes qui existent dans la littérature et qui sont supportés par des méthodes de conception des applications géomatiques voire télégéomatiques ainsi que des applications temps réel sont insuffisants et ne répondent pas au besoin du concepteur pour modéliser des applications géomatiques temps réel.

Notre démarche a été en premier lieu d'étudier et de spécifier plusieurs applications géomatiques temps réel, ces dernières ont tout naturellement engendré de nouveaux besoins en termes de conception et de développement. De même elles possèdent plusieurs caractéristiques géomatiques temps réel :

- la rapidité de réaction,
- les contraintes de temps, qui peuvent être plus ou moins impératives,
- la présence de plusieurs capteurs qui vont envoyer des données à un site central à des rythmes plus ou moins variables,
- l'importance des pannes, leur conséquence, leur prise en compte sur l'architecture du système et la nécessité de les réparer rapidement,
- la surveillance des systèmes qui nécessite la gestion d'états internes avec un enchaînement spécifique,
- la nécessité de visualisation en temps réel de l'évolution des systèmes.
- etc.

Ainsi donc, apparaît le besoin de tenir compte de ces caractéristiques dans la modélisation des applications géomatiques temps réel.

1.3 Contributions

Dès lors, le but de nos travaux a été de définir une méthode de conception adaptée aux spécificités des applications géomatiques temps réel. Afin d'atteindre cet objectif, nous nous sommes inspirée des méthodes destinées à la conception des applications géomatiques, ainsi que des méthodes pour la conception des applications temps réel.

De ce fait notre principale contribution est la définition d'un profil UML (Unified Modeling Language) dédié aux applications géomatiques temps réel, appelé GRT, ce dernier reprend certaines caractéristiques des méthodes MADS et T-OMEGA, ainsi que le profil TURTLE tout en intégrant de nouveaux autres concepts liés au temps réel. Par conséquent, au niveau de GRT nous avons effectué des extensions dans certains diagrammes pour qu'ils tiennent en compte des caractéristiques des applications géomatiques temps réel. Nous avons de même réalisé l'intégration du méta-modèle de données des différents diagrammes.

Pour parachever notre étude nous avons effectué la validation du profil avec plusieurs applications géomatiques temps réel.

Le profil GRT pour les applications géomatiques temps réel présentées dans ce mémoire est supporté par un outil de type Atelier de Génie Logiciel (AGL) appelé AGRT, que nous avons conçu et réalisé et qui permet de spécifier les diagrammes et de générer le code source.

Au niveau d'AGRT on donne le choix au concepteur de générer un code source en JAVA ou en SQL. Pour la génération du code SQL, nous avons conçu et implémenté des règles de passage de GRT au modèle logique.

1.4 Plan de la thèse

Après le chapitre 1 qui présente l'introduction générale, ce mémoire se décompose en trois parties : une partie dédiée au contexte de la thèse, une deuxième partie dédiée à notre contribution et une troisième partie dédiée aux annexes.

La première partie se compose du chapitre 2 qui est consacré à l'état de l'art. Au niveau de ce chapitre, dans un premier temps, nous présentons les méthodes de conception pour la modélisation des applications géomatiques. Par la suite, nous présentons les méthodes de conception pour des applications temps réel. Ensuite, une synthèse comparative est réalisée pour les principales méthodes

existantes. Enfin, nous mettons en lumière les problématiques soulevées et les contributions que nous présentons dans cette thèse.

La deuxième partie détaille notre contribution et elle s'articule autour de trois chapitres :

- Dans le chapitre 3 nous proposons un nouveau formalisme pour modéliser les applications géomatiques temps réel : le profil GRT. Ce nouveau formalisme va tenter de pallier certaines faiblesses des formalismes actuels. Il est basé sur la méthode de conception MADS, la méthode T-OMEGA et le profil TURTLE. Par la suite, nous avons défini les différents diagrammes de GRT ainsi que l'intégration au méta-modèle UML.
- Puis au niveau du chapitre 4 nous avons conçu et implémenté les règles de passage du profil GRT au niveau logique. Ces règles ont été implémentées sur le SGBD Oracle Spatial.
- Ensuite, nous présentons dans le chapitre 5 notre validation expérimentale par un AGL permettant de modéliser une application géomatique temps réel selon le profil GRT. Nous validerons aussi notre méthode de conception sur une étude de cas.

Enfin nous terminons notre travail par le chapitre 6 : conclusion et perspectives. Ce chapitre conclut ce mémoire par un bilan général des travaux de cette thèse ainsi que quelques pistes intéressantes en guise de perspectives de recherche pour une suite des travaux.

Pour bien clarifier certains aspects nous avons établi une troisième partie dédiée aux annexes, elle se compose de quatre annexes :

- Annexe A : les alarmes selon le système BSS ALCATEL B6.2
Pour alléger le volume de ce mémoire nous avons jugé nécessaire de spécifier en annexe les différents types d'alarmes dans un réseau GSM selon le système BSS ALCATEL B6.2.
- Annexe B : les règles de passage du modèle conceptuel MADS à un modèle logique
Etant donné que notre profil GRT est basé sur la méthode MADS et la méthode T-OMEGA, lors du passage d'un modèle conceptuel réalisé selon GRT au modèle logique nous pouvons utiliser quelques règles de passage du modèle conceptuel MADS au modèle logique, c'est la raison pour laquelle nous avons défini ces dernières en annexe.

- Annexe C : cahier des charges et spécification d'une application géomatique temps réel : « un système de détection d'alarmes dans les WSN»

Au niveau de cette annexe, nous présentons une application géomatique temps réel concernant la gestion d'alarmes dans les réseaux de capteurs sans fil «WSN (Wireless Sensor Networks)». Nous avons conçu cette application par le profil GRT en utilisant l'AGL AGRT. Enfin nous présentons une implémentation de cette application.

- Annexe D : convention de thèse en cotutelle

Comme ce travail a été effectué en cotutelle entre l'INSA de Lyon et SUP'COM de Tunis, il nous a semblé intéressant de mettre une copie de la convention de cotutelle et de son avenant en annexe.

PARTIE 1

CONTEXTE DE LA THESE

CHAPITRE 2.

ETAT DE L'ART

2.1 Introduction

Actuellement, il n'existe pas de méthode de conception spécifiquement définie pour les applications géomatiques temps réel. L'état de l'art présenté dans ce chapitre est divisé en trois parties. Celui ci consiste en une étude des méthodes de conception pour les applications géomatiques, en une étude des méthodes de conception pour les applications temps réel et en une étude des principaux langages de spécifications formelles. Notre but est de faire une analyse la plus exhaustive possible des méthodes existantes afin de bien dégager les difficultés rencontrées lors de la conception des applications géomatiques temps réel.

2.2 Les méthodes de conception pour les applications géomatiques

L'information spatiale est de plus en plus un élément de base pour la stratégie d'une organisation et de justification de ses actions et plus particulièrement les organisations de gestion de territoire et la mise en place de Systèmes d'Information Géographique (SIG) requièrent une coordination des ressources humaines, matérielles, etc. nécessaires à l'acquisition, le traitement et la diffusion de l'information. Les grandes phases pour réussir la mise en place d'un SIG sont : la définition des besoins, la rédaction du cahier des charges, le choix du SIG, la conception de la base de données géographiques et l'installation et le démarrage du projet [FAI 99].

La mise en place d'applications géomatiques suscite un cadre méthodologique organisationnel répondant aux besoins spécifiques à l'implémentation de SIG de part la nature des données manipulées, de leurs complexités, de leur volume, et du nombre d'acteurs intervenant à plusieurs niveaux dans l'organisation en faisant collaborer des décideurs, des élus, des géomaticiens ou informaticiens spécialisés en géomatique, etc.

Les Systèmes d'Information Géographique, malgré leurs apports incontestables aux techniques de traitement de l'information, présentent encore des problèmes au niveau conceptuel.

Les applications spatiales sont généralement complexes à plusieurs titres [PAR 97]:

- les applications requièrent des bases de données volumineuses,
- un grand nombre d'applications et d'utilisateurs potentiels ayant des préoccupations très différentes,
- les objets utilisés ont une structure complexe, on trouve des objets composés d'autres objets, des objets spécialisés ou génériques,
- l'espace n'est pas perçu de la même manière, un même phénomène peut avoir plusieurs représentations spatiales selon l'utilisateur, l'instant ou l'échelle.

Le modèle conceptuel adéquat doit être puissant, simple, visuel, formel et temporel. La modélisation conceptuelle, par sa lisibilité, facilite les échanges d'informations entre des partenaires d'organisations différentes.

Le modèle conceptuel spatio-temporel doit capter les sémantiques essentielles de la variation de l'information au cours du temps. Il faut que le modèle conceptuel spatio-temporel soit compatible avec la modélisation traditionnelle pour qu'il permette la modélisation des données qui ne sont ni spatiales ni temporelles [PAR 97]. Dans un modèle conceptuel spatio-temporel, nous devons connaître les différents types de liens (liens classiques, liens topologiques et liens temporels) entre les entités du monde réel.

La modélisation des données est une phase importante dans le processus d'élaboration des SIG. Elle s'exprime par des états successifs de modèles symbolisant différents niveaux d'abstraction entre la réalité et la base de données.

Il existe plusieurs formalismes et méthodes dédiés pour la modélisation des applications géomatiques : MODUL-R, Géo2, MECOSIG, Pollen, GeoOOA, OMEGA, T-OMEGA, MADs, GeoFrame, GeoUML, STER, etc. Ces derniers intègrent entre autres des moyens pour décrire les différents types géographiques dans les modèles. Nous réalisons un résumé des propositions de chaque formalisme ou méthode. Des comparatifs détaillés peuvent être trouvés dans [CLA97], [LBA97], [FRI01], [PIN02], [MIR06], [BEN 06], [BEN 07] et [DUB 07].

2.2.1 La méthode MODUL-R

MODUL-R [CAR 93] a été créé pour tenter de résoudre les lacunes rencontrées dans les modèles conceptuels de données à références spatiales et temporelles. Il est l'un des premiers formalismes complets, rencontrés dans la littérature, dédié à la conception d'applications géomatiques.

L'idée de base qui a suscité la création du formalisme MODUL-R est partie du constat suivant : l'adaptation du formalisme de Merise a été concrétisée par l'introduction, dans le MCD (Modèle Conceptuel des Données), d'entités

géométriques (polygone, ligne et point) permettant ainsi d'illustrer la référence spatiale.

Bien que Modul-R soit basé sur le formalisme de la méthode Merise, le développement de Modul-R a été influencé par l'approche Information Engineering [BED 96]. Cependant la lisibilité du schéma se trouvait inversement proportionnelle à la taille du modèle et à sa complexité.

Modul-R est composé de deux mots : «Modul» et «R». «Modul» signifie une structure modulaire de modélisation, alors que «R» signifie une modélisation intégrée de la réalité.

Le formalisme Modul-R ajoute trois nouveaux modules au formalisme individuel. Il est donc constitué de quatre modules : entité-relation, référentiel, agrégation et simplification.

L'approche Modul-R permet d'obtenir un langage complet et flexible :

- Modul-R est très complet pour les données des SIG du fait que différentes solutions permettent une modélisation en tenant compte de tous les aspects spécifiques à ce domaine : référence spatiale et temporelle des entités, agrégation / généralisation spatiale, etc.
- La flexibilité se manifeste en ce qui concerne le choix des composants sémantiques, le choix des sections à modéliser et le fait d'avoir plusieurs possibilités de visualisation.

2.2.1.1 *Les difficultés d'utilisation de la méthode MODUL-R*

La méthode Modul-R présente des inconvénients [LBA 97] :

- les traitements et les données sont séparés au niveau conceptuel,
- absence de représentation visuelle,
- absence de représentation dynamique,
- absence de contraintes topologiques,
- absence de représentation de relations topologiques,
- l'AGL support de MODUL-R est lié au SIG Microstation de la société Intergaph,
- absence de génération de traitements par cet AGL,
- etc.

2.2.2 La méthode Géo2

Le modèle de Géo2 [DAV 93] consiste à étendre le SGBD orienté-objet «SGBDOO» à la gestion des données géographiques. Un des avantages de ce modèle est qu'il intègre l'approche orientée-objet, le modèle de Géo2 s'étend aux types de données géométriques tels que le point, la ligne et la surface, aux fonctions telles que l'union, la distance, etc., et aux prédicats topologiques associés : l'adjacence, le chevauchement, etc.

La gestion d'informations géographiques nécessite des structures performantes et adéquates pour le stockage des primitives géométriques et topologiques et des relations topologiques entre ces primitives. Il y a trois structures de données parmi lesquelles l'utilisateur peut faire son choix : la structure de données filières «spaghetti», la structure de données réseau et la structure de données topologiques. Le caractère original dans Géo2 consiste à conserver dans la structure une hiérarchie entre ces trois structures. Grâce au mécanisme d'héritage, on peut utiliser d'une façon transparente l'une ou l'autre des structures. On aboutit ainsi à un modèle interne unique.

2.2.2.1 *Structure de Géo2*

Le modèle conceptuel Géo2 est constitué d'un modèle sémantique et d'un modèle de localisation.

- **Le modèle géographique sémantique**

Le modèle géographique sémantique doit permettre à l'utilisateur de s'abstraire au maximum des mécanismes utilisés pour implémenter le système et, en particulier, de la façon de gérer la localisation des entités géographiques. Cette localisation est une propriété importante des entités géographiques et peut être traitée comme un type abstrait de données.

Le modèle géographique sémantique étend le modèle entité-association avec le mécanisme d'héritage et de propagation.

Le mécanisme d'héritage est introduit à l'aide des deux concepts de spécialisation et de généralisation.

Quant au mécanisme de propagation, il est introduit dans le modèle à l'aide d'une information supplémentaire sur les attributs des entités et d'un type d'association particulier appelé «association de composition» qui permet d'associer une entité complexe à ses composants.

Aussi à chaque attribut d'une entité est associée une fonction d'agrégation pour cet attribut.

A partir de la connaissance des associations de composition et des fonctions d'agrégation, on peut propager automatiquement sur toutes les entités complexes des attributs calculés à partir des attributs définis sur les entités simples.

- **Le modèle de localisation**

L'intérêt d'un modèle géométrique réside essentiellement dans la possibilité de réaliser des traitements basés sur la localisation des données. Ces traitements sont définis par les opérations du type abstrait des données «TAD». Une opération du TAD doit être définie comme une fonction possédant un ou plusieurs paramètres et fournissant en résultat une valeur de localisation. Les fonctions d'intersection et d'union géométrique font partie de ces opérations.

Le choix d'un modèle fonctionnel impose une contrainte sur la définition du TAD, c'est-à-dire, le résultat de toute fonction doit être une valeur du TAD. De plus ce TAD doit permettre de représenter les données ponctuelles, linéaires et surfaciques. Ce TAD est appelé *Geometry*.

Ainsi, toute entité définie dans le modèle sémantique peut être localisée à l'aide d'un attribut de type *Geometry*.

2.2.2.2 Les difficultés d'utilisation de la méthode Géo2

Après l'utilisation de ce modèle, les chercheurs ont rencontré quelques problèmes parmi lesquels on peut citer [DAV 93] et [PAR 97] :

- le schéma complet des classes est complexe car il contient beaucoup de classes. Cette structuration a conduit à un long cheminement, de nombreux retours en arrière et beaucoup d'hésitations,
- il y a un autre problème qui réside dans l'utilisation du mécanisme d'héritage, ceci provient du fait qu'en orienté-objet, l'héritage est utilisé pour implémenter des mécanismes assez différents les uns des autres,
- les limites du SGBD O2 tels que :
 - l'inexistence de constructeurs de tableaux qui sont indispensables pour gérer les points intermédiaires des lignes,
 - l'inexistence de constructeurs d'associations qui éviteraient de gérer des références réciproques.
 - la faible diffusion du système O2.

2.2.3 La méthode MECOSIG

La méthode MECOSIG [PAN94], [PAN96] et [PAN97] « MEthode de COncption des Systèmes d'Information Géographique » est historiquement la première méthode orientée objets pour les SIG. Elle est basée sur un formalisme simple et puissant appelé CONGOO « COncption Géographique Orientée Objet ». Elle utilise quatre niveaux d'abstraction : un niveau descriptif, un niveau conceptuel, un niveau logique/physique et un niveau organisationnel. Le formalisme CONGOO est basé sur les formalismes MODUL-R et OOA/OOD (Object Oriented Analysis/ Object Oriented Design) [COA91] pour la modélisation statique des données et sur le Digramme de Flux de Données (DFD) [GAN 79] pour la modélisation des traitements.

MECOSIG se veut une méthode ouverte, c'est-à-dire susceptible d'être utilisée dans divers environnements, d'admettre l'interaction avec d'autres approches méthodologiques et d'accueillir un enrichissement ultérieur. La démarche méthodologique de MECOSIG est adaptable, c'est-à-dire qu'elle ne préconise pas un schéma déterministe de gestion de projets SIG. Au cours du cycle de développement, MECOSIG donne aux chefs de projets le choix entre plusieurs possibilités.

2.2.3.1 *Le formalisme CONGOO*

Le formalisme CONGOO sur lequel s'appuie MECOSIG, introduit certains points intéressants comme la distinction entre attributs logiques et attributs graphiques. Les attributs logiques sont les caractéristiques des objets du monde réel tels que : identifiant, nom, prénom, etc. Les attributs graphiques concernent les caractéristiques visuelles de la représentation de l'objet à l'écran tels que : couleur, forme "carto"-graphique, etc.

Les classes d'objets, sont à la fois des ensembles de toutes les instances représentées par des objets sémantiquement homogènes et des ensembles d'objets. Une classe peut contenir différents types d'objets correspondant à autant d'ensembles d'instances différents. Les caractéristiques graphiques, les attributs logiques ainsi que le mode d'implantation peuvent être différents pour les objets appartenant à la même classe. De plus, une classe peut inclure les contraintes topologiques s'appliquant entre toutes les instances représentées par les objets appartenant à la classe ainsi que les traitements qui s'appliquent aux objets eux-mêmes. Les attributs de la classe se rapportant à la population des instances de la classe peuvent être également mentionnés.

Un système intéressant de couches, est aussi introduit afin de regrouper, par thèmes des ensembles de classes. Les couches sont des ensembles de classes d'objets ayant en commun des relations, topologiques ou autres, et appartenant généralement à un même domaine du problème. Les couches sont donc des composants utilisés pour décrire la totalité de la base de données d'un système global, ainsi que l'ensemble des relations, notamment topologiques, communes entre les objets et/ou classes appartenant à la même couche.

Dans le formalisme CONGOO, un objet d'une classe peut avoir une géométrie ayant un type géographique. CONGOO distingue [PIN02] :

- les OGS (Objet Géographique Simple) dont le mode d'implantation peut être le point, la ligne, le polygone ou l'image notés respectivement OGS-P, OGS-L et OGS-Pol,
- les OGC (Objet Géographique Composé) correspondent à une réunion d'OGS ayant le même type d'implantation,
- les OGX (Objet Géographique Complexe) correspondent à la réunion d'objets pouvant avoir différents types d'implantation.

Le formalisme CONGOO, considère deux types de relations topologiques de base : le voisinage et la superposition. Elles peuvent être appliquées entre instances, objets, classes, couches et sous-couches, et leurs combinaisons. Le voisinage est considéré comme étant une superposition des frontières de la géométrie des objets, tandis que la superposition indique une intersection entre la géométrie des objets. Le voisinage et la superposition entre deux objets géographiques peuvent être d'une part nuls, partiels, ou totaux et d'autre part autorisés ou interdits. CONGOO considère que la relation « classique » de type

composé-composant peut être utilisée pour décrire un lien entre la géométrie des objets.

2.2.3.2 *Les difficultés d'utilisation de la méthode MECOSIG*

Parmi les difficultés rencontrées lors de l'utilisation de cette méthode :

- une limite forte du formalisme CONGOO est qu'il ne s'intéresse qu'aux propriétés spatiales des entités référencées. De ce fait, la temporalité et les combinaisons de spatialité et de temporalité ne peuvent pas être exprimées avec ce formalisme. De même les spatialités dérivées et les temporalités dérivées n'existent pas dans ce formalisme [MIR06],
- la méthode ne permet pas de définir des objets à implantation multiple,
- le concepteur est souvent confronté à une variété importante de concepts,
- la méthode s'intéresse aux aspects statiques et fonctionnels et ne propose pas de modèle dynamique pour les applications géomatiques et ce étant donné, que dans son approche objet, elle s'appuie sur la méthode OOA/OOD qui est orienté modèle statique,
- absence de représentation visuelle,
- la méthode est non supportée par un AGL,
- etc.

2.2.4 La méthode Pollen

La méthode Pollen «Procédure d'observation et de Lecture de L'environnement» [GAY97] a été développée dans le but d'intégrer des informations de nature différente dans un même système d'information. Elle a privilégié la représentation de l'espace et du temps indépendamment de tout système ou logiciel. Les techniques de conception de la méthode Pollen peuvent être utilisées par des non-informaticiens.

La méthode Pollen apporte un certain nombre de notions originales telles que la désagrégation, les objets opaques et les liens fonctionnels. Elle s'appuie sur une vision globale et abstraite du système d'information. Elle s'appuie, d'une part, sur des concepts théoriques de représentation du temps et de l'espace, et d'autre part, sur le formalisme de la méthode OMT «Object Modeling Technique» [RUM 91], et respecte donc le paradigme objet.

La méthode Pollen repose sur deux principes :

- une modélisation abstraite en phase d'analyse,
- le développement de systèmes informatiques multi-applications, basés sur la coopération entre logiciels.

2.2.4.1 *Modélisation des entités spatiales*

Le modèle objet de Pollen utilise le formalisme OMT enrichi du concept de classe opaque qui généralise les notions d'encapsulation et de réutilisation, et

qui décharge le concepteur de la complexité de la description des relations géométriques, topologiques, etc.

Les classes opaques sont des classes d'un type particulier. Elles disposent d'une structure interne qui n'est pas ou imparfaitement connue et rendent des services bien identifiés. Il s'agit en fait de représenter les classes d'objets complexes à partir de leur comportement et des services qu'ils rendent grâce aux opérations, et à ignorer leur structure, c'est-à-dire, leurs attributs.

Le concept d'objet opaque accroît considérablement le potentiel de la modélisation objet. Il permet d'intégrer n'importe quelle technique de traitement de données à l'intérieur d'un système d'information.

Il existe trois classes opaques abstraites spatiales fondamentales utilisées par Pollen : la classe Point, la classe Ligne et la classe Surface. Elles sont des classes abstraites, c'est-à-dire, elles sont destinées uniquement à donner naissance à d'autres classes par héritage.

Pollen distingue deux types de relations distinctes entre classes : les liens logiques et les liens fonctionnels. Les liens logiques sont les relations d'associations, d'agrégation et d'héritage. Quant aux liens fonctionnels, il existe un lien fonctionnel entre deux classes lorsqu'une opération d'une classe fait référence à la structure ou aux services de la seconde classe.

On trouve aussi la notion de désagrégation qui est très intéressante pour représenter des entités spatiales qui sont construites à partir de l'intersection d'autres entités spatiales. C'est en fait l'inverse de l'agrégation, on part d'objets connus de grandes tailles pour en construire des plus petits.

Dans Pollen, la vue continue d'un objet est exprimée par un lien multivalué vers un autre objet représentant le champ spatial continu, lui-même relié à un objet géométrique Point.

2.2.4.2 *Modélisation des entités temporelles*

L'aspect temporel est introduit au niveau du formalisme OMT par deux classes temporelles fondamentales : Instant et Intervalle [MIR 06].

Les classes Instant et Intervalle peuvent être munies de plusieurs opérations qui permettent de préciser des durées ou l'ordre des différents objets temporels. Ces classes vont permettre de positionner dans le temps deux types d'objets : les événements et les états. Un événement est défini comme un phénomène de durée nulle, alors qu'un état est défini comme un phénomène de durée non nulle.

2.2.4.3 *Modélisation de la dynamique du système*

Le modèle dynamique est un modèle hérité des diagrammes d'état d'OMT et des actigrammes de la méthode SADT «Structured Analysis and Design

Technique». Il enrichit le modèle objet par ajout d'opérations relatives aux processus modélisés.

Connaître la dynamique d'un système permet de prévoir son évolution, de comprendre son fonctionnement général et les interactions entre ses différents constituants.

Selon Pollen, on peut représenter les modèles dynamiques dans le modèle objet sous la forme de classes opaques reliées aux autres classes par l'intermédiaire des liens fonctionnels. Cette représentation a un double objectif :

- elle permet de préciser les services rendus par le modèle, sous la forme d'opérations, de tels modèles permettent d'évaluer la valeur de certaines variables en fonction du temps,
- elle permet de définir l'origine des informations qui vont servir à initialiser le modèle avant son utilisation.

2.2.4.4 *Les difficultés d'utilisation de la méthode Pollen*

Parmi les difficultés rencontrées lors de l'utilisation de cette méthode [MIR 06] et [PAR97] :

- l'absence de pictogramme est une faiblesse du formalisme de la méthode POLLEN car la mise en œuvre des classes opaques n'est pas aisée,
- il y a surcharge du schéma car il devient illisible du fait de la multiplication des liens,
- cette méthode ne permet d'associer la spatialité qu'aux types d'entité,
- la dimension du système de coordonnées n'est pas prise en compte,
- la méthode de modélisation de Système d'Information sur l'Environnement POLLEN n'a jamais été implémentée dans un atelier de génie logiciel.

2.2.5 La méthode GeoOOA

GeoOOA [KOS 95], [KOS 96], [KOS 97] est une extension d'OOA [SHL 88]. La méthode préconise une représentation des types géographiques par des pictogrammes. GeoOOA se focalise sur la modélisation des diagrammes de classes. La méthode permet de décrire les applications manipulant des données spatio-temporelles. Un formalisme a aussi été spécialement défini pour faciliter la description des structures de type réseau. Dans ce formalisme des objets de type lien relient des objets de type nœud.

2.2.5.1 *Modélisation des caractéristiques spatiales*

Il existe deux types de classes spécifiques aux informations géographiques :

- les géoclasses qui spécialisent une classe abstraite POINT, LINE, REGION ou RASTER,
- les classes réseaux qui spécialisent une classe abstraite NETWORK (réseau).

Chaque classe abstraite possède un ensemble d'attributs et d'opérations adaptées au type géographique associé. Chaque géoclasse et classe réseau sont spécifiées par un pictogramme représentant la géométrie de la classe qu'elle spécialise.

Les relations spatiales prédéfinies dans GeoOOA sont de type composé-composant. Elles portent sur les objets des géoclasses et sont de trois types : relation de couverture, relation de contenance et relation de partition.

2.2.5.2 *Modélisation des caractéristiques temporelles*

Selon la méthode GeoOOA, il est préférable que le concepteur modélise d'abord les aspects non temporels avant de les compléter par les aspects temporels. La méthode permet de spécifier des classes dites temporelles. Un objet d'une classe temporelle est associé à, ce que GeoOOA appelle, des estampilles temporelles.

Une estampille temporelle contient les valeurs des attributs d'un objet. Les estampilles correspondent à des sauvegardes des valeurs d'attributs des objets. Les estampilles temporelles d'un objet sont donc ordonnées dans le temps et constituent les versions successives de l'objet. Il est possible de spécifier quels attributs vont faire partie des estampilles d'un objet. Les classes temporelles sont décorées par un pictogramme spécifique.

Un lien de succession peut être spécifié entre une classe « Ancêtre » et une classe « Descendant » par une association particulière. Cette relation permet de définir qu'à la mort d'un objet de la classe « Ancêtre », certaines valeurs d'attributs sont transférées de l'objet de la classe « Ancêtre » vers un nouvel objet (ou plusieurs nouveaux objets) de la classe « Descendant ». Ce lien permet de spécifier les états par lesquels un objet peut passer.

2.2.5.3 *Les difficultés d'utilisation de la méthode GeoOOA*

Parmi les difficultés rencontrées lors de l'utilisation de cette méthode [PIN02] :

- l'absence de notation spécifique pour associer un objet d'une classe à une géométrie complexe ;
- l'insuffisance des relations topologiques décrites : il n'y a pas de relations d'intersection, de disjonction, etc.

2.2.6 La méthode OMEGA

La méthode OMEGA « Object Modeling for End-User Geographical Applications » [LBA 95], [LBA 97], [LBA 97b], [LBA 97c], [LBA 00], [LBA 00b], [LBA00c], [LBA 01], [LBA 02] et [LBA 02b] est la première méthode basée sur UML pour la conception d'applications géomatiques.

L'objectif principal de la méthode est d'apporter un support de modélisation d'applications géomatiques visuelles validées de manière incrémentale par

prototypage itératif avec l'utilisateur final et intégrant un formalisme à des niveaux de visibilité technique orientés vers le concepteur et des niveaux de visibilité utilisateur.

Aussi l'objectif de la méthode OMEGA est d'apporter une approche visuelle globale dès les phases d'analyse et conception en ciblant la conception d'applications géomatiques à l'aide de cet environnement d'interrogation visuelle.

Le formalisme utilisé est basé sur l'unification des méthodes orientées objets ainsi que sur les différents travaux sur les méthodes de conception pour les SIG.

OMEGA est une méthode visuelle qui donne, par exemple, la possibilité d'associer une « métaphore visuelle » à chaque classe.

Au niveau de la méthode OMEGA, quatre types de modèles sont définis :

- les modèles statiques : c'est la description des diagrammes de classes selon les différents points de vue (utilisateur, concepteur, etc.),
- les modèles dynamiques : décrivent les différents états d'un objet,
- les modèles fonctionnels : permettent la modélisation de traitements par des Diagrammes de Flux de Données,
- les modèles plates-formes : décrivent les plates-formes matérielles d'implantation.

Seulement les modèles statiques et les modèles dynamiques introduisent des notations vraiment spécifiques aux données géographiques.

2.2.6.1 *Modélisation des caractéristiques spatiales*

Trois types de références spatiales peuvent être associés à une classe :

- une référence à géométrie simple : tous les objets de la classe sont associés à une géométrie simple de même type (Point, Polyligne ou Polygone),
- une référence complexe : tous les objets de la classe sont associés à deux géométries simples,
- une référence alternative : chaque objet de la classe est associé à une seule géométrie simple mais le type de cette géométrie doit être choisi entre deux types définis pour la classe.

La référence spatiale détermine les types géographiques possibles d'un attribut nommé **obj** présent implicitement dans les classes. Chaque référence différente est représentée par un pictogramme. Les types de références spatiales utilisables dans OMEGA proviennent de la méthode de conception MODUL-R.

OMEGA utilise les contraintes topologiques de superposition et de voisinage définies dans MECOSIG. Les relations de type composé-composant sont utilisées pour exprimer l'agrégation spatiale.

2.2.6.2 *Les difficultés d'utilisation de la méthode OMEGA*

Parmi les difficultés rencontrées lors de l'utilisation de cette méthode [PIN02] :

- les aspects temporels n'existent que dans la description de l'évolution des objets dans le modèle dynamique,
- on ne peut pas exprimer des références complexes ou alternatives impliquant plus de deux types géographiques simples.

2.2.7 La méthode MADS

Le modèle MADS «Modélisation d'Applications à Données Spatio-temporelles» [PAR97], [PAR98], [PAR99], [PAR99b], [SOT 02], [SHU 03], [PAR04] et [PAR06] est un modèle conceptuel spatio-temporel, enrichi par un modèle dynamique : modèle des processus spatio-temporels. Cette extension est basée sur une analyse et une classification des processus spatio-temporels.

2.2.7.1 *Le modèle conceptuel MADS*

Le modèle conceptuel MADS a été guidé par les objectifs suivants [PAR97] :

- les schémas sont lisibles et faciles à comprendre,
- les trois dimensions structurelle, spatiale et temporelle sont orthogonales,
- des relations topologiques ou temporelles entre les entités peuvent être décrites d'une façon explicite,
- des concepts tels que l'agrégation spatiale ou temporelle, la généralisation/spécialisation et les mécanismes d'héritage associés sont définis de façon formelle,
- des champs spatiaux continus peuvent être décrits,
- la provision de types spatiaux génériques au-delà des types de base,
- les notations visuelles sont intuitives,
- dépasser le cadre de la modélisation cartographique.

MADS est basé sur un modèle combiné de type objet et relation. Il permet donc la modélisation des données classiques en offrant un ensemble de concepts tels que : objet, type d'objet, encapsulation des méthodes, association, type d'association, attribut (simple, complexe, monovalué, multivalué, obligatoire, facultatif et dérivé), méthode, généralisation, spécialisation, héritage, agrégation, raffinement et redéfinition.

MADS utilise aussi le principe de substituabilité, c'est-à-dire, tout rôle assumé par un sur-type peut être assumé par un sous-type.

Il gère des types d'objets spatiaux simples et complexes, des attributs temporels, des classes temporelles et des relations temporelles.

Les objets et les associations peuvent avoir une structure complexe grâce aux attributs complexes et multivalués. Le type d'association permet la modélisation explicite des liens n -aires entre objets. Le modèle permet de

représenter les champs continus dans l'espace grâce au concept d'attribut variable.

La spatialité et la temporalité sont visualisées dans les diagrammes par des pictogrammes, ce qui permet une appréhension visuelle immédiate et non ambiguë.

2.2.7.2 *Modélisation des caractéristiques spatiales*

2.2.7.2.1 Description de la spatialité

Le concept de spatialité englobe les notions d'emprise et de localisation. L'emprise décrit la forme associée à la représentation : de type point, ligne, surface, etc. Alors que la localisation permet de situer cette forme dans l'espace.

MADS fournit un ensemble de types abstraits spatiaux organisés en une hiérarchie de généralisation. La hiérarchie peut être étendue, selon les besoins de l'application, en créant des sous-types particuliers ou en regroupant plusieurs sous-types dans un même sur-type.

Ces types spatiaux sont : type spatial *Géo*, type spatial *Géosimple*, type spatial *Point*, type spatial *Ligne*, type spatial *Ligne orientée*, type spatial *Surface simple*, type spatial *Géocomposé*, type spatial *Semis*, type spatial *Graphe*, type spatial *Digraphe* et type spatial *Surface complexe*.

Chaque type spatial est associé à un ensemble de méthodes permettant de définir et de manipuler les instances du type.

Les types d'objets ainsi que les attributs peuvent être spatiaux. La spatialité d'un type d'objet est décrite en MADS par un attribut prédéfini, *géométrie*, dont le domaine de valeurs est un type abstrait spatial, soit prédéfini, soit défini explicitement pour l'application.

Un attribut spatial est un attribut simple, monovalué ou multivalué, dérivé ou non et dont le domaine de valeurs est un type abstrait spatial.

2.2.7.2.2 Les types d'associations spatiales

Les relations spatiales sont variées telles que les relations topologiques, les relations d'orientation, les relations métriques et les relations d'agrégation.

Dans MADS, un type d'association spatiale lie au moins deux objets spatiaux, et d'autres non-spatiaux, et dont la sémantique est une contrainte d'intégrité spatiale explicitement définie.

MADS offre deux types prédéfinis d'associations spatiales : les associations topologiques et les associations d'agrégation.

D'autres types d'associations spatiales peuvent être déclarés par l'utilisateur à l'aide des méthodes associées aux types abstraits spatiaux : les associations topologiques et les associations d'agrégation.

2.2.7.2.3 Les attributs variables

En MADS, un attribut est dit variable si sa valeur est définie par une fonction dont le domaine est l'ensemble d'éléments géométriques dans lesquels la géométrie de l'objet considéré a été décomposée.

2.2.7.2.4 Hiérarchie de généralisation

Le lien de généralisation peut relier des attributs spatiaux et non spatiaux. Par définition, les sous-types d'un type spatial sont spatiaux. A part le mécanisme d'héritage, il existe les mécanismes de raffinement et de réutilisation. Le mécanisme de raffinement permet de déclarer dans le sous-type un sous domaine pour un attribut hérité.

Aussi, le mécanisme de spécialisation multiple permet d'avoir plusieurs géométries différentes.

2.2.7.3 *Modélisation des caractéristiques temporelles*

Vu que la connaissance de l'évolution temporelle des données spatiales est primordiale [CLA 95], [CLA 02] et [PAR 97]. MADS offre les concepts permettant d'associer le temps à l'information suivant deux axes :

- l'estampillage : il est appliqué aux valeurs d'attributs, aux objets et aux associations. Il définit leur cycle de vie : création, désactivation, réactivation et destruction. Il peut être défini de manière absolue ou de manière relative,
- modélisation des liens dynamiques entre objets, des liens dont la sémantique a une composante temporelle inhérente.

2.2.7.3.1 L'approche temporelle de MADS

Selon l'approche MADS [SPA 98] et [PAR 97] :

- l'information modélisée peut être temporelle ou non,
- la temporalité peut être associée à tout concept du modèle,
- on ajoute au concept temporel conventionnel, la modélisation des liens dynamiques entre objets. Ces liens permettent de décrire des filiations entre objets, des règles de précedence temporelle et des situations d'agrégation particulières,
- d'autres principes sont adaptés par MADS tels que :
 - la représentation discrète et linéaire du temps, du passé au futur,
 - l'estampillage par le temps de validité,
 - la gestion de structures temporelles ponctuelles et de type intervalle, et plus généralement des éléments temporels,
 - le choix du calendrier est laissé à l'utilisateur,
 - la prise en compte de règles de cohérence sémantique, par une logique temporelle de premier ordre basée sur des intervalles.

2.2.7.3.2 Attributs temporels

Un attribut est dit temporel s'il est estampillé. Il sauvegarde l'évolution de ses différentes valeurs. La valeur d'un attribut est supposée indéfinie si elle n'appartient pas à sa période de validité. Tout attribut peut être temporel indépendamment de ses caractéristiques structurelles et spatiales. Si un attribut est temporel alors sa granularité doit être spécifiée. Les attributs non temporels peuvent être constants.

Pour les attributs multivalués temporels une période de validité est associée à chaque ensemble de valeurs de l'attribut. Quant aux attributs complexes, on peut attacher la temporalité à tout niveau. Un attribut temporel complexe peut avoir des composants temporels.

Il existe des contraintes temporelles sur les attributs complexes et leurs composants. Les contraintes les plus utilisées dans les applications sont : l'inclusion, la couverture et l'égalité.

2.2.7.3.3 Types d'objets temporels

Un type d'objet temporel garde trace du cycle de vie de ses instances, il est défini par les événements de création, de suspension, de réactivation et de destruction.

Le cycle de vie d'un objet temporel contient trois états : actif, suspendu et détruit. Chacun de ses états est associé à un élément temporel.

Un type d'objet temporel ou non peut avoir des attributs temporels et non temporels.

2.2.7.3.4 La généralisation

Le lien de généralisation est appliqué aux objets temporels et non temporels. Tous les sous-types d'un sur-type temporel sont obligatoirement temporels.

Le mécanisme de redéfinition est utile pour tous les cas où il s'agit de modéliser le cycle de vie des entités qui changent de classification.

2.2.7.3.5 Les associations temporelles

Un type d'association temporelle permet de garder trace du cycle de vie de ses instances. Un type d'association temporelle peut relier des types d'objets non temporels, et un type d'association non temporelle peut relier des types d'objets temporels. Comme pour les attributs, les cardinalités reliant les types d'objets aux types d'associations sont considérées comme statiques.

2.2.7.3.6 La modélisation temporelle dynamique

Vu l'importance des phénomènes dynamiques dans les applications spatio-temporelles, MADS propose quatre types d'associations dynamiques : association de transition, association de génération, association inter-temporelle et agrégation d'instantanés.

2.2.8 La méthode T-OMEGA

T-OMEGA [PIN01], [PIN02] et [PIN03] est une méthode de conception proposée pour la télégéomatique. Les travaux de recherches et industriels concernant cette méthode sont basés sur une extension d'UML.

La méthode T-OMEGA est composée :

- d'un formalisme étant une extension d'UML et destiné aux géomaticiens,
- d'une démarche méthodologique adaptant pour les besoins de la télégéomatique et la démarche de conception d'applications géomatiques définies dans OMEGA.

2.2.8.1 *Le formalisme T-OMEGA*

Tout comme OMEGA, la méthode T-OMEGA subdivise la modélisation des applications géomatiques en modèles statiques, fonctionnels, dynamiques et plates-formes, tout en y ajoutant les modèles de communication. Les types de modèles proposés par T-OMEGA et les types de diagrammes UML qui y sont rattachés sont les suivants :

Les modèles statiques : ces modèles correspondent à l'énumération des caractéristiques statiques des applications. Les modèles statiques sont décrits par les diagrammes de classes, les diagrammes d'objets et les diagrammes de cas d'utilisation d'UML. Les notations des diagrammes de classes sont étendues afin de faciliter la modélisation spatio-temporelle d'applications télégéomatiques au niveau conceptuel.

Le diagramme de classes T-OMEGA présente les concepts suivants :

- les classes applications : permettant de regrouper les diagrammes de classes par application, le concepteur pourra ainsi décrire la distribution des informations,
- les associations de connexion : elles permettent de décrire les relations entre les applications communicantes,
- les classes importées : permettant de décrire les objets accédés par des applications distantes. Le concepteur pourra ainsi définir les attributs et les opérations des objets accessibles entre les applications.

Les modèles de communication : il s'agit des modèles décrivant les interactions possibles entre les objets, ainsi que les interactions entre les applications. Ces modèles sont exprimés par les diagrammes de collaboration et les diagrammes de séquences.

Les modèles fonctionnels : les modèles fonctionnels présentent la description des opérations des applications. Ces modèles sont exprimés par les diagrammes d'activités. Dans ces diagrammes, on peut trouver des états début, finaux d'actions avec des transitions entre ces états, on trouve aussi des branchements conditionnels (par exemple 'si ... alors ... sinon ...'), comme on peut trouver des jonctions indiquant que plusieurs transitions mènent à un même état.

Les modèles dynamiques : les modèles dynamiques sont exprimés par les diagrammes états-transitions. Ces diagrammes présentent les différents états par lesquels passent les objets. Ces modèles peuvent prendre en compte les évolutions non déterministes des objets.

Les modèles plates-formes : les modèles plates-formes sont utilisés pour la description des différentes plates-formes matérielles supportant l'implémentation des objets. T-OMEGA intègre ces diagrammes pour permettre la spécification des choix matériels et des composants logiciels rattachés.

2.2.8.2 *La démarche méthodologique de T-OMEGA*

Les quatre phases de la démarche de T-OMEGA pour la conception d'applications télégéomatiques sont les suivants :

- phase d'analyse préalable : elle est consacrée à la mise en place du cadre du projet,
- phase de conception,
- phase de développement : elle intègre l'implémentation et la mise en place des applications,
- phase d'audit : elle est dédiée à l'évaluation et l'évolution des applications réalisées.

Pour chacune de ces phases, un document technique doit être émis afin de valider le niveau atteint dans le cycle de développement. Il est possible de revenir à une phase antérieure et même à des étapes au sein de cette phase afin d'y introduire des modifications.

2.2.8.3 *La représentation des types géographiques*

Le concepteur doit identifier tous les types d'entités géographiques externe et interne par rapport au fonctionnement de l'application qui doivent être modélisés. T-OMEGA offre la possibilité de modéliser chaque type d'entité par une classe d'objet et d'associer une géométrie à ces classes par l'ajout d'un attribut geo. Cet attribut possède un type géographique, et la valeur de cet attribut correspond à la géométrie associée à un objet. Afin de visualiser le type géographique associé à un attribut geo, un pictogramme est défini en entête de la classe. Nous appellerons une classe ayant un attribut geo, une classe

géographique. Une classe spécialisant une classe géographique est elle-même une classe géographique.

Le type géographique associé aux classes peut être vu comme un stéréotype d'UML. Les stéréotypes fournissent un mécanisme pour étendre le vocabulaire d'UML.

La classification par le concepteur des entités géographiques ne regroupe pas forcément, au sein d'une classe, des entités ayant le même type géographique de base. La méthode T-OMEGA permet, comme la méthode mère OMEGA, l'utilisation de deux opérations sur les types géographiques de base : l'opération ET et l'opération OU (exclusif).

2.2.8.4 *La représentation de la temporalité*

La méthode T-OMEGA tient compte au niveau conceptuel de la notion du temps. Dans T-OMEGA, une temporalité peut être associée à : un attribut, une classe et une relation.

Dans un diagramme de classes, un pictogramme spécifique est associé aux attributs pour spécifier qu'un historique des valeurs de l'attribut devra être conservé pour chaque objet. Un attribut ayant un tel pictogramme est appelé un attribut temporel. Plusieurs historiques pourront aussi être associés à l'attribut temporel d'un objet.

Si ce même pictogramme est placé au niveau du nom de la classe alors un historique devra être créé pour garder une trace (un historique) de tous les objets de la classe concernée ayant été instanciés depuis le démarrage de l'application. Cette trace d'objets devra contenir les historiques associés aux attributs temporels. Une telle classe est appelée classe temporelle.

Si ce même pictogramme est placé au niveau d'une association alors un historique devra être créé pour conserver les relations ayant existées entre les objets. Un tel type d'association est appelé une association temporelle. Plusieurs historiques pourront aussi être associés à une association temporelle.

2.2.8.5 *La visualisation de la géométrie des objets*

La description de la visualisation de la géométrie des objets peut être effectuée dès les premières étapes de conception. La façon dont seront présentées les informations géographiques influence directement la manière que l'utilisateur aura de percevoir l'application développée.

T-OMEGA propose la possibilité de définir des classes ayant un stéréotype <<map>>. On appelle ce type de classes, des classes de regroupement. Une classe de regroupement possède des relations d'agrégation avec classes géographiques. Cela indique qu'un objet d'une classe de regroupement est composé d'objets de classes géographiques.

Le concepteur doit pouvoir distinguer la géométrie des objets choisis pour le stockage des informations de celles utilisées pour l'affichage sur une carte. Lorsque le concepteur réalise la définition des classes géographiques, il faut donc qu'il ait la possibilité de spécifier deux types de géométries :

- la géométrie de stockage : c'est la géométrie choisie (au niveau conceptuel) pour le stockage de l'information. Il s'agit de la géométrie « stockée » par l'attribut geo des classes géographiques,
- la géométrie d'affichage : c'est la géométrie telle qu'elle apparaîtra sur une carte dans l'application finale. Cette géométrie n'est pas nécessaire pour les objets de classes géographiques modélisant des entités géographiques n'allant pas être affichées.

La géométrie d'affichage est généralement déterminée grâce à la géométrie de stockage. Les stéréotypes décrivant la géométrie d'affichage apparaissent à droite du stéréotype décrivant la géométrie de stockage, et sont tracés dans une couleur différente.

2.2.9 La méthode OMT-G

La méthode OMT-G [BOR99] et [BOR01] est une extension de OMT pour la description des applications manipulant des informations spatiales. En ce qui concerne les applications géomatiques, OMT-G distingue trois niveaux d'abstraction : le niveau monde réel, le niveau conceptuel et le niveau de l'implémentation.

OMT-G propose trois types spécifiques de diagrammes pour la modélisation conceptuelle :

- le diagramme de classes, comme en OMT permet de modéliser les différentes classes existantes,
- le diagramme de transformation permet de modéliser les conversions des géométries des objets,
- le diagramme de présentation permet de modéliser les attributs d'affichage à l'écran (ou sur papier) des données géographiques.

Grâce aux diagrammes cités ci dessus, OMT-G couvre la spécification d'un grand nombre d'aspects spatiaux (réseaux, évolution/conversion de la géométrie des objets, attributs de présentation, etc).

OMT-G considère deux groupes de classes : les classes conventionnelles et les classes géoréférencées. Chaque objet d'une classe géoréférencée représente un objet géographique du monde réel. Ainsi, chaque objet d'une classe géoréférencée est associé à une géométrie. Cet objet est soit :

- un « géo-objet » c'est-à-dire un objet ayant une géométrie de type point, ligne, polygone, nœud, ligne unidirectionnelle ou ligne bidirectionnelle,

- un « géo-champ » c'est-à-dire un objet ayant une géométrie de type réseau de triangles irréguliers (TIN), isolignes, ensemble de polygones adjacents, tessellations et nuages de points.

OMT-G donne la possibilité d'exprimer dans les diagrammes, sous la forme de relations entre classes des contraintes spatiales prédéfinies (contenir, couvrir, traverser, être devant, être à gauche de, être à droite de,...).

OMT-G introduit aussi la notion de généralisation conceptuelle.

2.2.10 La méthode GeoFrame

GeoFrame [JUG 98] se présente comme « une structure conceptuelle pour la modélisation orientée objets des données géographiques » [JUG 99]. GeoFrame étend le diagramme de classes d'UML et propose aussi un squelette de ce que l'on pourrait considérer comme étant une démarche méthodologique :

- en première étape, le concepteur regroupe les informations à modéliser par thèmes,
- en deuxième étape, pour chaque thème choisi, un diagramme de classes est réalisé. Le paquetage d'UML est utilisé pour regrouper les classes par thème. Le type général de géométrie associée à chaque classe est alors choisi,
- en dernière étape, le concepteur choisit précisément le type exact de géométrie associé à chaque classe (point, polygone,...).

2.2.10.1 Modélisation des caractéristiques spatiales

Une classe géographique spécialise une classe prédéfinie nommée GeographicObject ou une classe prédéfinie nommée GeographicField. Une classe GeographicObject peut être associée à une géométrie ayant comme type géographique : point, ligne, polygone ou complexe. Une classe spécialisant GeographicField peut être associée à une géométrie ayant comme type : grille de cellules, grille de points, polygones adjacents, réseau de triangles irréguliers, etc. L'association d'une géométrie à une classe est représentée par un pictogramme. Plusieurs types géographiques peuvent être associés à une classe.

2.2.10.2 Les difficultés d'utilisation de GeoFrame

Parmi les difficultés rencontrées lors de l'utilisation de cette méthode :

- on ne peut pas modéliser l'aspect temporel en utilisant GeoFrame,
- GeoFrame ne propose qu'un ensemble minimal de concepts et de notations pour la modélisation spatiale.

2.2.11 La méthode Geo-UML

La méthode Geo-UML [BED 99] est une extension des diagrammes de classes UML pour la définition des données spatio-temporelles. Elle peut être considérée comme la version orientée objet de la méthode MODUL-R.

2.2.11.1 *Modélisation des caractéristiques spatiales*

Geo-UML permet d'associer une géométrie à chaque classe d'objets. Les types géographiques associés aux classes sont représentés par des pictogrammes.

Les différentes géométries possibles sont :

- géométrie simple : les objets de classes sont associés à une et une seule primitive géométrique (point, ligne ou polygone),
- géométrie alternative : les objets de classes possèdent une ou l'autre des géométries proposées (deux géométries ou plus de façon mutuellement exclusive),
- géométrie multiple : chaque objet de classe possède autant de géométries qu'il y a de pictogrammes spécifiés pour la classe mais dont une seule est habituellement utilisée à la fois dans une représentation cartographique ou une requête spatiale,
- géométrie complexe : chaque objet de classe est associé en même temps à deux ou plusieurs géométries.

Geo-UML donne la possibilité de composer les pictogrammes à l'aide d'une multiplicité lorsque l'agrégat est composé de plusieurs géométries similaires.

Aussi, une notation est introduite pour permettre de spécifier qu'une géométrie d'une classe est « dérivée » (ou calculée) d'une géométrie d'une autre classe.

