



**Frédéric Fondement**

**Ingénieur de Recherche en Génie Logiciel**

8, impasse des Bagès  
70440 Servance  
03 84 63 80 19

[f.fondement@laposte.net](mailto:f.fondement@laposte.net)

Né le 27/12/1977 à Montbéliard (Doubs)  
Nationalité Française  
Permis B – véhicule

Version résumée sur [http://fondement.free.fr/CV\\_fondement.pdf](http://fondement.free.fr/CV_fondement.pdf)

## SITUATION ACTUELLE :

**Assistant d'enseignement et de recherche (ATER)** dans le domaine du **génie logiciel**, à l'Ecole Nationale Supérieure d'Ingénieurs Sud Alsace (**ENSISA**).

Mes recherches s'articulent autour de la modélisation des systèmes à dominante logicielle. Je m'intéresse tout particulièrement à la création de ces langages de modélisation (métamodélisation), à leur utilisation pratique (MDE), ainsi à la définition de leur syntaxe concrète, qu'elle soit graphique ou textuelle.

## DIPLOMES ET FORMATIONS

*Novembre 2007:* **Doctorat ès Sciences** dans le domaine du Génie Logiciel à l'Ecole Polytechnique Fédérale de Lausanne (EPFL) pour la thèse intitulée « Concrete syntax definition for modeling languages » ; <http://fondement.free.fr/lq/>

*Décembre 2001:* **Diplôme de Recherche Technologique** en Génie Electrique et Informatique Industrielle à l'Université de Haute Alsace (DRT GEII – diplôme universitaire bac +6), mention **très bien**

*Juillet 2000:* **Diplôme d'ingénieur** de l'Ecole Supérieure des Sciences Appliquées pour l'Ingénieur de Mulhouse (ESSAIM – désormais ENSISA), école habilitée en informatique et automatique, **2<sup>ème</sup> de promotion** ; <http://www.ensisa.uha.fr/>

*Juillet 1997:* **Diplôme Universitaire de Technologie** de l'Institut Universitaire de Technologie de Belfort en Génie Electrique et Informatique Industrielle (DUT GEII) option Electrotechnique ; <http://geii.iut-bm.univ-fcomte.fr/>

## COMPETENCES ET CONNAISSANCES

Informatique

Génie logiciel (**Ingénierie pilotée par les modèles, métamodélisation**, méthode Fondue, JUnit ...)

Architecture objet (**Java**, C++, UML, Rose, Objecteering,...)

Environnements de développement (**Eclipse**, JBuilder, **CVS**, ANT, NSIS, viewcvs, statcvs, ...)

Systèmes d'information (SGBDR comme MySQL, SQL, mapping objet-relationnel, JMI MDR)

Architectures n-tiers (RMI, Sockets, XML/DOM, SVG, HTML, Servlet, Apache, Tomcat, Resin)