La seule relation prédéfinie dans GeoUML est la relation d'agrégation. Cette dernière peut être interprétée comme une agrégation spatiale.

2.2.11.2 *Modélisation des caractéristiques temporelles*

En Geo-UML, la temporalité est représentée par un intervalle (une période) ou par un instant (une date). Tout type de temporalité est représenté dans les classes par des pictogrammes.

Deux types principaux de temporalité sont modélisés grâce à Geo-UML :

- la temporalité d'existence qui s'applique à une classe et qui est délimitée par une « naissance » et une « mort ». Le concept d'existence ne peut être appliqué qu'aux classes pour lesquelles l'utilisateur désire gérer un historique d'objets directement dans son implémentation,
- la temporalité d'évolution qui s'applique aux changements d'état de l'objet. Associer une temporalité d'évolution à un attribut ou à une géométrie signifie que le système conserve un historique des valeurs prises par l'attribut ou par la géométrie d'un objet.

2.2.12 Le formalisme STER

STER (Spatio-Temporal Entity/Relation) [CHEN 76] est un formalisme fondé sur le formalisme Entité/Relation. Il est dédié à la conception de bases de données spatio-temporelles. Les concepts de spatialité du formalisme sont Point, Line et Region. Ce formalisme distingue la temporalité de l'objet modélisé de celle de l'objet informatique (transaction time). En revanche, ce formalisme ne différencie pas les concepts Instant et Période. En effet, les concepts existence time et transaction time sont traités comme des périodes puisqu'ils ont un début et une fin [MIR 06].

À la différence d'autres formalismes, STER différencie la spatialité et la temporalité conjointe de la spatio-temporalité. Il permet donc la modélisation de la spatio-temporalité et de la temporalité conjointe.

Aussi au niveau de ce formalisme les propriétés spatiales et temporelles peuvent renseigner les entités mais aussi les attributs de ces dernières. Les propriétés spatiales et temporelles peuvent annoter les relations entre entités [TRY 97], [TRY 99], [TRY 99b] et [TRY 00].

2.3 Les méthodes de conception pour les applications temps réel

2.3.1 Système temps réel

Les systèmes temps réel sont des systèmes dont les temps de calcul doivent être bornés par une valeur donnée. Cette valeur est imposée par l'application que doit réaliser le système et fait partie du cahier des charges à mettre en œuvre [JOR 09].

Un système temps réel est un système (application ou ensemble d'applications) informatique dont le fonctionnement est assujéti à l'évolution dynamique d'un procédé extérieur qui lui est connecté et dont il doit contrôler le comportement. Un système temps réel inclut généralement différents sous systèmes chacun pouvant avoir ses propres contraintes temporelles.

Un système temps réel n'est pas un système «rapide» mais un système qui satisfait des contraintes temporelles (les contraintes de temps dépendent de l'application et de l'environnement alors que la rapidité dépend de la technologie utilisée, celle du processeur par exemple).

Un système temps réel interagit avec un environnement extérieur souvent complexe et en évolution. Il doit pouvoir interagir avec différents types d'éléments matériels. Il doit aussi respecter des échéances temporelles et il doit garantir une fiabilité importante.

L'objectif des systèmes temps réel est la gestion efficace des données tout en respectant les contraintes temporelles qui leur sont imposées. La plupart des systèmes temps réel sont des contrôleurs devant assurer un bon temps de réaction entre l'action et la réaction du contrôle. Ils sont souvent multi-tâches, la plupart des contrôleurs temps réel doivent assurer un ensemble de tâches simultanément ce qui demande un ordonnancement garantissant les délais sur toutes les tâches [JOR 09]. Les systèmes temps réel fournissent à cet égard des algorithmes efficaces pour ordonnancer les tâches temps réel de manière à respecter leurs échéances.

La raison d'être des systèmes temps réel est que la fiabilité du système dépend non seulement de l'exactitude des traitements effectués, mais également du temps dans lequel ces résultats sont fournis.

La limite de ce type de système est son incapacité de pouvoir manipuler des grandes quantités d'informations.

Généralement, un système de contrôle temps réel est un système connecté à un procédé physique externe dont il doit contrôler et commander le comportement. Comme l'illustre la Figure 1, un système temps réel se décompose alors en sous systèmes qui sont : le système de contrôle et le procédé contrôlé [IDO 06] :

- le système de contrôle est principalement composé d'un ordinateur temps réel et d'interfaces d'entrées/sorties servant à la communication avec le procédé à contrôler,
- le procédé à contrôler est composé du procédé physique externe ainsi que de capteurs et d'actionneurs.

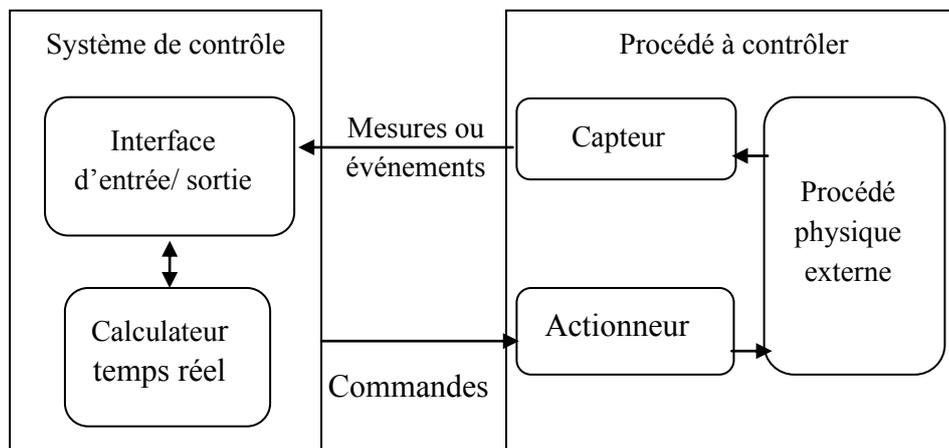


Figure 1. Architecture d'un système temps réel [IDO 06].

2.3.2 Caractéristiques des applications temps réel

Une application temps réel est une application pour laquelle les résultats produits ne doivent pas seulement être corrects en valeur mais également en temps, c'est-à-dire que le système doit respecter les contraintes temporelles spécifiées dans le cahier des charges. Une telle application est toujours en couplage fort avec son environnement et les contraintes temporelles peuvent s'exprimer sous diverses formes : temps de réponse à une sollicitation de l'environnement (simple, de bout en bout) et contraintes de débit à respecter (en entrée ou en sortie).

De nombreuses définitions ont été proposées pour clarifier la notion d'application temps réel. Cependant, comme les caractéristiques des systèmes temps réel sont très variées, aucune des définitions proposées n'est vraiment complète pour tenir compte de tous les domaines d'applications. Or toutes les applications temps réel ont en commun la prépondérance du facteur temps. Cette caractéristique fondamentale qui distingue globalement les applications temps réel des autres types d'applications informatiques (de gestion ou autres). En plus de l'existence de contraintes de temps, et selon les domaines d'applications, les applications temps réel doivent satisfaire d'autres contraintes primordiales, notamment la prédictibilité des comportements et la tolérance aux fautes.

La notion de « *criticité* » est utilisée comme critère pour pouvoir classer les applications temps réel selon la sévérité, en termes de coût engendré par le non respect des contraintes temporelles. Ainsi, on classe les applications temps réel en deux types : les applications à contraintes temporelles strictes, où le non-respect des contraintes de temps peut conduire à des défaillances avec des conséquences pouvant être graves et les applications à contraintes temporelles relatives, où le dépassement des échéances est considéré comme une faute bénigne. Il existe plusieurs types d'applications temps réel telles que le contrôle du trafic aérien, les applications militaires (gestion d'un champ de bataille), les applications de télécommunications, la gestion d'une centrale nucléaire, etc.

L'environnement d'une application temps réel possède sa propre dynamique. Cela signifie qu'il peut évoluer indépendamment de l'application logicielle. L'exécution de l'application et l'évolution de l'environnement peuvent être vues comme deux activités réalisées en parallèle et en constante interaction. Les contraintes temporelles de l'application sont dans la majorité des cas dérivées des propriétés physiques du système ainsi que de la fonction qu'il doit assurer [ROB 09].

2.3.3 Conception des applications temps réel

L'activité de conception des logiciels temps réel se découpe généralement selon deux points de vue complémentaires :

- la modélisation statique, qui consiste en une description systématique de tous les composants logiciels (données et fonctions) et de leurs interactions en terme de visibilité et de dépendance,
- la modélisation dynamique, qui se focalise sur la description de l'architecture temps réel du logiciel. L'objectif est de répondre aux exigences temporelles et de sécurité de fonctionnement du système.

On cite dans ce qui suit les principales méthodes de conception pour les applications temps réel.

2.3.3.1 UML

Le langage UML a été standardisé par l'OMG (Object Management Group). Il est maintenant largement utilisé dans les communautés scientifique et industrielle. Il est né de la mise en commun des trois plus importantes méthodes de modélisation orientées objet (la méthode Booch [BOO 92], la méthode OMT [RUM 91], et la méthode Object-Oriented Software Engineering (OOSE) [JAC 92]) du début des années 90 au sein d'un même langage [OMG 99] et [BOO 98].

Ces dernières années UML a connu de nombreuses extensions afin de faciliter la modélisation d'applications dans des domaines spécifiques tels que les systèmes distribués, les SIG [BRO 00], les systèmes multi-agents ou encore pour les documents hypermédias, etc.

Il est un langage graphique pour la visualisation, la spécification, la construction et la documentation des composants système. Il offre un standard pour la modélisation orientée objet [OMG 01], [OMG 01b], [OMG 02], [OMG 02b], [OMG 03], [OMG 05b] et [OMG 06].

Il existe neuf diagrammes composants UML : diagramme de classes, digramme d'objets, diagramme de cas d'utilisation, diagramme de séquences, diagramme de collaboration, diagramme d'états-transitions, diagramme d'activités, diagramme de composants et diagramme de déploiement.

La multitude des diagrammes UML, bien qu'elle permette la représentation des différentes vues du système, accentue la difficulté du processus de vérification et de validation. Les différents diagrammes dans un modèle UML se complètent et présentent plusieurs chevauchements sémantiques et syntaxiques entre eux. Ces chevauchements ne sont que partiellement pris en compte par les outils d'aide à la conception.

UML est un langage qui s'appuie sur le formalisme objet, formalisme dont le point fort réside dans son pouvoir de communication entre les analystes, les

concepteurs, les acteurs du domaine et les utilisateurs. Il a été conçu dans ce but. Ce sont ces différents diagrammes qui facilitent cette communication.

Bien qu'il possède une syntaxe bien définie basée sur le méta-modèle d'UML et complétée par des contraintes structurelles exprimées en OCL (Object Constraint Language [WAR 98]), la sémantique de ses notations reste ambiguë car elle est décrite en langage naturel.

UML est extensible à plusieurs domaines à l'aide de l'utilisation des stéréotypes. Un stéréotype permet de spécialiser un concept existant du méta-modèle d'UML. Un stéréotype est un moyen pour changer la sémantique d'un concept existant. De plus, il peut également posséder des valeurs marquées et des contraintes [IDO 06] et [BER 99].

En outre, le langage UML prévoit la possibilité d'associer à chaque stéréotype un pictogramme [OMG 99], [OMG 01], [OMG 05] et [OMG 06].

2.3.3.2 UML 2

UML 2 [OMG 03b], [OMG 03bc] a ajouté des métaclasse pour prendre en compte une forme simple du temps. UML définit deux types de données particuliers relatifs au temps qui peuvent être utilisées, en particulier, dans deux types de diagrammes : les diagrammes d'états et les diagrammes de séquences.

UML 2 a introduit les concepts suivants :

- le concept d'événement temporel (Time Event) : un événement temporel spécifie un point dans le temps représentant un moment absolu ou relatif du temps,
- la spécification des instants (Time Specification) et des durées (duration),
- le concept d'observation temporelle (observation),
- l'expression de contraintes temporelles.

2.3.3.3 UML-RT

UML-RT (UML for Real-Time) [SEL 98], [MAS 03], [GHE 06] et [KIR 03] s'appuie sur UML pour modéliser les systèmes temps réel. C'est une adaptation de la méthode ROOM (Real-Time Object_Oriented Modeling) [SEL 94].

UML-RT est basé sur l'utilisation de stéréotypes. Le stéréotype introduit dans UML-RT est la capsule pour prendre en compte les aspects contrôle des applications temps réel. UML-RT introduit aussi les notions de Port, de Protocole et de Connecteur [KHA 08] :

- capsules : correspondent aux concepts des acteurs ROOM. Ce sont des objets complexes, physiques, et peuvent être architecturaux distribués, qui interagissent avec leurs environnements par un ou plusieurs canal(aux) de communication basé(s) sur des signaux aux frontières de ces objets,

- port : implémente une interface d'une capsule. Chaque port d'une capsule joue un rôle particulier dans la collaboration entre cette capsule et les autres objets. Les ports sont associés à un protocole qui définit les flots d'information valides entre les ports connectés,
- connecteurs : vues abstraites des canaux de communication qui relient deux ou plusieurs ports.

UML-RT propose l'utilisation de Statecharts dans les diagrammes états-transitions, mais avec un stéréotype, nommé «chainState».

UML-RT introduit également une nouvelle représentation, le diagramme de structures, qui permet de décrire la structure d'une agrégation d'objets et surtout l'interconnexion de ces objets [FIS 01].

2.3.3.4 UML-MAST

UML-MAST est un profil UML fondé sur les concepts définis dans MAST (Modeling and Analysis Suite for Real Time Applications) [ESP 07], [MED 01] et [MED 05].

UML-MAST propose un cadre de modélisation temporel afin de réaliser une validation temporelle de type RMA (Rate Monotonic Analysis). Sachant que les techniques RMA sont basées sur les algorithmes d'ordonnancement temps réel permettant en outre de valider l'application d'un point de vue temps réel (respect des délais).

UML-MAST fournit des composantes pour modéliser les ressources matérielles (par exemple les réseaux), les composantes logiques de l'application et la charge de travail des activités et le choix des exigences.

UML-MAST définit un modèle capable de décrire le comportement temporel des nombreux systèmes temps réel, tel que les systèmes distribués.

La principale caractéristique de cette méthode est qu'elle permet de modéliser chaque aspect pertinent du système avec des composantes indépendantes une modélisation en temps réel. Par conséquent, il est possible de modéliser chaque logique de fonctionnement, les scénarios temps réel, etc, indépendamment de la plate-forme matérielle et les détails du fonctionnement du système.

2.3.3.5 TURTLE

TURTLE (Timed UML and RT-LOTOS Environment [APV 04]) est un profil UML [APV 01] dédié à la conception de systèmes temps réel sur une base formelle apportée par le langage RT-LOTOS (Real Time-LOTOS) [COUR 00]. Le profil TURTLE répond au besoin de doter la notation UML d'une sémantique formelle et d'adosser la validation formelle de modèles UML temps réel à des techniques éprouvées [APV 05].

La méthodologie supportée par TURTLE repose sur la réalisation d'une conception TURTLE constituée de diagrammes de classes et d'activités qui

décrivent la structuration du système et son comportement [APV 05b]. Ces dernières sont basées sur une extension des diagrammes de classes et d'activités d'UML 2.

Les diagrammes de classes ont été étendus par la définition de Tclasses qui sont autant de classes stéréotypées dotées de portes de communication par rendez-vous. Les associations entre Tclasses sont attribuées par des opérateurs de composition, ce qui permet d'exprimer explicitement les concepts de parallélisme, de séquence, de synchronisation, d'invocation et de préemption entre tâches décrites par des Tclasses.

Les diagrammes d'activités ont été étendus par des opérateurs temporels pour décrire un délai fixe et un délai non déterministe qui est associé au premier permet de travailler avec des intervalles temporels, et une offre limitée dans le temps [SAQ 04].

Le profil TURTLE est supporté par une chaîne d'outils qui va de l'édition de diagrammes à la validation d'un modèle par combinaison d'une exploration partielle sous forme de simulation et de vérification à base d'analyse d'accessibilité exploitée au moyen de « model checking » ou de vérification par abstraction [SAQ 00].

2.3.3.6 *SPT*

SPT (Scheduling Performance and Time [OMG05]) est un profil UML qui a été standardisé par l'OMG pour la modélisation des systèmes temps réel et il s'appuie sur UML 1.4.

SPT contient des constructions pour modéliser différents aspects des systèmes temps réel, tels que la temporisation, les ressources, la performance, l'ordonnancement, etc. Néanmoins, le profil se concentre plus sur les aspects syntaxiques des annotations, sans leur donner une sémantique formelle et sans faire le lien avec la sémantique de la partie fonctionnelle du modèle UML.

SPT ne considère qu'un temps métrique qui fait implicitement référence au temps physique. SPT introduit les concepts d'instant (instant) et de durée (duration), ainsi que ceux d'événements et de stimuli liés au temps. SPT modélise également des mécanismes temporels (clock et timer) et des services associés (démarrage, arrêt, suspension et reprise).

2.3.3.7 *SADT*

La méthode SADT (Structured Analysis and Design Technique) [ROS 77] est une méthode d'analyse qui couvre essentiellement la première partie du cycle de vie du logiciel. Elle permet non seulement de décrire les tâches du projet et leurs interactions, mais aussi de décrire le système que le projet vise à étudier, créer ou modifier, en mettant notamment en évidence les parties qui constituent le système, la finalité, le fonctionnement de chacune, et les interfaces entre les

diverses parties qui font qu'un système n'est pas une simple collection d'éléments indépendants, mais une organisation structurée de ceux-ci dans une finalité précise.

Le système étant défini, c'est-à-dire limité par sa frontière, on peut identifier :

- sa fonction,
- ses éléments constitutifs (sous-systèmes et composants),
- la matière d'œuvre sur laquelle s'exerce son action,
- les données d'entrée, ou de contrôle, qui provoquent ou modifient la mise en œuvre de la fonction.

Modéliser un système consiste à en donner une représentation qui énumère les quatre ensembles d'éléments ci-dessus en les distinguant les uns des autres, et en montrant les relations.

La méthode SADT propose une suite cohérente et hiérarchisée de diagrammes (datagrammes et actigrammes), obtenus par raffinements successifs (Figure 2) :

- un datagramme permet de représenter les données représentées par des boîtes, et montre les activités qui les créent ou les utilisent; celles-ci sont représentées par des flèches,
- un actigramme décrit l'enchaînement des activités, qui sont cette fois représentées par des boîtes ; les données qu'elles manipulent sont représentées par des flèches.

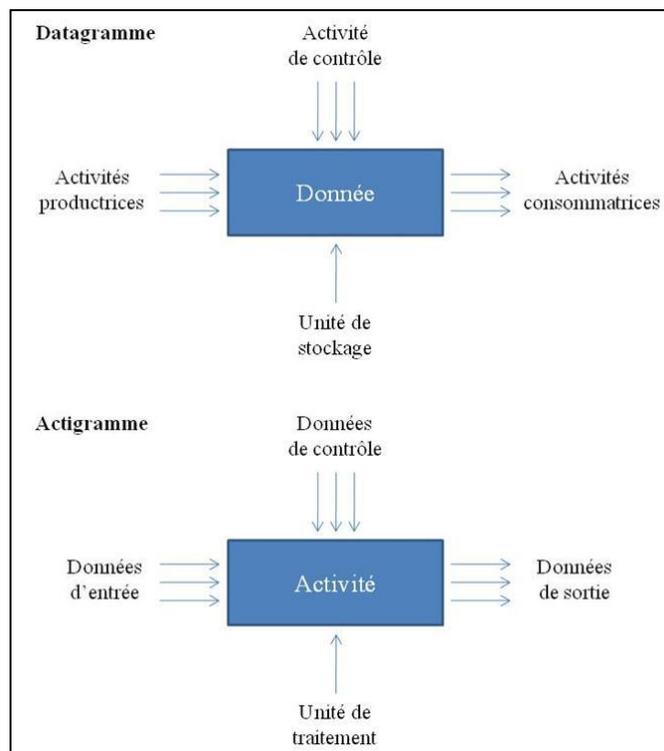


Figure 2. SADT: datagramme et actigramme.

Chaque boîte peut être décomposée en un diagramme plus détaillé. La vérification de chaque étape de décomposition peut se baser sur le respect des informations liées aux flèches : toutes les flèches d'un niveau doivent être conservées au niveau inférieur de décomposition.

Il faut contrôler systématiquement la cohérence entre données et traitements par contrôle croisé des actigrammes et des datagrammes.

2.3.3.8 SART

La méthode SART (Structured Analysis for Real Time) est une méthode de spécification des systèmes temps réel [WAR 85] et [WAR 86]. Elle permet d'obtenir une description complète et lisible des besoins. Elle permet de construire deux modèles d'une machine : le modèle des besoins et le modèle d'architecture :

- le modèle des besoins consiste en des DFD, des spécifications de traitements, des Diagrammes de Flots de Contrôle (DFC), des spécifications de contrôle, des spécifications de contraintes de temps et un dictionnaire de données. On construit les diagrammes par raffinements successifs. A chaque couche, on trouve un DFD, un DFC et une spécification de contrôle. Chaque processus utilisé dans les DFD et DFC fait l'objet d'une sous couche jusqu'à ce qu'un processus s'exprime de façon triviale en pseudo code dans une spécification de traitement,
- le modèle d'architecture a pour but d'attribuer les différents éléments du modèle des besoins (processus, spécifications de contrôles, données et contrôles) à des unités matérielles et logicielles qui composeront le système. Il intègre les éventuelles contraintes technologiques. Il est constitué de diagrammes de flots d'architecture, de spécifications de modules d'architecture, de diagrammes d'interconnexion d'architecture et d'un dictionnaire d'architecture.

SART est une méthode qui sépare les notions de données et de contrôles.

2.3.3.9 DARTS

La méthode DARTS (Design Approach For Real-Time System) est une méthode de conception particulièrement bien adaptée à la conception de systèmes temps réel [GOM 93]. C'est une méthode de conception fournissant des outils qui permet de décrire les interactions de tâches concurrentes en terme d'interfaces entre elles. Ces interfaces dépendent du fonctionnement du système, et peuvent être exprimées sous la forme de messages, événements ou modules de données.

Chaque tâche est représentée par un parallélogramme. Une tâche peut être soit une tâche matérielle, soit une tâche logicielle. L'activation des tâches matérielles est réalisée par un évènement alors que les tâches logicielles sont

activées par des synchronisations, boîtes aux lettres, ou rendez-vous par le biais d'une autre tâche, elle-même matérielle ou logicielle. Généralement les messages sont stockés dans une file de type FIFO (First In First Out).

Le Module De Données (MDD) est utilisé comme un système d'encapsulation et de protection de données. Le principe de ce module est de permettre à des tâches de partager des informations et d'y accéder en les protégeant contre des accès simultanés. Les données encapsulées ne peuvent être accédées qu'à travers l'interface. Le MDD doit garantir l'atomicité des opérations de cette interface les unes par rapport aux autres.

2.4 Les principaux langages de spécifications formelles

Les langages formels ont été proposés pour pallier le problème des langages semi-formels afin d'assurer un niveau aussi élevé que possible en matière de précision et de cohérence sémantique. Ils sont apparus dans les années 70 et se sont depuis développés en fonction des activités ciblées. Les langages formels sont basés sur des notations mathématiques qui fournissent un cadre précis et non ambigu pour la modélisation, ce qui permet de neutraliser tous les risques d'ambiguïté et d'incertitude, d'une part, et de permettre, d'autre part, de parvenir à un produit fini qui répond aux spécifications requises.

Les méthodes formelles utilisent des notations et des concepts spécifiques, qui génèrent souvent une faible lisibilité et une difficulté d'intégration dans les processus de développement et de certification. Et ce, contrairement aux méthodes semi-formelles qui sont plutôt privilégiées vu qu'elles permettent une vue synthétique, structurante et intuitive du système.

Les méthodes formelles consistent à intégrer une étape de vérification formelle dans les démarches de conception de logiciels. Elles reposent sur la formulation rigoureuse des besoins (spécification des besoins) sous forme de propriétés attendues du système et sur la modélisation du fonctionnement du système (spécification système).

2.4.1 Le langage Z

Le langage Z [ABR 77] est un langage de spécification formel à base de modèles. Il est basé sur la logique du premier ordre et sur la théorie des ensembles. En Z, le système est décrit sous forme d'une collection de variables et d'une liste d'opérations qui peuvent changer la valeur de ses variables. Les variables représentent l'état interne du système qui peut être accédé ou modifié uniquement par les opérations. Il y a donc encapsulation de l'état du système.

Z est enrichi par une construction modulaire très puissante : le schéma. Un schéma Z est une structure de spécification de données ou de traitements. Il est

caractérisé par un nom. Comme pour les variables et les constantes, le nom d'un schéma a un type dont les constituants sont ceux cités dans la partie de déclaration et qui vérifient les prédicats qui portent exclusivement sur les variables déclarées. La déclaration constitue le lexique des objets locaux du schéma qui est la définition de l'environnement d'application des prédicats de la deuxième partie. Les prédicats précisent les contraintes portant sur les variables ou la relation entre les états initiaux et finaux d'une opération.

Il existe deux types de schémas en Z : les schémas d'état et les schémas d'opération.

2.4.2 Le langage B

La méthode B [ABR 96], [ABR 96b] et [ABR00] est une méthode formelle avec preuves et raffinements, utilisée pour la réalisation de logiciels corrects par construction et pour la modélisation de systèmes dans leur environnement.

La méthode B utilise une notation fondée sur les concepts de la théorie des ensembles. C'est une notation uniformisée. Elle ne couvre pas les aspects initiaux de la programmation : le cahier des charges et la description des besoins. Elle porte sur l'ensemble des aspects du cycle de vie du logiciel tels que la spécification technique, la conception par raffinements successifs, l'architecture en couches et la génération du logiciel [IDA 05].

La méthode B est une façon de produire des logiciels en partant de spécifications abstraites des fonctionnalités des logiciels, puis, par des raffinements successifs, détailler de plus en plus cette spécification, jusqu'à obtenir une description qui soit directement implémentable, et complètement compatible avec la spécification initiale [LED 02].

Dans la méthode B un composant logiciel est une machine abstraite. Une machine abstraite contient des données (cachées) et propose à ses utilisateurs des opérations. On doit pouvoir comprendre ce que fait le composant. On doit argumenter sur sa cohérence et sur sa conformité au cahier des charges.

Une machine doit pouvoir être raffinée en une autre machine. On doit pouvoir passer graduellement du quoi au comment et on doit pouvoir être certain de la pertinence de la transformation.

2.5 Comparaison des méthodes existantes

La modélisation des données géographiques prises en temps réel est complexe, surtout que les méthodes existantes (géomatiques et temps réel) ne répondent pas suffisamment aux besoins du concepteur d'applications géomatiques temps réel.

Ces dernières présentent des avantages dans des domaines spécifiques, mais sont souvent limitées pour la conception des applications géomatiques temps réel, nous présentons dans le tableau 1 un récapitulatif des points forts et des points faibles de quelques méthodes citées précédemment [BEN 06] et [BEN 07].

Méthodes	Points forts	Points faibles
MODUL-R	Présentation graphique des données spatiales et des données temporelles.	Le schéma est difficilement lisible par un utilisateur non expert (très complexe). Les contraintes du temps réel ne sont pas prises en compte dans le modèle.
Géo2	Intègre l'approche orientée objet.	Le schéma complet des classes est complexe. Les contraintes du temps réel ne sont pas prises en compte dans le modèle.
MECOSIG	Présentation graphique des données spatiales, des contraintes topologiques et des relations topologiques.	On ne peut pas définir des objets à implantation multiple, absence de modèle dynamique et de représentation visuelle. Les contraintes du temps réel ne sont pas prises en compte dans le modèle.

Méthodes	Points forts	Points faibles
OMEGA	Présentation graphique des données spatiales et des données temporelles. Présentation dynamique des données. La méthode supporte un AGL.	On ne peut pas exprimer des références complexes ou alternatives impliquant plus de deux types géographiques simples. Les contraintes du temps réel ne sont pas prises en compte dans le modèle.
GeoOOA	La méthode préconise une représentation de types géographiques par des pictogrammes. Elle traite l'aspect temporel.	Absence de notation spécifique pour associer un objet d'une classe à une géométrie complexe, insuffisance des relations topologiques. Les contraintes du temps réel ne sont pas prises en compte dans le modèle.
MADS	Présentation graphique des données spatiales, des contraintes topologiques et des relations topologiques. Présentation graphique des données temporelles et des relations temporelles.	Les contraintes du temps réel ne sont pas prises en compte dans le modèle.
OMT-G	La méthode favorise le diagramme de classes, le diagramme de transformation et le diagramme de présentation.	Absence de temporalité.
GeoFrame	La méthode favorise le diagramme de classes.	GeoFrame ne propose qu'un ensemble minimal de concepts et de notations pour la modélisation spatiale.

Méthodes	Points forts	Points faibles
T-OMEGA	Conception des applications de télégéomatique. Présentation graphique des données spatiales et des données temporelles. Présentation dynamique des données. La méthode supporte un AGL.	Absence de concepts pour l'aspect temps réel.
UML	La conception de plusieurs diagrammes permettant d'englober le cycle de vie d'un logiciel.	Absence de représentation des concepts spatiaux.
UML-RT et TURTLE	Présentation des différentes contraintes spécifiques au temps réel.	Absence de représentation des concepts spatiaux.

Tableau 1. Récapitulatif de quelques méthodes de conception pour les applications géomatiques temps réel.

2.6 Conclusion

L'objectif de notre travail est de concevoir une nouvelle méthode de conception pour les applications géomatiques temps réel étant donné qu'il n'y a pas de méthode spécifique pour ces dernières.

Nous avons étudié les différentes méthodes existantes : les méthodes semi-formelles dédiées à la géomatique, les méthodes temps réel et les méthodes formelles. Nous avons constaté qu'elles présentaient des inconvénients et ne répondaient pas aux problèmes liés à la conception des applications temps réel manipulant des données spatio-temporelles.

A travers cette étude nous avons dégagé les concepts utiles pour réaliser un profil UML dédié aux applications manipulant des données spatio-temporelles. Le chapitre suivant va décrire les extensions d'UML que nous préconisons pour décrire, au niveau conceptuel, les aspects spatio-temporels des applications géomatiques temps réel.

PARTIE 2

CONTRIBUTIONS

CHAPITRE 3.

UNE METHODE DE CONCEPTION POUR LES APPLICATIONS GEOMANTIQUES TEMPS REEL

3.1 Introduction : Profil UML pour les applications géomatiques temps réel (GRT)

Dans ce chapitre, nous allons étudier un profil UML permettant de concevoir une application géomatique temps réel. Un profil UML est un mécanisme d'extension du méta-modèle du langage UML. Il permet d'enrichir le méta-modèle avec des concepts d'un domaine particulier sans en modifier le méta-modèle [ATK 00] et [MIR 06]. Le mécanisme de profil ([OMG 99], [OMG 03], [OMG 03b], [OMG 03c], [OMG 05], [OMG 06], [BLA 02], [BLA 05] et [MIR 06]) est effectué principalement grâce aux concepts UML de stéréotype, de valeur marquée et de contraintes.

Par ailleurs, le profil permet de personnaliser la notation UML pour les besoins d'un domaine d'application particulier.

A travers notre étude bibliographique nous avons montré qu'il existe des méthodes de conception dédiées aux applications géomatiques et des méthodes de conception dédiées aux applications temps réel et comme nous avons constaté que ces dernières ne répondent pas totalement aux besoins du concepteur des applications géomatiques temps réel, nous proposons un profil UML (Figure 3) pour permettre aux géomaticiens de modéliser ce type d'applications [BEN 09] et [BEN 09b].

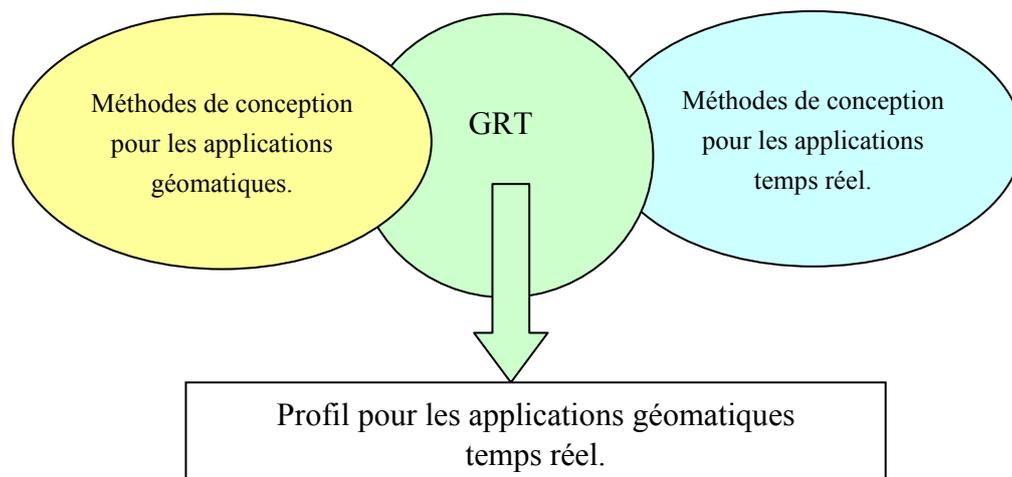


Figure 3. Présentation générale du profil GRT.

Le profil GRT a été proposé sur la base d'une extension de diagramme de classes, de diagramme d'activités, de diagramme d'état transition et l'ajout d'autres diagrammes tels que le diagramme de priorités (diagramme de cas d'utilisation avec succession d'actions) et le diagramme d'interventions.

3.2 Démarche méthodologique

Dans le contexte des méthodes orientées objets géomatiques, seules quelques unes proposent une démarche méthodologique telles que MECOSIG, OMEGA et T-OMEGA.

La démarche que nous proposons est basée sur une extension de celle de T-OMEGA.

Les différentes phases de la démarche de GRT pour la conception des applications géomatiques temps réel sont :

- la première phase est une phase d'analyse préalable qui est consacrée à la mise en place du cadre du projet. Au niveau de cette phase on élabore le programme de travail, des calendriers et des procédures de décisions. On définit aussi les propriétés du projet c'est-à-dire la fréquence d'acquisition minimale, la fréquence d'acquisition maximale, la fréquence de réaction minimale, la fréquence de réaction maximale, la vitesse de stockage, la vitesse d'affichage et le mode du projet synchrone ou asynchrone. De même on laisse le choix au concepteur de définir le type de temps utilisé (temps réel, temps mou et temps différé). A cette phase on définit une analyse des besoins et une étude de faisabilité. Au niveau de l'analyse des besoins nous permettons au concepteur de réaliser un digramme de priorités

dans lequel il définit ses acteurs avec les différents cas d'utilisation possibles en donnant un ordre de priorité pour chacun,

- la deuxième phase est la phase de conception. Toutes les spécifications de cette phase sont réalisées au niveau conceptuel. A ce niveau, le concepteur va décrire d'une manière détaillée les modèles statiques, dynamiques, fonctionnels et de communications,
- la troisième phase est la phase de développement. Nous avons réalisé un atelier génie logiciel permettant de générer du code qui permet d'aider le développeur à effectuer son développement,
- la quatrième phase est la phase d'audit.

A l'issue de chacune des phases, un document technique doit être émis afin de valider le niveau atteint dans le cycle de développement.

3.3 Types d'objets à modéliser

A l'aide de notre profil, nous modélisons des applications géomatiques temps réel, de ce fait nous utilisons des objets spatiaux qui varient en fonction du temps. Rappelons que la notion d'objet spatial est liée à l'espace. Les espaces les plus couramment utilisés sont de dimension deux ou trois, mais il n'est pas exclu d'augmenter le nombre de dimensions. Il existe donc plusieurs types d'objets spatiaux : des objets spatiaux à deux dimensions dans l'espace cartésien, des objets spatiaux à deux dimensions dans le globe terrestre, des objets spatiaux à trois dimensions dans l'espace cartésien, des objets spatiaux à trois dimensions dans le globe terrestre, etc.

Dans notre problématique nous nous sommes intéressée à la représentation de l'espace des données en deux dimensions (2D), une extension de la représentation des données en trois dimensions (3D) pourrait faire l'objet de travaux ultérieurs.

3.4 Présentation des différents modèles du formalisme étendu

L'extension d'UML que nous proposons permettra aux géomaticiens de modéliser des applications géomatiques temps réel. Au niveau de cette partie nous définirons les différents diagrammes utilisés dans notre formalisme tels que : le diagramme de classes, le diagramme de priorités, le diagramme d'activités, le diagramme d'interventions et le diagramme d'état transition.

3.4.1 Le diagramme de classes de GRT

3.4.1.1 Définition du diagramme de classes de GRT

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Une classe décrit un ensemble d'objets, une association décrit un ensemble de liens. Le diagramme de classes est alors un modèle permettant de décrire de manière abstraite et générale les liens entre objets.

Nous avons effectué une extension du diagramme de classes d'UML. Il met en œuvre principalement les concepts UML de Classe, d'Attribut et d'Opération, les concepts indispensables pour la description des classes, d'Association, d'Agrégation Partagée et d'Agrégation de composition, de Généralisation et de Paquetages.

Pour chaque classe, on définit une liste d'attributs en précisant leurs types, leurs visibilité (publique, protégée et privée), on précise aussi si l'attribut est statique ou non. On peut aussi définir l'un ou les attributs clés.

On définit pour chaque classe une liste de méthodes (opérations). Pour chaque méthode on peut définir une liste de paramètres. On peut préciser si la méthode est statique ou non, si elle est abstraite ou non, de même on peut définir sa visibilité.

La Figure 4 présente une classe définie à l'aide de diagramme de classes de GRT, dans laquelle nous avons défini un ensemble d'attributs : un attribut1 clé qui est public de type entier, un attribut2 qui est statique de type chaîne de caractères, un attribut3 qui est protégé de type caractère et un attribut4 qui est privé de type booléen. Nous avons aussi défini des méthodes : une méthode1 qui retourne un entier et elle possède deux paramètres, une méthode2 publique et qui ne retourne rien et une méthode3 qui est statique.

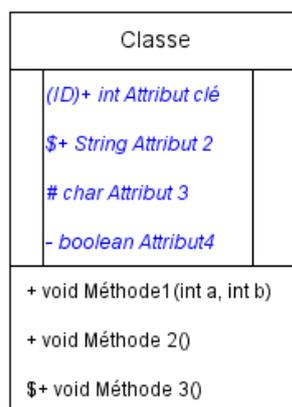


Figure 4. Présentation d'un exemple de classe définie par le diagramme de classes de GRT.

Le diagramme de classes de GRT indique tous les concepts de T-OMEGA (les classes applications, les classes importées et les associations de connexion), ainsi que les différents concepts géométriques et temporels définis dans la méthode MADS avec des extensions spécifiques aux applications géomatiques temps réel.

Dans le cas où le concepteur manipule des classes importées le concepteur est appelé à indiquer la classe source et la couleur qui identifie la connexion (Figure 5).

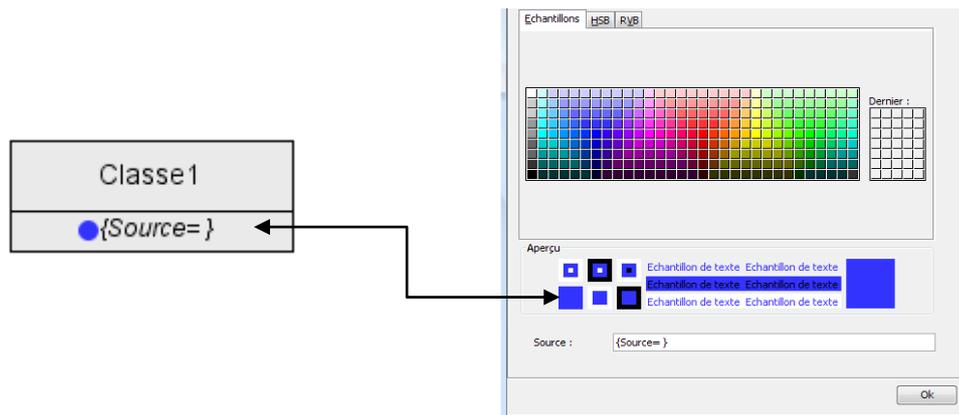


Figure 5. Présentation d'une classe importée au niveau du diagramme de classes de GRT.

La représentation des types géographiques est un concept primordial dans la phase de modélisation des applications géomatiques. Au niveau de ce diagramme, le concepteur identifie les différents types d'entités propres au fonctionnement interne de l'application.

On peut définir des entités de spatialité primitive, de temporalité primitive, de spatialité et temporalité conjointe ainsi qu'à celles de spatialité multiple et de spatialité alternative.

3.4.1.1.1 Les géométries adaptées

Au niveau de notre profil le concepteur doit pouvoir distinguer entre la géométrie de stockage et la géométrie d'affichage. Lors de la création d'une classe, on laisse le choix au concepteur de définir le type de géométrie utilisé, il peut même définir les deux types de géométries.

La géométrie de stockage est la géométrie pour le stockage de l'information spatiale, tandis que la géométrie d'affichage c'est la géométrie qui apparaîtra sur une carte dans l'application finale [PIN 02].

Nous proposons deux stéréotypes un pour la géométrie de stockage et un autre pour la géométrie d'affichage. Les stéréotypes décrivant la géométrie de

stockage apparaissent à droite du stéréotype décrivant la géométrie d’affichage, et sont tracés par une couleur de fond différente pour qu’on puisse les distinguer.

Dans l’exemple de la Figure 6 une ville se présente sur la carte comme étant un point (géométrie d’affichage), mais on la stocke dans la base comme une surface simple (géométrie de stockage).

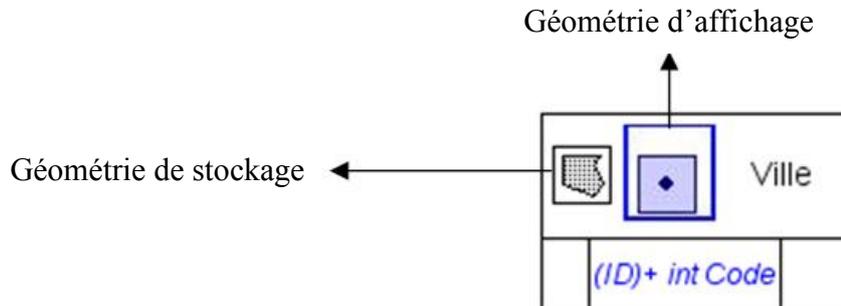


Figure 6. Exemple de spécification de géométrie de stockage et d’affichage.

3.4.1.1.2 Les types géographiques

Le profil doit offrir comme dans la plupart des méthodes de conception pour les SIG la possibilité d’associer une géométrie aux classes d’objets et aux attributs. Nous avons ajouté un attribut *geo* aux classes et la valeur de cet attribut correspond à la géométrie associée à un objet. Nous avons utilisé les différents types géographiques définis dans MADS ([PAR97], [PAR98] et [PAR99]), tels que *Géo*, *Géo simple*, *Géo composé*, *surface simple*, *surface composée*, *ligne*, *graphe*, *ligne orientée*, *digraphe*, *point* et *semis*. Afin de visualiser clairement pour une classe donnée, le type géographique associé à l’attribut *geo*, un pictogramme est défini en entête de la classe. Chaque type géographique peut être visuellement représenté par un stéréotype.

Le concept UML de stéréotype permet d’étendre et de préciser la sémantique de l’élément de modélisation. C’est un concept qui s’applique à tous les éléments de modélisation du langage UML [OMG 99], [OMG 01], [OMG 05] et [OMG 06].

Nous avons adopté l’élément de modélisation stéréotype pour bien exprimer la spatialité et la temporalité. Le langage UML prévoit la possibilité d’associer à chaque stéréotype un pictogramme. Nous offrons ainsi la possibilité d’associer un pictogramme à l’élément de modélisation stéréotype introduisant par là un langage visuel. Au niveau de notre profil les stéréotypes permettent de regrouper les classes d’objets selon leurs types géographiques et temporels.

Nous avons aussi défini des attributs spatiaux à l’aide des pictogrammes pour bien exprimer leurs spatialités.

Nous avons opté par la démarche de conception de stéréotypes traitée en T-OMEGA. Nous rappelons qu'au niveau de cette démarche nous définissons tout d'abord l'ensemble T des types à utiliser, puis nous définissons les stéréotypes, enfin nous définissons les différents opérateurs à utiliser pour définir la combinaison de types géographiques qui sont les opérateurs « ET » et « OU ».

Nous avons tout d'abord défini T comme étant l'ensemble de types de base existants dans le formalisme GRT :

$T = \{\text{Géo, Géo simple, Géo composé, surface simple, surface composée, ligne, graphe, ligne orientée, digraphe, point, semis}\}$.

L'ensemble des stéréotypes représentant graphiquement les différents types de T est représenté dans les Tableaux 2 et 3.

Type	Géo simple	Surface simple	Ligne	Point
Stéréotype				

Tableau 2. Présentation des types géométriques simples.

Type	Géo	Géo composé	Surface composée	Graphe	Ligne orientée	Digraphe	Semis
Stéréotype							

Tableau 3. Présentation des types géométriques complexes.

Les exemples des Figures 7 et 8 expriment la manipulation des stéréotypes au niveau d'une classe. Ils expriment le fait que la classe Terrain est une classe spatiale simple qui possède une géométrie de stockage de type Géo alors que la classe Route est une classe spatiale qui possède une géométrie de stockage de type graphe et possède un attribut spatial « Portion-route » de type ligne.



Figure 7. Représentation d'une classe spatiale.

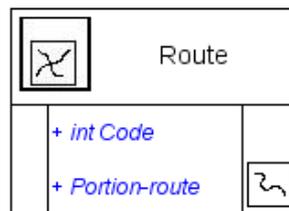


Figure 8. Représentation d'une classe spatiale avec un attribut spatial.

De même le concepteur peut faire une combinaison de types géographiques. Afin de définir les types géographiques combinés, nous avons utilisé les deux opérations «ET» et «OU» traités en T-OMEGA sur l'ensemble T des différents types géographiques cités précédemment. La définition des types géographiques combinés se fait par combinaison graphique de pictogrammes. L'exemple de la Figure 9 exprime que la classe *Bâtiment* ait un type géographique combiné, en se basant sur le fait que le type de sa spatialité est soit Géo simple, soit Géo composé.

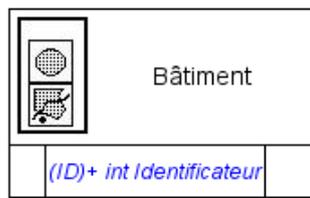


Figure 9. Représentation d'une classe spatiale en utilisant la combinaison de types géographiques.

Aussi au niveau de notre profil le concepteur peut définir les types géographiques combinés pour les deux types de géométrie traités : la géométrie de stockage et la géométrie d'affichage.

L'exemple de la Figure 10 exprime que la classe *Bâtiment* ait une géométrie de stockage et une géométrie d'affichage. La géométrie de stockage est soit Géo simple, soit Géo composé alors que la géométrie d'affichage est de type point et surface.

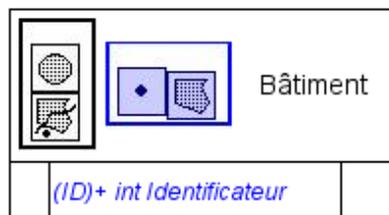


Figure 10. Représentation d'une classe spatiale en utilisant une géométrie d'affichage et une géométrie de stockage.

Nous donnons aussi le choix au concepteur de définir des associations spatiales. Les différentes associations topologiques sont : adjacence, disjonction, égalité, croisement, recouvrement et inclusion. Nous avons également défini un ensemble de stéréotypes (STsp) pour chaque type d'association spatiale citée précédemment (Tableaux 4 et 5).

Association	Adjacence	Disjonction	Egalité
Pictogramme STsp			

Tableau 4. Pictogrammes spécifiques aux associations spatiales (1).

Association	Croisement	Recouvrement	Inclusion
Pictogramme STsp			

Tableau 5. Pictogrammes spécifiques aux associations spatiales (2).

Dans l'exemple de la Figure 11 l'association entre les deux classes géographiques Région et Ville est une association spatiale de type inclusion. Elle exprime le fait que la spatialité de la classe Ville est incluse dans la spatialité de la classe Région.

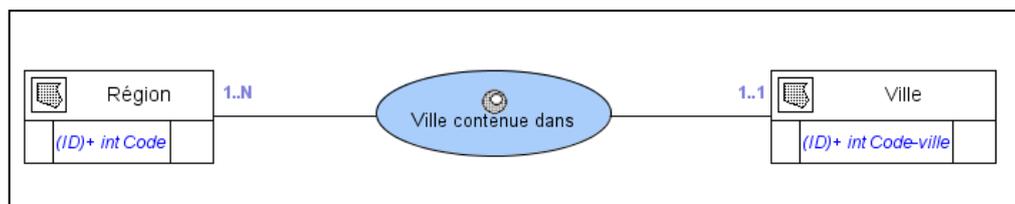


Figure 11. Représentation d'une association spatiale de type inclusion.

3.4.1.1.3 La temporalité

L'historique et la prévision sont les deux apports concrets de l'intégration temporelle pour la communauté des utilisateurs de SIG. La volonté de conservation de la trace de l'évolution d'une base de données est apparue naturellement avec la progression qualitative et quantitative des applications. La dynamique temporelle influence et enrichit le domaine spécifique des SIG [VID 94].

La dimension temps a été introduite dans les SIG car l'organisation de l'espace géographique évolue au cours du temps sous l'action de différents processus. Nous devons analyser les objets sur lesquels s'appuient ces processus et appréhender comment ils se transforment, se déforment ou se déplacent. Dans les SIG, on trouve plusieurs types d'entités spatiales : il y a les entités spatiales aux contours fixes pour lesquelles le problème de temps n'existe pas, les entités spatiales modifiables, déformables ou transformables pour lesquelles il faudrait tenir compte des déformations au cours du temps. Seulement, la gestion des changements d'états des entités transformables pose des problèmes, car il s'agit de gérer à la fois la variabilité des attributs, celle de la géométrie et de la topologie, et la filiation des identifiants. Il faut considérer le temps comme une variable active qui transforme les objets dont la filiation doit être analysée [CLA 95].

Un événement temporel est défini comme une action dans le temps où l'un ou plusieurs des attributs d'une entité évoluent notablement.

Un système d'information temporel est caractérisé par des intervalles temporels. La représentation d'un phénomène temporel dépend de son échelle d'appréciation et de son intervalle temporel de référence. Le choix des intervalles temporels doit autoriser un dynamisme relatif de l'application pour garantir le maintien de son intérêt.

Il faut distinguer la notion du temps réel qui représente la position dans le temps d'un événement du monde réel et du temps transactionnel qui définit la date de la prise en compte de cet événement dans la base de données.

Chaque entité est caractérisée par une date d'apparition et une date de disparition. Ainsi, les différents états passés de l'entité peuvent être conservés dans la base de données et restent visuellement actifs. Nous devons donc savoir pour chaque objet et association son cycle de vie : quand il a été créé, suspendu, réactivé et supprimé.

Plusieurs méthodes de conception des applications géomatiques proposent des moyens pour spécifier dans les diagrammes de classes que les évolutions des attributs des objets seront conservées par le système [PIN 02].

Dans notre formalisme nous avons utilisé les différents concepts de temporalité définis dans MADS et dans T-OMEGA. La temporalité peut être associée à tout concept du modèle. Une classe, un attribut et une association peuvent être temporels ou non. De même la spatialité peut être temporelle.

La Figure 12 exprime que la classe `Bâtiment` est temporelle de même sa spatialité est temporelle. Alors que La Figure 13 exprime que la classe `ville` contient un attribut `Aéroport` qui est temporelle c'est-à-dire on veut mémoriser, sa date de création, son activation, son réactivation et sa destruction.

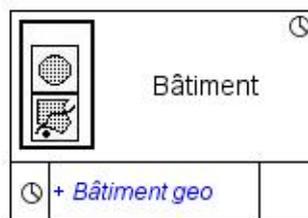


Figure 12. Représentation d'une classe temporelle ayant une spatialité temporelle.



Figure 13. Représentation d'une classe spatiale possédant un attribut spatio-temporel.

Au niveau de ce diagramme, nous pouvons définir des associations temporelles entre classes et des associations inter-temporelles (Tableau 6).

L'association inter-temporelle est le pendant en temporel des associations topologiques en spatial. Si on associe deux types d'objet par ce type d'association alors on spécifie une contrainte d'intégrité temporelle liant les cycles de vie des deux types d'objet [PAR 97].

Type	Association temporelle	Association inter-temporelle
Stéréotype		

Tableau 6. Pictogrammes spécifiques aux associations temporelles.

Etant donné que nous manipulons des applications géomatiques temps réel nous avons constaté l'importance des fréquences d'acquisition de données et des fréquences de réaction, au niveau de ce diagramme on laisse le choix au concepteur de définir la fréquence d'acquisition de données et la fréquence de réaction. De même le concepteur peut définir la vitesse de stockage, la vitesse d'affichage et le mode de travail entamé : synchrone ou asynchrone. Nous

avons défini ces attributs comme étant des attributs spéciaux qui sont spécifiques au profil GRT et qui peuvent être créés et initialisés dès la création du projet (Figure 14). Ces attributs peuvent être uniformes pour toutes les classes du projet mais le concepteur peut faire des changements de valeurs pour une classe donnée en respectant les bornes minimales et maximales.

Figure 14. Présentation des attributs spéciaux.

La Figure 15 exprime que la classe `Véhicule` qui est spatiale de type point possède un attribut simple (l'identificateur du véhicule) et des attributs spéciaux spécifiques.

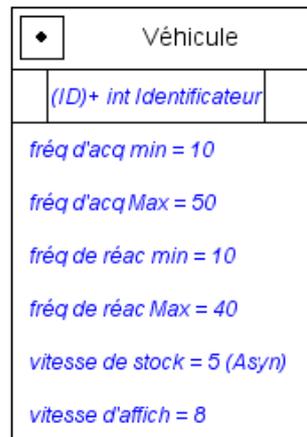


Figure 15. Exemple de représentation d'une classe avec des attributs spéciaux spécifiques au profil GRT.

3.4.1.2 Règles du modèle de classes de GRT

Dans la littérature le temps est considéré selon trois types ([SOU 98]) :

- temps absolu : une référence de temps est dite absolue lorsqu'elle est liée à un instant du calendrier,
- temps relatif : une référence temporelle est dite relative lorsqu'elle est liée à une autre référence temporelle,

- temps périodique : le temps périodique est défini comme une suite infinie d'instantanés ou d'intervalles où la distance temporelle entre deux éléments est exprimée par une durée.

Au niveau de notre formalisme, on traite deux catégories de temps : le temps d'acquisition de données et le temps de réaction. Au niveau de chaque catégorie on précise trois types de temps : le Temps réel, le Temps réel Mou et le Temps réel Différé, on aura donc neuf cas possibles.

- **Temps d'acquisition de données :**

Au niveau de ce mode on peut préciser le type de temps en fonction de la fréquence d'acquisition de données (Fra). On laisse le choix au concepteur de préciser :

- un seuil pour le Temps réel d'acquisition « Tra » tel que $Tra < seuil$,
- un deuxième seuil « seuil1 » pour le Temps réel Différé d'acquisition « TrDa » tel que $TrDa > seuil1$,
- un troisième seuil pour le Temps réel Mou d'acquisition « Trma » qui sera automatiquement compris entre Tra et TrDa.

- **Temps de réaction :**

Pour le temps de réaction aussi on doit spécifier les trois types de temps : Temps réel de réaction « Trr », Temps réel Différé de réaction « TrDr » et Temps réel Mou de réaction « TrMr ». On laisse le choix au concepteur pour spécifier les seuils car ils dépendent du type de l'application. Au delà des seuils précisés la réaction sera inutile.

3.4.1.3 Intégration au méta-modèle d'UML

Un méta-modèle décrit de manière formelle les éléments de modélisation et la syntaxe et la sémantique du formalisme qui permet de les manipuler. Il définit la sémantique du langage UML pour représenter des modèles UML. Le gain d'abstraction induit par la construction d'un méta-modèle facilite l'identification d'éventuelles incohérences et encourage la généralité. Le méta-modèle d'UML sert de description de référence pour la construction d'outils et pour le partage de modèles entre outils différents.

L'OMG maintient les différentes versions du méta-modèle d'UML.

Le méta-modèle proposé par [OMG 01] décrit les notations UML à l'aide de diagrammes de classes.

Le langage UML est représenté par trois paquetages (Figure 16) [PIN 02] :

- le paquetage « Fondement » décrivant la structure des éléments statiques du langage UML (Figure 17). Il est composé d'un paquetage « Noyau » qui désigne la description principale, d'un paquetage « Mécanismes d'extension » décrivant les mécanismes d'extension d'UML tels que les stéréotypes et d'un paquetage « Type de données » référençant les différents types de données de bases,

- le paquetage « Eléments du Comportement » décrivant la structure des éléments liés au comportement des modèles du langage UML (Figure 18). Il est composé des paquetages: « Cas d'utilisation », « Collaborations », « Machine d'Etats », « Graphe d'activités » et « Comportement commun »,
- le paquetage « Gestionnaire de Modèles » décrivant comment les éléments du langage sont organisés pour composer des modèles, des paquetages ou des sous-systèmes.

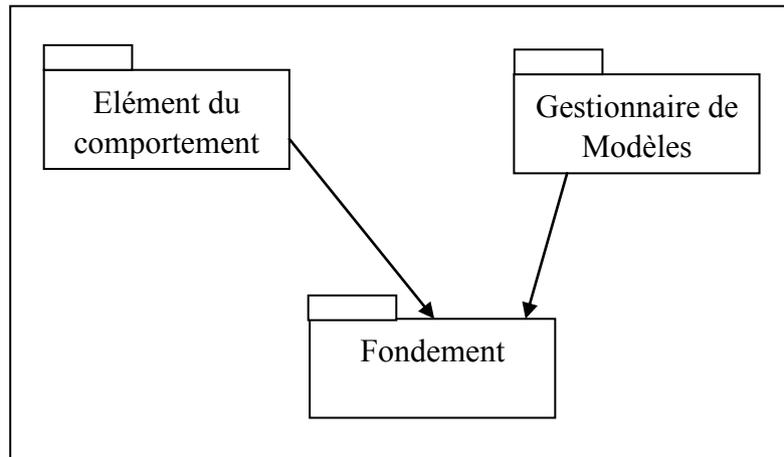


Figure 16. Méta-Modèle UML : Paquetages Principaux.

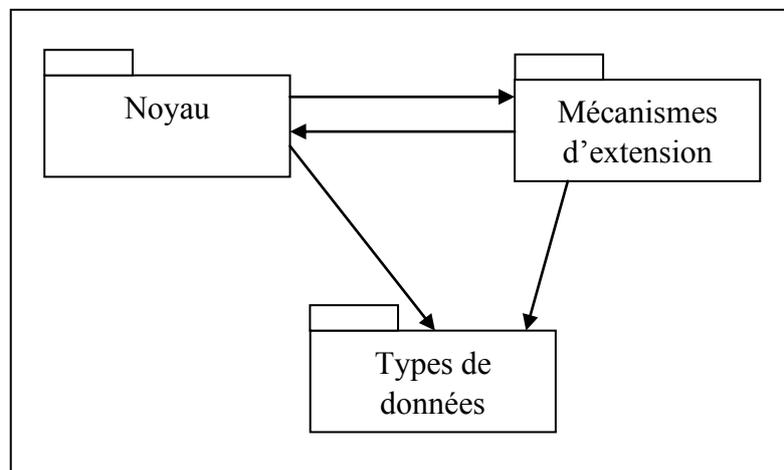
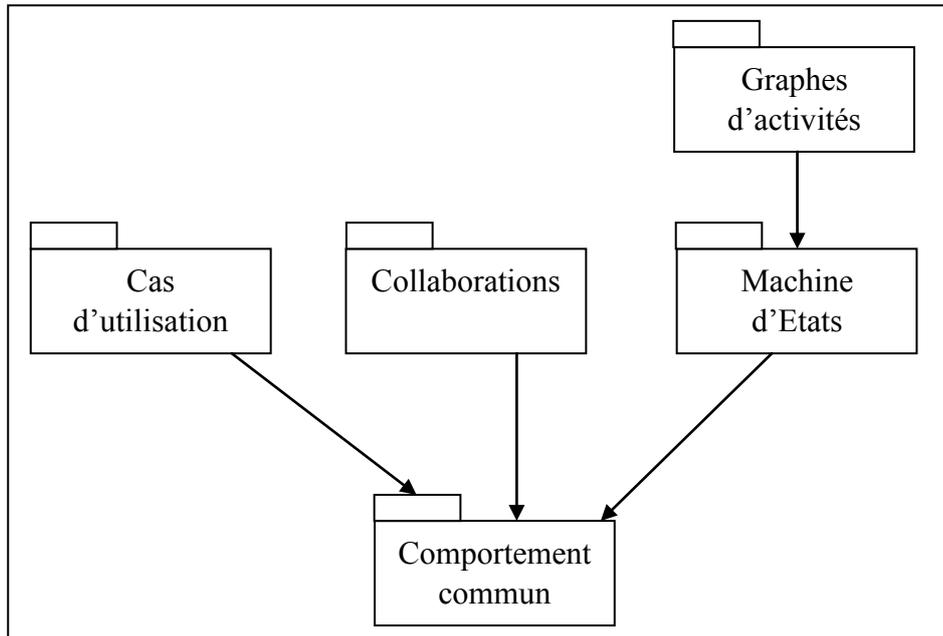


Figure 17. Méta-Modèle UML : Paquetage « Fondement ».



**Figure 18. Méta-Modèle UML :Paquetage
« Eléments du Comportement ».**

Comme indiqué précédemment, notre profil se base sur une extension du langage UML. Nous avons présenté une extension du diagramme de classes d'UML pour pouvoir répondre aux besoins du géomaticien lors de la conception d'une application géomatique temps réel. Nous allons présenter comment l'extension que nous proposons peut s'intégrer dans le méta-modèle d'UML. Les Figures 19 et 20 présentent l'extension réalisée, nous avons schématisé en gris les concepts de GRT.

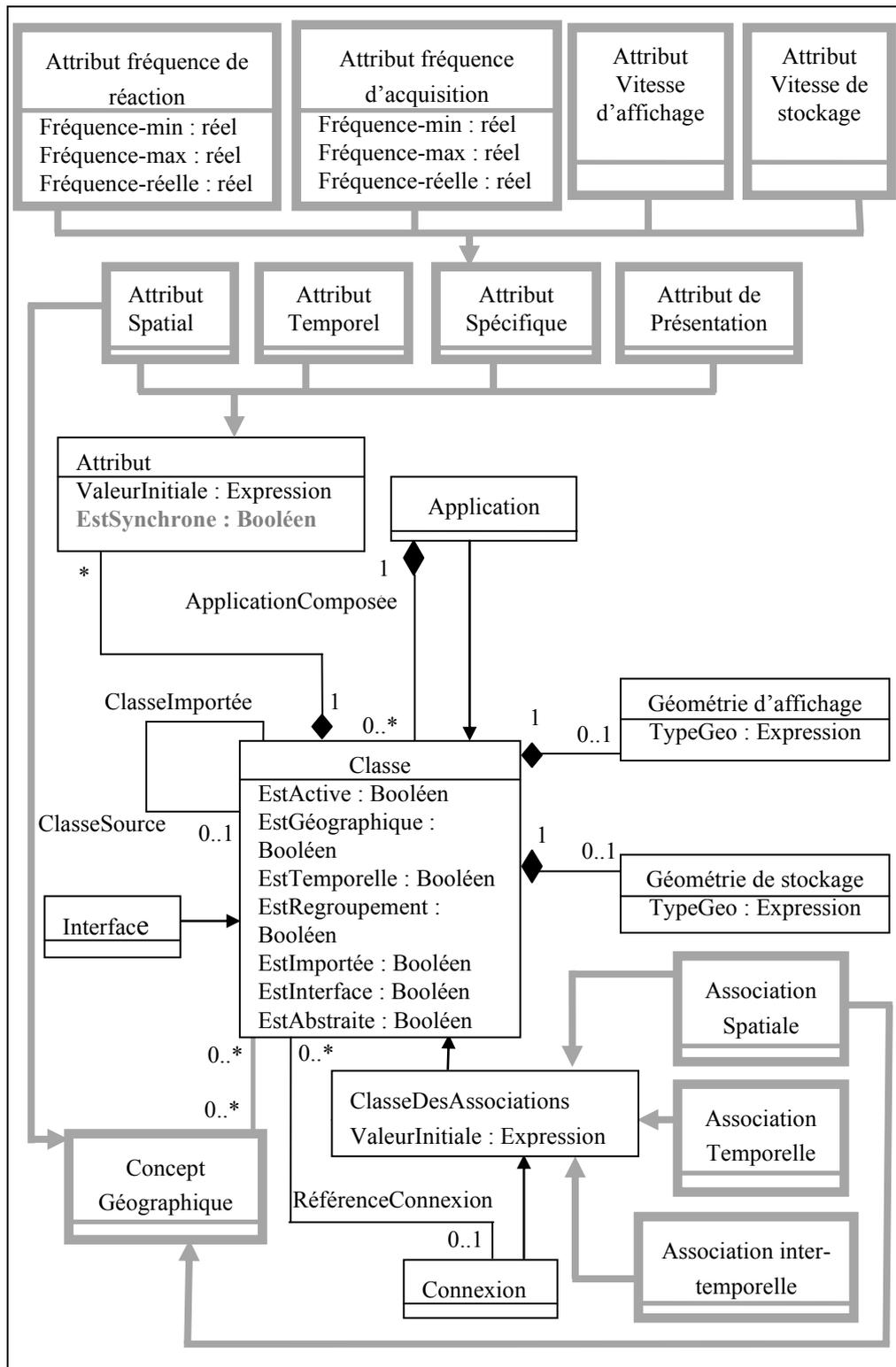


Figure 19. Extension du Méta-Modèle UML : Diagramme de classes.

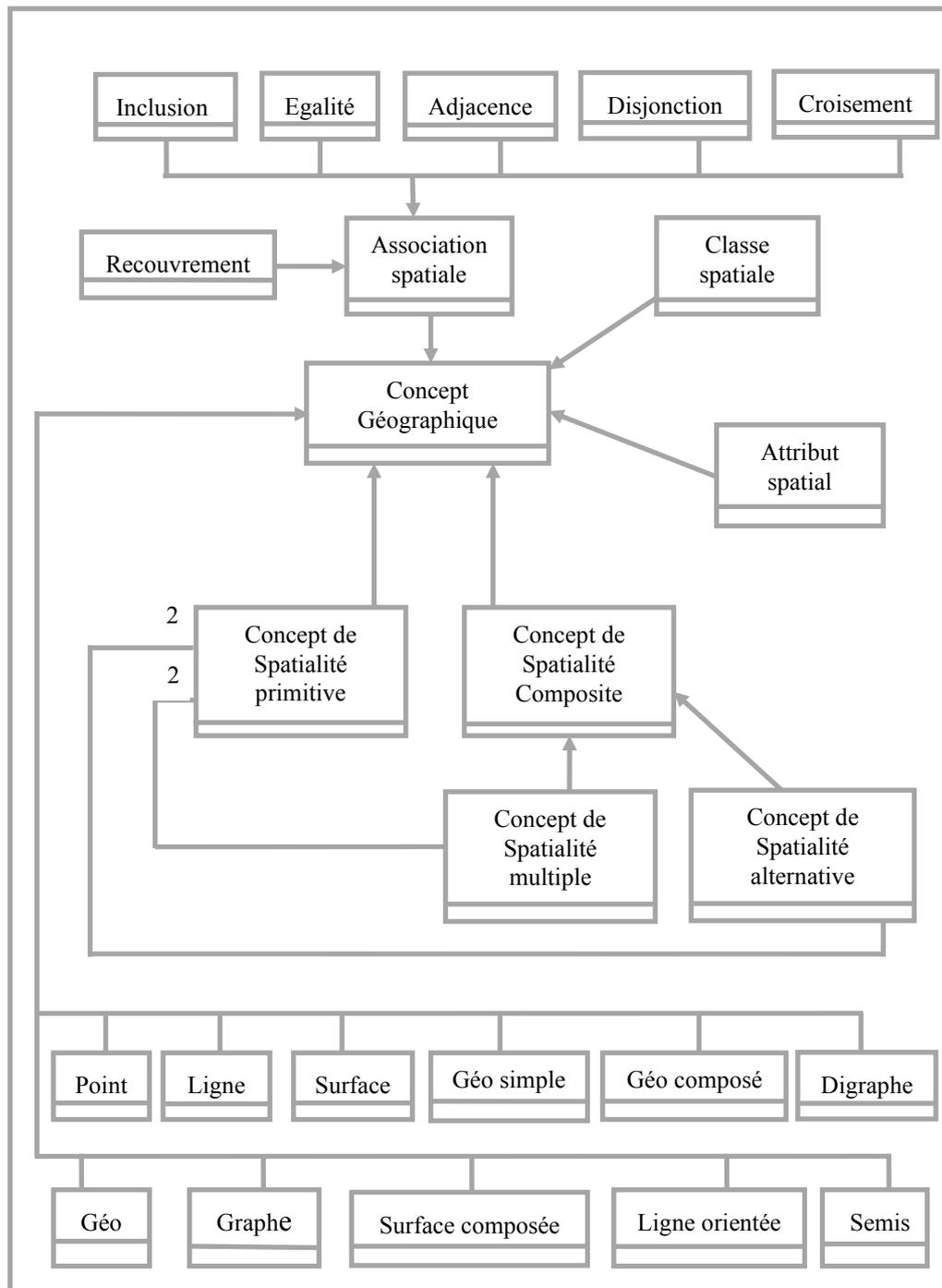


Figure 20. Extension du Méta-Modèle UML : Paquetage Géomatique.

3.4.2 Le diagramme de priorités

3.4.2.1 Définition du diagramme de priorités

En UML classique, les actions décrites dans le diagramme de cas d'utilisation peuvent arriver à n'importe quel moment. Dans les applications temps réel les actions doivent respecter un ordre bien précis, c'est la raison pour laquelle nous avons défini le diagramme de priorités que nous avons considéré comme une extension du diagramme de cas d'utilisation. C'est un diagramme de cas d'utilisation avec succession d'actions.

Ce diagramme est réalisé dès la phase d'analyse de besoins lors de la réalisation d'une application. Ce diagramme apporte une solution au problème de la détermination et de la compréhension des besoins. Il décrit le comportement du système de point de vue utilisateur sous la forme d'actions et de réactions. Il sert à structurer les besoins des utilisateurs et les objectifs correspondants du système. Lors de la modélisation des données pour des applications géomatiques temps réel, le concepteur a besoin de spécifier l'ordre d'importance de certaines actions par rapport à d'autres, c'est pour cette raison au niveau de ce diagramme on doit tenir compte des différentes priorités de certaines actions et réactions par rapport à d'autres.

Un diagramme de priorités est une extension du diagramme de cas d'utilisation d'UML dont des priorités sont affectées aux cas d'utilisations étant donné que les cas d'utilisations permettent aux utilisateurs de structurer leurs besoins. Ils les obligent à définir la manière dont ils voudraient interagir avec le système, à préciser quelles informations ils entendent échanger et à décrire ce qui doit être fait pour obtenir le résultat escompté. On peut donc définir toutes les relations existantes en diagramme de cas d'utilisation entre les cas d'utilisations telles que la relation d'inclusion «include», la relation d'extension «extend» et la relation généralisation/spécialisation [MUL 00].

Les acteurs sont des êtres humains ou autres présentés par des stéréotypes. Nous pouvons aussi définir un lien d'héritage entre les acteurs.

Aussi au niveau du diagramme de priorités on peut partitionner le modèle en paquetages.

La Figure 21 exprime un exemple de diagramme de priorités selon le profil GRT, sachant que les nombres représentent l'ordre de successions des différents cas d'utilisation.

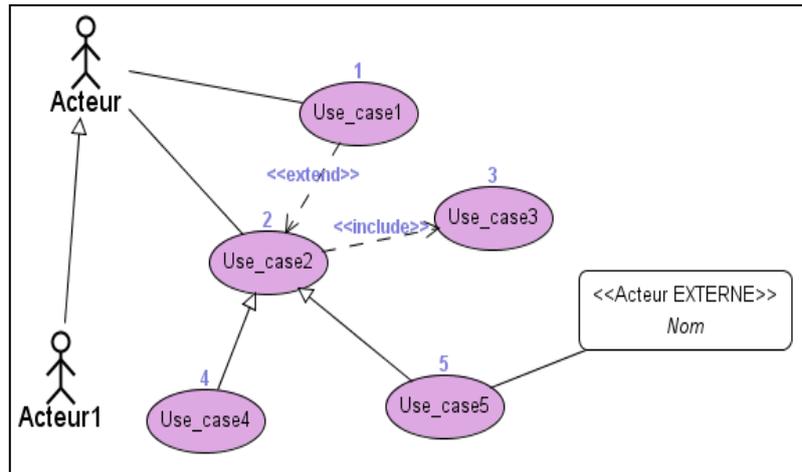


Figure 21. Présentation d'un diagramme de priorités selon GRT.

3.4.2.2 Règles du modèle de priorités

Au niveau de ce diagramme le concepteur aura la possibilité d'ajouter :

- des acteurs humains ou autres modélisés par des stéréotypes,
- des cas d'utilisations avec des valeurs de priorités qui sont successives et sans redondances, pour enlever toutes sortes d'ambiguïtés.

3.4.2.3 Intégration au méta-modèle d'UML étendu

Nous allons présenter l'extension du méta-modèle d'UML que nous proposons pour intégrer les différentes notions ajoutées dans le méta-modèle. La Figure 22 présente l'extension réalisée pour le diagramme de priorités, nous avons schématisé en gris les concepts de GRT. Elle présente donc les modifications du langage d'UML induite par nos travaux. Par ailleurs, la sémantique de notre extension a été présentée précédemment.

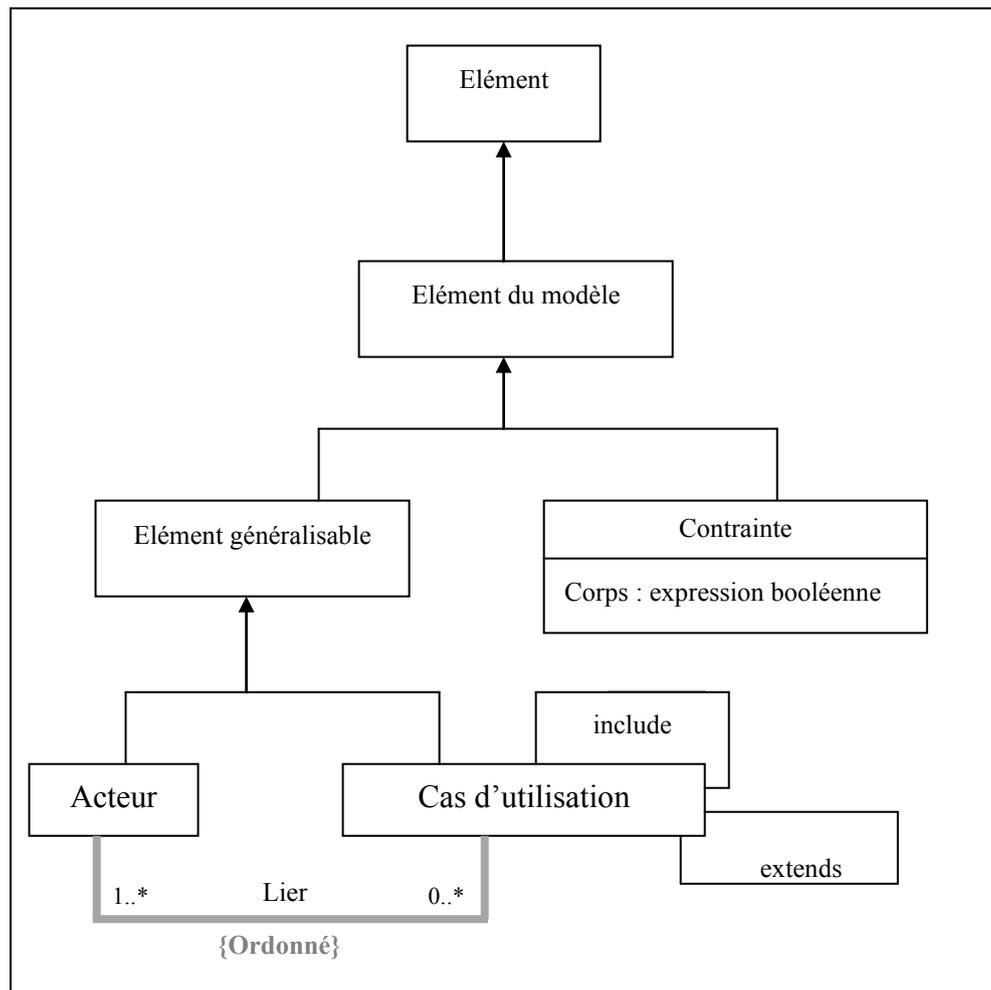


Figure 22. Extension du Méta-Modèle UML : Diagramme de priorités.

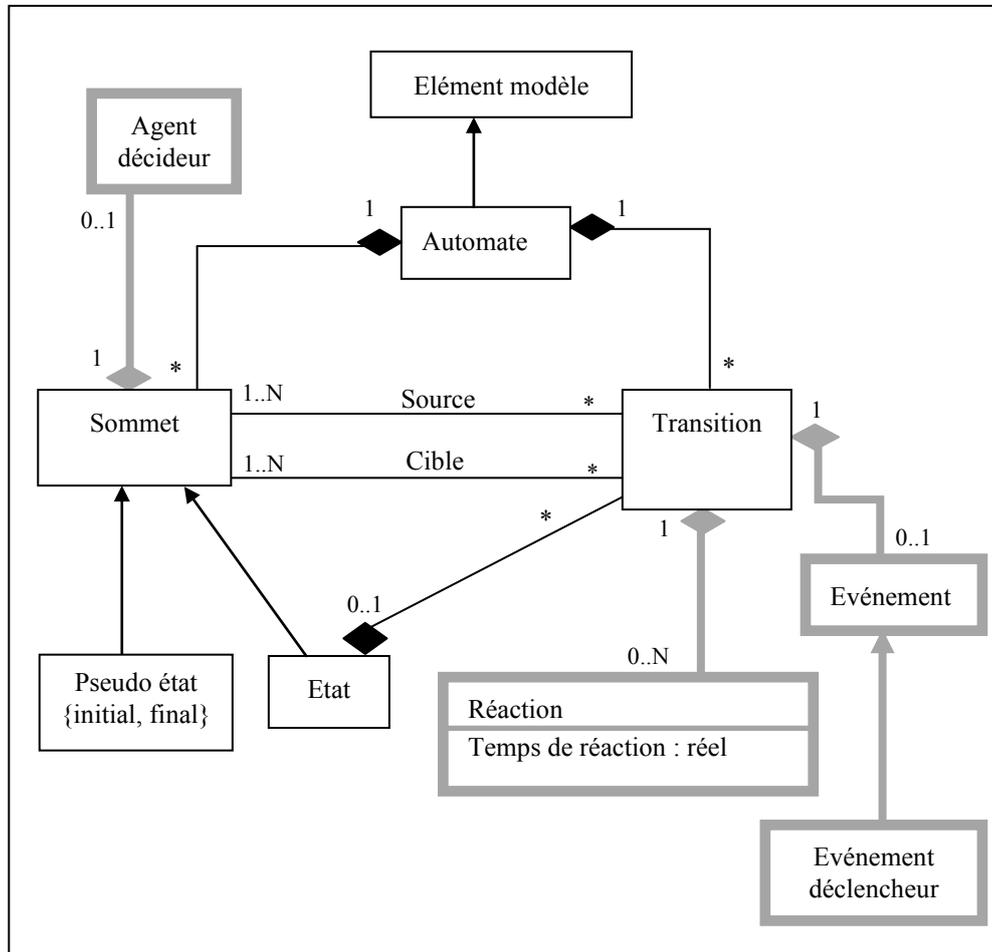
3.4.3 Le diagramme d'état transition de GRT

3.4.3.1 Définition du diagramme d'état transition

Les diagrammes d'états contiennent des états et des transitions entre les états. Ce type de diagrammes permet de modéliser les différents états par lesquels passe un objet. Une condition peut s'appliquer à une transition sous la forme d'un garde. Nous avons effectué une extension du diagramme d'état transition d'UML pour que les contraintes temporelles soient compatibles avec les applications géomatiques temps réel. Au niveau de ce diagramme nous pouvons définir un événement déclencheur, un agent décideur pour prendre la décision surtout s'il s'agit d'une réaction en temps réel, d'un état terminal et du temps de réaction.

3.4.3.2 Intégration au méta-modèle d'UML étendu

Nous allons présenter l'extension du méta-modèle d'UML que nous proposons pour intégrer les différentes notions ajoutées. La Figure 23 présente l'extension réalisée pour le diagramme d'état transition de GRT, nous avons schématisé en gris les concepts de GRT.



**Figure 23. Extension du Méta Modèle UML :
Diagramme d'état transition.**

3.4.4 Le diagramme d'activités de GRT

3.4.4.1 Définition du diagramme d'activités de GRT

Les diagrammes d'activités peuvent être vus comme des cas particuliers des diagrammes d'états transitions. Les Diagrammes d'activités et d'états transitions viennent compléter la panoplie des diagrammes permettant d'exprimer la dynamique du système. Ce sont deux diagrammes

complémentaires étant donné que le premier met l'accent sur les états du système alors que le second focalise l'intérêt du concepteur sur les transitions entre états [MUL 00].

D'une manière générale, les diagrammes d'activités modélisent le déroulement d'un procédé ou d'un algorithme. Ils représentent l'état de l'exécution d'un mécanisme, sous la forme d'un déroulement d'étapes regroupées séquentiellement dans des branches parallèles de flot de contrôle. Il représente la disjonction et la synchronisation d'événements mais aussi la dichotomie des transitions pouvant faire évoluer un objet dans un état vers deux états distincts. Une activité est un automate à deux états dont le franchissement de la transition entre ces derniers est conditionné par une fin d'activité.

Au niveau de notre profil nous avons effectué une extension au niveau de ce diagramme, pour répondre aux besoins du concepteur lors de la modélisation d'une application géomatique temps réel, nous avons utilisé les concepts utilisés au niveau du profil UML temps réel TURTLE [APV 04], [APV 05] et [APV 05b].

Ce diagramme permet de conduire une analyse du comportement du système à un haut niveau d'abstraction. Il supporte des opérateurs de choix, de parallélisme et de synchronisation.

Nous pouvons modéliser au niveau de ce diagramme les différents objets géographiques qui sont des instances de classes conçues au niveau du diagramme de classes du profil GRT.

Nous avons aussi mis l'accent sur les opérateurs temps réel par l'intégration de symboles pour la présentation des deux types de temps : le temps déterministe et le temps non déterministe (Tableau 7).

	Temps déterministe	Temps non déterministe
Symboles		

Tableau 7. Représentation de la notion du temps au niveau du diagramme d'activités GRT.

Nous définissons un temps déterministe quand la durée est bien précise, mais lorsqu'elle est aléatoire nous définissons un temps non déterministe.

Nous pouvons spécifier pour chaque type de temps la durée maximale, en précisant l'unité du temps utilisée (secondes, minutes, heures et jours).

Nous pouvons même définir pour chaque acteur un couloir d'activités dans lequel nous spécifions ses différentes activités et l'enchaînement entre elles.

La Figure 24 exprime un exemple de digramme d'activités selon le profil GRT.

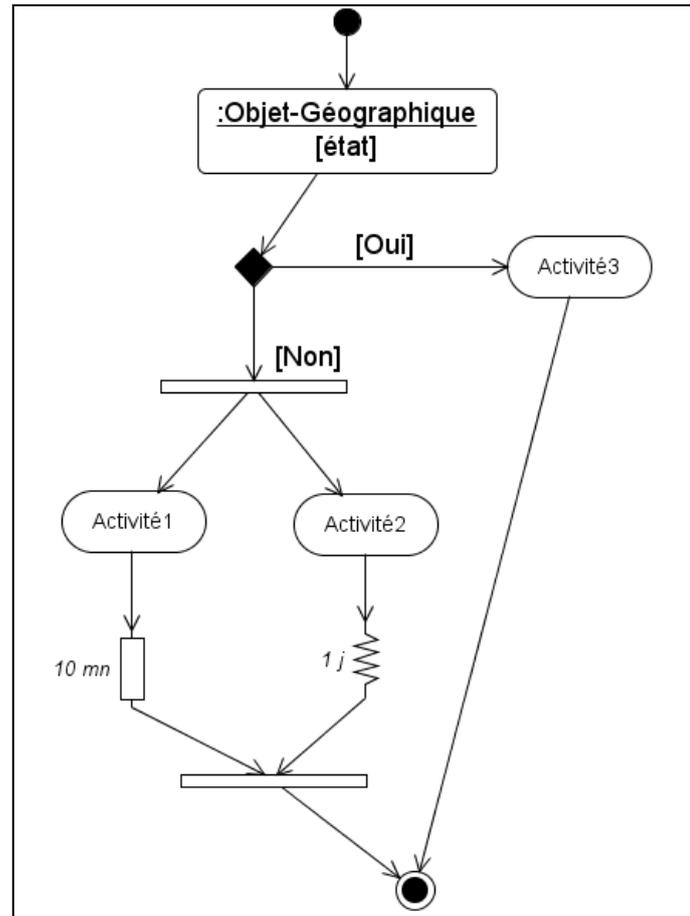


Figure 24. Présentation d'un diagramme d'activités selon GRT.

3.4.4.2 Règles du modèle d'activités de GRT

Au niveau de ce diagramme le concepteur aura la possibilité de spécifier des contraintes temporelles utiles pour des applications temps réel. Il peut définir deux types de délais : les délais déterministes et les délais non déterministes, en précisant la valeur maximale pour chacun.

Nous devons assurer la cohérence entre le diagramme de classes et le diagramme d'activités, de ce fait les objets utilisés au niveau de ce diagramme doivent être des instances des classes réalisées au niveau du diagramme de classes de GRT.

3.4.4.3 Intégration au méta-modèle d'UML étendu

Nous allons présenter comment l'extension que nous proposons peut s'intégrer dans le méta-modèle d'UML. La Figure 25 présente l'extension réalisée pour le diagramme d'activités de GRT, nous avons schématisé en gris les concepts de GRT.

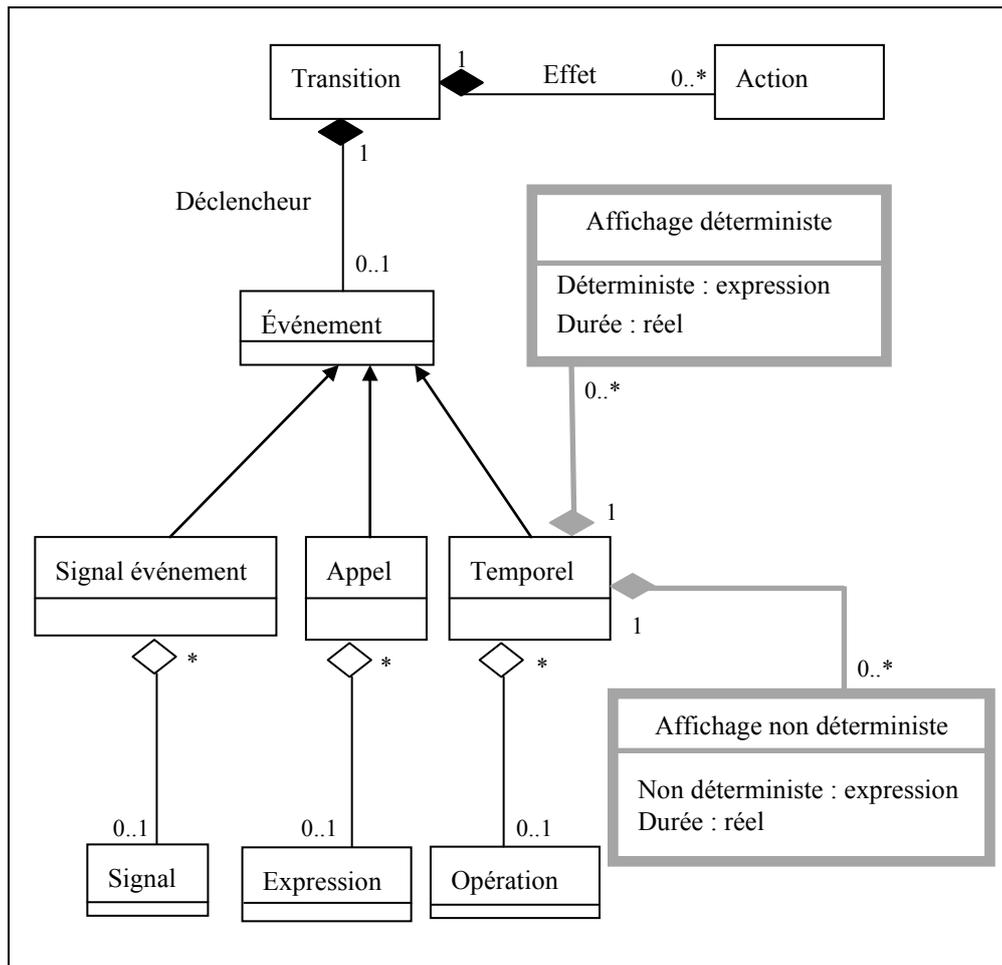


Figure 25. Extension du Méta-Modèle UML : Diagramme d'activités.

3.4.5 Le diagramme d'interventions

3.4.5.1 Définition du diagramme d'interventions

Vu l'importance des interventions pour les applications géomatiques temps réel, au niveau de notre profil on donne la possibilité au concepteur de définir un diagramme d'interventions dans lequel il précise les différentes interventions possibles selon un ordre chronologique.

Un diagramme d'interventions est une extension du diagramme de séquences, en ce basant sur les concepts du profil UML temps réel TURTLE ([APV 04], [APV 05] et [APV 05b]) pour qu'il soit adaptable aux applications géomatiques temps réel.

Le diagramme d'interventions montre des interactions entre objets selon un point de vue temporel. La représentation se concentre sur l'expression des interactions. Il représente une interaction entre objets en insistant sur la

chronologie des envois de messages. Les objets communiquent en échangeant des messages.

Outre les classiques messages synchrones et asynchrones, et les opérateurs de boucles et d’alternatives, ces diagrammes supportent des opérateurs temps réel permettant d’une part, d’associer à un événement un symbole « *tempo* » pour bien préciser qu’il se déroule en temps réel et d’autre part, d’associer à deux événements un intervalle de temps pouvant s’écouler entre l’occurrence de ces deux événements (Tableau 8).

	Tempo	Intervalle de temps
Symboles		 Durée

Tableau 8. Représentation des opérateurs du temps au niveau du diagramme d’interventions.

A part les contraintes temporelles ajoutées, nous pouvons aussi définir tout type d’objets : des objets classiques simples ou des objets de types géographiques. Les objets de type géographiques peuvent être représentés visuellement par des stéréotypes.

Au niveau de ce diagramme, nous donnons aussi la possibilité au concepteur d’exprimer graphiquement la combinaison de plus de deux types géographiques de base définis précédemment.

La Figure 26 exprime un exemple de digramme d’interventions selon le profil GRT.

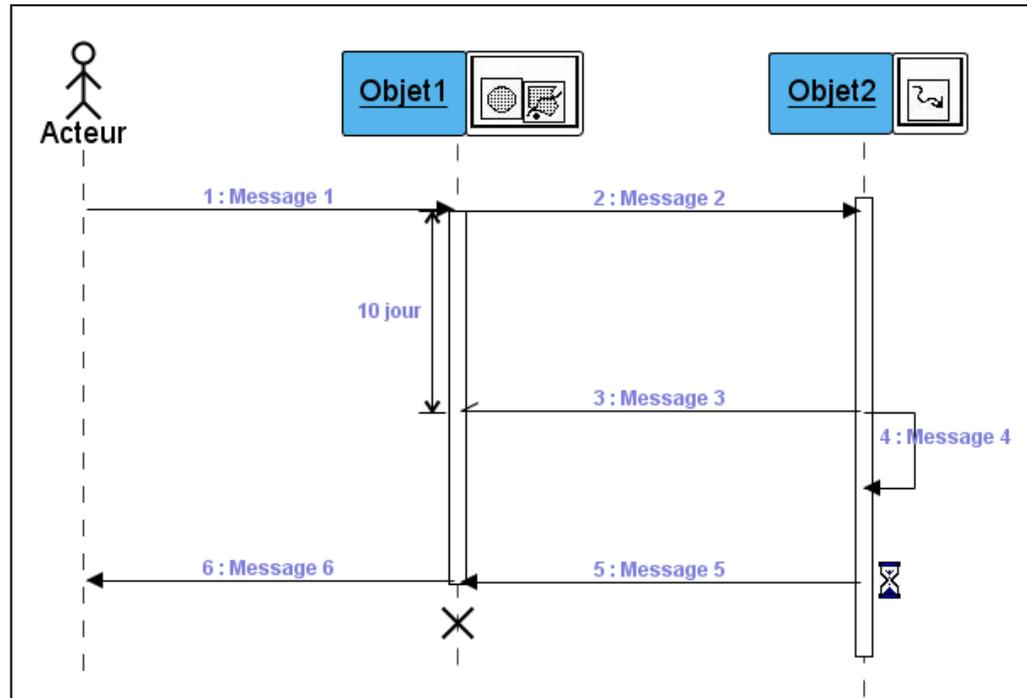


Figure 26. Présentation d'un diagramme d'interventions selon GRT.

3.4.5.2 Règles du modèle d'interventions

Au niveau de ce diagramme, le concepteur est amené à préciser la classe d'instanciation de l'objet à partir des classes définies au niveau du diagramme de classes. De même on lui permet d'associer un ou plusieurs stéréotypes aux objets géographiques. Le concepteur doit aussi attribuer obligatoirement à chaque message un numéro de séquence qui doit être unique au sein du diagramme. Il doit faire la distinction entre le temps réel, le temps mou et le temps différé. Ce diagramme doit être cohérent avec le diagramme de classes, c'est-à-dire les messages utilisés au niveau de ce diagramme sont des méthodes de classes définies dans le diagramme de classes.

3.4.5.3 Intégration au méta-modèle d'UML étendu

Nous allons présenter l'extension du méta-modèle d'UML que nous proposons pour intégrer les différentes notions ajoutées. La Figure 27 présente l'extension réalisée pour le diagramme d'interventions, nous avons schématisé en gris les concepts de GRT.

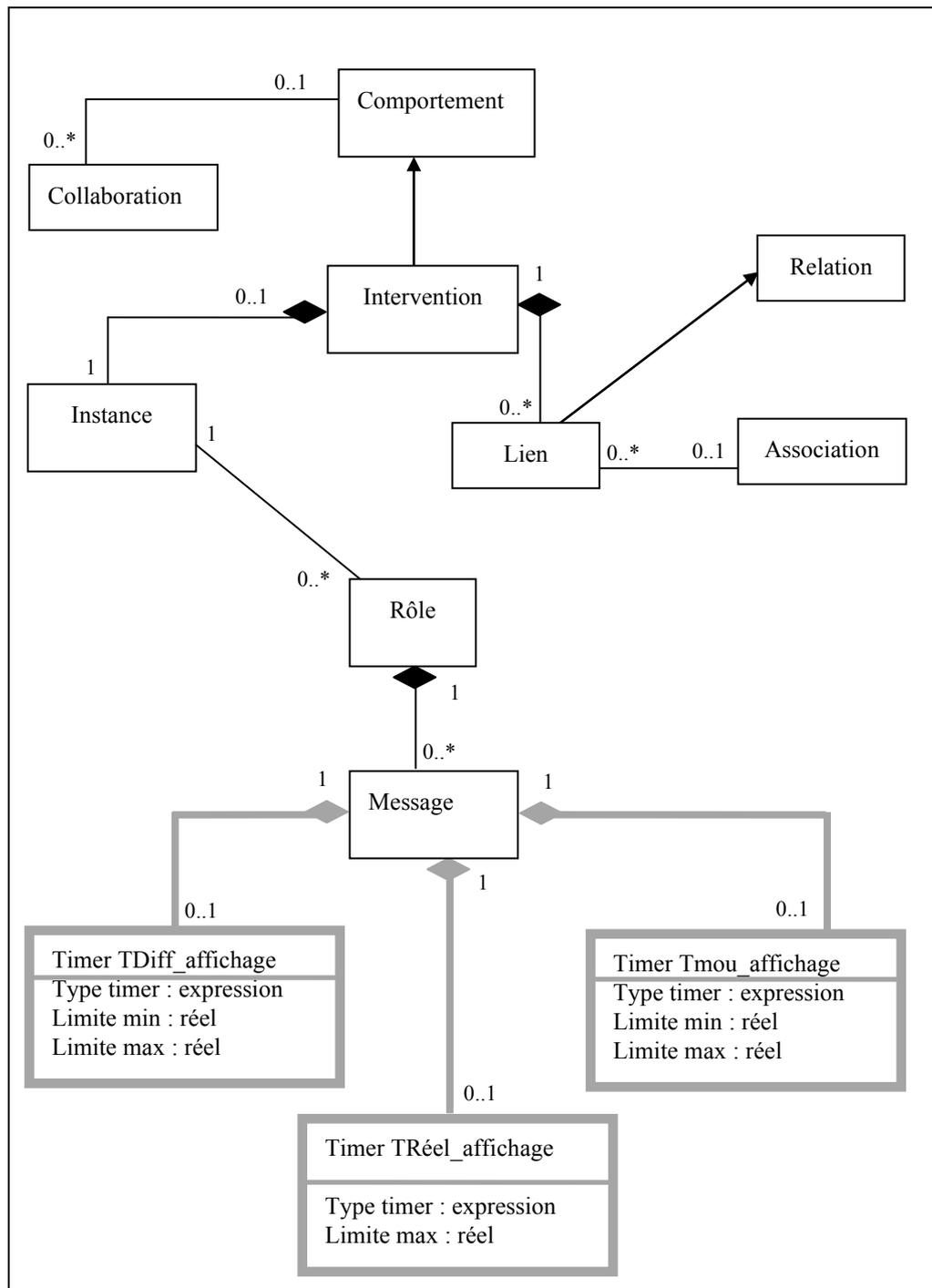


Figure 27. Extension du Méta-Modèle UML : Diagramme d'interventions.

3.5 Conclusion

Nous avons présenté dans ce chapitre, une extension du langage UML permettant de modéliser une application géomatique temps réel ceci en définissant un nouveau profil : le profil GRT [BEN 10]. Nous nous sommes inspirée pour réaliser ceci des travaux existants introduits dans le chapitre état de l'art. En effet, notre contribution, consiste en la prise en compte, dans un même formalisme, des besoins du concepteur pour modéliser une application géomatique temps réel. Une des difficultés majeures de nos travaux a été d'essayer d'enrichir notre formalisme sans pour autant complexifier ses notations pour que le modèle conceptuel réalisé soit visuel et intuitif. Afin de compléter notre étude, nous avons présenté l'intégration de notre formalisme au méta-modèle d'UML pour les différents diagrammes.

Une fois le modèle conceptuel établi, il est nécessaire de passer à l'étape suivante : la modélisation logique. Le passage au niveau logique est bien sûr une étape complexe et implique l'identification de certaines catégories d'objets.

Le chapitre suivant exprime les règles de passage d'un diagramme de classes réalisé par le profil GRT au modèle logique.

CHAPITRE 4.

REGLES DE PASSAGE DU MODELE CONCEPTUEL AU MODELE LOGIQUE DANS GRT

4.1 Introduction

Une fois le modèle conceptuel établi, il est nécessaire de passer à l'étape suivante : la modélisation logique. La modélisation logique des données est une représentation des données issue de la modélisation conceptuelle qui construit une structure de données adaptée au SGBD ou au SIG. Le schéma logique doit reproduire la totalité de la sémantique existante dans le schéma conceptuel en adaptant la description aux concepts offerts par le SGBD ou le SIG. Par exemple, si l'on adapte un SGBD relationnel, il s'agit de représenter, par un formalisme précis l'ensemble des tables qu'il faudrait créer pour réaliser le projet décrit dans le modèle conceptuel de données. L'adaptation à l'environnement concret sera faite en aval de cette opération. Ainsi, tout changement de système, d'équipe ou d'outils de développement pourra s'appuyer sur le modèle logique sans remettre en cause le travail réalisé en amont.

Vu l'importance du niveau logique, nous allons traiter les différentes règles générales de transformation d'un diagramme de classes réalisé à l'aide du profil GRT à un niveau logique. La transformation se réalise en plusieurs étapes, suivant un certain nombre de règles automatisables pour qu'on puisse avoir à partir d'un diagramme de classes généré selon GRT un code exécutable.

Pour définir ces règles nous avons utilisé les règles décrites en [BEN 99] qui sont les règles de passage d'un modèle conceptuel MADS à un niveau logique avec l'ajout de nouvelles règles spécifiques au profil GRT. Ce sont les règles classiques de passage du modèle entité-association vers le modèle relationnel (voir par exemple [HAI 91]) auxquelles nous avons ajouté les règles de transformation des données spatiales et temporelles.

On passe ainsi des concepts évolués, ceux du profil GRT, vers des concepts plus simples et moins puissants, ceux du SGBD. Il s'agit donc d'un passage d'un concept de haut niveau à un concept de bas niveau. Parfois, les concepts au niveau logique sont insuffisants pour décrire le modèle conceptuel, et on est obligé d'introduire des contraintes d'intégrité pour obtenir un schéma logique

sémantiquement équivalent. Ces contraintes sont implémentées selon le système cible par des triggers (déclencheurs), des checks, ou des procédures associées à l'application.

Le trigger permet une mise à jour automatique cohérente, une maintenabilité facile des tables et une sécurité des données de la base. Il permet au concepteur de s'assurer de la cohérence de la base de données. Il agit sur les nouvelles données et sur les mises à jour ou suppressions des données [ABD 95]. Dans la plupart du temps les triggers sont déclenchés par la modification du contenu d'une table.

Le check est un mot clé associé à une condition qui doit être vérifiée pour chaque valeur insérée [ABD 95]. La contrainte check permet que toutes les valeurs d'une colonne satisfassent des conditions données.

Les différentes règles de transformation traitées sont présentées dans la section suivante.

4.2 Les règles de transformation dans GRT

Cette section exprime les différentes règles de transformation d'un diagramme de classes réalisé selon le profil GRT au modèle logique. Nous avons tout d'abord cité les règles structurelles, celles qui transforment la structure des données sans faire référence ni aux informations spatiales ni aux informations temporelles, qui sont valables quelque soit le type des informations (thématique pur, spatial et temporel), puis les règles qui ont des variantes spatiales ou temporelles, puis les règles spatiales et enfin les règles temporelles.

4.2.1 Transformations structurelles

Nous avons traité les différentes règles de transformations structurelles [BEN 99] telles que : la transformation d'une entité en une table, la transformation d'une association n -aire (une association n -aire dont tous les rôles sont multivalués et une association n -aire qui a au moins un rôle monovalué), la transformation des liens *is-a*, la transformation d'une contrainte de disjonction entre liens *is-a*, la transformation d'une contrainte de couverture entre liens *is-a* et la transformation d'une contrainte de partition entre liens *is-a*.

Certaines règles citées seront exprimées en annexe B.

4.2.2 Transformations d'attributs

Nous avons aussi traité les règles permettant les transformations d'attributs telles que la transformation d'un attribut en dépendance fonctionnelle avec un autre attribut, la transformation d'un attribut complexe, la transformation d'un attribut complexe en dépendance fonctionnelle avec un attribut clé, la transformation d'un attribut multivalué, la transformation d'un attribut de domaine énuméré, la transformation d'un attribut spatial et la transformation d'un attribut temporel.

Au niveau de ce paragraphe nous allons exprimer quelques règles de transformation et les autres règles sont citées en annexe B.

4.2.2.1 Transformation d'un attribut de domaine énuméré

S'il existe dans une entité E un attribut (A) de domaine énuméré ayant pour type *ENUM* ('V₁', 'V₂',...), alors il faut ajouter un trigger pour vérifier la valeur de l'attribut lors de l'insertion (Figure 28) ou bien mettre un check.

```

CREATE      TRIGGER  domaine_énuméré
BEFORE     INSERT
ON         E
WHEN      E.A <> V1
          AND      E.A <> V2
          AND      E.A <> V3
          |
          AND      E.A <> Vn
ABORT     TRANSACTION.
```

Figure 28. Définition d'un trigger permettant l'élimination d'un attribut énuméré.

4.2.2.2 Transformation d'un attribut spatial

Un attribut spatial dans GRT peut prendre les types géographiques suivants : *Géo*, *Géo simple*, *Géo composé*, *surface simple*, *surface composée*, *ligne*, *graphe*, *ligne orientée*, *digraphe*, *point* et *semis*. En utilisant Oracle Spatial [GRE 04], lors de la transformation d'un attribut spatial, nous remplaçons son type par *SDO_GEOMETRY* qui est un type complexe comportant cinq attributs simples et qui englobe tous les types spatiaux cités précédemment (Figure 29).

```
CREATE TYPE SDO_GEOMETRY AS OBJECT (
  SDO_GTYPE NUMBER,
  SDO_SRID NUMBER,
  SDO_POINT SDO_POINT_TYPE,
  SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,
  SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

Figure 29. Création du type SDO_GEOMETRY.

Le type SDO_GTYPE indique le type de la géométrie spécifié, il est formé par quatre chiffres de la forme *dltt* avec :

- d : identifie le nombre de dimension (2, 3, ou 4),
- l : identifie la mesure de dimension linéaire de référence,
- tt : identifie le type de géométrie.

Le tableau 9 décrit les types existants au niveau d'Oracle Spatial [GRE 04] que nous avons manipulée suivant les valeurs de SDO_GTYPE :

Valeur	Type géométrique	Description
dl00	Géométrie inconnue	Spatial ignore cette géométrie
dl01	Point	La géométrie est de type point
dl02	Ligne ou courbe	La géométrie contient une ligne ou une courbe
dl03	Polygone	Un polygone avec ou sans trou
dl04	Collection	Un sur-ensemble contenant plusieurs géométries
dl05	Multipoint	Un ensemble de points
dl06	Multi-ligne	Un ensemble de lignes ou de courbes

Tableau 9. Liste des types géométriques suivant SDO_GTYPE.

Les définitions des types SDO_POINT_TYPE, SDO_ELEM_INFO_ARRAY, et SDO_ORDINATE_ARRAY qui sont utilisés lors de la création de SDO_GEOMETRY sont décrites dans la Figure 30.

```
CREATE TYPE SDO_POINT_TYPE AS OBJECT
(
  X NUMBER,
  Y NUMBER,
  Z NUMBER);
CREATE TYPE SDO_ELEM_INFO_ARRAY AS
VARRAY (1048576) OF NUMBER;
CREATE TYPE SDO_ORDINATE_ARRAY AS
VARRAY (1048576) OF NUMBER;
```

Figure 30. Création du type SDO_POINT_TYPE.

SDO_ORDINATES est un tableau stockant les coordonnées des différents points de la géométrie.

SDO_ELEM_INFO est un tableau de taille variable dont les éléments sont groupés par triplets. Chaque triplet indique l'emplacement de la première coordonnée de la géométrie dans le tableau SDO_ORDINATES, le type de la géométrie et si elle est complexe ou non.

Lorsque les deux tableaux SDO_ELEM_INFO et SDO_ORDINATES sont tous les deux nuls et que l'attribut SDO_POINT est déclaré alors les valeurs X et Y représentent les coordonnées d'une géométrie point, sinon cet attribut est ignoré.

4.2.2.3 Transformation d'un attribut temporel

Un attribut temporel au niveau de GRT est un attribut dont la valeur évolue au cours du temps et dont on veut mémoriser son évolution passée, présente et future. Au niveau conceptuel, c'est une fonction du temps vers le domaine de valeurs de l'attribut. Lorsqu'il existe dans une entité E un attribut temporel (A), l'attribut (A) devient un attribut complexe de type liste de 1 à n et non temporel (A_liste). Les attributs composants de l'attribut (A_liste) seront : un attribut simple spécifiant la valeur (A) et deux attributs de type date : `date_début` et `date_fin` de la validité de cette valeur. Nous devons aussi ajouter deux triggers pour vérifier la cohérence des dates [BEN 99].

- Un trigger pour vérifier que la date de début est inférieure à la date de fin (Figure 31).

```
CREATE TRIGGER attribut_temporel
BEFORE INSERT
ON E
WHEN New(E.A_liste).Date_début > New(E.A_liste).Date_fin
OR
      New(E.A_liste).Date_début < Last(E.A_liste).Date_fin
ABORT TRANSACTION.
```

Figure 31. Trigger1 vérifiant la cohérence des dates.

- Un trigger pour vérifier lors de la fin de la valeur d'un attribut que la date de fin est supérieure à la date de début (Figure 32).

```
CREATE TRIGGER changement_de_valeur
BEFORE UPDATE
ON E
WHEN
      New(E.A_liste).Date_fin < Last(E.A_liste).Date_début
ABORT TRANSACTION.
```

Figure 32. Trigger2 vérifiant la cohérence des dates.

Après avoir effectué cette transformation, nous devons appliquer la règle de transformation d'un attribut multivalué pour pouvoir passer au niveau relationnel.

4.2.3 Transformation d'une association spatiale

Une association spatiale est une association qui lie des classes spatiales et qui force ces dernières à respecter une relation spatiale prédéfinie, telle que par exemple l'inclusion, l'adjacence ou l'égalité. Lors de la transformation d'une association spatiale nous devons appliquer les règles de transformation d'une association classique et ajouter un trigger vérifiant que les tuples des classes vérifient la contrainte spatiale spécifiée par l'association.

Nous présentons dans la Figure 33 un exemple d'une association spatiale de type égalité entre deux classes spatiales qui possèdent une géométrie de

stockage¹ de type surface, ensuite nous définissons le trigger qui exprime l'élimination de la contrainte égalité au niveau de l'association.

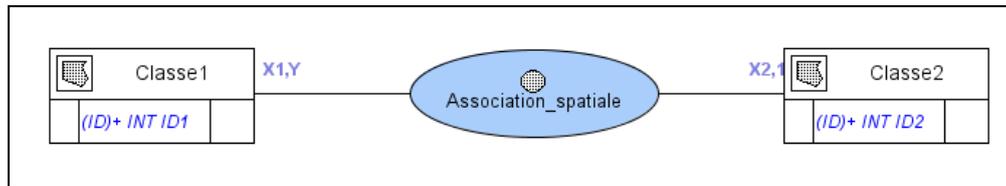


Figure 33. Exemple d'une association spatiale de type égalité.

Lors de la transformation de ce modèle conceptuel, nous devons appliquer la règle de transformation d'une association classique et ajouter un trigger pour assurer le respect d'une association spatiale de type égalité (Figure 34).

```
CREATE TRIGGER association_spatiale_egalite
BEFORE INSERT
ON Classe2
WHEN
NOT EXIST ( SELECT ID1, geometrieStock
FROM Classe1, Classe2
WHERE Classe1.ID1 = Classe2.ID1
AND
SDO_Equal (Classe1.geometrieStock, Classe2.geometrieStock)
= 'True' );
ABORT TRANSACTION.
```

Figure 34. Définition d'un trigger assurant le respect d'une association spatiale de type égalité.

4.2.4 Transformation d'un attribut spatial d'une classe spatiale en une nouvelle classe

Si une classe A spatiale possède un attribut spatial alors on peut transformer cet attribut en une nouvelle classe spatiale B liée à A par une association.

Après avoir effectué cette transformation, nous devons appliquer la règle de l'élimination d'une association N-aire avec un rôle monovalué pour pouvoir passer au niveau relationnel.

¹ La géométrie de stockage, c'est la géométrie choisie (au niveau conceptuel) pour le stockage de l'information [PIN 02].

4.2.5 Transformation d'une classe spatiale

La spatialité d'une classe dans GRT est décrite soit par sa géométrie d'affichage², soit par sa géométrie de stockage et soit par les deux. De ce fait la transformation d'une entité ayant une géométrie d'affichage est faite en ajoutant l'attribut `geometrieAff` de type `SDO_GEOMETRY` et la transformation d'une entité ayant une géométrie de stockage est réalisée en ajoutant l'attribut `geometrieStock` de type `SDO_GEOMETRY`.

4.2.6 Transformation d'une classe à géométries combinées

Le profil GRT permet de définir des classes à plusieurs géométries liées entre elles par le connecteur logique « ET » (Figure 35) ou par le connecteur logique « OU » (Figure 36).

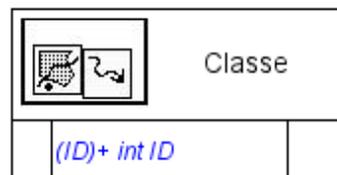


Figure 35. Présentation d'une classe spatiale ayant des géométries combinées par le connecteur « ET ».

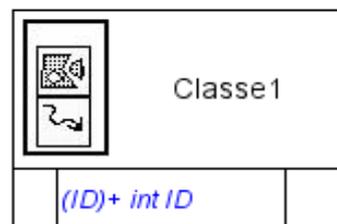


Figure 36. Présentation d'une classe spatiale ayant des géométries combinées par le connecteur « OU ».

La transformation d'une classe ayant des géométries liées par le connecteur « ET » se fait en ajoutant des attributs de types `SDO_GEOMETRY` ayant les géométries déclarées dans la classe.

² La géométrie d'affichage, c'est la géométrie telle qu'elle apparaîtra sur une carte dans l'application finale [PIN 02].

La transformation d'une classe ayant des géométries de stockage liées par le connecteur « OU » se fait en déclarant un attribut de type SDO_GEOMETRY et en ajoutant un trigger qui vérifie lors de l'insertion de l'attribut géométrique déjà cité que ses valeurs coïncident avec l'une des géométries de la classe (Figure 37). On procède de même pour une classe ayant des géométries d'affichages liées par le connecteur « OU » (Figure 38).

```
CREATE TRIGGER stockage_double_geometrie_OU_nom_entité
BEFORE INSERT ON nom_entité
WHEN
(nom_entité.geometrieStock.SDO_GTYPE<>
                                geometrieStock1.SDO_GTYPE)
AND
|
AND
(nom_entité.geometrieStock.SDO_GTYPE<>
                                geometrieStockn.SDO_GTYPE)
ABORT TRANSACTION.
```

Figure 37. Définition d'un trigger permettant la vérification pour une classe ayant des géométries combinées de stockage liées par le connecteur « OU ».

```
CREATE TRIGGER affichage_double_geometrie_OU_nom_entité
BEFORE INSERT ON nom_entité
WHEN
(nom_entité.geometrieAff.SDO_GTYPE<>geometrieAff1.SDO_GTYPE)
AND
|
AND
(nom_entité.geometrieAff.SDO_GTYPE<>geometrieAffn.SDO_GTYPE)
ABORT TRANSACTION.
```

Figure 38. Définition d'un trigger permettant la vérification pour une classe ayant des géométries combinées d'affichage liées par le connecteur « OU ».

4.3 Implémentation des différentes règles de transformation dans GRT

Les différentes règles citées précédemment ont été implémentées pour pouvoir générer à partir d'un diagramme de classes généré selon le profil GRT un code source en SQL (Structured Query Language) [BRO 08] interprétable sous le système de gestion de base de données Oracle Spatiale.

Nous présentons quelques exemples de modèles conceptuels conçus à l'aide de notre atelier de génie logiciel AGRT que nous avons implémenté selon le profil GRT. Nous présentons ensuite le code SQL généré automatiquement par AGRT pour les différents schémas traités.

4.3.1 Exemple 1

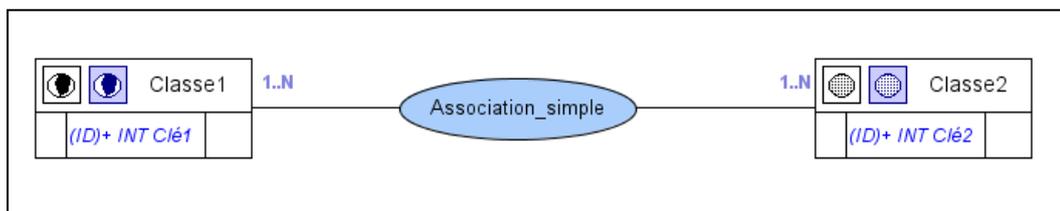


Figure 39. Schéma conceptuel présentant deux classes spatiales liées par une association simple.

Dans l'exemple de la Figure 39, nous présentons deux classes spatiales qui ont chacune une géométrie d'affichage et une géométrie de stockage qui seront représentées dans le code par les attributs `geometrieAff` et `geometrieStock` de type `SDO_GEOMETRY`. Ces dernières sont liées par une association simple.

Le code SQL, généré automatiquement par AGRT, pour ce schéma conceptuel est récupéré dans un fichier « .txt » que nous allons présenter dans la Figure 40.

```
/* Code généré par AGRT
*/
CREATE OR REPLACE PROCEDURE abort_transaction IS
BEGIN
RAISE_APPLICATION_ERROR (-20000, ' ');
END abort_transaction;
/* Classe1
*/

CREATE TABLE Classe1 (
Cle1          INT NOT NULL,
geometrieAff  SDO_GEOMETRY ,
geometrieStock SDO_GEOMETRY ,
CONSTRAINT Classe1_PRIMARY_KEY PRIMARY KEY (Cle1) );

CREATE TABLE Association_simple (
Cle1          INT NOT NULL,
Cle2          INT NOT NULL,
CONSTRAINT Association_simple_FOREIGN_KEY_1 FOREIGN KEY
(Cle1) REFERENCES Classe1 ( Cle1),
CONSTRAINT Association_simple_FOREIGN_KEY_2 FOREIGN KEY
(Cle2) REFERENCES Classe2 ( Cle2),
CONSTRAINT Association_simple_PRIMARY_KEY PRIMARY KEY
(Cle1 ,Cle2) );
/* Classe2
*/

CREATE TABLE Classe2 (
Cle2          INT NOT NULL,
geometrieAff  SDO_GEOMETRY ,
geometrieStock SDO_GEOMETRY ,
CONSTRAINT Classe2_PRIMARY_KEY PRIMARY KEY (Cle2) );
```

Figure 40. Le code source généré à partir du modèle conceptuel.

4.3.2 Exemple 2

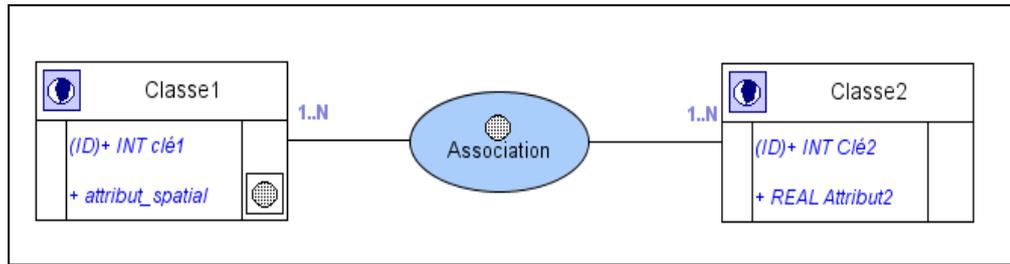


Figure 41. Schéma conceptuel présentant une association spatiale de type égalité.

Dans l'exemple de la Figure 41, Les deux classes possèdent une géométrie de stockage de type Géo. L'attribut spatial de la Classe 1 est de type Géo simple, il aura donc dans le code le type `SDO_GEOMETRY`. L'association, étant spatiale de type égalité, va engendrer la création d'un trigger qui vérifie lors de l'insertion l'égalité des géométries des deux classes. Le code généré est décrit dans la Figure 42.

```

/* Code généré par AGRT
*/
CREATE OR REPLACE PROCEDURE abort_transaction IS
BEGIN
RAISE_APPLICATION_ERROR (-20000, ' ');
END abort_transaction;
/* Classe1
*/
CREATE TABLE Classe1 (
Cle1                INT NOT NULL,
attribut_spatial    SDO_GEOMETRY,
geometrieStock      SDO_GEOMETRY ,
CONSTRAINT Classe1_PRIMARY_KEY PRIMARY KEY (Cle1) );

CREATE TABLE Association (
Cle1                INT NOT NULL,
Cle2                INT NOT NULL,
CONSTRAINT Association_FOREIGN_KEY_1 FOREIGN KEY ( Cle1)
REFERENCES Classe1 ( Cle1),
CONSTRAINT Association_FOREIGN_KEY_2 FOREIGN KEY ( Cle2)
REFERENCES Classe2 ( Cle2),
CONSTRAINT Association_PRIMARY_KEY PRIMARY KEY ( Cle1 ,Cle2)
);

CREATE OR REPLACE TRIGGER association_spatiale_egalite
BEFORE INSERT ON Classe2
FOR EACH ROW
WHEN
(NOT EXISTS ( SELECT Cle1,geometrieStock
              FROM Classe1,Classe2
              WHERE
SDO_EQUAL(Class1.geometrieStock,Classe2.geometrieStock)
= 'True'))
BEGIN
ABORT_TRANSACTION;
END;
/* Classe2
*/
CREATE TABLE Classe2 (
Cle2                INT NOT NULL,
Attribut2           REAL,
geometrieStock      SDO_GEOMETRY,
CONSTRAINT Classe2_PRIMARY_KEY PRIMARY KEY (Cle2) );

```

Figure 42. Le code source généré à partir du modèle conceptuel.

4.3.3 Exemple 3

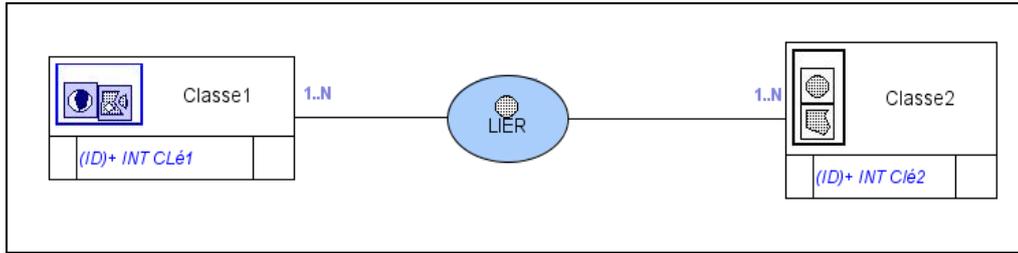


Figure 43. Schéma conceptuel présentant des entités à géométries combinées.

L'exemple de la Figure 43 aborde des classes ayant plus d'une géométrie. La Classe1 possède deux géométries d'affichage liées par le connecteur « ET ». Leurs transformations consistent à déclarer deux attributs de type SDO_GEOMETRY correspondant chacun à l'une des géométries. La Classe 2 possède deux géométries de stockage liées par le connecteur « OU ». Le code SQL correspondant à ce diagramme de classes est décrit dans la Figure 44.

```

/* Code généré par AGRT
*/
CREATE OR REPLACE PROCEDURE abort_transaction IS
BEGIN
RAISE_APPLICATION_ERROR (-20000, ' ');
END abort_transaction;

CREATE TABLE Classe1 (
Cle1                INT NOT NULL,
geometrieAff1      SDO_GEOMETRY ,
geometrieAff2      SDO_GEOMETRY ,
CONSTRAINT Classe1_PRIMARY_KEY PRIMARY KEY (Cle1) );

CREATE TABLE Lier (
Cle1                INT NOT NULL,
Cle2                INT NOT NULL,
CONSTRAINT Lier_FOREIGN_KEY_1 FOREIGN KEY ( Cle1) REFERENCES
Classe1 ( Cle1),
CONSTRAINT Lier_FOREIGN_KEY_2 FOREIGN KEY ( Cle2) REFERENCES
Classe2 ( Cle2),
CONSTRAINT Lier_PRIMARY_KEY PRIMARY KEY ( Cle1 ,Cle2) );

CREATE OR REPLACE TRIGGER association_spatiale_egalite
BEFORE INSERT ON Classe2
FOR EACH ROW
WHEN
(NOT EXISTS ( SELECT Cle1,geometrieStock
              FROM Classe1,Classe2
              WHERE
SDO_EQUAL(Class1.geometrieStock,Classe2.geometrieStock)
= 'True'))
BEGIN
ABORT_TRANSACTION;
END;

CREATE TABLE Classe2 (
Cle2                INT NOT NULL,
geometrieStock1     SDO_GEOMETRY
CONSTRAINT Classe2_PRIMARY_KEY PRIMARY KEY (Cle2) );

CREATE OR REPLACE TRIGGER
stockage_double_geometrie_OU_Classe2
BEFORE INSERT ON Classe2
FOR EACH ROW
WHEN
Get_GType(new.geometrieStock1) <> GGeoSimple AND
Get_GType(new.geometrieStock1) <> GSurfaceSimple)
BEGIN
ABORT_TRANSACTION;
END;

```

Figure 44. Le code source généré à partir du modèle conceptuel.

4.4 Conclusion

Dans ce chapitre, nous avons défini les différentes règles de transformation d'un diagramme de classes décrit selon le profil GRT à un niveau logique. Ces règles permettent le passage d'un modèle conceptuel spatio-temporel vers celui d'un système d'information géographique. Elles ont été implémentées dans l'atelier génie logiciel AGRT permettent de générer un code source en SQL. Nous présentons dans le chapitre suivant notre atelier AGRT.

CHAPITRE 5.

EXPERIMENTATION :

IMPLEMENTATION D'UN AGL

(AGRT)

5.1 Introduction

Afin de permettre au concepteur de réaliser les différents diagrammes du profil GRT que nous avons proposés, nous avons défini un outil logiciel de conception pour les applications géomatiques temps réel doté des différents concepts déclarés dans le profil GRT.

Le but de ce chapitre est de présenter cet outil. En effet, nous allons en premier lieu décrire notre AGL, son architecture et sa conception, ensuite nous expérimenterons notre méthode par la modélisation de quelques applications géomatiques temps réel.

5.2 Un outil pour la conception des applications géomatiques temps réel

Un AGL est un ensemble d'outils logiciels structurés au sein d'une même interface qui permettent la conception, le développement et le débogage de logiciels.

Nous pouvons dire qu'un AGL est un ensemble intégré d'outils articulés autour d'un référentiel afin d'automatiser l'ensemble des phases du cycle de vie d'un logiciel. Il doit donc faciliter le développement d'un produit tout le long de son cycle de vie

L'utilisation d'un AGL se justifie par plusieurs raisons [LBA 97], nous pouvons résumer ces dernières comme suit :

- couvrir la totalité du cycle de développement de logiciel en automatisant certaines tâches telle que la génération de code et en assistant les concepteurs dans les phases d'analyse et de conception,
- diminuer le coût de maintenance qui représente souvent les deux tiers du coût d'un projet informatique en assurant une gestion intégrée de la maintenance évolutive et corrective,
- avoir une approche qualité et sûreté de fonctionnement plus fiable en offrant des outils de production automatique et de documents accompagnant le logiciel tout au long de son cycle de vie,

- la réduction des délais de livraison,
- la réduction du coût de développement du logiciel en réduisant le temps de production de code et en augmentant la qualité du code source produit ce qui entraîne une double économie sur l'effort et le temps de développement d'une part et sur la diminution des efforts de maintenance d'autre part.

Nous avons développé un AGL destiné à la conception des applications géomatiques temps réel, il est nommé AGRT. Ce dernier est basé sur notre profil GRT qui est basé sur la méthode MADS, la méthode T-OMEGA et le profil TURTLE avec l'ajout de nouveaux concepts pour les applications géomatiques temps réel.

5.2.1 Description de l'AGL

Nous avons développé AGRT à l'aide du langage JAVA en utilisant le JDK version 6.0. Le développement a été fait en utilisant l'environnement de développement intégré « NetBeans IDE 5.5 », alors que les icones et les logos sont faits en utilisant les outils « AAA Logo » et « Axialis IconWorkshop ».

L'AGL se compose de trois modules qui s'articulent autour d'un référentiel :

- le premier module est un éditeur de diagrammes paramétré pour la conception des diagrammes,
- le deuxième module est un module de réutilisation et d'enrichissement de diagrammes,
- le troisième module est un générateur de code,
- le référentiel permet de gérer, maintenir et archiver les différents méta-modèles.

5.2.2 Architecture de l'AGL

Vu l'importance de la phase de modélisation, nous avons été amenée à réaliser un grapheur qui permet au concepteur de concevoir les applications géomatiques temps réel en spécifiant quatre types de diagrammes : les diagramme de priorités (diagrammes de cas d'utilisation avec succession d'actions), les diagrammes de classes, les diagrammes d'interventions et les diagrammes d'activités, auxquels nous avons ajouté quelques notations spécifiques pour qu'ils soient adaptés à la conception des applications géomatiques temps réel.

5.2.3 Modèle conceptuel d'AGRT

Nous avons effectué la conception de la partie statique d'AGRT, nous nous sommes intéressée aux diagrammes de cas d'utilisation et de classes. Le diagramme de cas d'utilisation permet de cerner nos objectifs, tandis que le diagramme de classes contient toutes les classes utilisées pour élaborer l'AGL.

5.2.3.1 *Le diagramme de cas d'utilisation*

Nous avons étudié deux diagrammes de cas d'utilisation : un premier diagramme pour la création et le sauvegarde d'un projet et les différents diagrammes traités dans notre profil (Figure 45) et un deuxième diagramme pour l'édition des différents diagrammes et la génération du code source (Figure 46).

5.2.3.1.1 Le diagramme de cas d'utilisation pour la création et le sauvegarde du projet

Ce diagramme énumère les différentes fonctionnalités offertes au concepteur pour gérer les diagrammes de son projet. Pour créer un projet, le concepteur est amené à spécifier le chemin de sauvegarde de ces diagrammes. Une fois le projet créé, le concepteur peut ajouter des diagrammes (soit en les important d'un autre projet soit en créant de nouveaux diagrammes), peut même retirer des diagrammes, peut aussi supprimer des diagrammes et peut renommer des diagrammes sachant que le nom du diagramme doit être unique au sein du même projet.

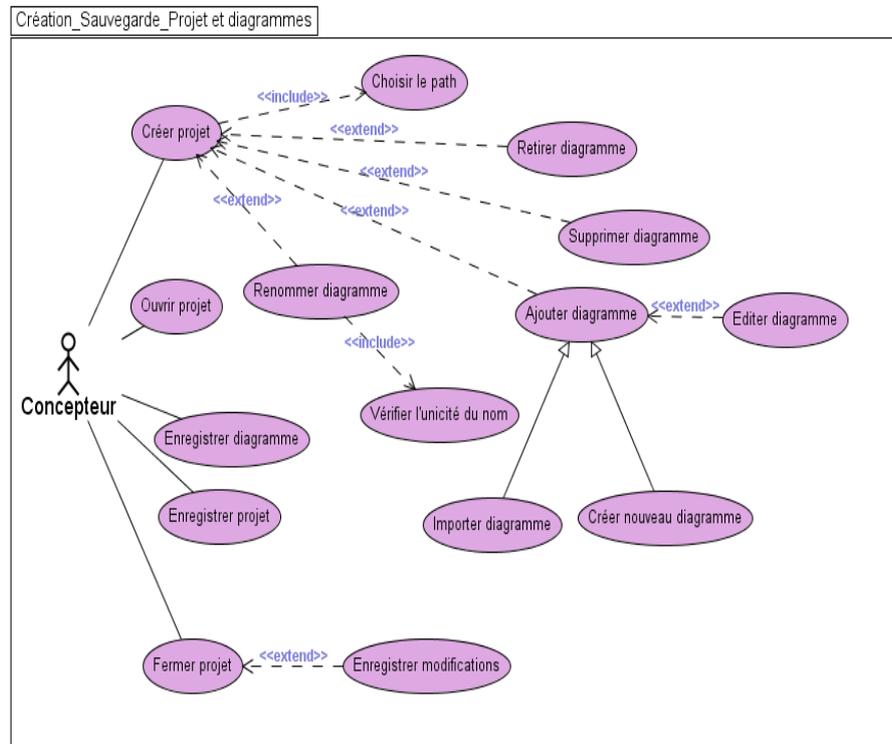
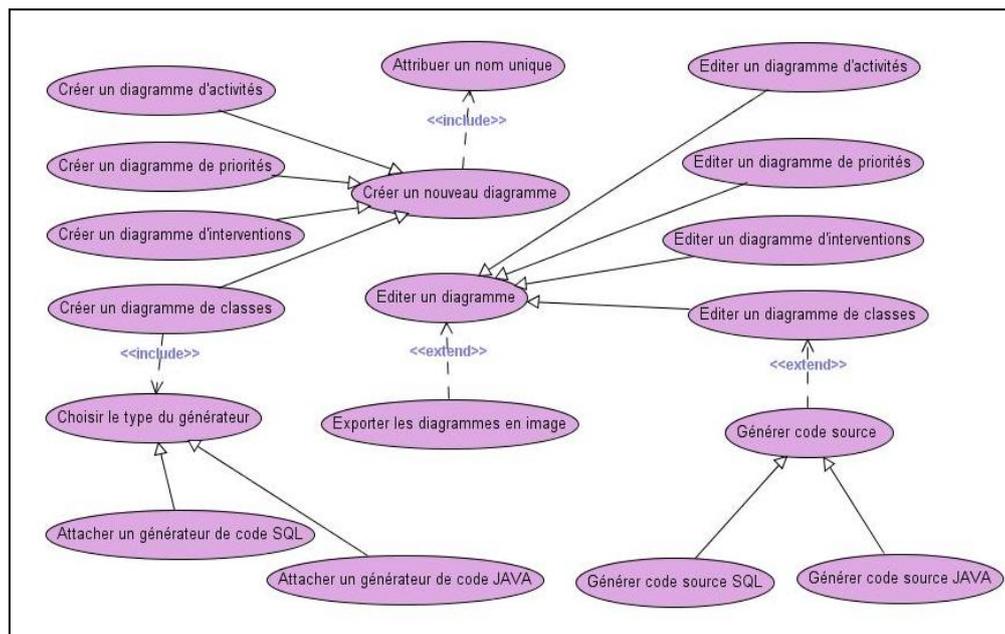


Figure 45. Diagramme de cas d'utilisation d'AGRT : création et sauvegarde du projet et des diagrammes (ce diagramme a été créé à l'aide du module graphique d'AGRT).

5.2.3.1.2 Le diagramme de cas d'utilisation pour l'édition des différents diagrammes et la génération du code source

Ce diagramme décrit les fonctionnalités offertes aux concepteurs pour l'édition des diagrammes et la génération du code source.

Le concepteur peut ainsi concevoir quatre types de diagrammes : diagramme de classes, diagramme d'interventions, diagramme de priorités et diagramme d'activités. Une fois conçus, les diagrammes peuvent être exportés en images. Au niveau des diagrammes de classes, le concepteur peut générer un code source en JAVA ou en SQL selon le type de générateur attaché au diagramme de classes lors de sa création.



**Figure 46. Diagramme de cas d'utilisation d'AGRT :
édition des diagrammes et génération du code.**

5.2.3.2 Le diagramme de classes

Pour avoir un modèle conceptuel complet et lisible nous avons partagé le travail en trois modules, chaque module correspondant à une partie de l'AGL : un premier diagramme pour décrire les composants graphiques utilisés (Figure 47), un deuxième diagramme pour décrire les classes permettant l'affichage, la manipulation et le stockage des diagrammes (Figure 48) et un troisième diagramme pour décrire la génération du code source (Figure 49).

5.2.3.2.1 Le diagramme des composants graphiques

Le diagramme des composants graphiques est un diagramme qui décrit les différents composants graphiques utilisés au niveau d'AGRT. Ce dernier décrit la hiérarchie des classes responsables de la description des composants graphiques. La hiérarchie utilisée rend l'outil facilement extensible en donnant la possibilité d'ajouter d'autres types de diagrammes. Tous les composants graphiques héritent d'une classe abstraite appelée « *AbstractDrawable* ». Cette classe définit l'attribut « *Selected* » de type « *Boolean* » qui prend la valeur « *True* » si le composant est sélectionné. Elle implémente l'interface « *IMovableDrawable* » qui hérite elle-même de l'interface « *IDrawable* ».

Les classes appartenant à l'hierarchie de racine « `IDrawable` » définissent les méthodes suivantes :

- la méthode « `draw()` » qui permet de définir la forme du composant,
- la méthode « `getRectangle()` » qui retourne le plus petit rectangle contenant le composant graphique,
- la méthode « `restaureDialogue()` » qui est responsable de la réinitialisation des dialogues de paramétrage des composants.

Les classes appartenant à l'hierarchie de racine « `IMovableDrawable` » définissent les méthodes suivantes :

- la méthode « `setPosition()` » qui définit la position du composant graphique dans le diagramme,
- la méthode « `getPosition()` » qui retourne la position du composant graphique dans le diagramme.

Cette interface rend facile la modification des positions des composants. Ceci va être exploité ultérieurement dans l'implémentation des classes de gestion des actions de la souris pour déplacer les composants.

Au niveau de la classe abstraite « `FormDrawable` » nous avons défini les méthodes « `setPosition()` » et « `getPosition()` » imposées par l'interface « `IMovableDrawable` ». Nous avons également défini des attributs communs pour toutes les formes. Par exemple l'attribut de type vecteur « `linkers` » contient tous les « `LineDrawable` » modélisant les relations du composant avec les autres composants. Cette classe définit une méthode abstraite « `getBorderLines()` » qui doit être redéfinie dans les classes filles et qui sert à décrire précisément le profil des composants. Les méthodes « `draw()` » et « `restaureDialogue()` » imposées par l'interface « `IDrawable` » doivent être implémentées au niveau des classes filles de « `FormDrawable` ».

Au niveau de la classe abstraite « `LineDrawable` » nous avons aussi défini les méthodes « `setPosition()` » et « `getPosition()` » imposées par l'interface « `IMovableDrawable` ». Les attributs communs à tous les types des lignes sont définis dans cette classe. Cette classe définit les méthodes abstraites « `getSourceIntersection()` » et « `getDestinationIntersection()` » qui doivent être redéfinies au niveau des classes filles de cette classe en plus des méthodes « `draw()` » et « `restaureDialogue()` » imposées par l'interface « `IDrawable` ».

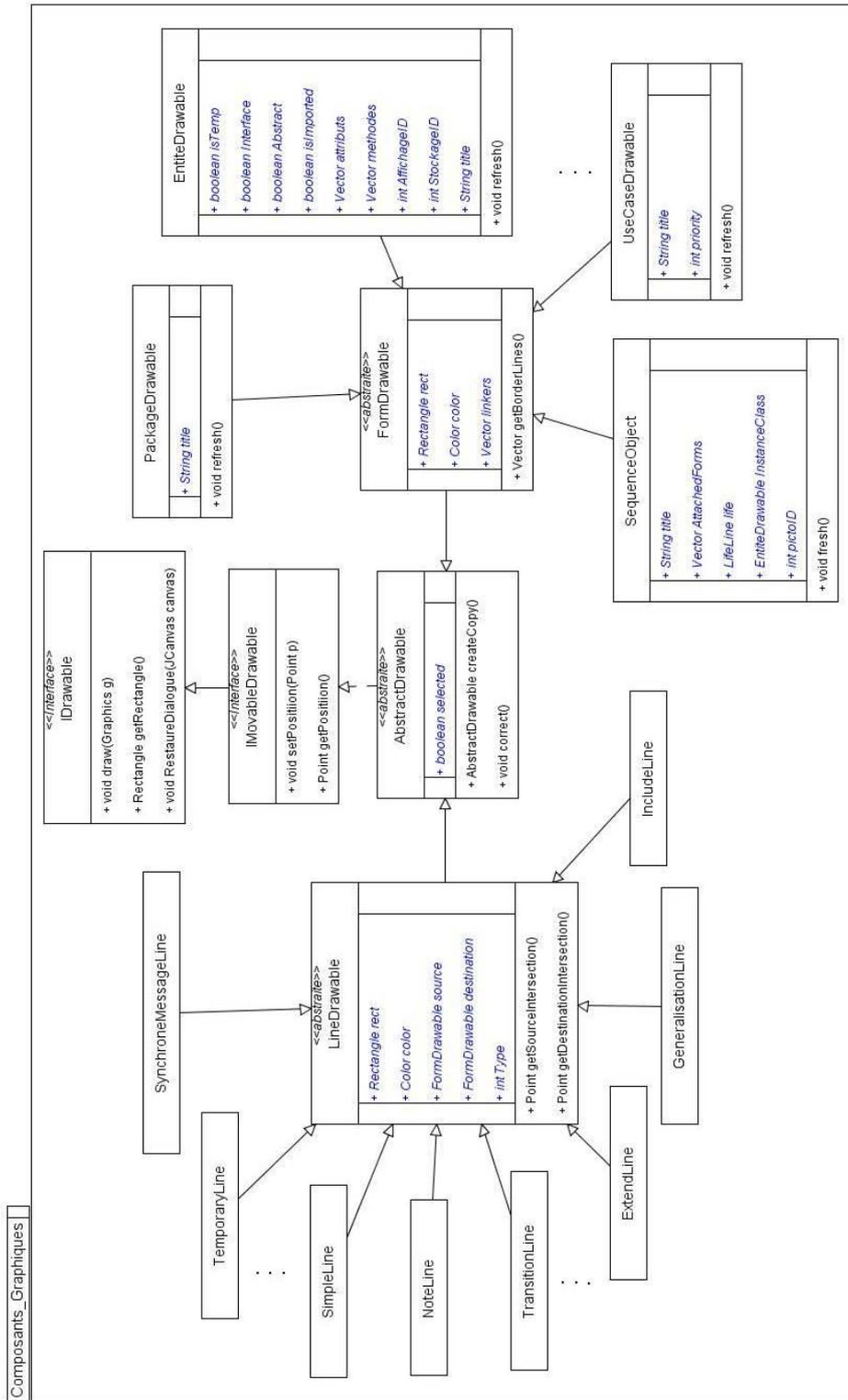


Figure 47. Diagramme de classes présentant les composants graphiques d'AGRT (méta-modèle d'AGRT).

5.2.3.2.2 Le diagramme de classes permettant l'affichage, la manipulation et le stockage des diagrammes

Ce diagramme définit les classes responsables de l'affichage, de la manipulation et du stockage des composants graphiques. La classe fondamentale est la classe « JCanvas », elle hérite de la classe « Javax.Swing.JPanel ». Ces composants sont affichés dans l'écran grâce à la surcharge de la méthode « paint() » héritée de la classe « Javax.Swing.JPanel » qui l'hérite à son tour de la classe « Javax.Swing.JComponent ». La classe « JCanvas » gère les deux attributs « undo » et « redo » de type « Stack » pour annuler ou répéter une action d'ajout, de suppression ou de modification d'un composant graphique. La classe « JCanvas » définit aussi la méthode « computePreferedSize() » qui retourne sa dimension optimale suivant la disposition des composants graphiques.

A une instance de la classe « JCanvas » on associe un « SimpleMouseListener » et un « JCanvasMouseListener ». La classe « SimpleMouseListener » gère les actions simples de la souris en surchargeant les méthodes « leftClickAction() » et « rightClickAction() » déjà définies dans la classe mère « JCanvasMouseListener ». La classe « JCanvasMouseListener » implémente l'interface « java.awt.event.MouseMotionListener » qui impose les méthodes « mouseDragged() » et « mouseMoved() ». Elle hérite de la classe « JCanvasMouseListener » et surcharge ses méthodes « mousePressed() » et « mouseReleased() ». Les objets de type « JCanvasMouseListener » attachés aux « JCanvas » sont soit des instances de la classe « TraceSequLineMouseListener » s'il s'agit d'un diagramme d'interventions, soit des instances de la classe « TraceLineMouseListener » s'il s'agit d'un autre type de diagramme. Ces deux classes sont définies pour gérer les actions de traçage des liens entre les « FormDrawable ».

Chaque « JCanvas » est encapsulée dans une feuille de dessin. Il y a quatre types possibles de feuille de dessin : « ClassFeuillePaint », « PriorityFeuillePaint », « SequenceFeuillePaint » et « ActivityFeuillePaint ». A chaque feuille de dessin est associé un diagramme.

La classe « UInterface » modélise l'interface principale d'interaction avec l'utilisateur. Les classes « Attribut » et « Methode » sont des structures de données définies pour faciliter la gestion des composants graphiques de type « EntiteDrawable ». Les classes « Diagramme » et « Projet » sont aussi des structures de données qui facilitent la tâche de sauvegarde des projets et des diagrammes.

Pour simplifier ce diagramme on n'a pas présenté les classes de dialogue qui servent à paramétrer les composants graphiques. Elles sont simplement des interfaces graphiques de type « `Javax.Swing.JDialog` » instanciées à partir de l'interface principale soit pour créer des nouvelles instances des composants graphiques soit pour modifier un composant graphique.

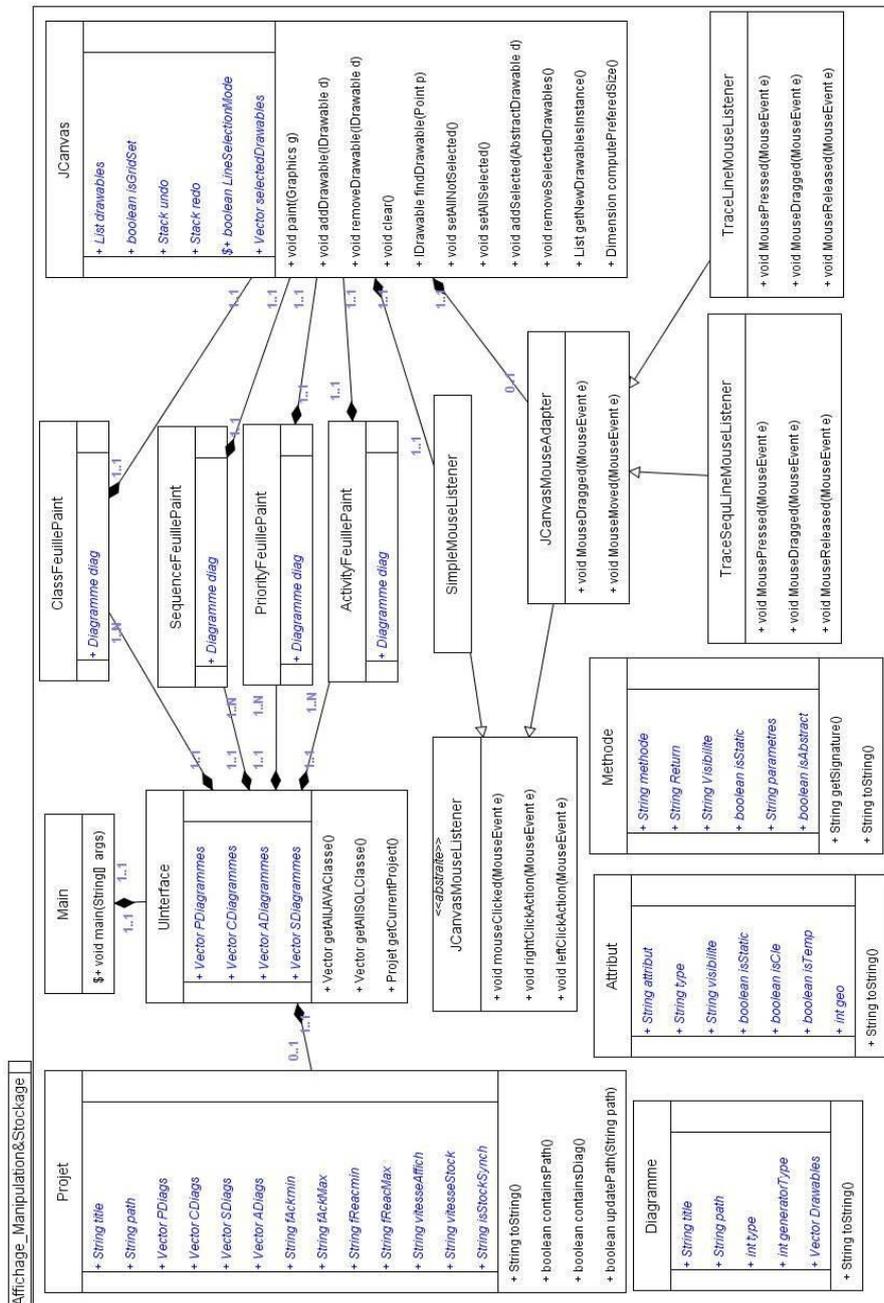


Figure 48. Diagramme de classes d'AGRT : affichage, manipulation et stockage des diagrammes.

5.2.3.2.3 Le diagramme de classes permettant la génération du code source

Ce diagramme décrit les classes permettant la génération du code source. Le code source peut être généré en JAVA ou en SQL.

Un squelette de code peut être généré à partir des diagrammes de classes dessinés avec AGRT, ce dernier peut être complété lors de la phase de développement. Le concepteur peut choisir le type de code selon le langage cible choisi. Dès que le concepteur choisit le langage spécifié, un répertoire nommé « `src` » est créé automatiquement sous le répertoire du projet.

Tous les types de générateurs de code implémentent l'interface « `Generateur` ». Il faut donc définir pour chaque générateur les méthodes « `genererCode()` » et « `exporterCode()` ».

Le code source en JAVA est généré grâce à la classe « `JAVAEntiteGenerator` » qui interprète une à une les instances de la classe « `EntiteDrawable` » et crée un fichier (*.java*) portant le nom de la classe.

Le code source en SQL est généré en passant par deux étapes. Dans une première étape les instances de « `EntiteDrawable` » sont interprétées pour générer un modèle logique à l'aide de la classe « `SQLLogicalModelGenerator` ». La deuxième étape consiste à interpréter le modèle logique en modèle physique à l'aide de la classe « `SQLCodeGenerator` ». Le code source est exporté dans un fichier texte portant le nom du projet.

Les deux classes « `CleMultiple` » et « `Association` » sont des structures de données simplifiant l'interprétation des instances de « `EntiteDrawable` » en modèle logique pour la génération du code SQL.

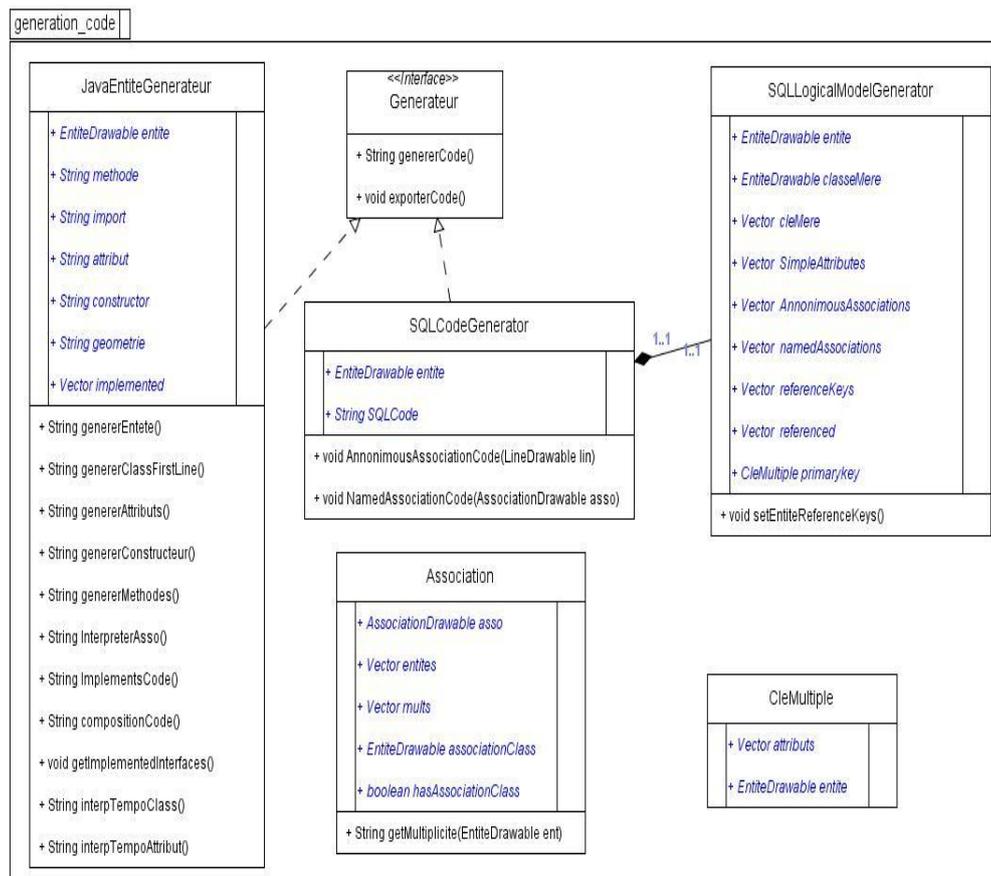


Figure 49. Diagramme de classes d'AGRT : génération du code source.

5.2.3.2.4 Exemple de génération du code source en JAVA

Considérons le diagramme de classes de la Figure 50 créé à l'aide de notre atelier AGRT.

Nous présentons la génération automatique du code source en JAVA généré par AGRT à partir de ce diagramme. Nous obtenons sept fichiers : deux fichiers pour les interfaces « interface1 » (Figure 51) et « interface2 » (Figure 52), trois fichiers pour « Classe1 » (Figure 53), « Classe2 » (Figure 54) et « ClasseMere » (Figure 55) et deux fichiers pour définir les types « Point » (Figure 56) et « Surface » (Figure 57).

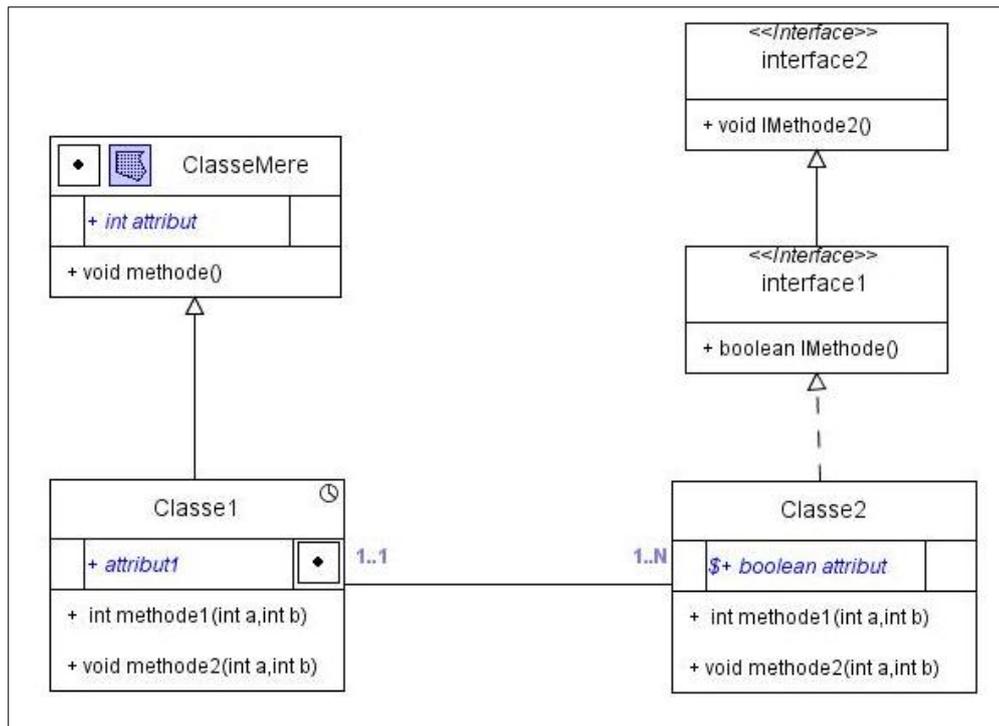


Figure 50. Exemple de diagramme de classes réalisé par AGRT.

```

/* interfacel.java
 * généré à l'aide d'AGRT */
public interface interfacel extends interface2 {
    public boolean IMethode();
}
  
```

Figure 51. Code source de l'interface « interface1 ».

```

/* interface2.java
 * généré à l'aide d'AGRT */
public interface interface2 {
    public void IMethode2();
}
  
```

Figure 52. Code source de l'interface « interface2 ».

```
/* Classe1.java
 * généré à l'aide d'AGRT */
import java.util.ArrayList;
import java.util.Date;
public class Classe1 extends ClasseMere {
    public GPoint attribut1;
    private ArrayList <Classe2> arClasse2;
    private Date dateCreation;
    private Date dateDestruction;
    private Date dateActivation;
    private Date dateDesactivation;
    public Classe1() {
        arClasse2 = new ArrayList<Classe2>();
        dateCreation=new Date(System.currentTimeMillis());
    }
    public int methode1(int a,int b){
        return null;}
    public void methode2(int a,int b){
        throw new UnsupportedOperationException("Not yet
implemented");}
    public void setActivationDate(Date d){
        if (d.after(dateCreation)){
            this.dateActivation=d;
        }
    }
    public void setDesactivationDate(Date d){
        if (!dateActivation.equals(null)){
            if (dateActivation.before(d){
                this.dateActivation=d;
            }}
    }
    public void setDestructionDate(Date d){
        if (!dateDesactivation.equals(null)){
            if (dateDesactivation.before(d){
                this.dateDestruction=d;
            }}
    }
    public void addClasse2(Classe2 iClasse2) {
        if(! arClasse2.contains ( iClasse2)) {
            iClasse2.setClasse1 (this);
            arClasse2.add ( iClasse2 );
        }
    }
    public ArrayList <Classe2> getARClasse2(){
        return ( arClasse2 );
    }
}
```

Figure 53. Code source de la classe « Classe1 ».

```
/* Classe2.java
 * généré à l'aide d'AGRT */
public class Classe2 implements interface1 {
    public static boolean attribut;
    private Classe1 iClasse1;
    public Classe2() {
    }

    public int method1(int a,int b){
        return null;
    }

    public void method2(int a,int b){
        throw new UnsupportedOperationException("Not yet
implemented");
    }

    public void addClasse1(Class1 iClasse1) {
        if ( iClasse1 != null ) {
            if( ! iClasse1.getARClasse2().contains(this)){
                this.setClasse1( iClasse1 );
                this.iClasse1.getARClasse2().add(this);
            }
        }
    }

    public void setClasse1 (Class1 iClasse1 ){
        this.iClasse1= iClasse1;
    }

    public boolean IMethode(){
        return null;
    }

    public void IMethode2(){
        throw new UnsupportedOperationException("Not yet
implemented");
    }
}
```

Figure 54. Code source de la classe « Classe2 ».

```
/* ClasseMere.java
 * généré à l'aide d'AGRT */
public class ClasseMere {
    public int attribut;
    public GSurfaceSimple geometrieAff ;
    public GPoint geometrieStock ;
    public ClasseMere() {
    }
    public void methode(){
        throw new UnsupportedOperationException("Not yet
implemented");
    }
}
```

Figure 55. Code source de la classe « ClasseMere ».

```
/* GPoint.java
 * généré à l'aide d'AGRT */
public class GPoint {
    private float x;
    private float y;
    public GPoint(float x, float y){
        this.x=x;
        this.y=y;    }
    public GPoint(float x){
        this.x=x;
        this.y=0;    }
    public GPoint(){
        this.x=0;
        this.y=0;    }
    public float getX() {
        return x;    }
    public float getY() {
        return y;    }
    public void setX(float x){
        this.x=x;    }
    public void setY(float y){
        this.y=y;    }}}
```

Figure 56. Code source de la classe « GPoint ».

```
/* GSurface.java
 * généré à l'aide d'AGRT */
import java.util.ArrayList;

public class GSurface {
    private ArrayList <GPoint> points;

    public GSurface(ArrayList <GPoint> points) {
        this.points=points;
    }
    public ArrayList <GPoint> getContour() {
        return points;
    }
    public void setContour(ArrayList <GPoint> points) {
        this.points=points;
    }
}
```

Figure 57. Code source de la classe « GSurface ».

5.3 Etudes de cas

Une application géomatique temps réel est une application qui a les caractéristiques d'une application géomatique et d'une application temps réel.

Une application géomatique se caractérise par la manipulation de données à références spatiales d'où l'introduction d'une difficulté supplémentaire.

Tandis que les applications temps réel ont en commun la prépondérance du facteur temps, cette caractéristique fondamentale distingue globalement les applications temps réel des autres types d'applications informatiques. En plus de l'existence de contraintes de temps, les systèmes temps réel doivent satisfaire d'autres contraintes primordiales, notamment la prédictibilité des comportements et la tolérance aux pannes.

Afin de valider notre AGL, nous avons conçu et réalisé plusieurs applications géomatiques temps réel à l'aide d'AGRT. Cette section applique notre profil GRT à une application géomatique temps réel choisie parmi ces dernières. Sachant qu'une autre application est détaillée en annexe C.

Nous allons tout d'abord présenter cet ensemble d'applications que nous avons conçu à l'aide d'AGRT de façon sommaire. Ensuite nous détaillons une application géomatique temps réel : « gestion des alarmes dans un réseau GSM

(Global System Mobil) » en spécifiant son cahier des charges et en présentant la partie conception et l'implémentation de cette dernière.

5.3.1 Quelques exemples d'applications géomatiques temps réel

Nous avons commencé notre démarche par la définition des applications géomatiques temps réel strictes et des applications géomatiques temps réel mou. Nous avons identifié les différents types d'applications, classé les applications par type et identifié les spécifications spatio-temporelles (données, traitements, distributions, contexte, contraintes, liens spatio-temporel et temps réel, etc). Nous avons choisi une application par classe d'applications et nous avons réalisé la conception des applications à l'aide d'AGRT. Parmi ces applications on cite : application concernant la pollution atmosphérique, application concernant l'épuration des eaux, application concernant la gestion de flotte de véhicules terrestres par GPS (Global Positioning System), application pour l'implémentation d'un système de détection d'alarmes dans les réseaux de capteurs sans fil, etc.

5.3.1.1 *Application concernant la pollution atmosphérique*

Le réseau national tunisien de contrôle de l'air contient cinq stations fixes et une station mobile. Les cinq stations sont réparties sur l'ensemble du territoire tunisien. En général on utilise la station mobile pour contrôler la qualité de l'air suite à une réclamation dans cet endroit géographique ou en cas de panne de l'une des stations fixes. Plusieurs mesures sont effectuées : des mesures tous les quarts d'heures, toutes les demi-heures, toutes les heures, des mesures journalières et des mesures mensuelles.

Il y a environ soixante paramètres à mesurer, mais il y en a qui sont plus importants que d'autres.

Comme il y a des normes à respecter, il doit y avoir des messages d'alarmes qui se déclenchent dès que les paramètres mesurés dépassent le seuil prévu. Aussi à la suite d'une panne ou d'une défaillance dans l'un des appareils de mesures dans les stations, un signal d'alarme sera affiché pour pouvoir intervenir le plus tôt possible.

5.3.1.2 *Application concernant l'épuration des eaux au niveau de l'Office National d'Assainissement (ONAS)*

Etant donné que l'ONAS est devenu le principal intervenant dans le domaine de la protection de l'environnement hybride et de la préservation des ressources hydrauliques à Tunis, celui-ci se charge du contrôle de la qualité des eaux usées qu'elles soient traitées ou non, pour s'assurer de leur conformité aux normes en vigueur. L'ONAS possède 70 stations d'épuration et 511 stations de pompage. Cette application permet de mettre en place un outil de

suivi et d'analyse des différents paramètres prélevés et mesures collectées pour contrôler la qualité de l'eau et le fonctionnement de la station, prévenir tout dépassement de seuil et éviter les procédures manuelles de transmission de données afin de limiter les erreurs dues au travail de saisie des résultats, ceci à l'aide d'un système de gestion des alarmes qui se déclenche à chaque défaillance.

5.3.1.3 *Application concernant la gestion de flotte de véhicules terrestres par GPS (Global Positioning System)*

La complexification d'un système de transport urbain a considérablement alourdi la tâche des experts du transport et des décideurs publics qui doivent faire face à plusieurs problématiques. Outre les préoccupations d'investissement et de congestion, ils doivent intégrer les problèmes de sécurité, d'efficacité du système de transport, d'équité et d'environnement durable [BOU 07].

Face aux problèmes de congestion principalement corrélés à celui du contexte économique de raréfaction des ressources publiques, à la multiplication des accidents, les solutions qui s'imposaient au début des années soixante, reposaient essentiellement sur un élargissement des voies ou sur la réalisation de rocade de contournement [MAS 00].

De ce fait, nous avons décidé de traiter une application de gestion de flotte de véhicules terrestres par GPS. La gestion de flotte de transports terrestres est très importante puisqu'elle offre à celui qui va gérer les véhicules, et qui sera confronté à de nombreux problèmes d'ordre technique et organisationnel dans la localisation en temps réel ou en temps différé, l'occasion de garantir la sécurité du conducteur, de tout l'environnement et de gérer d'une manière centralisée les flottes mobiles. Les systèmes de gestion de flotte sont des méthodes techniquement avancées pour la poursuite et la surveillance des véhicules à distance. Ils permettent de gérer une flotte de véhicules par l'acquisition des positions et du temps du trajet, etc.

Ces systèmes améliorent la gestion de la consommation d'énergie, la vitesse du trafic et le positionnement des véhicules. L'application permet d'assurer les tâches suivantes :

- le suivi de flotte des véhicules permettant de traduire les changements affectant les coordonnées d'un véhicule suivi par GPS par le repositionnement des objets correspondants sur une carte,
- la gestion des routes fermées : cette tâche nécessite une vérification permanente des états des routes par l'interrogation de la base des données, lorsqu'une route est fermée l'application se charge d'afficher un message d'alerte suivi d'une modification de couleur des routes concernées,

- la correction des états de routes : elle consiste à vérifier si les routes fermées sont revenues à leurs états fonctionnels,
- la gestion des véhicules en panne,
- le calcul des distances parcourues,
- la déclaration du débordement des véhicules,
- le calcul du plus proche service demandé.

5.3.1.4 *Application pour l'implémentation d'un système de détection d'alarmes dans les réseaux de capteurs sans fil (WSN : Wireless Sensor Networks)*

Les WSN connaissent un fort engouement ces dernières années. Les capteurs traditionnels, mesurant une grandeur physique, sont présents depuis longtemps dans les domaines comme l'industrie, l'aéronautique ou l'automobile. Ils sont en général reliés à la base de traitements par liaison filaire. La nouveauté avec les réseaux de capteurs, est qu'ils ont la possibilité de communiquer par ondes radios avec d'autres capteurs proches distants de quelques mètres tout au plus. Chaque capteur collaborant avec ses voisins proches, un réseau de capteur peut couvrir une étendue assez vaste. Comme les ressources d'un capteur sont très limitées, il serait envisageable que la réalisation d'une application puisse être le résultat de la collaboration de tous les capteurs qui exécutent un service plus simple. Afin de résister aux déploiements, ces capteurs doivent être très solides et de plus, ils doivent aussi pouvoir survivre dans les conditions les plus extrêmes dictées par leur environnement d'utilisation (feu ou inondation par exemple). En plus des contraintes environnementales, une contrainte très importante est l'économie de batterie. Les réseaux de capteurs présentent des intérêts considérables pour le secteur industriel, mais aussi pour les organisations civiles où la surveillance et la reconnaissance de phénomènes physiques est une priorité. En effet, un réseau de capteurs peut être mis en place dans le but de surveiller une zone géographique plus ou moins étendue pour détecter l'apparition de phénomènes ou mesurer une grandeur physique (température, pression, vitesse, etc.). Par exemple, on aurait pu imaginer un réseau de capteurs, situé au large des côtes indonésiennes, capable de mesurer l'amplitude de la houle dans le but de prévenir l'apparition de tsunamis. Les applications sont très nombreuses, avec des impacts importants dans des domaines aussi variés que l'industrie, la recherche, l'environnement, ou la médecine. Les chercheurs électroniciens sont confrontés à de nombreux challenges, au niveau de la taille des capteurs (miniaturisation maximale), mais aussi au niveau des performances (l'émission de messages la plus performante possible opposé à la consommation la moins importante).

Les capteurs sans fils sont des éléments indépendants les uns des autres. Par conséquent, ils doivent également disposer d'une alimentation autonome. Leur

durée de vie est limitée par la durée de vie de leur batterie puisqu'ils utilisent leur réserve d'énergie à des fins de calcul et de transmission de données. Cette contrainte forte a une influence majeure sur l'ensemble des techniques mises en place pour le déploiement de tels réseaux. Un effet majeur de cette limitation énergétique est la limitation maximale des transmissions par voie hertzienne, très coûteuses. Il est donc primordial d'effectuer tant que possible le traitement de l'information localement au niveau du nœud. Dans un WSN, chaque nœud joue le rôle d'émetteur et de routeur. Une défaillance énergétique d'un capteur peut changer significativement la topologie du réseau et imposer une réorganisation coûteuse de ce dernier.

De plus pour la tolérance aux pannes, certains nœuds peuvent être défaillants à cause d'un manque d'énergie, d'un dommage physique ou d'une interférence. Ceci ne doit pas affecter la globalité de la tâche du réseau de capteurs. En cas de défaillance, de nouveaux liens et routes doivent être établis pour assurer la collecte des données. La redondance peut également être utilisée, tout en veillant à conserver une faible consommation d'énergie. Enfin, dans diverses applications, la donnée doit être transmise dans une certaine plage de temps, après quoi, elle n'aura aucune importance. Pourtant, dans la plupart des applications, la durée de vie du réseau est favorisée au détriment de la qualité d'émission des données. Les problématiques sont donc très éloignées de celles des réseaux classiques, telle la maximisation du débit mais il s'agit plutôt d'étendre la durée de vie du système et sa robustesse, en cas de chute de certains nœuds seulement.

Le but de cette application est de répondre aux exigences d'un réseau de capteurs sans fil en l'implémentant dans une zone géographique bien déterminée. L'application traitée est capable de détecter un changement dans l'environnement et de réagir en fonction notamment des besoins de l'utilisateur afin d'acheminer vers des points de traitement les différentes données collectées. Elle permet aussi de traiter les tolérances aux pannes puisque l'absence de sécurité physique pour ce type de capteurs, et la nature vulnérable des communications radios sont des caractéristiques qui augmentent les risques de pannes sur ce type de réseau.

5.3.2 Cahier des charges et spécification pour une application géomatique temps réel : « gestion des alarmes dans un réseau GSM »

Vu la diversité des applications géomatiques temps réel, notre choix s'est porté sur une application de gestion des alarmes dans un réseau GSM.

Un environnement mobile est un système composé de sites mobiles et qui permet à ses utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques. La communication sans fil est moins fiable que la

communication dans les réseaux filaires. La propagation du signal subit des perturbations (erreurs de transfert, micro-coupure, timeout, etc.) dues à l'environnement, qui altèrent l'information transférée. Il s'ensuit alors, un accroissement du délai de transit de messages à cause de l'augmentation du nombre de retransmissions. La connexion peut aussi être rompue ou altérée par la mobilité des sites.

Un usager peut sortir de la zone de réception ou entrer dans une zone de haute interférence. Le nombre d'unités mobiles dans une même cellule (dans le cas des réseaux cellulaires), par exemple lors d'un rassemblement populaire, peut entraîner une surcharge du réseau. Rappelons que la norme GSM est aujourd'hui l'une des technologies d'accès radio mobile les plus répandues. La gestion et la maintenance du réseau GSM est l'un des plus grands soucis des opérateurs cherchant à maintenir un certain niveau de qualité afin d'augmenter le nombre de clients et de satisfaire les abonnés de leurs réseaux.

De ce fait nous avons conçu une interface qui facilite la tâche de supervision des alarmes du réseau GSM. Elle permet au superviseur de suivre la situation du réseau en affichant sur une carte les alarmes dès qu'elles se déclenchent. Elle offre au superviseur la possibilité de localiser l'entité concernée par l'alarme, de distinguer s'il s'agit d'une alarme critique, majeure ou mineure, de voir en détail le type de panne qui a causé l'alarme et l'historique des alarmes de l'entité en question. Ceci rend l'intervention de l'équipe de maintenance plus rapide et plus efficace en donnant de toute évidence une priorité aux pannes critiques.

5.3.2.1 *La technologie GSM*

Nous allons présenter tout d'abord le concept cellulaire, puis l'architecture du réseau GSM en présentant brièvement ses entités et leurs fonctions.

5.3.2.1.1 Le concept cellulaire

Les réseaux de première génération possédaient des cellules de grande taille (50 kilomètres de rayon) au centre desquelles se situait une station de base (antenne d'émission). Au tout début, ce système allouait une bande de fréquences de manière statique à chaque utilisateur qui se trouvait dans la cellule qu'il en ait besoin ou non. Ce système ne permettait donc de fournir un service qu'à un nombre d'utilisateurs égal au nombre de bandes de fréquences disponibles. La première amélioration consiste à allouer un canal à un utilisateur uniquement à partir du moment où celui-ci en avait besoin permettant ainsi d'augmenter le nombre d'abonnés, étant entendu que tout le monde ne téléphone pas en même temps. Mais ce système nécessitait toujours des stations mobiles de puissance d'émission importante et donc des appareils mobiles de taille et de poids conséquents. De plus, afin d'éviter les

interférences, deux cellules adjacentes ne peuvent pas utiliser les mêmes fréquences. Cette organisation du réseau utilise donc le spectre fréquentiel d'une manière sous optimale [DEM 04].

C'est pour résoudre ces différents problèmes qu'est apparu le concept de cellule. Le principe de ce système est de diviser le territoire en petites zones, appelées cellules, et de partager les fréquences radio entre celles-ci (Figure 58). Une cellule est de forme circulaire mais dépend en réalité de la topographie de la région qui est servie par l'antenne de la cellule. Pour plus de clarté, on peut les illustrer par des hexagones. Au centre d'une cellule on retrouve un ou un ensemble d'émetteurs-récepteurs correspondant à une bande de fréquences.

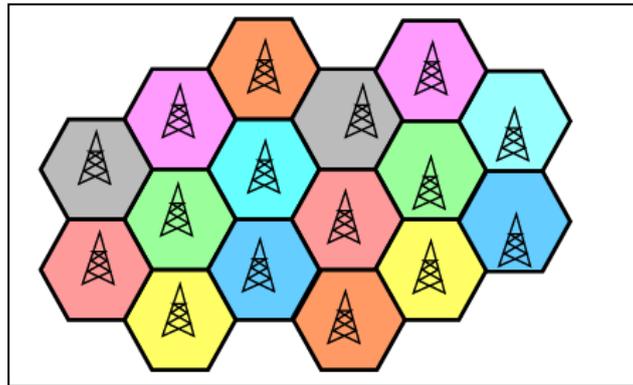


Figure 58. Le concept cellulaire.

Ces fréquences ne peuvent pas être utilisées dans les cellules adjacentes afin d'éviter les interférences. Ainsi, on définit des motifs, aussi appelés clusters, constitués de plusieurs cellules, dans lesquels chaque fréquence est utilisée une seule fois. Le nombre de cellules dans un cluster doit être déterminé de manière à ce que ce dernier puisse être reproduit continuellement sur le territoire à couvrir. Typiquement, le nombre de cellules par clusters est de 4, 7, 12 ou 21. La forme et la dimension des clusters et le nombre de cellules est en fonction du nombre de fréquences disponibles. La taille des cellules peut varier entre 0.5 et 35 km et dépend de la densité d'utilisateur et de la topographie.

La dimension d'une cellule est en fonction de la puissance de son émetteur-récepteur. Si un émetteur-récepteur est très puissant, alors son champ d'action sera très vaste, mais sa bande de fréquence peut être rapidement saturée par des communications. En revanche, en utilisant des cellules plus petites, c'est-à-dire émetteur-récepteur moins puissant alors la même bande de fréquence pourra être réutilisée plus loin, ce qui augmente le nombre de communications possibles.

Dans la conception d'un réseau cellulaire, il faut considérer plusieurs aspects parmi lesquels on cite :

- la topographie (bâtiments, collines, montagnes, etc.),
- la densité de la population (ou de communications) pour établir la dimension de la cellule,
- deux cellules adjacentes ne peuvent pas utiliser la même bande de fréquence afin d'éviter les interférences. La distance entre deux cellules ayant la même bande doit être de deux à trois fois le diamètre d'une cellule.

Un opérateur devra donc tenir compte des contraintes du relief topographique et des contraintes urbanistiques pour dimensionner les cellules de son réseau.

On distingue pour cela trois services principaux [DEM 04] :

1. Le service « Outdoor » : il indique les conditions nécessaires pour le bon déroulement d'une communication téléphonique en extérieur.
2. Le service « Incar » : il tient compte des utilisateurs se trouvant dans une voiture. On ajoute typiquement une marge supplémentaire de 6 décibel Watt, dans le bilan de puissance pour en tenir compte.
3. Le service « Indoor »: il permet le bon déroulement des communications à l'intérieur des bâtiments. Cette catégorie de service se subdivise à son tour en deux services :
 - le « Soft Indoor » lorsque l'utilisateur se trouve juste derrière la façade d'un bâtiment,
 - le « Deep Indoor » lorsque l'utilisateur ne se trouve plus à l'intérieur.

5.3.2.1.2 L'architecture du réseau GSM

L'architecture du réseau GSM peut être divisée en trois sous-systèmes (Figure 59) : le sous-système radio, le sous-système réseau ou d'acheminement et le sous-système opérationnel ou d'exploitation et de maintenance [LAG 02].

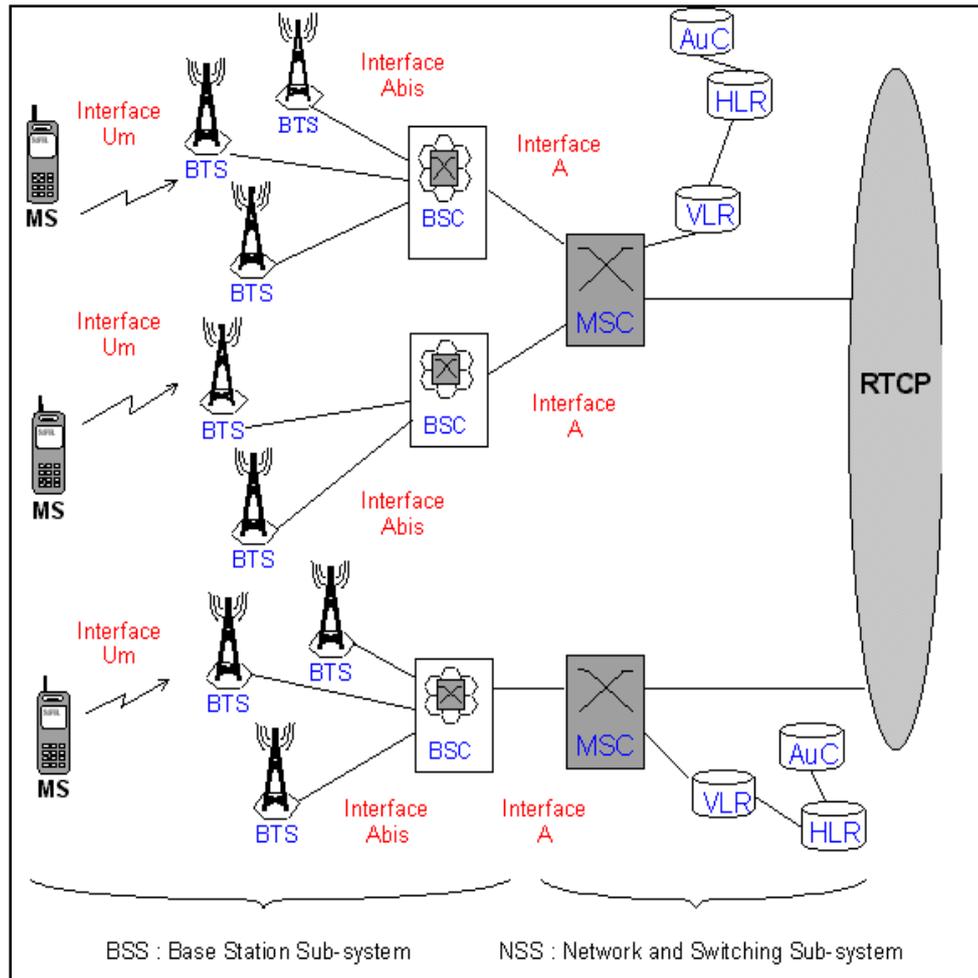


Figure 59. L'architecture du Réseau GSM.

A. Le sous système radio

Le sous-système radio « BSS (Base Station Subsystem) » gère la transmission radio. Il est constitué de plusieurs entités dont le mobile, le contrôleur de station de base BSC (Base Station Controller) et la ou les stations de base BTS (Base Transceiver Station) qui en dépendent.

A.1. *Le mobile*

Dans un réseau GSM, le terminal mobile peut prendre trois aspects :

- le téléphone de voiture,
- le portable d'une puissance de 8 W,
- le portatif (terminal de poche), d'un poids compris entre 150 et 350 grammes et d'une puissance d'environ 2W.

Le terminal est scindé en deux parties :

- le combiné téléphonique identifié par un numéro unique l'IMEI (International Mobile Equipment Identity) qui est l'identité internationale spécifique à chaque combiné. En pratique ce numéro n'est que peu utilisé,
- la carte SIM (Subscriber Identity Module) contient toutes les informations d'abonnement parmi lesquelles on cite :
 - le numéro d'identification temporaire attribué par le réseau qui permet la localisation et qui est utilisé sur les canaux radio TMSI (Temporary Station Identity),
 - les algorithmes de chiffrement,
 - le code permettant son blocage, une carte SIM se bloque automatiquement après un certain nombre d'erreurs sur le code entré par l'utilisateur.

Cet ensemble permet d'accéder aux services d'un PLMN (Public Land Mobile Network) GSM. Cette découpe permet à l'utilisateur d'utiliser n'importe quel terminal GSM car son identification complète est portée par la carte SIM. L'établissement d'une communication commence toujours par une phase d'authentification durant laquelle le réseau dialogue avec la carte SIM.

A.2. La station de base ou BTS

La station de base est l'élément central, que l'on pourrait définir comme un ensemble émetteur-récepteur pilotant une ou plusieurs cellules. Dans le réseau GSM, chaque cellule principale au centre de laquelle se situe une station de base peut être divisée, grâce à des antennes directionnelles, en plus petites cellules qui sont des portions de celle du départ et qui utilisent des fréquences porteuses différentes.

C'est la station de base qui fait le relais entre le mobile et le sous système réseau. Comme le multiplexage temporel est limité à huit intervalles de temps, une station de base peut gérer tout au plus huit connexions simultanées par cellule. Elle réalise les fonctions de la couche physique et de la couche liaison de données.

Parfois on peut exploiter une station de base localement ou par télécommande à travers son contrôleur de station de base.

A.3. Le contrôleur de station de base ou BSC

Le contrôleur de station de base est l'organe «intelligent» du BSS. Il a pour fonction principale de gérer la ressource radio. Ce contrôleur remplit différentes fonctions tant au niveau de la communication qu'au niveau de l'exploitation.

Pour les fonctions des communications des signaux en provenance des stations de base, le BSC agit comme un concentrateur puisqu'il transfère les communications provenant des différentes stations de base vers une sortie

unique. Dans l'autre sens, le contrôleur commute les données en les dirigeants vers la bonne station de base.

En même temps, le BSC remplit le rôle de relais pour les différents signaux d'alarme destinés au centre d'exploitation et de maintenance. Il alimente aussi la base de données des stations de base. Enfin, une dernière fonctionnalité importante est la gestion des ressources radio pour la zone couverte par les différentes stations de base qui y sont connectées. En effet, le contrôleur gère les transferts inter-cellulaires des utilisateurs dans sa zone de couverture, c'est-à-dire quand une station mobile passe d'une cellule à une autre. Il doit alors communiquer avec la station de base qui va prendre en charge l'abonné et lui communiquer les informations nécessaires tout en avertissant la base de données locale VLR (Visitor Location Register) de la nouvelle localisation de l'abonné.

C'est donc un maillon très important de la chaîne de communication et il est, de plus, le seul équipement de ce sous système à être directement gérable (via l'interface X25 qui le relie au sous-système d'exploitation et de maintenance).

B. Le sous système réseau

Le sous-système réseau, appelé Network Sub-System (NSS), joue un rôle essentiel dans un réseau mobile. Le rôle principal de ce sous-système est de gérer les communications entre les abonnés et les autres usagers. Alors que le sous-réseau radio gère l'accès radio, les éléments du NSS prennent en charge toutes les fonctions de contrôle et d'analyse d'informations contenues dans des bases de données nécessaires à l'établissement de connexions utilisant une ou plusieurs des fonctions suivantes : chiffrement, authentification ou roaming, etc.

Le NSS est constitué du centre de commutation mobile « MSC (Mobile-services Switching Centre) », de l'enregistreur de localisation nominale « HLR (Home Location Register) », du centre d'authentification « AuC (Authentication Center) », de l'enregistreur de localisation des visiteurs « VLR (Visitor Location Register) » et de l'enregistreur des identités des équipements « EIR (Equipment Identity Register) ».

B.1. Le centre de commutation mobile (MSC)

Le centre de commutation mobile peut être considéré comme le cœur d'un système cellulaire étant donné qu'il fait la gestion des appels et de tout ce qui est lié à l'identité des abonnés, à leur enregistrement et à leur localisation. Le MSC agit comme un nœud d'un réseau commuté.

Son principal rôle est d'assurer la commutation entre les abonnés du réseau mobile et ceux du réseau commuté public (RTC « Réseau Téléphonique Commuté ») ou de son équivalent numérique, le réseau RNIS « Réseau Numérique à Intégration de Service ». D'un point de vue fonctionnel, il est semblable à un commutateur de réseau ISDN « Integrated Services Digital

Network », mis à part quelques modifications nécessaires pour un réseau mobile. De plus, il participe à la fourniture des différents services aux abonnés tels que la téléphonie, les services supplémentaires et les services de messagerie. Il y a quatre bases de données qui sont associées au MSC : le HLR, le VLR, l'EIR et l'AuC.

B.2. L'enregistreur de localisation nominale (HLR)

Il existe au moins un enregistreur de localisation par réseau PLMN. Il s'agit d'une base de données contenant les informations essentielles sur les abonnés appartenant à la région desservie par le commutateur de services mobiles. Cette base de données contient également la position courante de ses abonnés.

Le HLR contient :

- toutes les informations relatives aux abonnés : le type d'abonnement, la clé d'authentification Ki (cette clé est connue d'un seul HLR et d'une seule carte SIM), les services souscrits, l'identité internationale de l'abonné utilisée par le réseau (IMSI « International Mobile Subscriber Identity »), etc,
- ainsi qu'un certain nombre de données dynamiques telles que la position de l'abonné dans le réseau et l'état de son terminal (allumé, éteint, en communication, libre, etc.).

B.3. Le centre d'authentification (AuC)

La base de données AuC, est une base de données protégée qui contient une copie d'une clé secrète inscrite sur la SIM de chaque abonné. Cette dernière est utilisée pour vérifier l'authenticité de l'abonné et pour l'encryptage des données envoyées. Ce contrôle se fait par l'intermédiaire de canaux radio. Un AuC est en général associé à chaque HLR.

B.4. L'enregistreur de localisation des visiteurs (VLR)

Cette base de données ne contient que des informations dynamiques et est liée à un MSC. Elle contient des données dynamiques qui lui sont transmises par le HLR avec lequel elle communique lorsqu'un abonné entre dans la zone de couverture du centre de commutation mobile auquel elle est rattachée. Par conséquent, lorsque l'abonné quitte cette zone de couverture, ses données sont transmises à un autre VLR.

B.5. L'enregistreur des identités des équipements (EIR)

Malgré les mécanismes introduits pour sécuriser l'accès au réseau et le contenu des communications, le téléphone mobile doit potentiellement pouvoir accueillir n'importe quelle carte SIM de n'importe quel réseau. Il est donc imaginable qu'un terminal puisse être utilisé par un voleur sans qu'il ne puisse être repéré. Pour éviter ce risque, chaque terminal reçoit un identifiant unique IMEI qui ne peut pas être modifié sans altérer le terminal. En fonction de données au sujet d'un terminal, un opérateur peut décider de refuser l'accès au

réseau à un terminal qui a été déclaré perdu ou volé. Sachant que tous les opérateurs n'implémentent pas une telle base de données.

C. Les interfaces

Il existe plusieurs types d'interfaces telles que l'interface « Um », l'interface « Abis », l'interface « A » et l'interface « Q3 ».

C.1. Interface Um

L'interface Um est appelée aussi interface Air ou radio, entre BTS et MS (Mobile Station) s'appuie sur le protocole LAPDm (Link Access Protocol on the D mobile channel), qui est un protocole de liaisons de données. Il est utilisé pour le transport du trafic et des données de signalisation. Le téléphone mobile et le sous système radio communiquent par l'intermédiaire de l'interface Um, qui est une liaison radio.

C.2. Interface Abis

L'interface Abis est l'interface entre les deux composants du sous système BSS : la BTS et le BSC.

C.3. Interface A

L'interface A sépare le NSS du BSS.

C.4. Interface Q3

Le profil Q3 possède la structure en sept couches du modèle OSI (Open System Interconnexion). Les couches basses qui transportent les informations possèdent un profil X25.

5.3.2.2 Modélisation de l'application

Etant donné que les alarmes dans un réseau GSM diffèrent d'un système à un autre, nous modélisons les différents types d'alarmes selon le sous système BSS Alcatel B6.2. Ces différentes classes d'alarmes sont détaillées en annexe A.

Au niveau de cette partie nous avons modélisé trois types de diagrammes selon le profil GRT et en utilisant notre AGL AGRT :

- le diagramme de priorités qui décrit l'environnement du système, ses objectifs et ses besoins. Ce diagramme est un diagramme de cas d'utilisation avec succession d'actions,
- le diagramme de classes qui exprime de manière générale la structure statique de notre système en exprimant les différentes classes utilisées,
- le diagramme d'interventions, qui permet de modéliser les échanges d'informations entre les différents objets. Ce diagramme exprime les interactions entre objets selon un point de vue temporel en insistant sur la chronologie des envois de messages.

5.3.2.2.1 Le diagramme de priorités

Le système est composé du sous système BSS (BTS et BSC) et du centre d'opération et maintenance OMC (Operation and Maintenance Center).

En fonction des différentes tâches assurées, nous avons partagé nos besoins en quatre packages : package supervision, package authentification, package système et package maintenance (Figure 60).

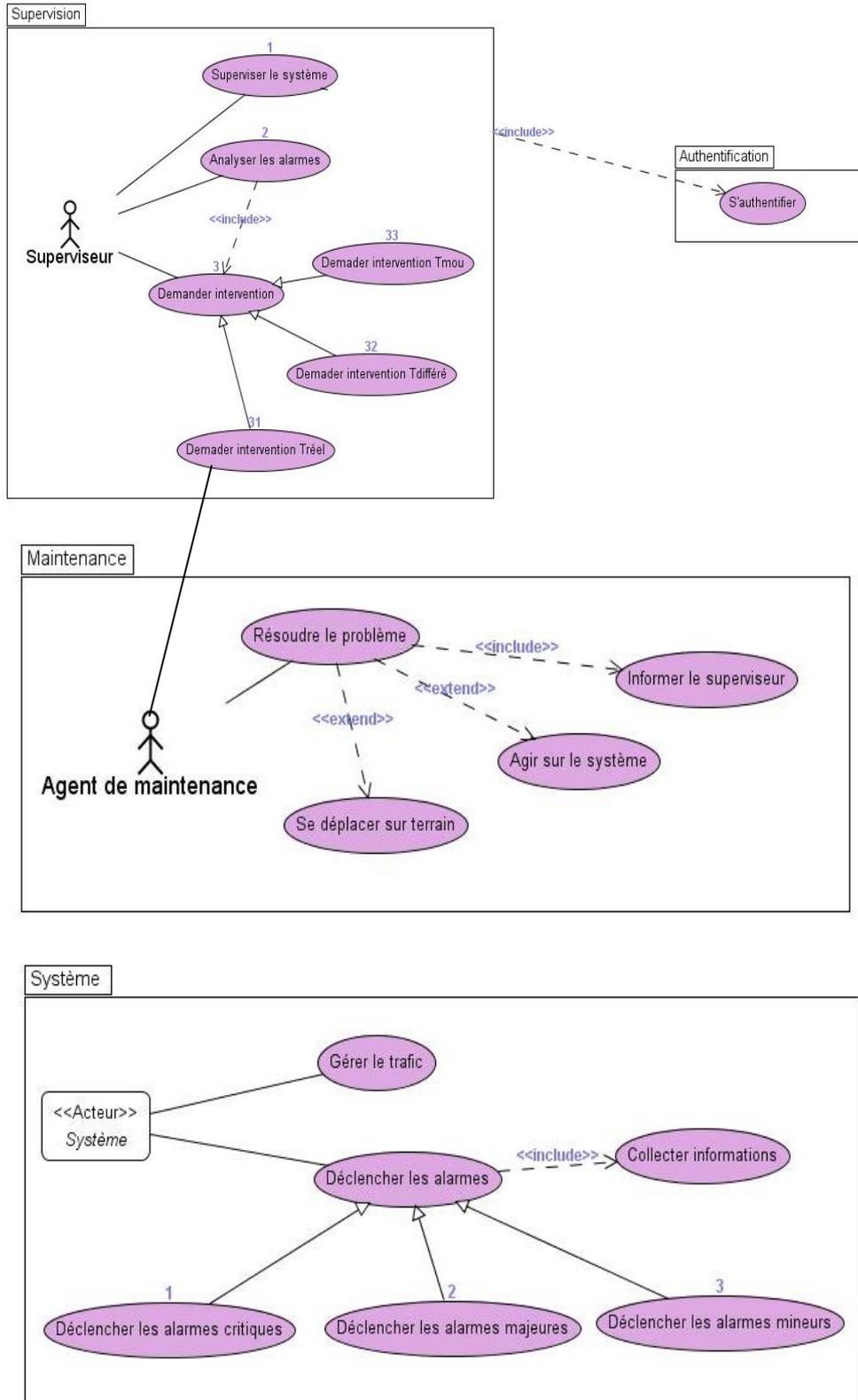


Figure 60. Diagramme de priorités.

Au niveau du package supervision le superviseur est appelé à superviser en premier lieu tout le système, puis à analyser les alarmes, pour savoir s'il s'agit d'une alarme critique, majeure ou mineure et enfin à consulter l'équipe de maintenance pour demander une intervention en temps réel, en temps mou ou en temps différé selon le type d'alarme déclenchée. Tous les cas d'utilisation de la tâche supervision sont précédés par une phase d'authentification. Ceci est modélisé par la relation d'inclusion entre le package supervision et le package authentification. Le package système décrit les tâches du système qui sont la gestion du trafic et la collecte d'informations pour déclencher les alarmes en cas de problème. Pour le déclenchement d'alarmes il y a un ordre de priorités qu'on doit respecter il faut en premier lieu déclencher les alarmes critiques c'est-à-dire là où on doit réagir en temps réel, puis déclencher les alarmes majeures (quand on doit réagir en temps différé) et en fin déclencher les alarmes mineurs (réagir en temps mou). La réparation des pannes est la tâche de l'équipe de maintenance. Elle est décrite par le package maintenance. Le cas d'utilisation « Résoudre le problème » permet de résoudre le problème soit par un déplacement sur terrain, soit en agissant sur le système si la panne est technique. A la fin de l'intervention l'équipe de maintenance doit informer le superviseur pour qu'il vérifie l'état du système.

5.3.2.2.2 Le diagramme de classes

Le diagramme de classes que nous avons conçu est composé de trois classes d'applications : « SGBD-Géographique », « Site-de-supervision » et « SGBD-Alarmes ». Ces dernières sont reliées entre elles par des associations de connexions (Figure 61).

Sachant que les classes d'applications permettent de définir les différentes applications, selon un découpage choisi par le géomaticien.

Les deux associations de connexions utilisées sont les suivantes : « Connexion_SGBD » et « Connexion_alarms ».

Connexion_SGBD : chaque Site-de-supervision doit être capable de communiquer avec le SGBD-Géographique qui peut avoir d'un à plusieurs clients.

Connexion_alarms : chaque Site-de-supervision doit être capable de communiquer avec le SGBD-Alarmes qui peut avoir d'une à plusieurs alarmes.

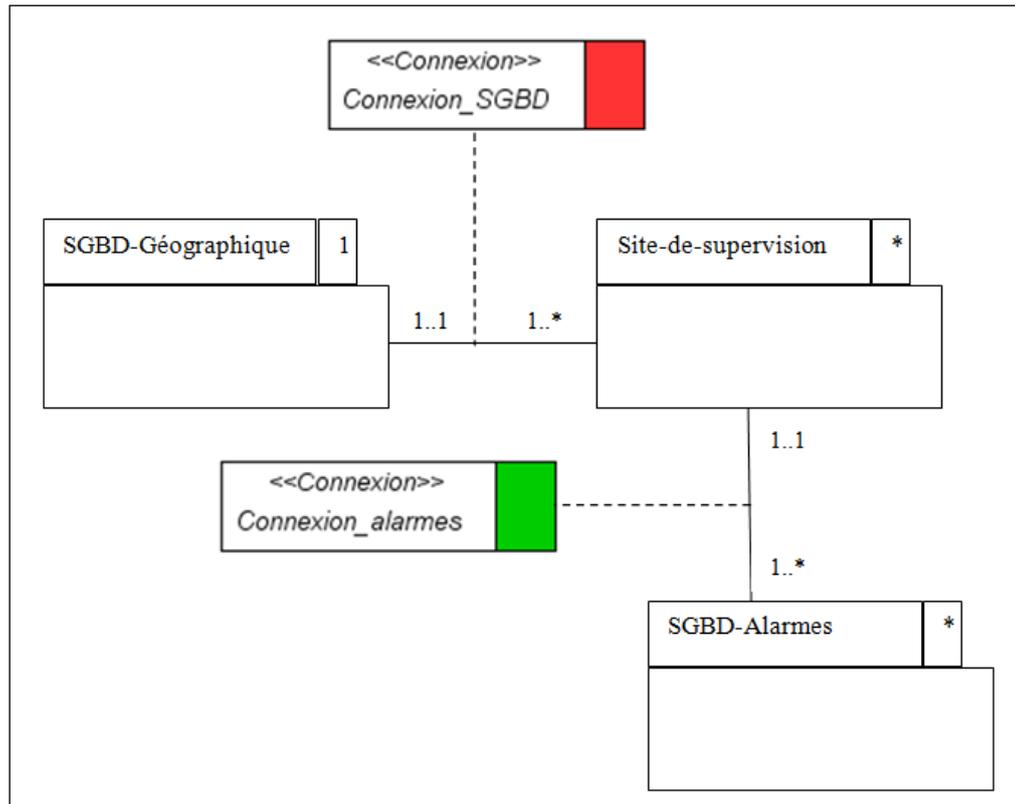


Figure 61. Diagramme de classes d'applications.

Présentation des différentes classes d'applications :

La classe d'application SGBD-Géographique : elle est composée des différentes classes géographiques qui interviennent dans le système (Figure 62).

Nous avons défini différentes classes géographiques :

- une classe « BTS » qui désigne les BTS, pour laquelle nous avons défini une géométrie de stockage de type point,
- une classe « BSC » qui désigne les BSC, pour laquelle nous avons défini une géométrie de stockage de type point,
- une classe « OMC » qui spécifie l'OMC pour laquelle nous avons défini une géométrie de stockage de type point,
- une classe « Zone_localisation » qui spécifie la zone de localisation pour laquelle nous détectons les différentes alarmes dans le réseau. Pour cette classe nous avons défini une géométrie de stockage de type surface,

Nous avons aussi défini des classes d'associations topologiques de type ligne : la classe d'association « Lien_Abis » (modélisant le lien entre BTS et BSC) et la classe d'association « Liaison_X25 » (lien entre BSC et OMC).

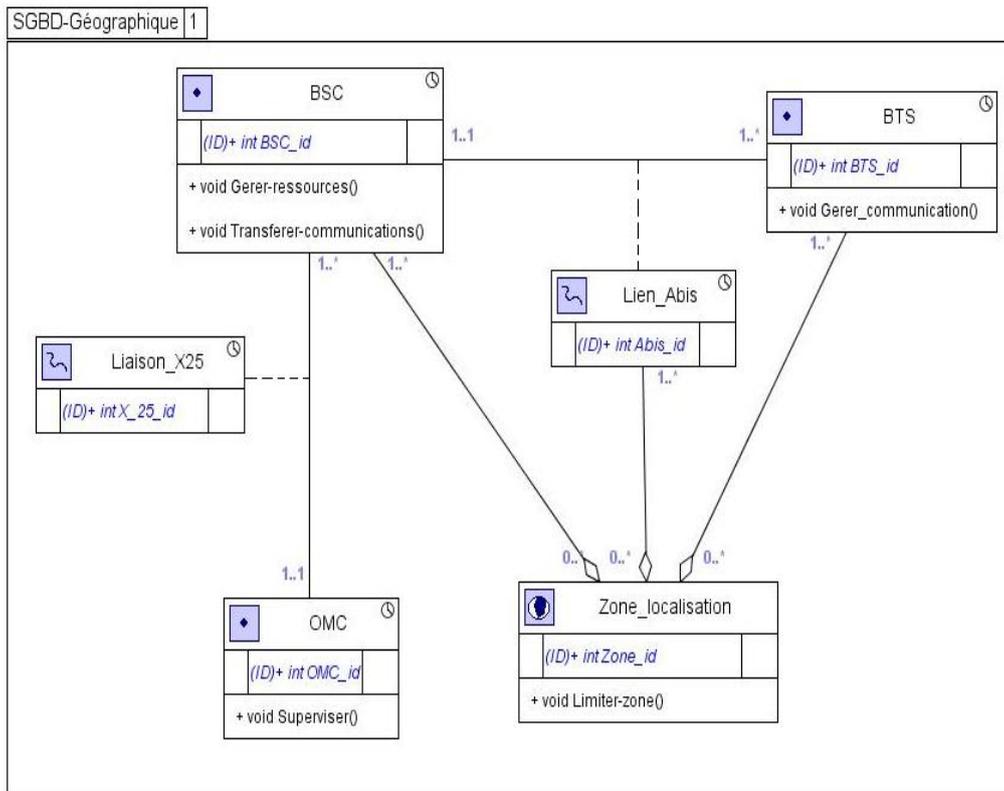


Figure 62. La classe d'application SGBD-Géographique.

La classe d'application Site-de-supervision : elle est composée de la classe principale « Plan_reseau » qui assure la visualisation des alarmes sur le plan de localisation grâce aux informations sémantiques et géographiques. Ces informations sont représentées respectivement par les classes importées « BTS_site », « BSC_site », « OMC_site », « Lien_Abis_site », « Liaison_X25_site » et « Zone_local_site » de l'application « SGBD-Géographique » et « Alarme_BTS_site », « Alarme_BSC_site », « Alarme_X25_sie » et « Alarme_Abis_site » de la classe d'application « SGBD-Alarmes » (Figure 63).

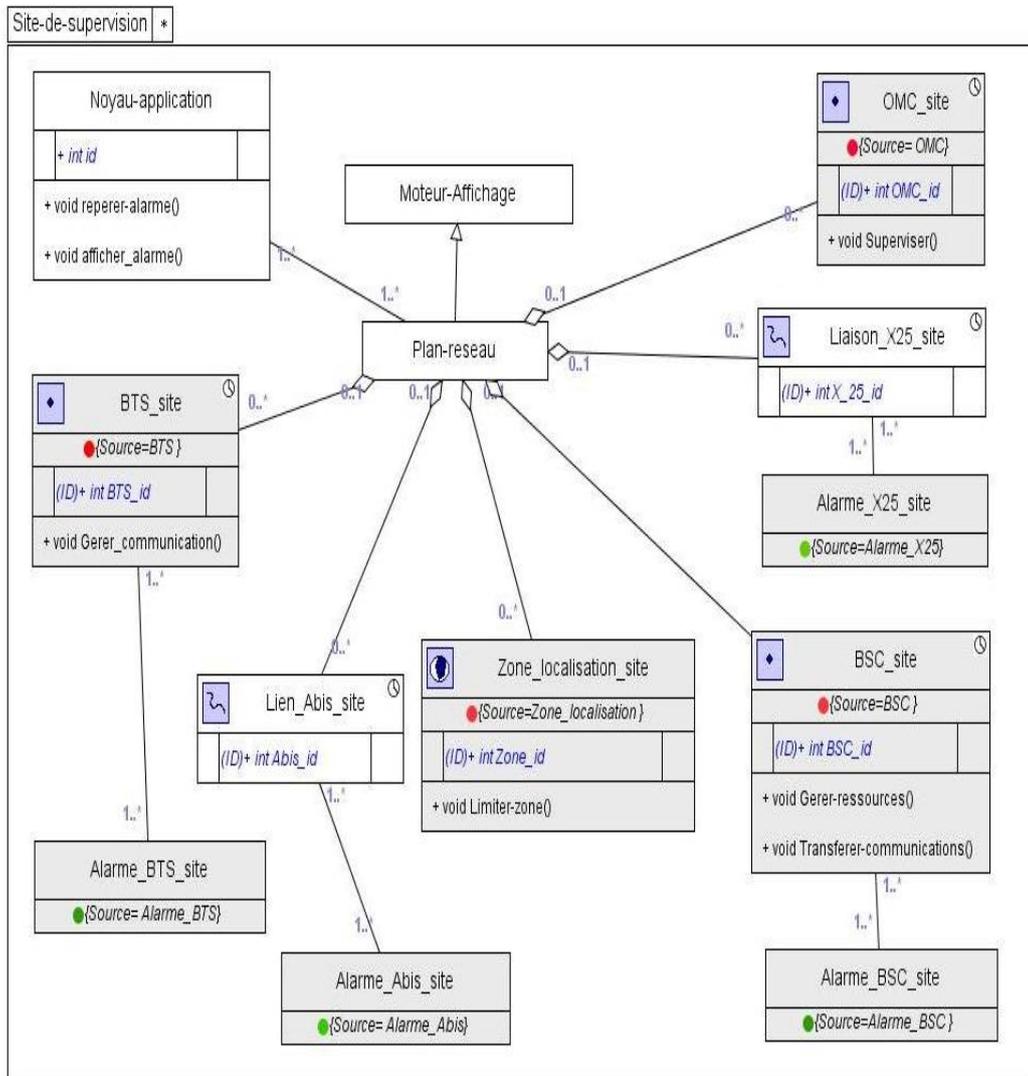


Figure 63. La classe d'application Site-de-supervision.

La classe d'application SGBD-Alarmes : elle est essentiellement composée des différentes classes d'alarmes qui affectent le système (Figure 64).

Dans notre conception, nous avons ajouté une autre classification d'alarmes selon leurs gravités et l'urgence de l'intervention des agents de l'OMC et en se basant sur la classification du système BSS Alcatel B6.2.

Cette classe contient une super-classe la classe temporelle « Alarme » qui est une classe abstraite puis grâce à un lien d'héritage on définit quatre classes filles : la classe « Alarme_BTS », la classe « Alarme_BSC », la classe « Alarme_Abis » et la classe « Alarme_X25 ».

Chaque classe fille sera elle-même une classe mère de plusieurs autres classes filles qui spécifieront la localisation de l'alarme au niveau du réseau BSS. Pour que le modèle conceptuel soit lisible nous avons modélisé les classes filles des différents types d'alarmes séparément.

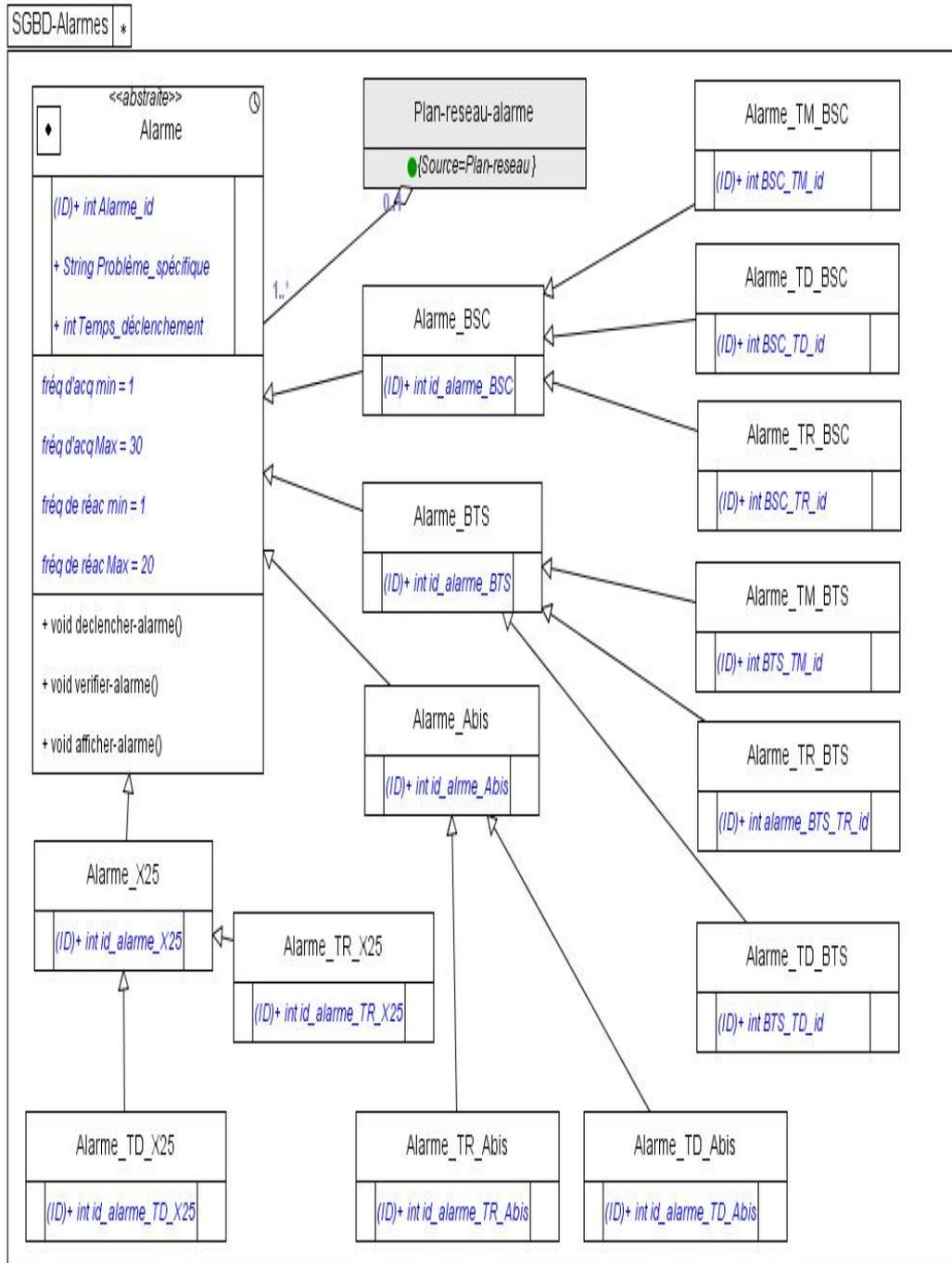


Figure 64. La classe d'application SGBD-Alarmes.

- La classe « Alarme_BTS » concerne les différentes alarmes affectant la BTS. Elle possède trois sous classes filles :
 - la classe « Alarme_TR_BTS » concerne les alarmes affectant la BTS et qui exigent une correction immédiate. Elle même est une classe mère d'autres classes filles (Figure 65),
 - la classe « Alarme_TD_BTS » caractérise les alarmes affectant la BTS qui doivent être traitées en temps différé. Elle possède trois classes filles (Figure 66),
 - la classe « Alarme_TM_BTS » spécifie les alarmes qui n'affectent pas vraiment le fonctionnement du réseau mais diminuent, tout de la qualité de service où touchent la sécurité de ces éléments. Elle possède deux classes filles (Figure 67).

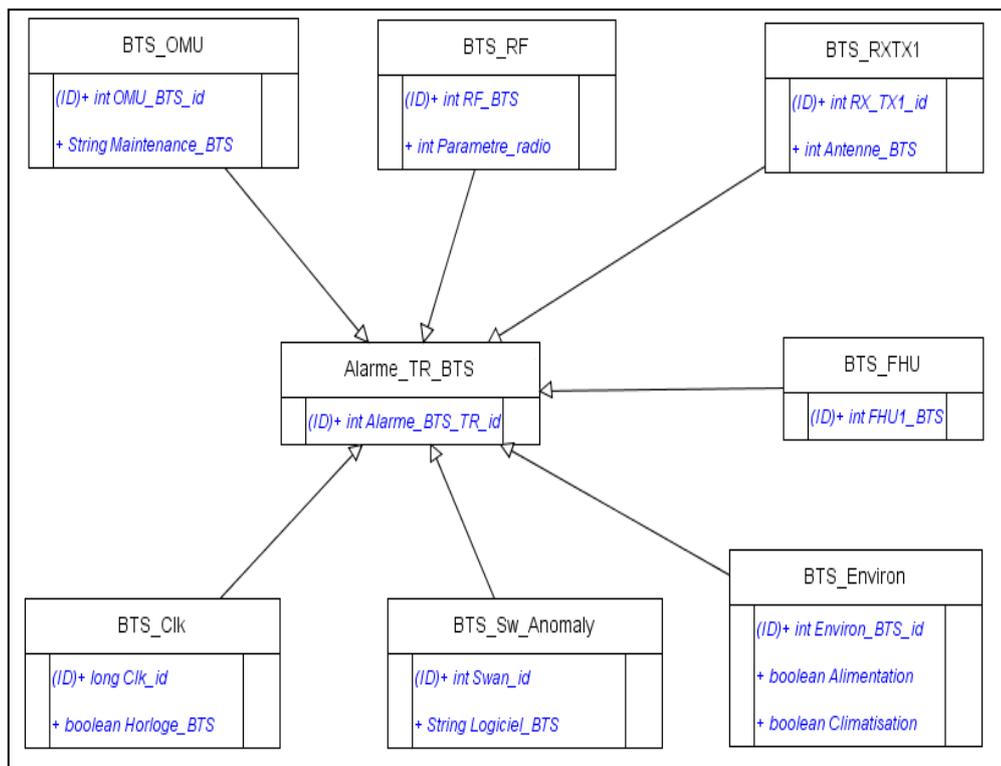


Figure 65. Les sous classes de la classe « Alarme_TR_BTS ».

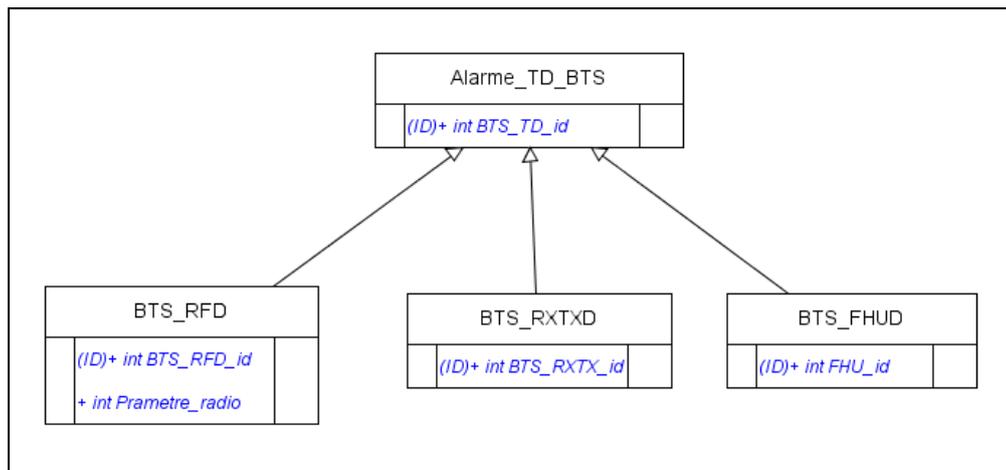


Figure 66. Les sous classes de la classe « Alarme_TD_BTS ».

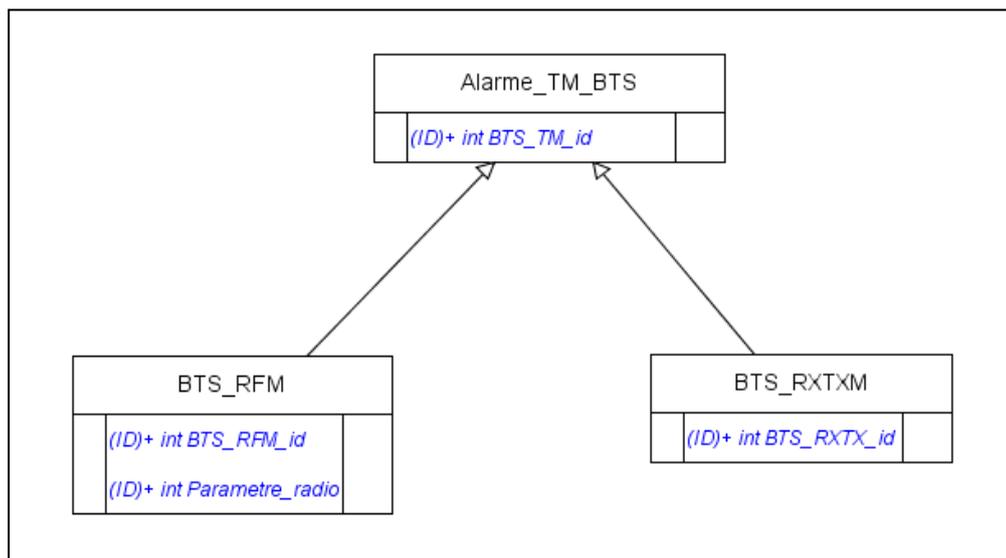


Figure 67. Les sous classes de la classe « Alarme_TM_BTS ».

- La classe « Alarme_BSC » concerne les différentes alarmes affectant le BSC. Elle possède aussi trois sous classes filles :
 - la classe « Alarme_TR_BSC » concerne les alarmes affectant le BSC et qui exigent une correction en temps réel. Elle est une classe mère de plusieurs classes filles (Figure 68),
 - la classe « Alarme_TD_BSC » concerne les alarmes affectant le BSC et qui doivent être traitées en temps différé. Elle possède trois classes filles (Figure 69),

- la classe « `Alarme_TM_BSC` » caractérise les alarmes qui peuvent être traitées en temps mou. Elle contient trois classes filles (Figure 70).

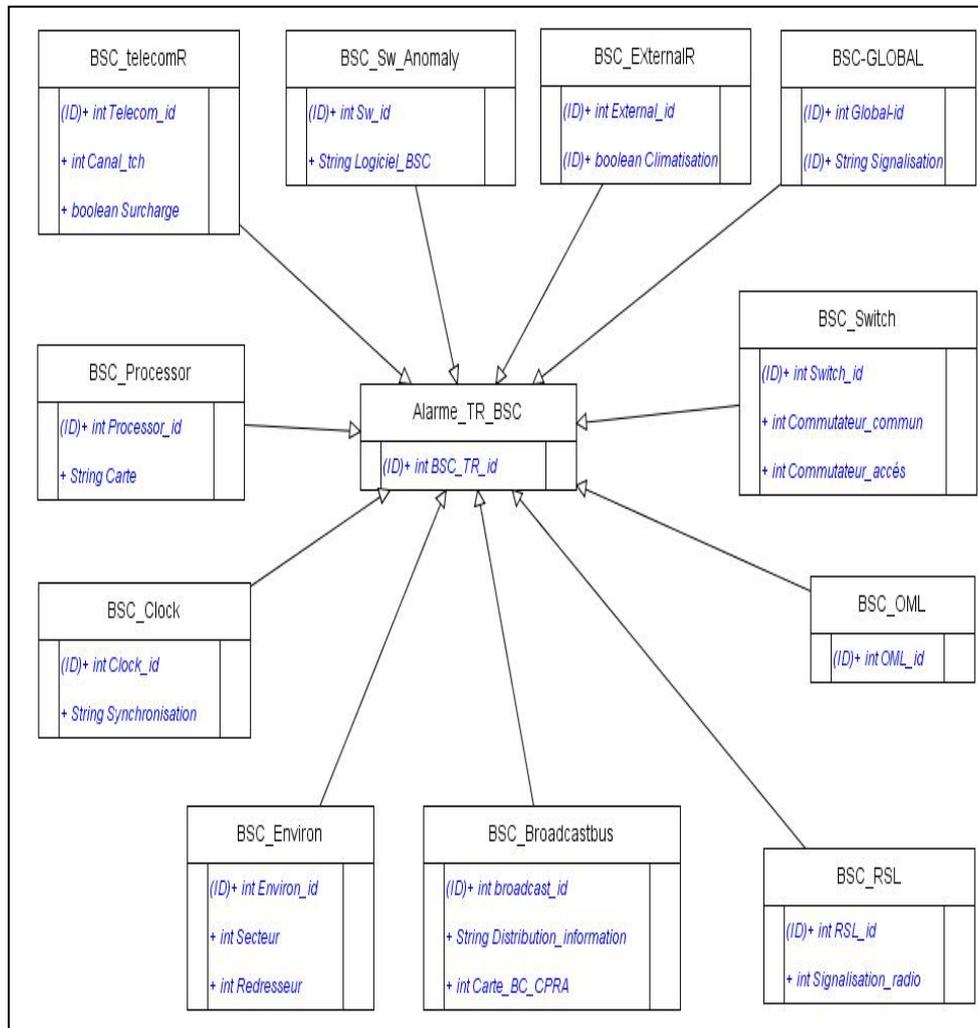


Figure 68. Les sous classes de la classe « `Alarme_TR_BSC` ».

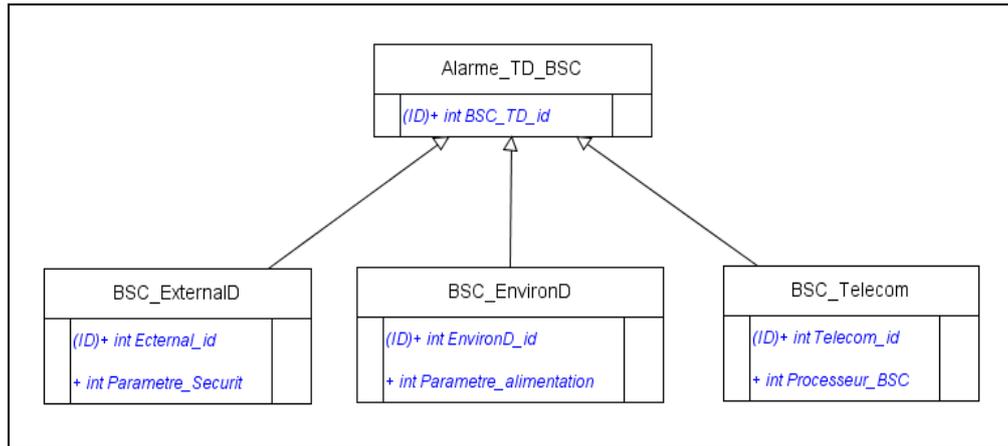


Figure 69. Les sous classes de la classe « Alarme_TD_BSC ».

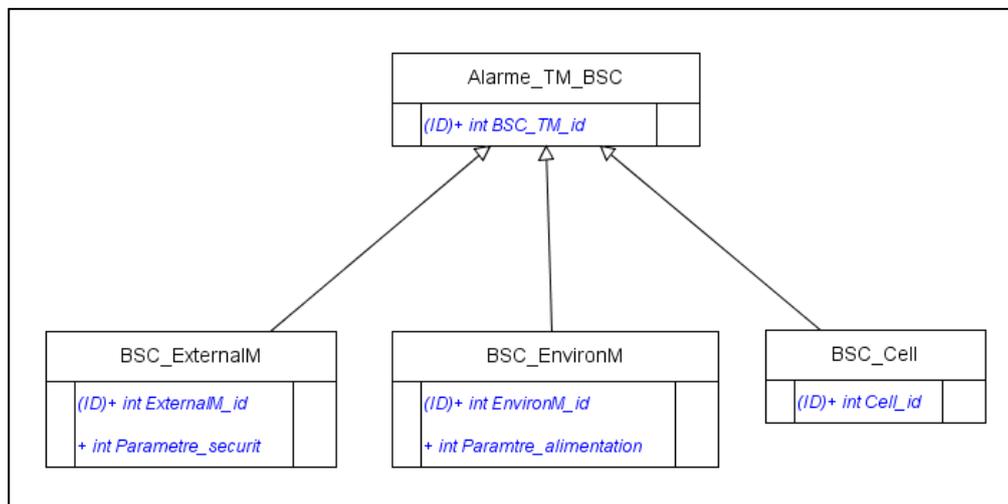


Figure 70. Les sous classes de la classe « Alarme_TM_BSC ».

5.3.2.2.3 Les diagrammes d'interventions

Le diagramme d'interventions permet de donner une vue dynamique du système à travers des scénarios. Un scénario décrit les échanges d'informations entre les différents objets et/ou acteurs. Pour notre application, nous avons décrit trois scénarios.

Dans un premier scénario nous avons simulé l'apparition d'une façon générale d'une alarme qui doit être traitée en temps réel que nous l'avons considéré comme alarme critique et nous avons décrit les interactions entre le superviseur, l'équipe de maintenance et le système qui est un objet géographique combiné pour réparer la panne. Pour mettre l'accent sur le fait que l'intervention doit être en temps réel (panne critique), un temporisateur

(représenté par un sablier) est armé dès que l'équipe de maintenance reçoit un ordre d'intervention immédiate. Après une première intervention, l'équipe de maintenance consulte le superviseur qui consulte à son tour le système pour vérifier l'état du système. Si le problème persiste, le superviseur demande une autre fois l'intervention de l'équipe de maintenance jusqu'à réparation de la panne (Figure 71).

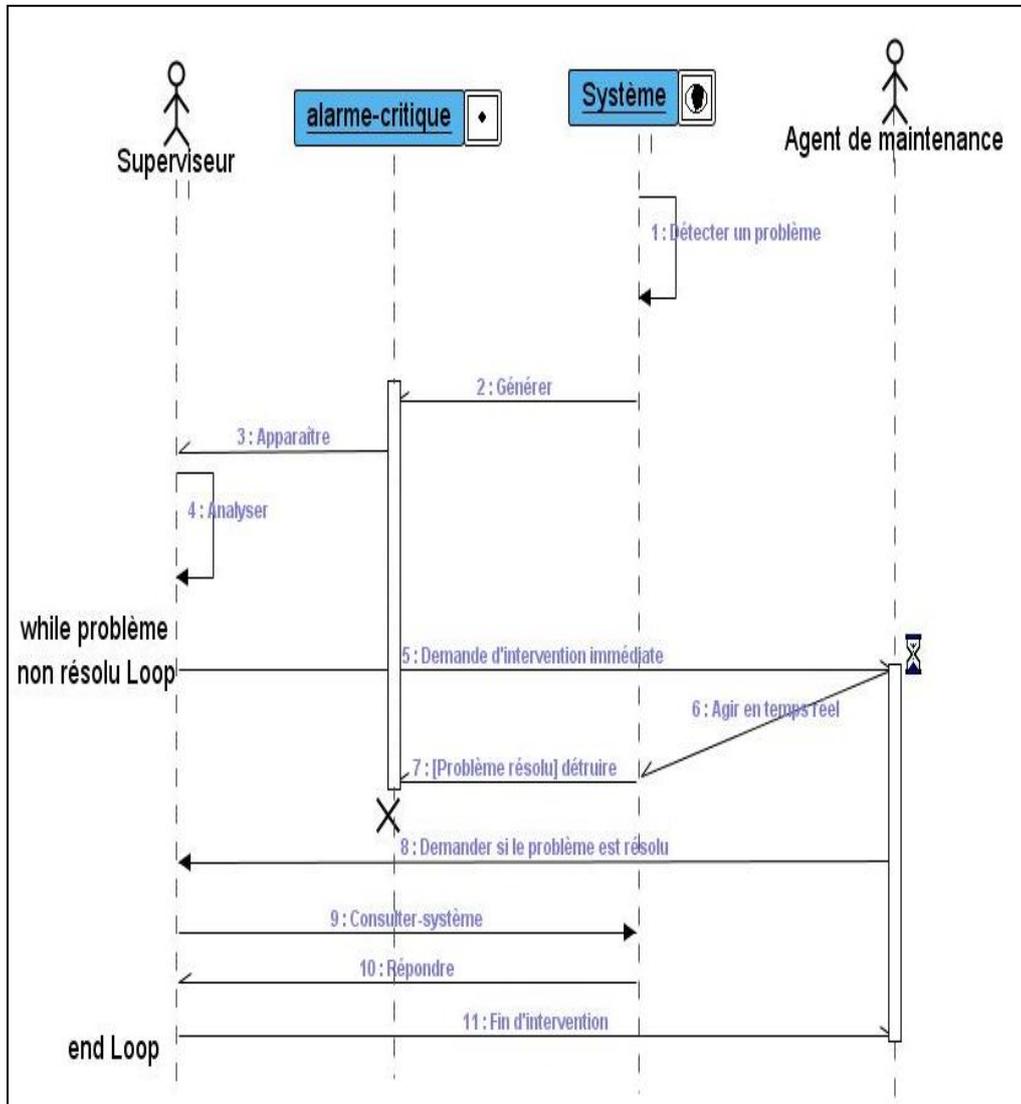


Figure 71. Scénario 1 : réaction en cas de déclenchement d'une alarme critique.

Dans un deuxième scénario, nous avons fait la même chose mais pour une panne majeure, c'est-à-dire une panne qui demande une intervention en temps différé. La seule différence est que la contrainte temporelle est moins stricte

que dans le premier scénario. On parle ici d'intervention en temps différé, elle est représentée dans le diagramme par une durée de 35 minutes (Figure 72).

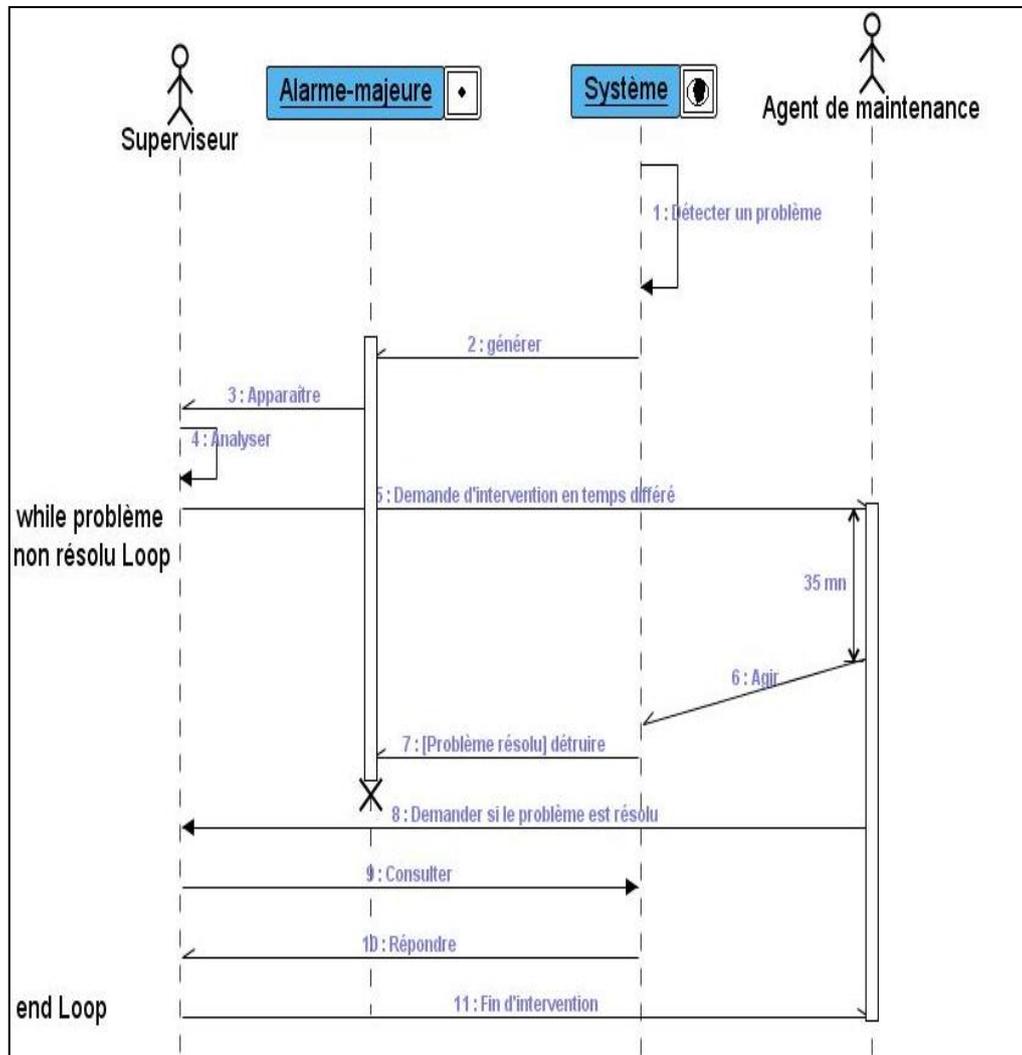


Figure 72. Scénario 2 : réaction en cas de déclenchement d'une alarme majeure.

Dans un troisième scénario, nous avons simulé l'apparition simultanée de deux alarmes : une alarme critique et une alarme mineure, sachant qu'une alarme mineure est une alarme traitée en temps mou. Nous avons supposé qu'il existe une seule équipe de maintenance disponible. Donc l'intervention se fait sur le système pour résoudre d'abord la panne critique en temps réel, ensuite la panne qui a déclenché l'alarme mineure en temps mou (Figure 73).

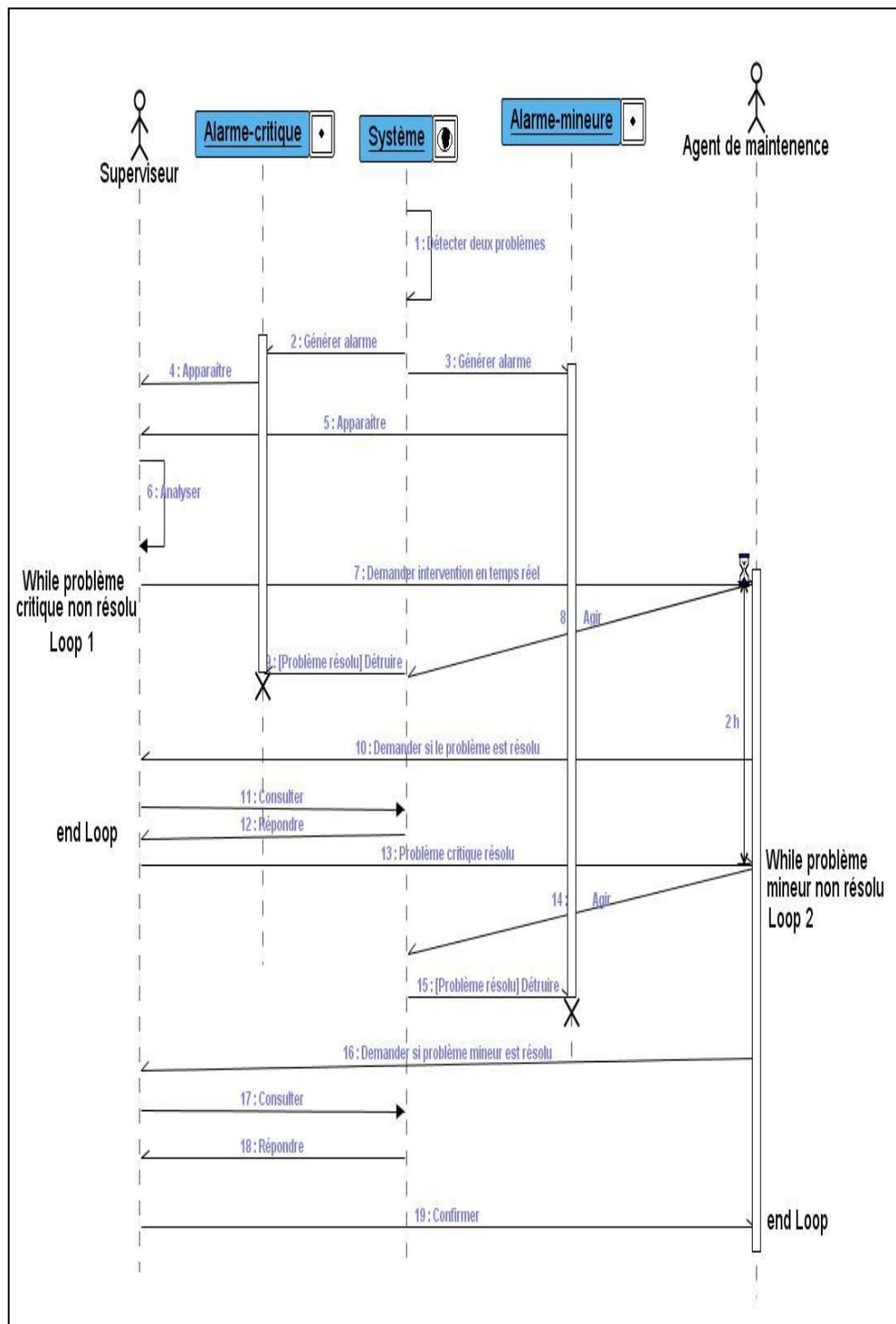


Figure 73. Scénario 3 : réaction en cas de déclenchement simultané d'une alarme critique et d'une alarme mineure.

5.3.2.3 Implémentation

Pour réaliser la phase d'implémentation nous avons utilisé le système d'information géographique ArcView qui est un puissant outil pour gérer, visualiser, interroger et analyser toutes les données disposant d'une composante spatiale. ArcView fonctionne sur une grande variété de systèmes informatiques. Il est composé de cinq modules ayant chacun ses propres menus : les vues (VIEWS), les tables, les diagrammes (CHARTS), la mise en page et restitution (LAYOUTS) et les SCRIPTS.

Nous avons tout d'abord commencé par la répartition des classes de sous système réseau en quatre thèmes ou couches d'informations.

Un premier thème « Ghazela.img » qui est considéré comme plan de référence. C'est une image satellitaire de zone de repérage, que nous avons choisie comme cité pour détecter les alarmes du réseau GSM de la cité El-Ghazela (Figure 74).

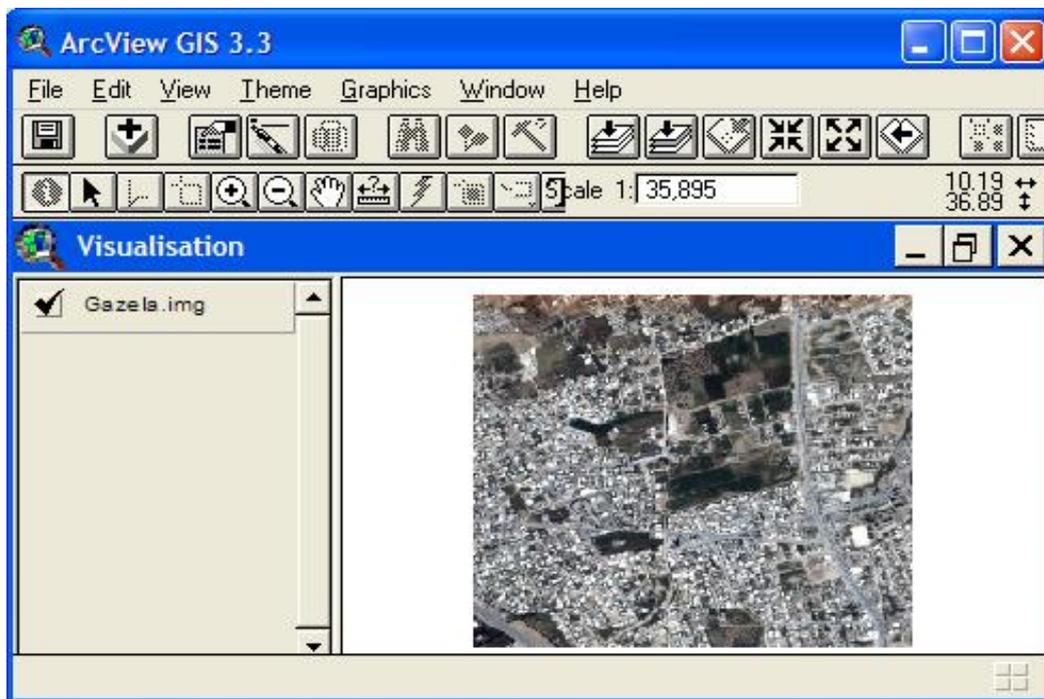


Figure 74. Présentation du thème « El-Ghazela ».

Un deuxième thème « bts.shp » schématise la répartition des BTS dans cette zone (Figure 75).

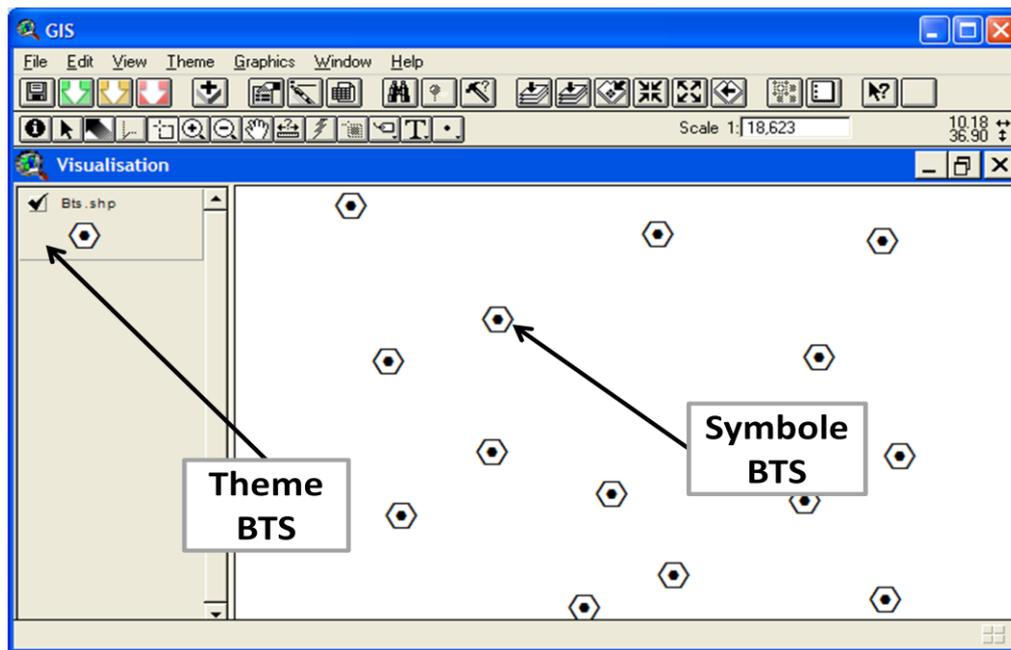


Figure 75. Présentation du thème BTS.

Un troisième thème « bsc.shp » schématise la répartition des BSC (Figure 76).

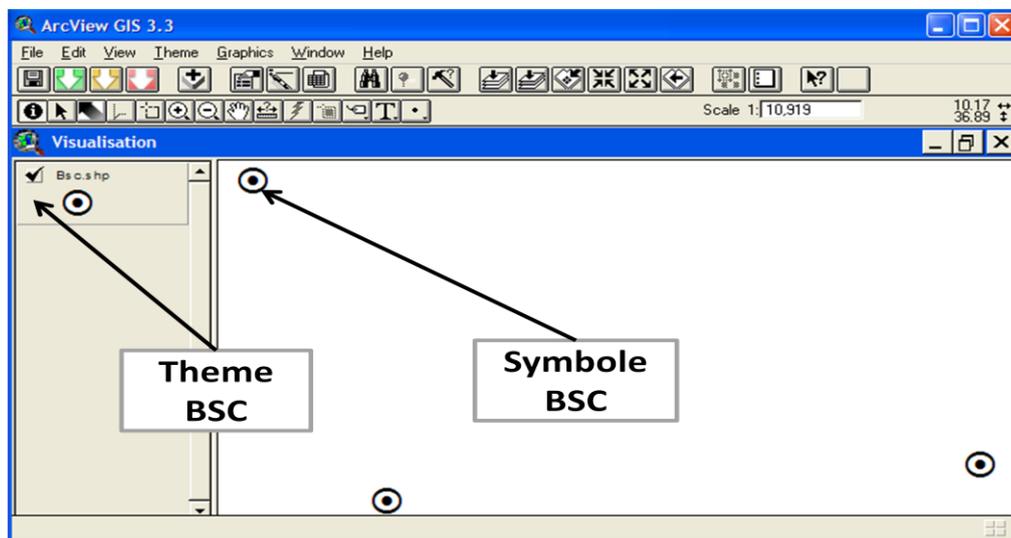


Figure 76. Présentation du thème BSC.

Un quatrième thème « Lienabis.shp » pour schématiser les liaisons Abis entre les BTS et les BSC (Figure 77).

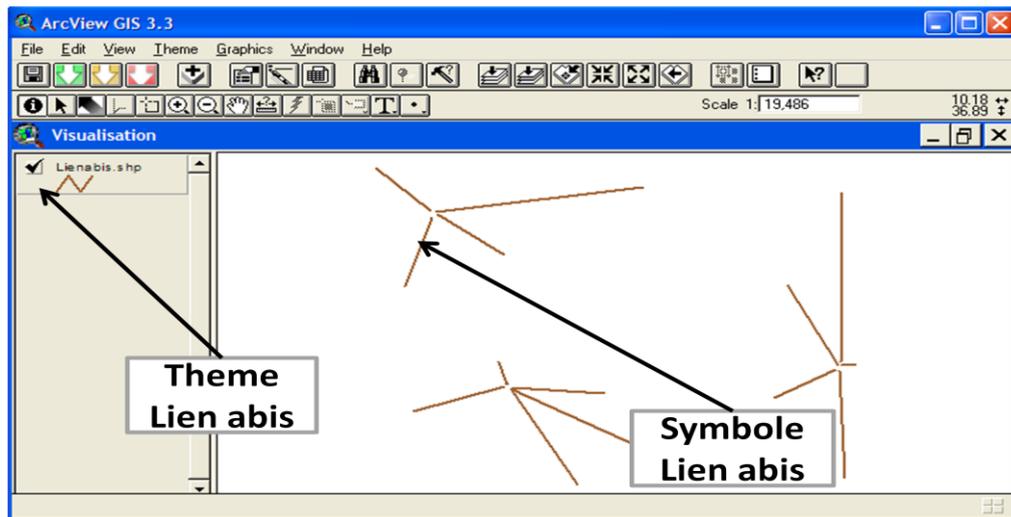


Figure 77. Présentation du thème lien Abis.

Après avoir défini les quatre thèmes nous avons superposé ces derniers dans une seule vue (Figure 78).

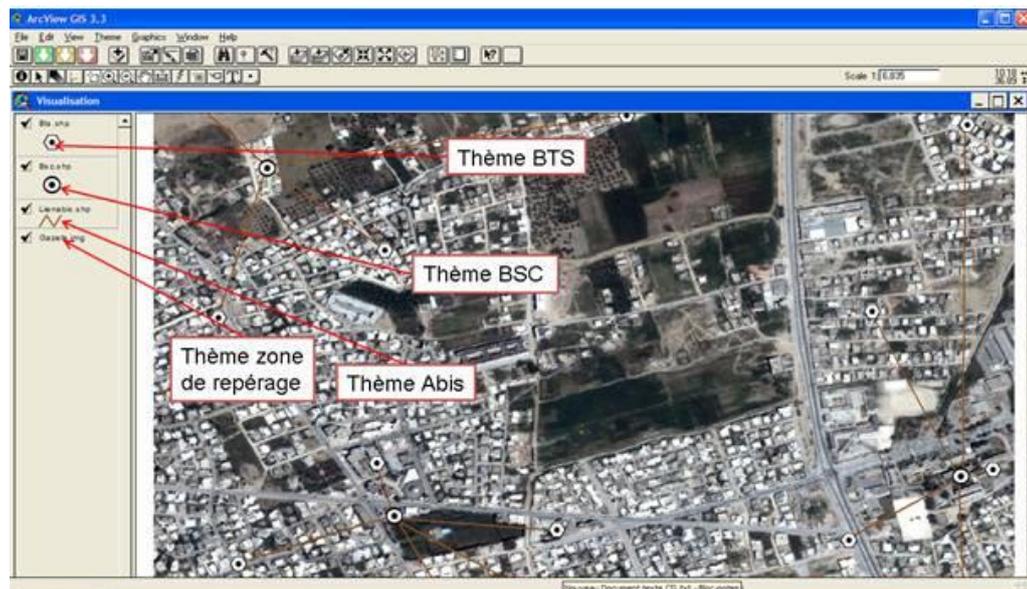


Figure 78. Présentation de l'ensemble des thèmes.

Une fois établies les tables des données descriptives des classes, la visualisation des alarmes nécessite l'accès à l'information sur l'état système à travers une connexion à la base de données. C'est pour cela que nous avons

rédigé des scripts permettant l'envoi des requêtes SQL à la base et la vérification de l'existence de nouvelles alarmes.

Pour visualiser les nouvelles alarmes parvenues, nous avons écrit des scripts qui permettent de se focaliser sur la classe concernée par l'alarme. Cette dernière clignote et change de couleur selon la gravité de l'alarme qui l'a affecté : rouge pour critique (Figure 79), orangé pour majeure et vert pour mineure.

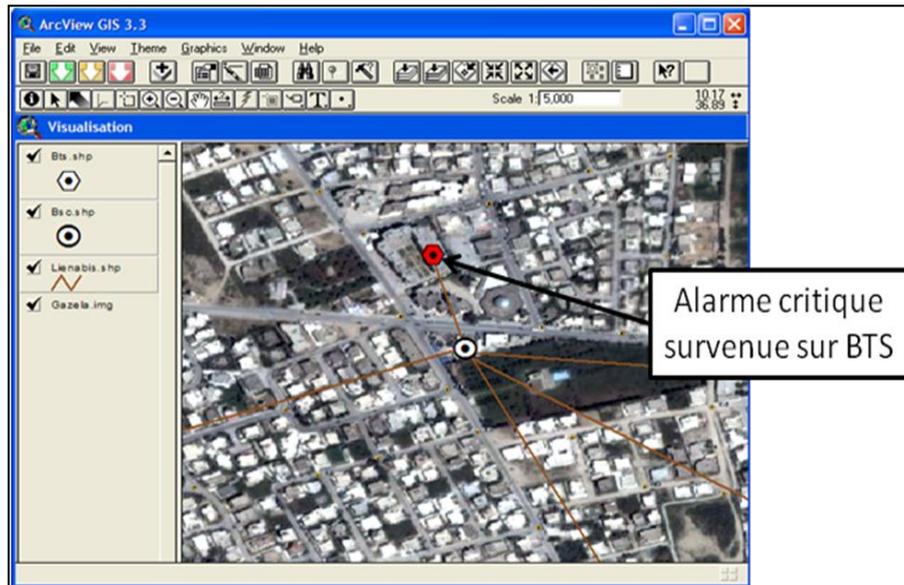


Figure 79. Exemple de manifestation d'une alarme critique.

Afin d'assurer une meilleur gestion de flux d'alarmes, nous avons ajouté à l'interface de visualisation trois boutons qui informent le superviseur sur l'historique des différentes alarmes critiques, majeures et mineures. Nous avons aussi introduit des boutons à la catégorie « tools » qui permettent au superviseur de consulter l'historique des alarmes qui ont affecté la classe d'alarme sélectionnée (Figures 80 et 81).



Figure 80. Différents boutons pour visualiser l'historique des alarmes.

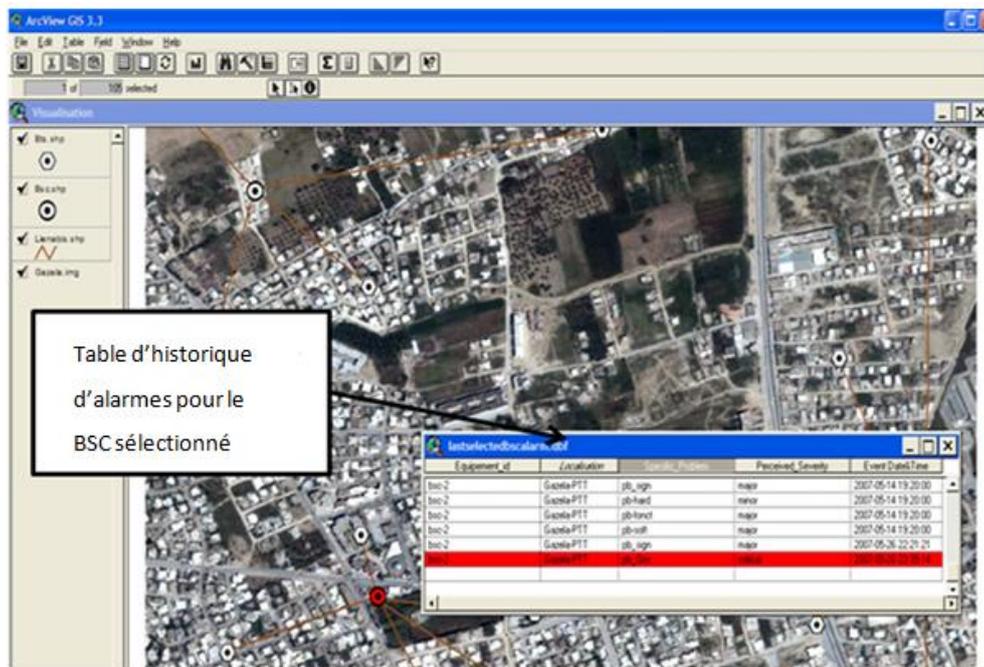


Figure 81. Exemple d'historique d'alarmes pour un BSC.

5.4 Conclusion

Dans ce chapitre, nous avons présenté les résultats de nos travaux intégrés dans l'atelier AGRT. Nous avons décrit AGRT, présenté son architecture et son modèle conceptuel. La conception et la mise en place d'une application géomatique temps réel est une tâche assez difficile aussi bien pour les concepteurs que pour les utilisateurs finaux. Nous avons expérimenté notre profil GRT sur une application géomatique temps réel intitulée «Gestion des alarmes dans un réseau GSM». Nous avons pris en considération l'aspect géomatique et l'aspect temps réel. De nombreuses applications développées avec cette solution nous ont permis de mesurer l'intérêt de notre profil tout en étudiant de manière empirique la pertinence des notations de T-OMEGA, MADS et TURTLE.

Les composants logiciels de cet atelier ont été développés en Java afin d'assurer la portabilité des applications.

CHAPITRE 6.

CONCLUSION ET PERSPECTIVES

6.1 Conclusion

Les Systèmes d'Information Géographiques possèdent des domaines d'applications très variés et sont souvent très difficilement personnalisables pour un utilisateur final non expert.

De plus, les données gérées dans les SIG peuvent être de nature très complexe (alphanumérique, vecteur, raster, etc), ce qui entraîne une difficulté au niveau de la conception, une difficulté supplémentaire si on introduit la notion temps réel aux SIG. Les applications temps réel ont souvent besoin de stocker et de traiter très rapidement des volumes importants d'informations. Elles manipulent également des données actives dont la mise à jour provoque une réaction du système qui se matérialise par le déclenchement d'actions (mises à jour d'autres données, déclenchement d'une alarme, ...) et des données temporelles c'est-à-dire datées ce qui permet de gérer l'historique de leur évolution et ainsi de procéder à des anticipations.

Face aux difficultés des applications géomatiques temps réel, les méthodes de conception consacrées aux applications géomatiques et les méthodes de conception dédiées aux applications temps réel ne répondaient pas aux besoins du concepteur, d'où l'objectif principal de notre travail est de définir une méthode de conception spécialisée dans les applications géomatiques temps réel.

Nous avons ainsi défini un profil UML pour les applications géomatiques temps réel, c'est le profil GRT. Ce profil est basé sur des extensions et fusion des méthodes de conception pour les applications géomatiques MADS et T-OMEGA et les méthodes de conception pour les applications temps réel telles que le profil TURTLE. Etant donné que le temps réel est relatif, nous avons laissé la possibilité au concepteur de qualifier la notion du temps réel en précisant la fréquence d'acquisition et la fréquence de réaction, c'est-à-dire on précise les intervalles minimum et maximum pour savoir quel type de temps il manipule. De même le concepteur peut définir la vitesse de stockage et la vitesse d'affichage et s'il travaille en mode synchrone ou asynchrone.

Par conséquent, au niveau de GRT nous avons effectué des extensions dans certains diagrammes d'UML pour qu'ils tiennent en compte des caractéristiques des applications géomatiques temps réel. Nous avons de même réalisé l'intégration du méta-modèle de données des différents diagrammes. Ce profil a été validé avec plusieurs applications géomatiques temps réel telles que

la gestion de flotte de véhicules terrestres par GPS, l'implémentation d'un système de détection d'alarme dans les réseaux de capteurs sans fil WSN, la gestion des alarmes dans un réseau GSM, etc.

Le profil GRT pour les applications géomatiques temps réel présentée dans ce mémoire est supportée par un outil de type AGL appelé AGRT permettant de schématiser les diagrammes et de générer le code source en JAVA ou en SQL selon le choix du concepteur.

Les travaux présentés dans cette thèse ont donc permis de définir précisément une méthode de conception pour les applications géomatiques temps réel supportée par une solution AGL, et validée par la conception d'une architecture d'applications géomatiques temps réel.

6.2 Perspectives

Après avoir effectué ces années de thèse, on peut dire que la recherche dans le domaine de conception des applications géomatiques temps réel est très vaste surtout qu'il existe énormément de contraintes à respecter, ce qui peut faire l'objet de plusieurs axes de recherches et de ce fait nous présenterons des perspectives sur plusieurs domaines.

6.2.1 Perspectives d'applications

La conception des applications traitant des données en trois dimensions sur le globe terrestre telles que :

- la surveillance maritime : il s'agit de réaliser un tableau de bord qui effectue la gestion du trafic des bateaux et d'afficher des signaux d'alarmes en cas de problèmes.
- la surveillance aérienne : il s'agit de surveiller en temps réel le trafic aérien et de suivre les trajectoires des avions en trois dimensions sur le globe terrestre.

6.2.2 Perspectives du profil

Concernant la problématique de conception nous pouvons citer les perspectives suivantes :

- la prise en compte des données en 3D sur le globe terrestre au niveau de la modélisation peut représenter un enjeu important car les ordinateurs deviennent de plus en plus puissants et les SIG proposent de plus en plus des fonctionnalités intégrant les données en trois dimensions,
- une autre perspective du profil GRT est de spécifier durant la phase de modélisation conceptuelle un ensemble de traitements au niveau physique

qui se déroulera dans un temps donné, par exemple en moins d'une 1/10 de secondes.

- la finalisation de nos contributions en vue d'une proposition à l'OMG/OGC (Open Geospatial Consortium), afin que notre profil destiné aux applications géomatiques temps réel soit validé par l'OMG.

6.2.3 Perspectives de l'AGL

Des améliorations de l'outil de personnalisation de l'outil AGRT sont possibles :

- l'ajout d'autres diagrammes tels que le diagramme d'état transition et le diagramme de déploiement ; à la différence de la méthodologie UML classique la garantie du respect de certaines contraintes temps réel ne sera faite qu'au niveau du diagramme de déploiement notamment pour tenir compte des caractéristiques temporelles des plates-formes dans le cas du temps réel dur.
- la réalisation d'une version en ligne d'AGRT permettant de faire de la conception sur le web ainsi que la génération de code.

REFERENCES BIBLIOGRAPHIQUES

Bibliographie

- [ABD 95] ABDELLATIF A., LIMAME M., ZEROUAL A. ORACLE 7 Langages-Architecture-Administration. Paris : Eyrolles, 1995.
- [ABR 77] ABRIAL J.R. Manuel du langage Z (Z/13). Rapport de recherche, Electricité de France, 1977.
- [ABR 96] ABRIAL J.R. Extending B without Changing it (for Developing Distributed Systems). In Henri Habrias, Proceedings of 1st Conference on the B method, November 25-27, 1996, Nantes, pp. 169-190.
- [ABR 96b] ABRIAL J.R. The B-Book. Cambridge : Cambridge University Press, 1996.
- [ABR 00] ABRIAL J.R. " B : 2000 et plus ". Ecole Jeunes chercheurs en programmation. Lyon : Ecole Normale Supérieure de Lyon, janvier 2000.
- [APV 01] APVRILLE L., SAQUI-S P., LOHR C., SENAC P., COURTIAT J.P. A New UML Profile for Real-time System Formal Design and Validation. In 4th International Conference on the Unified Modeling Language (UML 2001), Toronto, Canada, Springer Berlin, 2001, pp. 287-301.
- [APV 04] APVRILLE L., COURTIAT J.P., LOHR C., SAQUI-S P. TURTLE : A Real-Time UML Profile Supported by a Formal Validation Toolkit. IEEE Transactions on Software Engineering, 2004, Vol. 30, n°7, pp. 473-487.
- [APV 05] APVRILLE L., SAQUI-S P., KHENDEK F. TURTLE-P : a UML Profile for the Formal Validation of Critical and Distributed systems. SOSYM, Journal of Software and System Modeling, Springer Berlin, 2006, Vol. 5, n°4, pp. 449-466.
- [APV 05b] APVRILLE L., SAQUI-S P., KHENDEK F. Synthèse d'une conception UML temps-réel à partir de diagrammes de

- séquences. In CFIP'2005 : Colloque Francophone sur l'Ingénierie des Protocoles, Bordeaux, France, 2005, 14p.
- [ATK 00] ATKINSON C., KÜHNE T. Strict Profiles : Why and How. In Third International Conference on The Unified Modeling Language (UML 2000), York, UK, 2000, pp. 309-322.
- [BED 96] BEDARD Y., CARON C., MAAMAR Z., MOULIN B., VALLIERE D. Adapting Data Models For The Design Of Spatio-Temporal Databases. International Journal of Computer, Environment and Urban Systems, 1996, Vol. 20, n°1, pp.19-41.
- [BED 99] BEDARD Y. Visual Modelling of Spatial Databases Towards Spatial PVL and UML. Geomatica, 1999, Vol. 53, n°2, pp. 169-186.
- [BEN 99] BEN YOUSSEF M. Modélisation orientée-objet et spatio-temporelle dans un système d'information géographique : rapport de DEA. DEA Systèmes de communications, option Télédétection et Systèmes d'Information Spatiale. Tunis : l'Ecole Nationale d'Ingénieurs de Tunis (ENIT), 1999, 141 p.
- [BEN 06] BEN YOUSSEF M., LBATH A., BELHADJ Z., LAURINI R. From Design of Real Time Geomatic Software to Mobile Urban Applications. In 25th Urban Data Management Symposium, Aalborg, Denmark, 2006, 9p.
- [BEN 07] BEN YOUSSEF M., LBATH A., BELHADJ Z. A Problem of Real Time GeoProcessing Applications Design and Generation. IEEE ICTIS'07 (Information and Communication Technologies International Symposium), FEZ, Maroc, 2007, 4p.
- [BEN 09] BEN YOUSSEF M., LBATH A., BELHADJ Z. A GRT UML Profile for Real Time GeoProcessing Applications. Proceeding of the International Conference on Software, Knowledge and Information Management and Applications (SKIMA 2009). Maroc, Fes, 2009, 4p.

- [BEN 09b] BEN YOUSSEF M., LBATH A., BELHADJ Z. Using UML for Real Time Geomatic Modelisation. IEEE, Proceeding of the 5th International Conference on Signal Image Technology&Internet Based Systems (SITIS 09). Marrakech, Maroc, 2009, 6p.
- [BEN 10] BEN YOUSSEF M., LBATH A., BELHADJ Z. A dedicated UML Profile for Real Time Geomatic modelisation and code generation. Journal of Communication and Computer, 2010, pp.7. Accepté à paraître.
- [BER 99] BERNER S., GLINZ M., JOOS S. A Classification of Stereotypes for Object-Oriented Modeling Languages. Proceedings of the 2nd Conference on the Unified Modeling Language. Fort Collins, USA, 1999, pp. 249-264.
- [BOO 92] BOOCH G. Object Oriented Design. Inc. Redwood City (California), 1992, 45p.
- [BOO 98] BOOCH G., RUMBAUGH J., ACOBSON I. The Unified Modeling Language. User Guide. London : Addison-Wesley, 1998, 482 p.
- [BOR 99] BORGES K.A.V., LAENDER A.H.F., DAVIS JR C.A. Spatial Data Integrity Constraints in Object Oriented Geographic Data Modeling. Proceedings of the International Symposium on Geographic Information System. ACM Press, USA, 1999, pp. 1-6.
- [BOU 07] BOUSSIÉ J.M. Modélisation de comportements dans les systèmes dynamiques: Application à l'organisation et à la régulation de stationnement et de déplacement dans les Systèmes de Trafic Urbain. Thèse Informatique. La Rochelle : Université de La Rochelle, 2007, 351p.
- [BRO 00] BRODEUR J., BEDARD Y., PROULX M.J. Modeling Geospatial Application Databases Using UML-based Repositories Aligned with International Standards in Geomatics. Proceeding of the 8th ACM Symposium on Advances in Geographic Information Systems, Washington D.C., USA, 2000, pp. 39-46.

- [BRO 08] BROUARD F., BRUCHEZ R., SOUTOU C. SQL. 2ème ed. Paris : Pearson Education, 2008.
- [BOR 01] BORGES K.A.V., DAVIS JR C.A., LAENDER A.H. F. OMT-G : An Objected-Oriented Data Model for Geographic Applications. *Geoinformatica*, 2001, Vol. 5, n°3, pp. 221-260.
- [BLA02] BLANC X., ESFRAY P. Du profil UML au composant MDA (4), Développeur Référence [en ligne], 2002, v2.22, pp. 2-5. Disponible sur <http://wwwwesto.ump.ma/mounir/mda/these%20mda/MDAv1.8.pdf> (consulté le 15.09.2009).
- [BLA 05] BLANC X. MDA en action ingénierie logicielle guidée par les modèles. Paris : Eyrolles, 2005, 270p.
- [CAR 93] CARON C., BEDARD Y., GAGNON P. MODUL-R : un formalisme individuel adapté pour les SIRS. *Revue Internationale de Géomatique*, 1993, Vol. 3, n°3, pp. 283-306.
- [CAR 06] CARTRON M. Vers une plate-forme efficace en énergie pour les réseaux de capteurs sans fil. Thèse Traitement du Signal et Télécommunication. Rennes : l'Université de Rennes 1, 2006, 132p.
- [CHE 76] CHEN, P. P.-S. The entity-relationship model : toward a Unified View of Data. *ACM Transactions on Database Systems*, 1976, Vol. 1, n°1, pp. 9-36.
- [CLA 95] C CLARAMUNT., THERIAULT M. Managing Time in GIS An Event-Oriented Approach. London, UK : Springer-Verlag, 1995, pp. 22-43.
- [CLA 97] CLARAMUNT C., COULONDRE S., LIBOUREL T. Autour des méthodes orientées objet pour la conception des SIG. *Revue Internationale de Géomatique*, 1997, Vol. 7, n°3-4, pp. 233-257.

- [COA 91] COAD P., YOURDON E. Object-Oriented Analysis. New Jersey : Prentice Hall, 1991, 194p.
- [COU 00] COURTIAT J.P., SANTOS C.A.S., LOHR C., OUTTAJ B. Experience with RT-LOTOS, a Temporal Extension of the LOTOS Formal Description Technique. Computer Communications, 2000, Vol. 23, n°12, pp. 1104-1123.
- [DAV 93] DAVID B., RAYNAL L., SCHORTER G., MANSART V. Géo2 : modélisation objet de données géographiques. Revue Internationale de Géomatique, 1993, Vol. 3, n°3, pp. 265-281.
- [DEM 04] DEMOULIN C., VAN DROOGENBROECK M. Principes de base du fonctionnement du réseau GSM. Revue de l'AIM, 2004, Vol. 4, n°4, pp 3-18.
- [DUB 07] DUBOISSET M. Un Système de Contraintes d'Intégrité OCL pour les Bases de Données Spatiales, Application à un Système d'Information pour l'Épandage Agricole. Thèse Informatique. Clermont-Ferrand : Université Blaise Pascal, 2007, 135p.
- [ESP 07] ESPINOZA O.H.D. An Integrated Model-Driven Framework for Specifying and Analyzing Non-Functional Properties of Real-Time Systems. Thèse Sciences. Evry : Université d'Evry, 2007, 191p.
- [FAI 99] FAIZ S. Systèmes d'Informations Géographiques : Information Qualité et Data Mining. Tunis : Editions C. L. E, 1999.
- [FIS 01] FISCHER C., OLDEROG E.R., WEHRHEIM H. A CSP View on UML-RT Structure Diagrams. Proceedings of the 4th International Conference on Integrated Fundamental Approaches to Software Engineering, Berlin Heidelberg. New York : Springer-Verlag, 2001, Vol. 2029/2001, pp. 91-108.
- [FRI 01] Friis-Christensen A., Tryfona N., Jensen C. S. Requirements and Research Issues in Geographic Data Modeling.

- Proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems. Atlanta, Georgia, USA, 2001, pp. 2-8.
- [GAN 79] GANE C., SARSON T. Structured systems analysis : tools and technics. New Jersey : Prentice Hall, Inc., 1979, 240p.
- [GAY 97] GAYTE O., LIBOUREL T., CHEYLAN J., LARDON S. Conception des systèmes d'information sur l'environnement, Paris : HERMES, 1997.
- [GHE 06] GHERBI A., KHENDEK F. UML Profiles for Real-Time Systems and their Applications. Journal of Object Technology, 2006, Vol. 5, n°4, pp. 149-169.
- [GOM 93] GOMAA H. Software Design Methods for Concurrent and Real-Time System. Boston, MA, USA : Addison Wesley, 1993. (SEI Series in Software Engineering).
- [GRE 04] GREGAN M. Tutorial Oracle Spatial [en ligne]. Disponible sur:
http://www.iict.ch/Tcom/Presentations/EI2004/Oracle_Spatial_Tutorial.pdf (consulté le 05-01-2009).
- [HAI 91] HAINUT J.L. Entity-Generating Schema Transformation for Entity Relationship Models. Proceeding of the 10th International Conference on the Entity Relationship Approach. San Mateo, CA, USA, 1991, 28p.
- [IDA 05] IDANI A., LEDRU Y. Object Oriented Concepts Identification from Formal B Specifications. Proceedings of 9th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'04). Elsevier, Linz, Austria, 2005, Vol. 133 of ENTCS, pp. 159-174.
- [IDO 06] IDOUDI N., DUVALLET C., SADEG B., BOUAZIZ R. et GARGOURI F. Un modèle objet pour les SGBD temps réel. Actes du Colloque International sur l'Informatique et ses Applications (IA 2006). Oujda, Maroc, 2006, pp. 285-292.

- [JAC 92] JACOBSON I. Object-Oriented Software Engineering. Workingham (UK) : Addison-Wesley, 1992, 236p.
- [JOR 09] JORIS R. Gestion du temps par le raffinement. Thèse Informatique. Nancy : Université Henri Poincaré Nancy 1, 2009, pp. 122.
- [JUG 98] JUGURTA L.F., CIRANO I., KATE B. Applying Analysis Patterns in GIS Domain. Proceedings of the 10th Colloquim of Spatial Information Research Center. Otago, New Zealand, 1998, 8p.
- [JUG 99] JUGURTA L.F., CIRANO I. Specifying Analysis Patterns for Geographic Databases on the Basis of a Conceptual Framework. Proceedings of the 7th ACM Symposium on Advances in Geographic Information Systems. Kansas City, USA, 1999, pp.7-13.
- [KHA 08] KHANH H.N. Aide au développement de systèmes temps réel à l'aide d'un langage graphique flots de données. Thèse Sciences pour l'Ingénieur. Poitiers : Université de Poitiers, 2008, 222p.
- [KIR 03] KIRESTEN B. Using UML 2.0 in Real-Time Development : A Critical Review. Proceedings of SVERTS workshop. San Francisco, USA, 2003, pp. 41-54.
- [KOS 95] KOSTERS G., PAGEL B.U, SIX H.W. Object-Oriented Requirements Engineering for GIS Applications. Proceedings of the 3rd ACM Workshop on Advances in Geographic information systems. Baltimore, Maryland, 1995, pp. 61-68.
- [KOS 96] KOSTERS G., PAGEL B.U. The GeoOOA-Tool and Its Interface to Open Software Development Environments for GIS. Proceedings of the 4th Workshop on Advances in Geographic Information System. Rockville, USA, 1996, pp.163-171.
- [KOS 97] KOSTERS G., PAGEL B.U, SIX H.W. GIS-Application Development with GeoOOA. International Journal of

- Geographical Information Science, 1997, Vol. 11, n°4, pp. 307-335.
- [LAG 02] LAGRANGE X., GODLEWSKI P., TABBANE S. Réseaux GSM 5ème édition revue et augmentée. Paris : Hermès Sciences Publication, 2000, 521p.
- [LBA 95] LBATH A., AUFAURE-PORTIER M.A. AIGLE : Un atelier visuel de génération d'applications géographiques. Revue Internationale de Géomatique, 1995, Vol. 5, n°2, pp.179-195.
- [LBA 97] LBATH A. AIGLE, un environnement visuel pour la conception et la génération automatique d'applications géomatiques. Thèse Ingénierie Informatique. Lyon : INSA de Lyon, 1997, 282p.
- [LBA 97b] LBATH A., AUFAURE M.A., LAURINI R. Using a Visual Language for the Design and Query in GIS Customization. Proceedings of International Conference on Visual Information Systems : VISUAL'97. San Diego, USA, 1997, pp. 197-205.
- [LBA 97c] LBATH A. A Generic Graphical User Interface for Customization of GIS Applications. Proceedings of Joint European Conference on Geographical Information. Vienna, Austria, 1997, pp. 116-124.
- [LBA 00] LBATH A., PINET F. The Development and Customization of GIS-based Applications and Web-based GIS Applications with the CASE tool AIGLE. Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems. Washington D.C., USA, 2000, pp. 194-196.
- [LBA 00b] LBATH A., PINET F. Towards Conceptual Modeling of TeleGeoprocessing Applications. Proceedings of the 2nd Symposium on TeleGeoProcessing. Nice France, 2000, pp. 25-39.
- [LBA 00c] LBATH A., PINET F. Automatic Generation of Geographic Urban Applications for the Web with the CASE Tool

- AIGLE. Proceedings of the 22nd Symposium on Urban and Regional Data Management. Delft, Netherlands, 2000, 6p.
- [LBA 01] LBATH A. Process and Device for Automatic Generation of geomatic applications. Brevet international, 2001. Brevet WO/2003/048,931.
- [LBA 02] LBATH A. OMEGA : Object modeling for end-user geographic applications : A visual UML based methods for GIS. IEEE International Symposium on Signal Processing and Information Technology, Marrakech, Maroc, 2002.
- [LBA 02b] LBATH A., OUZIRI M. MobSQL, An SQL like query language for mobile object database. IEEE International Symposium on Signal Processing and Information Technology. Marrakech, 2002.
- [LED 02] LEDANG H. Traduction systématique de spécifications UML vers B. Thèse Informatique. Nancy : université LORIA, 2002.
- [MAS 00] MASSON S. Les interactions entre système de transport et système de localisation en milieu urbain et leur modélisation. Thèse Sciences Economiques. Lyon : Université Lyon II, 2000, 570p.
- [MAS 03] MASSE J., KIM S., HONG S. Tool set implementation for scenario-based multithreading of UML-RT models and experimental validation. Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2003). Toronto, Canada, 2003, pp. 70-77.
- [MED 01] MEDINA J., GONZALEZ.H.M., DRAKE J.M. MAST Real-Time View: A Graphic UML Tool for Modeling Object-Oriented Real-Time Systems. Proceedings of the 22nd IEEE Real-Time Systems Symposium (RTSS'01). IEEE Computer Society Press, 2001, pp. 245-256.

- [MED 05] MEDINA J. Methodology and UML tools for modeling and analysis of object oriented real-time systems. Thèse Informatique. Espagne : Université de Cantabria, 2005, 382p.
- [MIR 06] MIRALLES A. Ingénierie des modèles pour les applications environnementales. Thèse Informatique. Montpellier : Université de Montpellier II, 2006, 344p.
- [MUL 00] MULLER P.-A, GAERTNER N. Modélisation objet avec UML, 2ème édition. Paris : Eyrolles, 2000, 520p.
- [OMG 99] OMG. Unified Modeling Language-Specification - Version 1.3, Juin 1999.
- [OMG 01] OMG, Unified Modeling Language-Specification - Version 1.4, 2001. <http://www.omg.org/cgi-bin/apps/doc?formal/01-09-67.pdf> (consulté le 02.05.2008).
- [OMG 01b] OMG. Common Warehouse Metamodel - Version 1.0, 2001. <http://www.omg.org/cgi-bin/doc?ad/2001-02-01.pdf> (consulté le 05.01.2009).
- [OMG 02] OMG. Meta Object Facility Specification - Version 1.4, 2002. <http://www.omg.org/cgi-bin/doc?formal/2002-04-03.pdf> (consulté le 04.06.2009).
- [OMG 02b] OMG. MOF 2.0 Query/Views/Transformations ad/2002-04-10, 2002. <http://www.omg.org/docs/ad/02-04-10.pdf> (consulté le 04.06.2009).
- [OMG 03] OMG. Unified Modeling Language-Specification - Version 1.5, Mars 2003. <http://www.omg.org/cgi-bin/apps/doc?formal/03-03-01.pdf> (consulté le 08.07.2009).
- [OMG 03b] OMG. UML 2.0 Infrastructure Specification, 2003. <http://www.omg.org/cgi-bin/apps/doc?ptc/03-09-15.pdf> (consulté le 02.03.2009).
- [OMG 03c] OMG. UML 2 OCL Final Adopted Specification, 2003. <http://www.omg.org/docs/ptc/03-10-14.pdf> (consulté le 04.05.2009).

- [OMG 05] OMG. UML Profile for Schedulability, Performance, and Time Specification, Version 1.1, Needham, MA, U.S.A : OMG, Janvier 2005. Disponible à : <http://www.omg.org/cgi-bin/doc?formal/2005-01-02> (consulté le 07.07.2009).
- [OMG 05b] OMG. Unified Modeling Language: Superstructure - Version 2.0, 2005. <http://www.omg.org/cgi-bin/apps/doc?formal/05-07-04.pdf> (consulté le 04.08.2009).
- [OMG 06] OMG. Unified Modeling Language: Infrastructure - Version 2.0, 2006. <http://www.omg.org/cgi-bin/apps/doc?formal/05-07-05.pdf> (consulté le 04.08.2009).
- [PAN 94] PANTAZIS D. Analyse Méthodologique des Phases de Conception et de Développement d'un SIG. Thèse Informatique. Liège : Université de Liège, 1994, 556p.
- [PAN 96] PANTAZIS D., DONNAY J.P. La conception de SIG-Méthode et formalisme. Paris : Hermès, 1996.
- [PAN 97] PANTAZIS D., DONNAY J.P. Objets géographiques à limites indéterminés-Modélisation et intégration dans un modèle conceptuel de données. *Revue Internationale de Géomatique*, 1997, Vol. 7, n°2, pp. 159-186.
- [PAR 97] PARENT C., SPACCAPIETRA S., ZIMÁNYI E., DONINI P., PLAZANET C., VANGENOT C., ROGNON N., RAUSAZ P. MADS, modèle conceptuel spatio-temporel. *Revue Internationale de Géomatique*, 1997, Vol. 7, n°3-4, pp. 317-351.
- [PAR 98] PARENT C., SPACCAPIETRA S., ZIMÁNYI E., DONINI P., PLAZANET C., VANGENOT C. Modeling Spatial Data in the MADS Conceptuel Model. *Proceedings of the International Symposium on Spatial Data Handling. SDH 98*, Vancouver, Canada, 1998, pp. 138-150.
- [PAR 99] PARENT C., SPACCAPIETRA S., ZIMÁNYI E., Spatio-Temporal Conceptual Models : Data Structures + Space + Time. *Proceedings of the International ACM Symposium on*

- Advances in Geographic Information Systems. USA, 1999, pp. 26-33.
- [PAR 99b] PARENT C., SPACCAPIETRA S., ZIMÁNYI E., DONINI P., PLAZANET C., VANGENOT C. MADS ou l'Information Spatio-Temporelle à Portée de ses Utilisateurs. Quatrièmes rencontres de Théo Quant, Besançon, France, 1999.
- [PAR 06] PARENT C., SPACCAPIETRA S, ZIMÁNYI E. Conceptual Modeling for Traditional & Spatio-temporal Applications : the MADS Approach. Berlin : Springer, 2006, 450p.
- [PIN 01] PINET F., LBATH A. An Algorithm for Stereotype Deduction in UML-Based Formalism and its Application in Geographic Information Systems. Proceedings of the IEEE Symposium on Visual/Multimedia Approaches to Programming and Software Engineering. Stresa, Italy, 2001, pp. 296-303.
- [PIN 02] PINET F. Vers une Méthode de Conception pour la Télégéomatique : Intégration et Validation au Sein d'un Atelier de Développement. Thèse Informatique. Lyon : INSA de Lyon, 2002, 243p.
- [PIN 03] PINET F., LBATH A. Object-Oriented Modelling Of Communicating Gis-Based Applications. Proceedings of the 6th International Conference AGILE. Lyon, France, 2003, 7p.
- [ROB 09] ROBERT T. Détection d'erreur au plus tôt dans les systèmes temps-réel une approche basée sur la vérification en ligne. Thèse Systèmes Informatiques Critiques. Toulouse : Institut National Polytechnique de Toulouse, 2009, 135p.
- [ROS 77] ROSS D. T. Structured Analysis (SA) : a language for communicating ideas. IEEE Transactions on Software Engineering, 1977, Vol. Se-3, n°1, pp. 16-34.

- [RUM 91] RUMBAUGH J., BALAHA M., PREMERLANI W., EDDY F., LORENSEN W. Object-Oriented Modeling and Design. Londres : Prentice Hall, 1991, 515p.
- [SAQ 00] SAQUI-S P., LOHR C., COURTIAT J.P., SAMPAIO P. TURTLE: A Timed UML and RT-LOTOS Environment. Workshop on Formal Methods for Real-Time UML, UML'2000 Conference. York, UK, 2000.
- [SAQ 04] SAQUI-S P., APVRILLE L., LOHR C., COURTIAT J.P. TURTLE : un pont entre UML et RT-LOTOS. Approches Formelles dans le Développement du Logiciel (AFADL'04). Besançon, France, 2004, 4p.
- [SEL 94] SELIC B., GULLEKSON G., WARD P.T. Real-Time Object-Oriented Modeling. New-York : John Wiley and Sons, Inc., 1994.
- [SEL 98] SELIC B., RUMBAUGH J. Using UML for Modeling Complex Real-Time Systems. Rational Software Corporation, 1998.
- [SHL 88] SHLAER S., MELLOR S.J. Object-Oriented Systems Analysis : Modeling the World in Data. Yourdon Press, Londres : Prentice Hall, 1988, 350 p.
- [SHU 03] SHU H., SPACCAPIETRA S., PARENT C., QUESADA SEDAS S. Uncertainty of Geographic Information and its Support in MADS. Proceedings of the 2nd International Symposium on Spatial Data Quality (ISSDQ'03). Hong Kong, China, 2003, 13p.
- [SOT 02] SOTNYKOVA A., MONTIES S., SPACCAPIETRA S. Semantic Integration in MADS Conceptual Model. In Heterogeneous Information Exchange and Organizational Hubs, Boston : Kluwer Academic Publishers, 2002, 22p.
- [SPA 98] SPACCAPIETRA S., PARENT C., ZIMÁNYI E. Modeling Time From a Conceptual Perspective. Proceedings of the 7th International Conference on Information and Knowledge Management. Bethesda, Maryland, USA, 1998, pp. 432-440.

- [TRY 97] TRYFONA N., PFOSER D., HADZILACOS T. Modeling Behavior of geographic objects : an experience with the object modeling technique. Advanced information systems engineering, Internaional Conference no9. Espagne, Barcelone, 1997, vol. 1250, pp. 347-359.
- [TRY 99] TRYFONA N., ANDERSEN S., MOGENSEN S. R., JENSEN C. S. A Methodology and a Tool for Spatiotemporal Database Design. Proceedings of Seventh Hellenic Conference on Informatics. Ioannina, Greece, 1999, pp. 53-60.
- [TRY 99b] TRYFONA N., JENSEN C.S. Conceptual Data Modeling for Spatiotemporal Applications. GeoInformatica, 1999, Vol. 3, n°3, pp. 245-268.
- [TRY 00] TRYFONA N., JENSEN C. S. Using Abstractions for Spatio-Temporal Conceptual Modeling. ACM Symposium on Applied Computing. Villa Olmo, Italy, 2000, pp. 313-322.
- [VID 94] VIDALE L., CLARAMUNT C., DE SÈDE M., PRÉLAZ-DROUX R. Sémantique et logique spatio-temporelles. Revue Internationale de géomatique, 1994, Vol. 4, n°2, pp. 165-180.
- [WAR 98] WARMER J. KLEPPE A. The Object Constraint Language : Precise Modeling with UML. Boston : Addison Wesley, 1998.
- [WAR 85] WARD P.T., MELLOR S.J. Structured Development for Real-Time Systems, Vol. 1-3. New York : Yourdon Press, , 1985.
- [WAR 86] WARD P.T., MELLOR S.J. Structured development for real time systems, Vol. 3. Englewood Cliffs : Prentice-Hall, 1986, 503p.

ANNEXES

- Annexe A.** Les alarmes selon le système BSS Alcatel B6.2.
- Annexe B.** Les règles de passage du modèle conceptuel MADS à un modèle logique.
- Annexe C.** Cahier des charges et spécifications d'une application géomatique temps réel : « Un système de détection d'alarmes dans les WSN ».
- Annexe D.** Convention de thèse en cotutelle.

ANNEXE A.

LES ALARMES SELON LE SYSTEME BSS ALCATEL B6.2

A.1 Introduction

Etant donné que les alarmes dans un réseau GSM diffèrent d'un système à un autre, nous présentons dans cette annexe les différents types d'alarmes selon le sous système BSS Alcatel B6.2.

A.2 Les classes d'alarmes pour le BSS 6.2

Les alarmes sont regroupées suivant les unités qui constituent le BSS Alcatel qui sont principalement la BTS et le BSC.

A.2.1 Les alarmes de la BTS

La BTS au niveau du BSS 6.2 possède onze classes d'alarmes qui sont réparties de la façon suivante :

- CLK : elle contient les alarmes qui se rapportent aux problèmes de synchronisation et d'horloge dans la BTS.
- OMU : elle contient les alarmes de l'unité d'exploitation et de maintenance au niveau de la BTS.
- SW-AN : elle contient les alarmes qui peuvent affecter la partie logicielle pour les éléments qui gèrent la partie radio de la BTS.
- FHU : cette classe regroupe les alarmes qui peuvent affecter l'unité de saut de fréquence.
- ENVIR : cette classe regroupe les alarmes qui affectent l'environnement du fonctionnement de la BTS telles que l'alimentation, la climatisation, etc.
- RF : cette classe regroupe les alarmes qui affectent la partie radio de la BTS.
- G-BTS : cette classe regroupe les alarmes qui affectent l'unité de trame de la BTS.
- BTS-EXTERNAL : cette classe d'alarmes renferme celles qui touchent la partie externe du local de la BTS telles que l'ouverture de la porte du local de la BTS, etc.
- RX-TX : cette classe contient les alarmes qui affectent les équipements qui gèrent les fonctions de l'antenne de la BTS.

- BTS- TRANS : cette classe regroupe les alarmes de la BTS qui affectent la partie transmission vers le BSC, ces alarmes concernent les défauts qui peuvent affecter la liaison de l'interface Abis tels que les défauts de l'horloge et les taux d'erreurs.

A.2.2 Les alarmes du BSC

Le BSC au niveau du BSS 6.2 contient plusieurs classes d'alarmes, parmi lesquelles on cite :

- BSC-ENVIRON : cette classe regroupe les alarmes qui peuvent toucher l'alimentation en énergie électrique du BSC.
- PROCESSOR : cette classe regroupe les alarmes qui peuvent affecter les différents processeurs des cartes du BSC.
- SWITCH : cette classe englobe les défaillances qui peuvent affecter les commutateurs d'accès et les commutateurs communs pour un BSC suite aux interventions d'un opérateur.
- X25 : une telle classe rassemble les alarmes qui peuvent affecter la liaison X25 entre l'OMC et le BSC.
- BSC-SW-ANOMALY : cette classe regroupe les anomalies qui affectent la partie logicielle du BSC.
- TELECOM : cette classe regroupe les alarmes qui affectent la partie télécommunication du BSC, on peut détecter les différentes alarmes de cette classe lorsqu'il y a une surcharge au niveau de la liaison avec le MSC ou une surcharge au niveau du processeur du BSC, ainsi que la perte des canaux du trafic TCH (Traffic CHannels).
- CELL : cette classe regroupe les alarmes qui peuvent affecter les cellules qui sont sous le contrôle de la BSC, ces alarmes concernent surtout la perte des canaux dédiés « BCCH (Broadcast Call Control) et SDCCH (Stand alone Dedicated Control Channel)» ainsi que les canaux du trafic TCH, ces alarmes affectent surtout la qualité du service offerte.
- BTS-OM : cette classe regroupe les alarmes qui affectent la liaison d'exploitation et de maintenance entre le BSC et la BTS.
- BACKUP-MEDIUM : dans cette classe sont regroupées les alarmes qui peuvent survenir lors de l'exécution d'une sauvegarde du backup.
- GLOBAL : cette classe regroupe les alarmes qui touchent la fonction globale du BSC telles que insuffisance de ressource pour de nouveaux liens de signalisation N7, etc.
- TR-OM : cette classe regroupe les alarmes qui affectent la transmission des messages d'exploitation et de maintenance entre BSC et BTS.
- ALERTER : cette classe d'alarmes regroupe les anomalies qui sont déclenchées suite à des comparaisons sur la disponibilité de canal SDCCH

d'autres cellules du BSS, généralement cette classe regroupe les alarmes qui affectent la qualité du service.

- EXTRENAL : cette classe regroupe les alarmes qui affectent la partie externe du BSC telles que la défaillance de climatisation, l'ouverture des portes de bâties, etc.
- SWITCH G2 : cette classe d'alarmes regroupe les défaillances qui peuvent affecter les commutateurs du BSC suite aux défauts logiciels ou matériels.
- BSC-CLOCK : cette classe d'alarmes regroupe les anomalies qui peuvent affecter le fonctionnement d'horloge du BSC.
- BROADCAST-BUS : cette classe regroupe les alarmes qui peuvent affecter la distribution des informations entre les différents modules du BSC.
- RSL : cette classe regroupe les alarmes qui peuvent gêner le fonctionnement du protocole de signalisation radio entre la BTS et le BSC.
- OML : cette classe regroupe les alarmes qui affectent la signalisation OML (Operation Maintenance Link) pour les tâches d'exploitation et de maintenance.

ANNEXE B.

LES REGLES DE PASSAGE DU MODELE CONCEPTUEL MADS A UN MODELE LOGIQUE

B.1 Introduction

Une fois le modèle conceptuel établi, il est nécessaire de passer à l'étape suivante : la modélisation logique. La modélisation logique des données est une représentation des données issue de la modélisation conceptuelle qui construit une structure de données adaptée au SGBD ou au SIG. Le schéma logique doit reproduire la totalité de la sémantique existante dans le schéma conceptuel en adaptant la description aux concepts offerts par le SGBD ou le SIG.

Nous présentons dans cette annexe quelques règles de passage d'un modèle conceptuel MADS à un modèle logique [BEN 99].

B.2 Eliminations d'une association n -aire

Elles consistent à éliminer une association n -aire dont tous les rôles sont multivalués et à éliminer une association n -aire avec un rôle monovalué [BEN 99].

B.2.1 Elimination d'une association n -aire dont tous les rôles sont multivalués

Lors de l'élimination d'une association n -aire dont tous les rôles sont multivalués entre plusieurs entités $E1$, $E2$, ... et En , nous devons créer une nouvelle table T . Cette dernière aura comme clé une composition des clés $ID1$, $ID2$, $ID3$, ... et IDn des différentes entités qui deviennent des clés externes référençant respectivement $E1$, $E2$, $E3$, ... et En et la clé ID de l'association A (si elle existe). Elle aura comme attributs ceux de l'association (s'ils existent). Si l'association ne contient pas de clé alors nous devons ajouter dans la table un attribut système qui spécifie la clé de l'association. Cet attribut sera géré automatiquement par le système comme une sorte de compteur.

B.2.2 Elimination d'une association n -aire avec un rôle monovalué

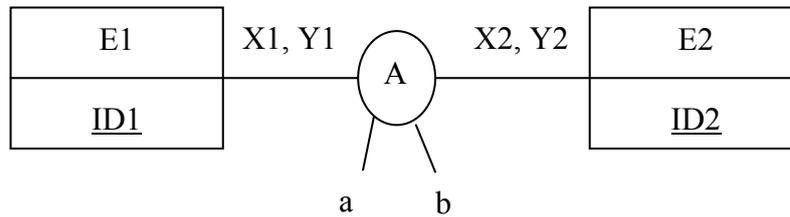


Figure 82. Exemple 1 d'une association n -aire avec un rôle monovalué.

Lors de la transformation d'une association n -aire avec un rôle monovalué (Figure 82) nous devons ajouter dans la table $E2$ la clé de $E1$ ($ID1$) qui devient une clé externe référençant $E1$, ainsi que les attributs de l'association A (s'ils existent).

Pour l'exemple de la Figure 83, si $X2=0$ et $N > 2$ alors il faut ajouter un trigger qui vérifie que les clés des $n-1$ tables sont présentes ou absentes en même temps dans la table E (Figure 84). Sachant qu'ils deviennent des clés externes référençant $E1, E2, E3, \dots, En-1$.

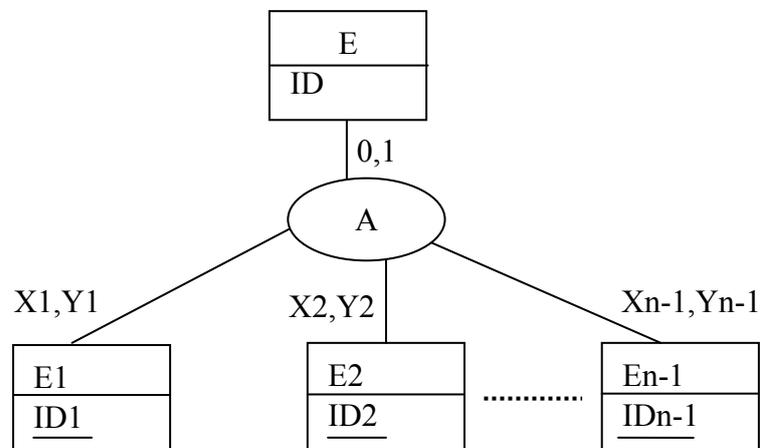


Figure 83. Exemple 2 d'une association n -aire avec un rôle monovalué.

```

CREATE TRIGGER association_n_aire
BEFORE INSERT
ON E
WHEN
    (ID1≠NULL
    AND
    (ID2= NULL
    OR
    |
    OR
    IDn-1 = NULL))
OR
    (ID2≠ NULL
    AND
    (ID1= NULL
    OR
    ID3= NULL
    AND
    |
    AND
    IDn-1 = NULL))
OR
    |
OR
    (IDn-1 ≠ NULL
    AND
    ID1= NULL
    OR
    ID2= NULL
    OR
    |
    OR
    IDn-2 = NULL)
)
ABORT TRANSACTION;

```

Figure 84. Définition d'un trigger permettant l'élimination d'une association *n*-aire avec un rôle monovalué.

Cette règle de transformation est valable pour toutes les associations avec un rôle monovalué, même les associations d'agrégation.

B.3 Transformation d'un lien d'héritage

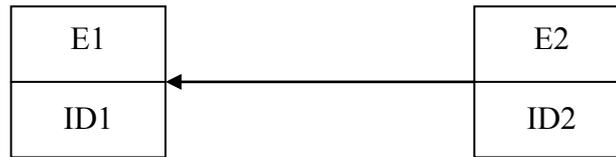


Figure 85. Présentation d'un lien d'héritage.

Lors de la transformation du lien *is-a* entre les deux entités de la Figure 85, nous devons ajouter dans la table *E2* la clé de *E1* (*ID1*) qui devient une clé externe référençant *E1* et une clé primaire de *E2*.

B.4 Eliminations d'attributs

Elles consistent à éliminer un attribut spatial ou temporel en dépendance fonctionnelle avec un autre attribut, à éliminer un attribut complexe spatial ou temporel en dépendance fonctionnelle avec un attribut clé, à éliminer un attribut multivalué et à éliminer un attribut spatial d'une entité spatiale en une nouvelle entité [BEN 99].

B.4.1 Elimination d'un attribut spatial ou temporel en dépendance fonctionnelle avec un autre attribut

Une dépendance fonctionnelle (DF) est un lien entre les valeurs de plusieurs attributs. Formellement : X et Y sont deux attributs d'un type d'entité (ou d'un type d'association). Il y a dépendance fonctionnelle : $X \rightarrow Y$ si et seulement si la connaissance de la valeur de X détermine une seule valeur de Y .

Pour transformer un attribut spatial ou temporel en dépendance fonctionnelle avec un attribut on applique les règles citées ci-dessous.

Règle générale : soit une entité $E1$ qui ne possède que des attributs simples. S'il y a dans $E1$ un groupe d'attributs reliés par des dépendances fonctionnelles, alors on en fait un autre type d'entité $E2$ et on relie les deux entités $E1$ et $E2$ par une association sachant que cette dernière aura une cardinalité $1-1$ du côté de $E1$ et $1-n$ du côté de $E2$.

Règle spatiale1 : s'il existe parmi les attributs qui sont en dépendance un et un seul qui est spatial alors on élimine l'attribut spatial, et $E2$ devient spatiale, de même spatialité que l'attribut éliminé. De plus s'il existe une contrainte d'intégrité spatiale entre l'entité $E1$ et l'attribut éliminé alors l'association entre $E1$ et $E2$ devient spatiale.

Règle spatiale2 : s'il existe parmi les attributs qui sont en dépendance plusieurs qui sont spatiaux alors on élimine seulement celui qui caractérise la

spatialité de $E2$. Donc $E2$ devient spatiale, de même spatialité que l'attribut éliminé et les autres attributs spatiaux seront des attributs de l'entité $E2$. De plus, s'il existe une contrainte d'intégrité spatiale entre l'entité $E1$ et l'attribut éliminé alors l'association entre $E1$ et $E2$ devient spatiale.

Règle temporelle : s'il existe parmi les attributs qui sont en dépendance ceux qui définissent le cycle de vie de l'entité $E2$ alors on peut les éliminer, en rendant l'entité $E2$ temporelle. De plus s'il existe une contrainte d'intégrité temporelle entre l'entité $E1$ et les attributs éliminés alors l'association entre $E1$ et $E2$ devient inter-temporelle.

B.4.2 Elimination d'un attribut complexe spatial ou temporel en dépendance fonctionnelle avec un attribut clé

Dépendance à source double : on appelle dépendance à source double s'il y a dans une entité $E1$ un attribut qui dépend fonctionnellement de l'attribut clé de $E1$ et d'un autre attribut.

Pour transformer un attribut complexe spatial ou temporel en dépendance fonctionnelle avec un attribut clé on applique les règles citées ci-dessous.

Règle générale : s'il y a dans une entité $E1$ une dépendance à source double tel qu'il existe dans $E1$ un attribut complexe (b) qui dépend fonctionnellement d'un autre attribut clé (a) de $E1$ et si l'attribut (b) est composé des attributs (c) et (d) et si (c) est en dépendance fonctionnelle de (a) et (d) alors on fait de (b , d) un autre type d'entité $E2$ et l'attribut (c) devient un attribut de l'association reliant $E1$ et $E2$.

Règle temporelle : si l'attribut (b) est temporel et la temporalité dépend de lui et de l'attribut clé de $E1$ alors l'association entre $E1$ et $E2$ devient temporelle.

B.4.3 Elimination d'un attribut multivalué

Dans le cas de transformation d'un attribut multivalué, complexe ou simple, on applique les règles ci-dessous.

Règle générale : soit une entité $E1$ sans dépendance à source double. S'il y a dans $E1$ un attribut multivalué (a), alors on peut en faire un autre type d'entité $E2$ relié à $E1$ par une association.

Règle spatiale1 : si l'attribut (a) contient un et un seul attribut spatial alors on élimine cet attribut spatial et l'entité $E2$ devient spatiale et de même spatialité que l'attribut éliminé. De plus, s'il existe une contrainte d'intégrité spatiale entre l'entité $E1$ et l'attribut (a) alors l'association entre $E1$ et $E2$ devient spatiale.

Règle spatiale 2 : si l'attribut (a) contient plusieurs attributs spatiaux alors on élimine seulement celui qui caractérise la spatialité de l'entité $E2$. Donc l'entité $E2$ devient spatiale et de même spatialité que l'attribut éliminé et les autres

attributs spatiaux seront des attributs de l'entité $E2$. De plus, s'il existe une contrainte d'intégrité spatiale entre l'entité $E1$ et l'attribut (a) alors l'association entre $E1$ et $E2$ devient spatiale.

Règle temporelle : si l'attribut (a) est temporel et que la temporalité ne dépend que de lui alors l'entité $E2$ devient une entité temporelle. De plus, s'il existe une contrainte d'intégrité temporelle entre l'entité $E1$ et l'attribut (a) alors l'association entre $E1$ et $E2$ devient inter-temporelle.

B.4.4 Transformation d'un attribut spatial d'une entité spatiale en une nouvelle entité

Si une entité E spatiale possède un attribut spatial alors on peut transformer cet attribut en une nouvelle entité spatiale $E2$ liée à E par une association.

B.5 Transformation d'une entité dont la spatialité est temporelle en une entité classique

S'il existe une entité E dont la spatialité est temporelle alors il faut la transformer en une autre entité dans laquelle on ajoute un attribut complexe de type liste de 1 à n qui contient un attribut géométrie spécifiant la spatialité et deux attributs de type date qui définissent la date de début et la date de fin de la valeur de géométrie. Nous devons ajouter deux triggers pour vérifier la cohérence des dates : un trigger pour vérifier que la date de début est inférieure à la date de fin et un trigger pour vérifier lors de la fin de la valeur de l'attribut Géométrie que la date de fin est supérieure à la date de début. Puis, nous devons appliquer la règle de transformation d'un attribut multivalué pour pouvoir passer au niveau relationnel.

B.6 Elimination de la temporalité d'une association temporelle

Comme pour la transformation d'une entité temporelle en entité non temporelle, lors de l'élimination de la temporalité d'une association temporelle A , nous devons ajouter dans A des attributs spécifiant le cycle de vie de l'association. Nous devons aussi ajouter des triggers pour vérifier la cohérence des dates.

Après avoir effectué cette transformation, nous devons appliquer la règle de transformation d'une association n -aire, selon le rôle de l'association : monovalué ou multivalué, pour pouvoir passer au niveau relationnel.

B.7 Elimination d'un cas de raffinement

Si une entité spatiale E super-classe de n sous-classes $E1, E2, E3, \dots, En$ spatiales dont la géométrie est raffinée alors lors de la transformation de E en table nous ne devons pas spécifier la géométrie de E car la géométrie sera représentée dans les sous classes.

ANNEXE C.

CAHIER DES CHARGES ET SPECIFICATION D'UNE APPLICATION GEOMATIQUE TEMPS REEL : « UN SYSTEME DE DETECTION D'ALARME DANS LES WSN »

C.1 Introduction

Les réseaux de capteurs sans fil (WSN) connaissent une évolution ces dernières années. Ces réseaux ont un intérêt particulier pour les applications militaires, environnementales, domotiques, médicales, et bien sûr les applications liées à la surveillance des infrastructures critiques. Ces applications ont souvent besoin d'un niveau de sécurité élevé. Or, de part leurs caractéristiques (absence d'infrastructure, contrainte d'énergie, topologie dynamique, nombre important de capteurs, sécurité physique limitée, capacité réduite des nœuds,...), la sécurisation des réseaux de capteurs est à la source, aujourd'hui, de beaucoup de défis scientifiques et techniques.

Nous présentons en premier lieu leurs principes de fonctionnement, leurs domaines d'applications et leurs avantages majeurs mais aussi la problématique de ce type de réseau. Nous présentons ensuite une étude conceptuelle de l'application concernant la détection d'alarmes dans les WSN en utilisant notre profil GRT. Et finalement nous présentons l'implémentation de cette application.

C.2 Les Réseaux WSN

Les récentes avancées dans les domaines des technologies sans fil et électroniques ont permis le développement à faible coût de minuscules capteurs consommant peu d'énergie. Ces capteurs ont trois fonctions principales qui sont le captage des données (de type son, vibration, lumière,...), le calcul des informations à l'aide de ces valeurs collectées et la communication à travers un réseau de capteurs. Pour ce faire, les micro-capteurs, intègrent une unité de

capteur chargée de capter des grandeurs physiques et de les transformer en grandeurs numériques, une unité de traitement informatique et de stockage de données et un module de transmission sans fil. De ce fait, les micro-capteurs sont de véritables systèmes embarqués. Le déploiement de plusieurs d'entre eux, en vue de collecter et de transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome, forme un réseau de capteurs sans fil.

C.2.1 Architecture

Les réseaux de capteurs sans fil, sont considérés comme un type spécial des réseaux ad hoc. La position des nœuds n'est pas obligatoirement prédéterminée. Ils sont dispersés aléatoirement à travers une zone géographique, appelée champ de captage, qui définit le terrain d'intérêt pour le phénomène capté. Les données captées sont acheminées à un nœud considéré comme un "point de collecte", appelé nœud puits (ou sink). Ce dernier peut être connecté à l'utilisateur du réseau via Internet ou satellite. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais des nœuds puits (Figure 86).

Un réseau de capteurs sans fil consiste en une grande population, relativement dense, de capteurs dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée.

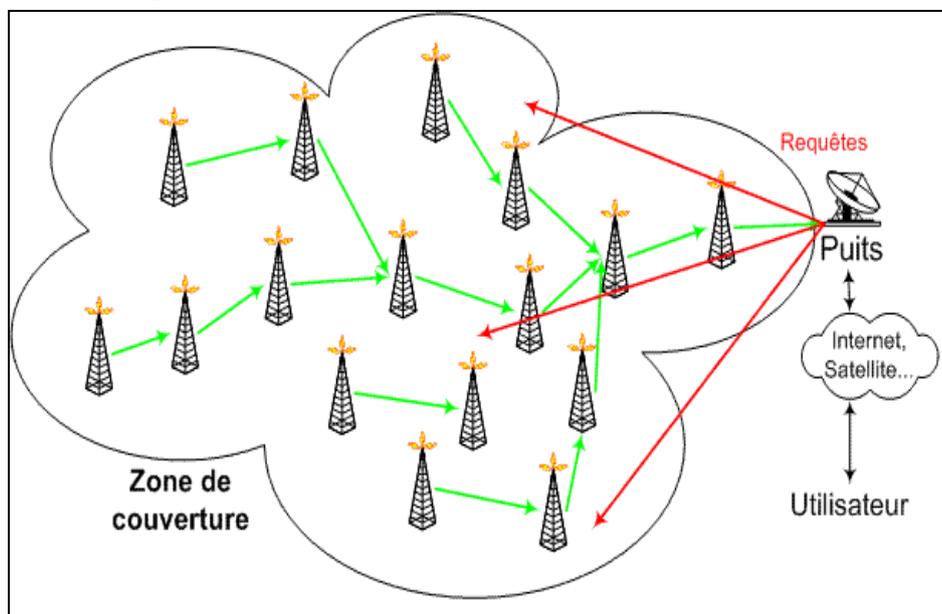


Figure 86. Architecture du WSN.

C.2.2 Composition d'un Capteur

Les capteurs sont des dispositifs de tailles extrêmement réduites avec des ressources très limitées, autonomes, capable de traiter des informations et de les transmettre, via les ondes radio, à une autre entité (capteurs, unité de traitements...) sur une distance limitée à quelques mètres. Un capteur analyse son environnement, et propage les données récoltées aux capteurs appartenant à sa zone de couverture. Chaque capteur reliant l'information sur sa propre zone de couverture, le réseau se trouve entièrement couvert. Les réseaux de capteurs sans fil sont capables d'analyser et de surveiller une zone définie, présentent des avantages indéniables dans divers domaines. Des transmetteurs sont prévus pour recueillir et transférer les mesures issues de capteurs sur lesquels ils se connectent. Un nœud d'un réseau de capteur est composé de quatre sous-systèmes principaux. Il s'agit des sous-systèmes de calcul, de communication, de capteurs, et de génération de courant. La Figure 87 présente l'architecture d'un nœud de réseau de capteurs, et ses différentes composantes [CAR 06].

L'architecture comporte nécessairement un composant de communication pour pouvoir diffuser les rapports vers le reste du réseau et recevoir des informations des autres éléments. La plupart du temps, ce composant de communication est bidirectionnel et peut non seulement émettre, mais aussi recevoir des informations. De plus, dans la plupart des cas, ce composant utilise des communications radio, bien que quelques exceptions existent dans certaines implémentations. La partie contrôle est constituée d'un microcontrôleur basse consommation avec une capacité en mémoire RAM (Random Access Memory) plutôt limitée ainsi qu'une mémoire flash contenant le programme. Ce dernier est la plupart du temps construit autour d'un système d'exploitation adapté pour des réseaux de capteurs. Un système d'alimentation permet d'alimenter en énergie les différents systèmes. L'alimentation comprend des batteries et éventuellement un système de conversion de tension si les tensions d'alimentation des systèmes sont incompatibles entre elles et dans le but de mieux exploiter tous les états de charge des batteries (pleine charge, charge minimale). Certaines implémentations incluent un récupérateur d'énergie.

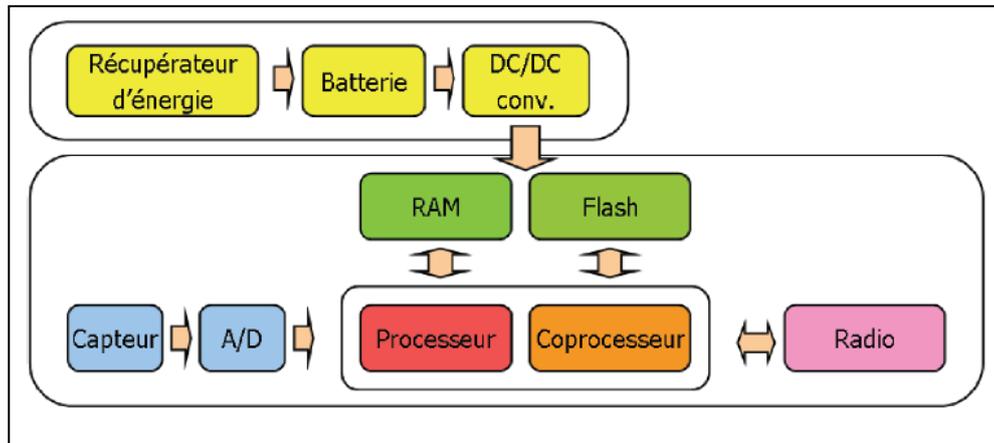


Figure 87. Architecture d'un nœud de réseau de capteurs [CAR 06].

C.2.3 Problématique

Les réseaux de capteurs étant constitués de nombreux nœuds, ils sont en mesure de tirer parti de la densité du réseau et des redondances pour apporter de la tolérance aux pannes et de la précision. La taille réduite des capteurs favorise en outre le déploiement en nombre de ces derniers.

Les capteurs sans fils sont des éléments indépendants les uns des autres. Par conséquent, ils doivent également disposer d'une alimentation autonome. Leur durée de vie est limitée par la durée de vie de leur batterie puisqu'ils utilisent leur réserve d'énergie à des fins de calcul et de transmission de données. Cette contrainte forte a une influence majeure sur l'ensemble des techniques mises en place pour le déploiement de tels réseaux. Un effet majeur de cette limitation énergétique est la limitation maximale des transmissions par voie hertzienne, très coûteuses. Il est donc primordial d'effectuer tant que possible le traitement de l'information localement au niveau du capteur. Dans un WSN, chaque capteur joue le rôle d'émetteur et de routeur. Une défaillance énergétique d'un capteur peut changer significativement la topologie du réseau et imposer une réorganisation coûteuse de ce dernier.

De plus pour la tolérance aux pannes, certains nœuds peuvent être défaillants à cause d'un manque d'énergie, d'un dommage physique ou d'une interférence. Ceci ne doit pas affecter la globalité de la tâche du réseau de capteurs. En cas de défaillance, de nouveaux liens et routes doivent être établis pour assurer la collecte des données. La redondance peut également être utilisée, tout en veillant à conserver une faible consommation d'énergie.

Enfin, dans diverses applications, la donnée doit être transmise dans une certaine plage de temps, après quoi, elle n'aura aucune importance. Pourtant,

dans la plupart des applications, la durée de vie du réseau est favorisée au détriment de la qualité d'émission des données.

Les problématiques sont donc très éloignées de celles des réseaux classiques, telle la maximisation du débit mais il s'agit plutôt d'étendre la durée de vie du système et sa robustesse, en cas de chute de certains nœuds seulement.

Les problèmes majeurs qui peuvent causer des défaillances sont : le problème du niveau d'énergie dans un capteur, le problème de destruction d'un capteur, le problème de panne d'un capteur, etc.

C.2.3.1 Problème du niveau d'énergie dans un capteur

Un capteur a normalement une durée de vie égale à une année, cependant on doit contrôler son niveau de batterie afin de maximiser cette durée. Lorsqu'un capteur détecte que son niveau d'énergie est faible (30%), une alarme se déclenche au niveau de la carte.

Le problème du niveau d'énergie présente des priorités, à savoir trois priorités :

- 1^{ère} accordée au Sink,
- 2^{ème} accordée à un capteur qui se trouve à l'extrémité du réseau,
- 3^{ème} accordée à un capteur qui est à l'intérieur du réseau.

Selon la priorité du capteur concerné, l'alarme peut changer et l'administrateur doit réagir selon cette priorité.

- Si c'est un Sink (priorité la plus élevée) alors il faut faire une intervention immédiate.
- Si la priorité est inférieure c'est-à-dire un capteur se trouve à l'extrémité du réseau ou à l'intérieur du réseau alors l'administrateur prend sa décision selon les ressources qu'il dispose.

C.2.3.2 Problème de destruction d'un capteur

Les capteurs sont placés dans l'environnement extérieur, donc ils sont en face des perturbations climatiques, des interventions humaines ou d'autres. Si un capteur est détruit, suite à l'une des causes précédemment citées, il doit être remplacé immédiatement et on doit récupérer les données qui seront encore dans sa mémoire et non transmises.

C.2.3.3 Problème de panne d'un capteur

On distingue deux types de pannes d'un capteur :

- une panne qui affecte le comportement du capteur, mais il reste toutefois fonctionnel. Dans ce cas, l'administrateur va se rendre compte par des informations erronées envoyées par ce capteur ou par des retards de transmission ou de détection. Un agent interviendrait sur le champ pour résoudre le problème identifié par l'administrateur,

- une panne qui affecte le fonctionnement du capteur, c'est-à-dire le capteur ne fonctionne plus. Ce type de panne est généralement matériel. Dans ce cas le capteur n'envoie plus de données, et l'administrateur se rend compte du problème dès qu'il reçoit les données concernant la zone couverte par ce capteur. Un agent de maintenance interviendrait dans ce cas pour réparer le capteur en question.

C.3 Modélisation de l'application

Dans cette partie nous détaillons l'étude conceptuelle de l'application. Nous avons modélisé trois types de diagrammes (le diagramme de priorités, le diagramme de classes et le diagramme d'interventions) selon le profil GRT et en utilisant notre AGL AGRT.

C.3.1 Diagramme de priorités

Après l'étude des spécifications, nous avons élaboré le diagramme de priorités qui est représenté dans la Figure 88. Il existe essentiellement trois acteurs dans le système de détection de pannes dans les réseaux WSN.

- **Agent de maintenance** : il recevra des ordres de réparation de la part de l'administrateur et interviendra pour faire le nécessaire afin de garantir un bon fonctionnement du réseau. Il ne possède donc que des privilèges relatifs à la réparation des capteurs défectueux.
- **Administrateur** : c'est le premier responsable de la gestion du réseau, il possède tous les privilèges et a un accès total au centre de traitement automatique, pour collecter les informations ou pour superviser l'état du réseau.
- **Centre de traitement automatique** : c'est une partie software qui permet de récupérer les données à partir des capteurs et donnera à l'administrateur l'accès à ces dernières.

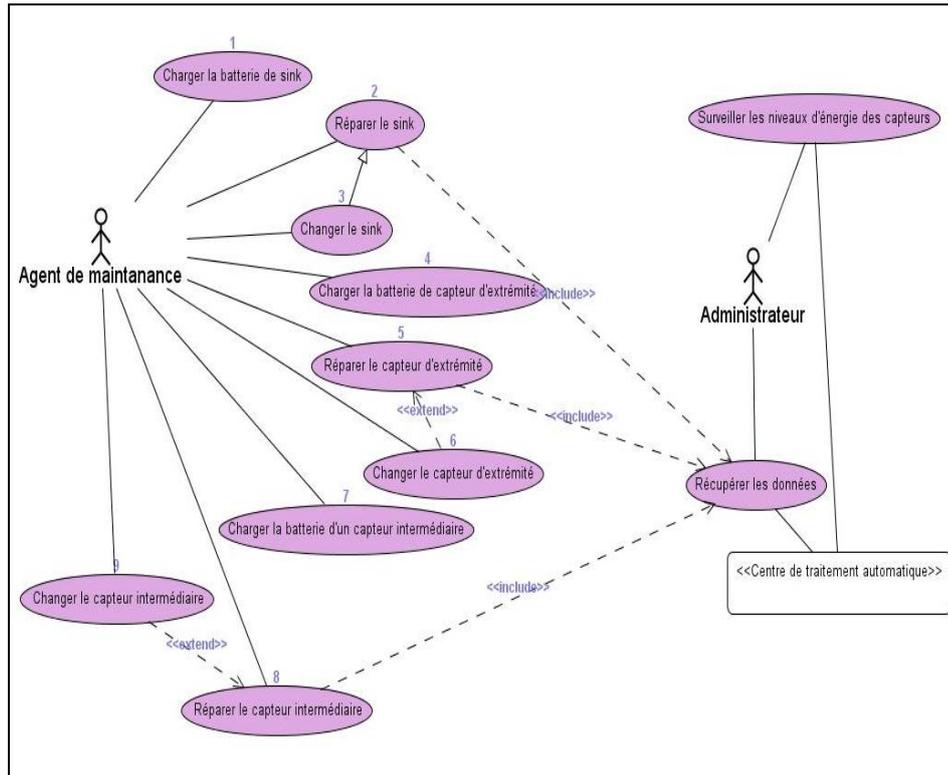


Figure 88. Diagramme de priorités.

Pour chaque type de capteur nous avons associé trois cas d'utilisation classés selon leurs degrés de priorités, mais le « sink » reste l'élément le plus important d'où la nécessité de sa réparation le plus rapide possible, nous avons donc affecté les trois premières priorités aux opérations qui lui concerne. Les autres priorités sont affectées aux opérations appliquées sur les capteurs d'extrémités et les capteurs intermédiaires car leurs traitements ne nécessitent pas une réaction immédiate.

C.3.2 Diagramme de classes

Le diagramme de classes présente l'architecture conceptuelle du système, il décrit les classes que le système utilise ainsi que les différentes associations entre elles. Il est illustré par la Figure 89. Nous présentons d'une façon globale les différentes classes :

- Capteur : cette classe désigne un capteur dans le réseau WSN.
- Sink : cette classe admet comme attributs ceux de sa classe mère ainsi qu'une méthode `Diffusion_infos ()` puisqu'il est un capteur puits.
- Capteur_d'extrémité : cette classe spécifie les capteurs d'extrémités.

- Capteur_intermédiaire: cette classe spécifie les capteurs intermédiaires.
- Batterie: cette classe possède trois attributs Référence, Capacité et Niveau_energie qui spécifient le traitement des pannes de chargements.
- Alarme: cette classe désigne les alarmes, elle est la classe-mère de trois types d'alarmes: A_batterie (alarmes engendrés par les batteries), A_panne (alarmes dues aux pannes dans un capteur) et A_destruction (alarmes engendrés par la destruction d'un capteur).
- Centre-Traitement: cette classe désigne le centre du traitement.
- Agent_maintenance: cette classe spécifie l'agent de maintenance.
- Pôle: représente la surface qui englobe tout le système d'un réseau de capteurs.

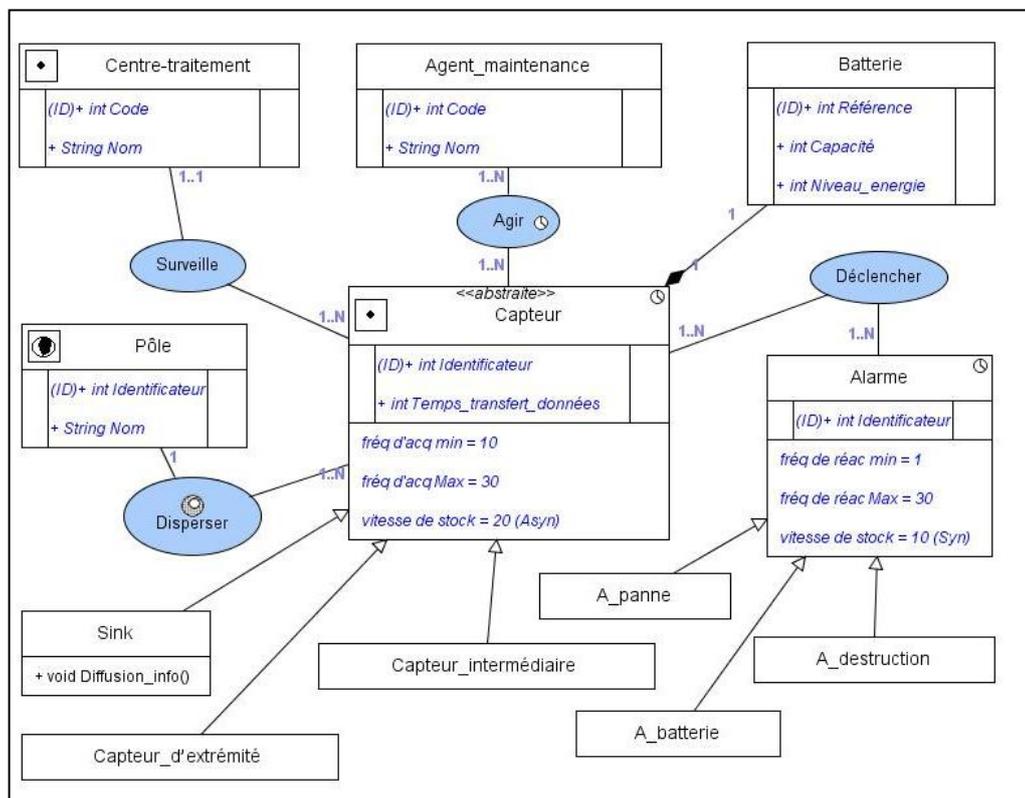


Figure 89. Diagramme de classes.

C.3.3 Diagrammes d'interventions

Les diagrammes d'interventions permettent d'étudier la dynamique du système à travers des scénarios. Pour cette application nous avons décrit quatre scénarios pour bien exprimer le fonctionnement du système.

Dans un premier scénario nous avons simulé la réception d'une alarme de chargement de batterie (on a pris comme exemple le cas d'alarme du « Sink »). Chaque capteur dispose d'un détecteur du niveau d'énergie de batterie qui se déclenche lorsqu'il atteint un niveau de 30%, quand l'administrateur reçoit l'alarme à partir du centre du traitement il vérifie le type d'alarme et sa priorité pour préciser la durée maximale de réparation du capteur. Toutes ces informations seront envoyées à l'agent de maintenance pour agir selon la marge temporelle indiquée (Figure 90).

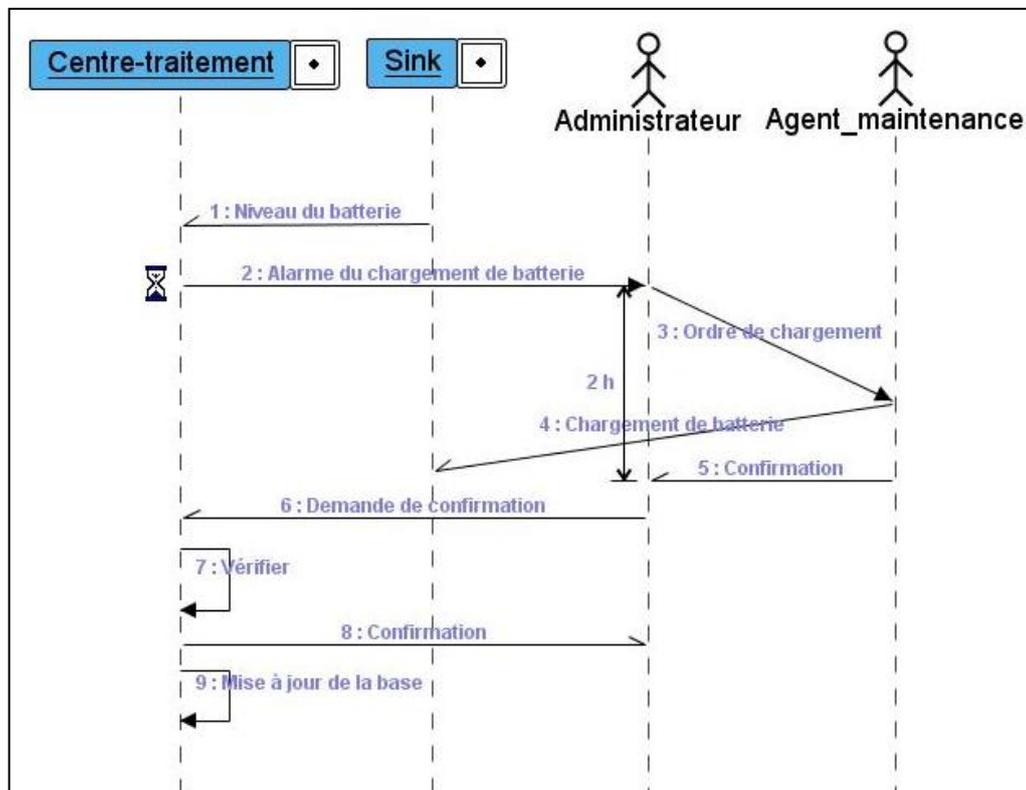


Figure 90. Scénario 1 : déclenchement d'une alarme du « Sink ».

Dans un deuxième scénario, nous avons décrit la réception des deux alarmes de chargement de batterie de deux capteurs différents (Figure 91). Nous donnons toujours la priorité au capteur Sink puisqu'il est le plus prioritaire. L'administrateur fait son raisonnement en se basant sur l'importance de

l'emplacement de capteur dans le réseau, ces degrés d'importances sont classés comme suit : capteur Sink, capteur d'extrémité, et capteur intermédiaire. Ce raisonnement ce fait quelque soit le type d'alarme.

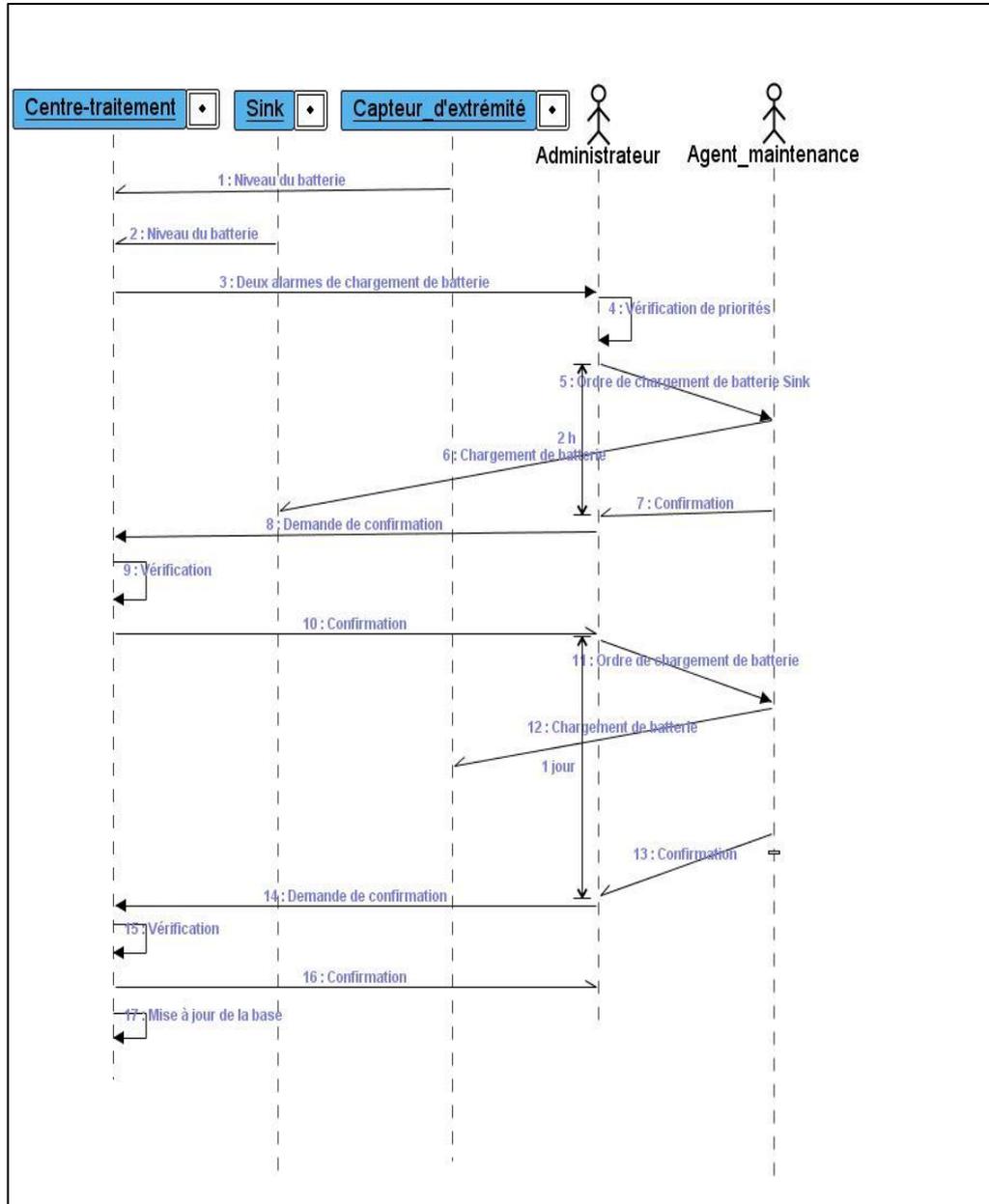


Figure 91. Scénario 2 : déclenchement de deux alarmes de priorités différentes.

Dans un troisième scénario, nous avons décrit la réception d'une alarme de panne de capteur (Figure 92). En cas de panne de capteur, le centre du traitement détecte un retard au niveau du temps de transfert des données du capteur défaillant ou une absence totale des informations. Il y a un message d'alarme qui se déclenche pour que l'administrateur agisse en temps réel. Ce dernier envoie un ordre de réparation à l'agent de maintenance et lui demande de récupérer les données non émises enregistrées dans la mémoire du capteur.

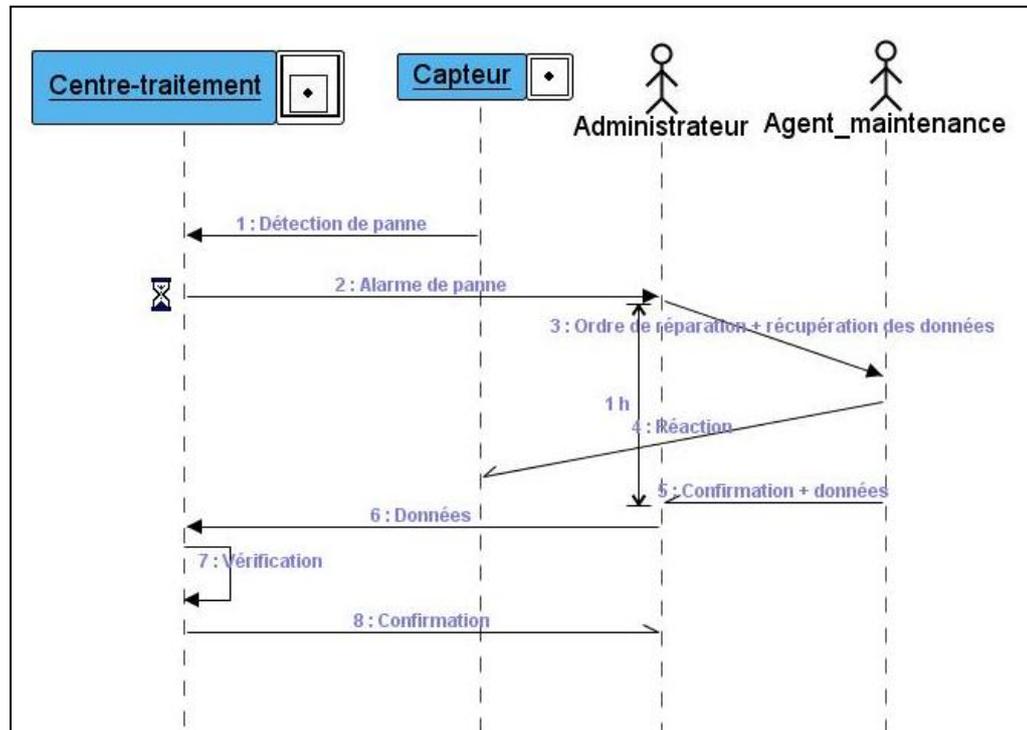


Figure 92. Scénario 3 : déclenchement d'une alarme spécifiant une panne d'un capteur.

Dans un quatrième scénario, nous avons décrit la réception d'une alarme de destruction de capteur (Figure 93). En cas de destruction de capteur, le centre du traitement détecte une absence totale de transfert de données. L'administrateur n'exige pas une récupération des données puisque la destruction peut affecter la mémoire. D'où une opération de changement de capteurs est suffisante.

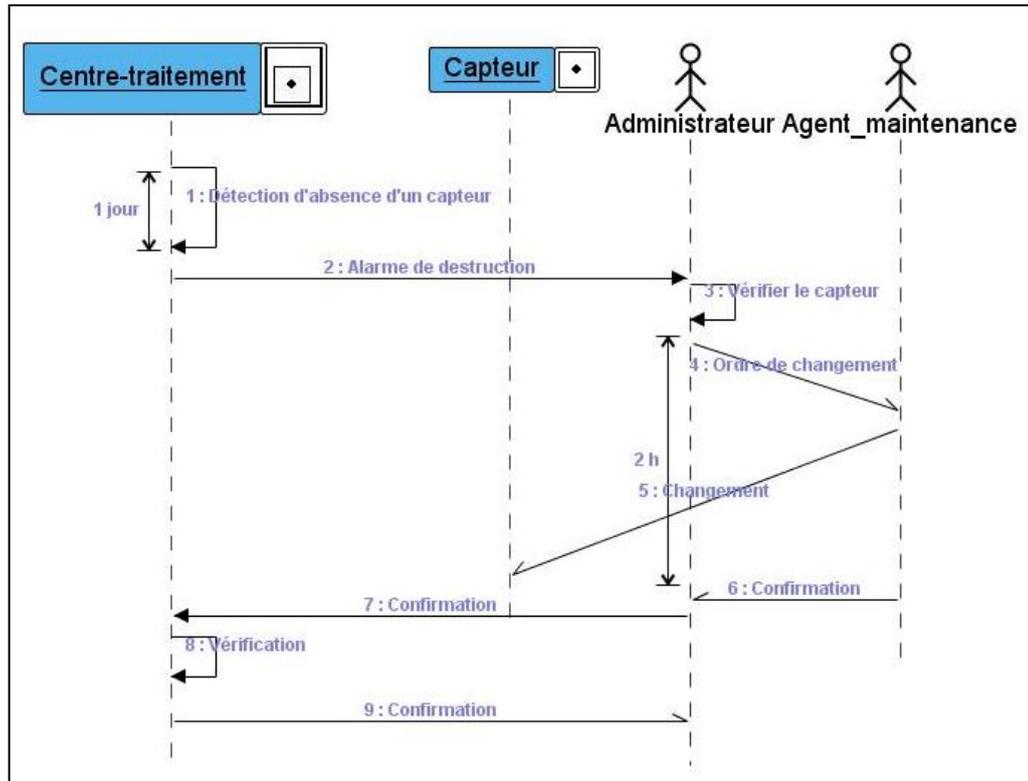


Figure 93. Scénario 4 : déclenchement d'une alarme spécifiant une destruction d'un capteur.

C.4 Implémentation de l'application

L'application vise à permettre la supervision des capteurs dans un réseau de capteurs sans fils afin de prévoir les chutes de batterie ou les défaillances critiques au niveau de ces derniers. Pour le faire, nous avons réalisé une carte représentant un réseau de capteur dans le pôle technologique El Ghazela destiné à la supervision, à partir de laquelle nous aurons la possibilité de voir l'état des capteurs en temps réel et donc vérifier les réparations faites au niveau du réseau. Pour réaliser la phase d'implémentation nous avons utilisé le système d'information géographique ArcView qui est un puissant outil pour gérer, visualiser, interroger et analyser des données géographiques.

L'interface représentée par la Figure 94 montre la cartographie du pôle El Ghazela avec les capteurs disséminés près des différentes entreprises.



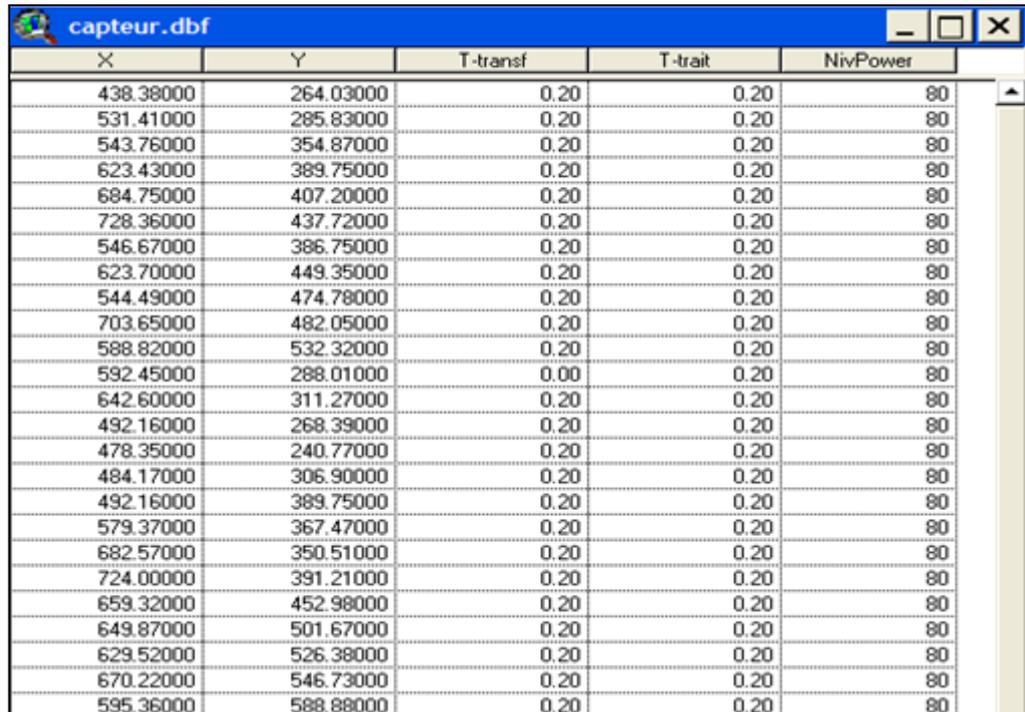
Figure 94. Cartographie du réseau de capteurs sans fils.

Afin d'insérer la cartographie, il faut tout d'abord commencer par créer une vue (*View*) à laquelle nous avons ajouté les thèmes. Sachant qu'un thème est un ensemble de données géographique à visualiser sous forme de carte.

Ensuite nous avons ajouté les capteurs sur la cartographie en conservant leurs spatialités, par conséquent ils seront représentés sur la carte comme des points. Chaque capteur aura un symbole différent de l'autre pour être distinguer sur la carte :

- les capteurs ordinaires sont représenté par le symbole « **C** » avec la couleur vert foncé,
- les capteurs d'extrémité sont représentés quant à eux par le symbole « **#** » avec la couleur vert clair,
- et pour le Sink il est représenté par le symbole « **S** » avec la couleur noir.

Il faut spécifier pour chaque type de capteur une table qui contient ses différentes informations (Figure 95).



X	Y	T-transf	T-trait	NivPower
438.38000	264.03000	0.20	0.20	80
531.41000	285.83000	0.20	0.20	80
543.76000	354.87000	0.20	0.20	80
623.43000	389.75000	0.20	0.20	80
684.75000	407.20000	0.20	0.20	80
728.36000	437.72000	0.20	0.20	80
546.67000	386.75000	0.20	0.20	80
623.70000	449.35000	0.20	0.20	80
544.49000	474.78000	0.20	0.20	80
703.65000	482.05000	0.20	0.20	80
588.82000	532.32000	0.20	0.20	80
592.45000	288.01000	0.00	0.20	80
642.60000	311.27000	0.20	0.20	80
492.16000	268.39000	0.20	0.20	80
478.35000	240.77000	0.20	0.20	80
484.17000	306.90000	0.20	0.20	80
492.16000	389.75000	0.20	0.20	80
579.37000	367.47000	0.20	0.20	80
682.57000	350.51000	0.20	0.20	80
724.00000	391.21000	0.20	0.20	80
659.32000	452.98000	0.20	0.20	80
649.87000	501.67000	0.20	0.20	80
629.52000	526.38000	0.20	0.20	80
670.22000	546.73000	0.20	0.20	80
595.36000	588.88000	0.20	0.20	80

Figure 95. Exemple d'une table relative à un capteur.

Afin de tester la possibilité de détecter les pannes possibles en temps réel, nous avons créé un bouton faisant appel à un script pour la mise à jour des données sur les états des capteurs.

Après avoir cliqué sur ce bouton, les données au niveau des tables des différents capteurs seront modifiées. Si un capteur présente un niveau d'énergie inférieur à 30% donc sa couleur changera automatiquement en rouge et une boîte de dialogue se lance indiquant le type d'alarme pour que l'administrateur puisse réagir selon la priorité d'alarme.



Figure 96. Détection des pannes de capteurs.

La Figure 96 montre un exemple d'exécution : deux capteurs ordinaires et un capteur d'extrémité sont affectés par un manque d'énergie. Donc ils sont colorés en rouge jusqu'à une nouvelle mise à jour qui montre la réparation des capteurs.

C.5 Conclusion

Cette application a pour objectif l'implémentation d'un système de détection de pannes dans un réseau de capteurs sans fil. Ce système est capable de détecter un changement dans l'environnement et de réagir en fonction notamment des besoins de l'utilisateur afin d'acheminer vers des points de traitement les différentes données collectées, ainsi qu'une conception d'une procédure pour la tolérance aux pannes puisque l'absence de sécurité physique pour ce type de capteurs, et la nature vulnérable des communications radios sont des caractéristiques qui augmentent les risques de pannes sur ce type de réseau.

ANNEXE D.
CONVENTION DE THESE EN
COTUTELLE

CONVENTION DE THESE

ENTRE LES SOUSSIGNES :

L'Ecole Supérieure des communications de Tunis ci après dénommée « **Sup'Com** », située Route de Raoued Km 3.5, 2083 cité El Ghazala Tunisie, relevant de l'Université 7 Novembre à Carthage, représentée aux fins des présentes par son directeur, Monsieur **Lotfi Ammar** ;

ET :

L'Institut National des sciences appliquées de Lyon ci après dénommée « **INSA-Lyon** » ; situé avenue Albret Einstein 69621 Villeurbanne Cedex **France**, représentée aux fins des présentes par son directeur, Monsieur **Alain STORK**.

IL EST D'ABORD EXPOSE CE QUI SUIVIT :

Vu le décret n°97-1801 du 03 septembre 1997, modifiant et complétant le décret n° 93-1823 du 6 septembre 1993, fixant les conditions d'obtention des diplômes nationaux (tunisiens) sanctionnant les études doctorales.

Vu la loi n° 84-52 du 26 janvier 1984 modifiée sur l'enseignement supérieur, vu l'arrêté du 25 avril 2002 relatif aux études de troisième cycle, vu l'arrêté du 18 janvier 1994 relatif à la création d'une procédure de cotutelle de thèse, prévus par la législation en vigueur en France,

Les deux parties, animées par la volonté de favoriser les échanges de doctorants entre elles et de renforcer, ainsi, la coopération scientifique et universitaire entre la Tunisie et la France, décident d'un commun accord, dans le cadre de la législation dans leurs pays respectifs, d'utiliser la procédure de codirection (cotutelle) concernant :

Mme Malek Ben Youssef ep Ben Jebara, née le 19-07-1974 à Tunis, Tunisie, de nationalité tunisienne.

Adresse dans le pays d'origine : **Ecole supérieure des Communications de Tunis – Route de Raoued, Km 3.5 2083 Cité El Ghazala Tunis – Tunisie.**

Adresse dans le Pays d'accueil : **Bâtiment Blaise Pascal, INSA de Lyon, 7 avenue Capelle F-69621 Villeurbanne Cedex France.**

CECI ETANT EXPOSE, IL A ETE CONVENU ET ARRETE CE QUI SUIVIT

Article 1

Le Doctorant doit être inscrit dans les deux établissements, à l'INSA-Lyon et à Sup'Com. Il doit s'acquitter des droits d'inscription dans un des établissements à savoir Sup'Com et exonéré de ces mêmes droits dans l'autre à savoir l'INSA-Lyon.

Dans ce contexte, les deux parties prennent acte et enregistrent les données suivantes :

1. Date de l'inscription en thèse du doctorant, sous le régime de cotutelle : rentrée universitaire 2004/2005.
2. Durée prévisionnelle des travaux de Recherche dans le cadre de la thèse en accord avec la législation en vigueur trois ans. Cette durée peut être prolongée d'une année au maximum après accord spécifique établi entre les deux établissements signataires de la convention, dans le respect des conditions prévues par les législations en vigueur dans les pays respectifs et après accord du Président de l'Université du 7 novembre à Carthage :
 - Périodes prévues à l'établissement de recherche en Tunisie : 9 à 10 mois par an.
 - Périodes prévues à l'établissement de recherche en France : 2 à 3 mois par an.

Article 2

Lors de son séjour en France, la couverture sociale du doctorant est assurée par lui-même conformément à la législation en vigueur.

Article 3

Dans chacun des établissements concernés, le doctorant effectuera ses travaux de recherche sous la direction et la responsabilité des directeurs de thèse suivants :

- Monsieur Ziad BELHADJ, Maître de conférence, Ecole Supérieure des Communications de Tunis « Sup'Com » ;
- Monsieur Robert Laurini, Professeur, Institut National des Sciences appliquées de Lyon «INSA-Lyon».
- Monsieur Ahmed Lbath, Professeur contractuel, HDR, Institut National des Sciences appliquées de Lyon «INSA-Lyon».

Les directeurs de thèse s'engagent à exercer pleinement et conjointement, auprès du doctorant, les compétences qui leur sont attribuées par la réglementation en vigueur et les traditions universitaires dans leurs pays respectifs.

Article 4

La composition du jury de soutenance obéit à la réglementation en vigueur dans chacun des pays impliqué dans la cotutelle. Le jury de soutenance, désigné par les deux établissements partenaires, sera composé à parité par des représentants scientifiques des deux pays. En tout état de cause, le jury doit comprendre obligatoirement les directeurs de thèse.

Article 5

La thèse, préparée en codirection, sera rédigée en **Français**, sera soutenue en Français et complétée par un résumé d'une page écrite en **anglais** et en **arabe**.

Sup'Com, s'engage à délivrer au doctorant, le titre de docteur, et à transmettre une copie du dossier complet de soutenance à l'institution partenaire, l'INSA-Lyon, qui s'engage à délivrer au doctorant, à son tour, le titre de Docteur.

Article 7

Les modalités de dépôt, signalement et reproduction de la thèse ainsi que l'autorisation de la soutenir obéissent à la réglementation en vigueur dans le pays où a lieu la soutenance.

La date et le lieu de soutenance sont fixés d'un commun accord et notifiés par écrit par les codirecteurs de thèse au chef de l'établissement concerné.

Article 8

La protection du sujet de thèse du doctorant ainsi que la publication, l'exploitation et la protection des résultats de recherche issus des travaux du doctorant dans les deux établissements sont assujetties à la réglementation en vigueur et assurées conformément aux procédures spécifiques à chaque pays impliqué dans la cotutelle.

Les résultats obtenus au cours de la préparation de cette thèse pourront être publiés, mais resteront la propriété conjointe des deux établissements Sup'Com et l'INSA-Lyon.

Article 9

La conclusion de la présente convention a été préalablement, autorisée par la cotutelle du *Président de l'Université du 7 Novembre à Carthage* en date du *1^{er} décembre 2004* sous le n° *22...* et par Monsieur le *Directeur de l'INSA-Lyon*.

Article 10

Soucieux de l'intérêt des doctorants et du développement de la coopération entre leurs pays respectifs, les établissements d'enseignement supérieur et de recherche sus-indiqués s'engagent à respecter les dispositions ci-dessus et à faire tout ce qui est nécessaire pour l'application de la présente convention dans les meilleures conditions.

En cas de litige, les parties à la présente convention s'engagent à rechercher toute solution amiable avant d'en décider la résolution.

Article 11

Au cas où le régime de cotutelle viendrait à être dénoncé par une des parties concernées, celle-ci devra le notifier par écrit à son établissement d'origine (Sup'Com) en indiquant les raisons de sa décision.

L'établissement d'origine (Sup'Com) devra en informer l'établissement d'accueil (INSA-Lyon) et l'Université du 7 Novembre à Carthage dans un délai d'un mois.

Le Directeur de Sup'Com
L.Ammar



Le Directeur de l'INSA-Lyon
par délégation Jolion Jean Michel

Le Directeur de Thèse
Z.Belhadj

Le Directeur de Thèse
R.Laurini

Le Directeur de Thèse
A.Lbath

AVENANT RELATIF A LA CONVENTION DE THESE EN CO-TUTELLE DE MALEK BEN YOUSSEF

En complément à la convention de thèse en co-tutelle pour Mme Malek BEN YOUSSEF, signée le 01 décembre 2004 entre l'Institut National des Sciences Appliquées de Lyon (INSA), et L'Ecole Supérieure des Communications de Tunis ci après dénommée « Sup'Com », relevant de l'Université 7 Novembre à Carthage, l'article 6 est ainsi modifié.

Article 6.1 : Composition du jury

Le jury sera composé de six personnes dont trois nommés par l'INSA et trois nommés par Sup'Com. Seront membres de droit du jury :

- les directeurs de thèse
- deux rapporteurs, l'un proposé par l'INSA et l'autre par Sup'Com ; celui choisi par l'INSA devra être au moins habilité à diriger des recherches ; celui choisi par Sup'Com doit être avoir reçu l'habilitation universitaire tunisienne.

Article 6.2 : Président du jury

Le président du jury sera choisi parmi les examinateurs avant la soutenance. Il ne devra être ni un directeur ou co-directeur de thèse ni rapporteur.

Article 6.3 : Absence d'un des membres du jury

De manière à garantir la parité du jury, si, au dernier moment, un membre du jury était dans l'obligation de ne pas pouvoir participer à la soutenance, la soutenance aura lieu malgré tout et il devra d'urgence communiquer au président du jury un rapport en français donnant son jugement sur la thèse.

Article 6.4 : Soutenance

Une unique soutenance aura lieu à Lyon, sera effectuée en langue française et durera environ 45 minutes.

Article 6.5 : Procès-verbal et mentions possibles

Un seul rapport sera rédigé par le président et signé par tous les membres du jury. Ce rapport, long d'une page environ, sera rédigé en français.

Conformément à la décision du Conseil Scientifique de l'INSA de Lyon, il ne sera délivré aucune mention.

Le 3 octobre 2008

Pour l'INSA de Lyon

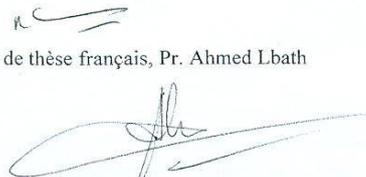
Pour le directeur de l'INSA, Pr. Alain STORCK,
par délégation Pr. Daniel BARBIER, directeur
adjoint à la Recherche, chargé des études
doctorales



Le directeur de thèse français, Pr. Robert
LAURINI



Le directeur de thèse français, Pr. Ahmed Lbath



Pour Sup'Com de Tunis

Le directeur de l'Ecole Supérieure des
Communications de Tunis, Naceur AMMARI



Le directeur de thèse tunisien, Pr. Ziad BELHADJ




FOLIO ADMINISTRATIF

THESE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE LYON

NOM :	Ben Jebara née Ben Youssef -	Prénoms : Malek	DATE de SOUTENANCE : 07/07/2010
TITRE :	Modélisation conceptuelle pour les applications géomatiques temps réel		
NATURE :	Doctorat Numéro d'ordre : 2010 ISAL 44 - Ecole doctorale : Infomaths - Spécialité : Informatique		
Cote B.I.U. - Lyon :	T 50/210/19 /	et	bis CLASSE :
RESUME :			
<p>Le domaine des applications géomatiques est en plein essor et le nombre de logiciels spécialisés dans la manipulation des données spatio-temporelles a connu une croissance considérable cette dernière décennie. Cette évolution a été accompagnée, dans la littérature, par la proposition de plusieurs méthodes de conception dédiées aux applications géomatiques. Les applications temps réel connaissent également une grande évolution et font référence à la localisation spatiale.</p> <p>L'avènement des télécommunications a suscité l'intérêt de prendre en compte la distribution et la répartition de l'information pour les applications géomatiques. La discipline de télégéomatique est née d'une fusion de la géomatique et des télécommunications en se caractérisant par l'usage des systèmes modernes de positionnement de type GPS (Global Positioning System).</p> <p>Actuellement, l'émergence des systèmes d'information pervasifs ouvre plusieurs axes de recherche. Ainsi les applications géomatiques temps réel soulèvent plusieurs problématiques : l'architecture logicielle de telles applications doit supporter des modèles mathématiques et des algorithmes qui ont des capacités de calculs importantes, la prise en compte de la granularité du temps réel et la conception et la modélisation de telles applications doit permettre de formaliser les spécifications et d'apporter des outils logiciels cohérents par rapport aux besoins des utilisateurs et des contraintes liées à la réactivité et éventuellement la mobilité des composants du système.</p> <p>Les différents formalismes qui existent dans la littérature et qui sont supportés par des méthodes de conception des applications géomatiques et des applications temps réel sont insuffisants et ne répondent pas au besoin du concepteur.</p> <p>Un travail de recherche dans la prise en compte au niveau conceptuel des contraintes liées à la dimension temps réelle de la nouvelle génération des applications géomatiques temps réel est certainement prometteuse.</p> <p>Les recherches que nous avons menées nous ont permis de proposer des concepts de base s'appuyant sur des extensions d'UML (Unified Modeling Language) afin de prendre en compte au niveau conceptuel les problématiques de communications et de localisation des données et des applications.</p> <p>Nous proposons un profil UML pour la conception des applications géomatiques temps réel. C'est le profil GRT (Geographic Real Time) dédié à la conception des applications géomatiques temps réel. Ce travail débouche sur un prototype doté d'un atelier de génie logiciel supportant le formalisme proposé. Une validation à travers des exemples concrets nous permettra d'enrichir ce prototype.</p>			
MOTS-CLES :	Système d'Information Géographique – modélisation conceptuelle – base de données spatiales – base de données temporelles – profil UML – temps réel.		
Laboratoire (s) de recherche :			
Laboratoire d'Informatique en Image et Systèmes d'Information et Unité de Recherche en Imagerie Satellitaire et ses Applications.			
Directeurs de thèse: R. LAURINI, Professeur (INSA Lyon), A. LBATH, Professeur (Université Joseph Fourier Grenoble) et			
Z. BELHADJ Professeur (SUP'COM Tunis)			
Président de jury : Professeur Mohamed Rached BOUSSEMA, ENIT (Tunisie), président,			
Composition du jury :			
Professeur Ziad BELHADJ, SUP'COM (Tunisie), directeur de thèse,			
Docteur Sami FAIZ, Université de Jendouba (Tunisie), rapporteur,			
Professeur Robert LAURINI, INSA de Lyon (France), Directeur de thèse,			
Professeur Ahmed LBATH, Université Joseph Fourier de Grenoble (France), co-directeur,			
Professeur Thérèse LIBOUREL, Université de Montpellier II (France), rapporteur,			
Docteur Mohamed Saber NACEUR, INSAT de Tunis (Tunisie), examinateur,			
Professeur Christine PARENT, Université de Lausanne (Suisse), examinatrice.			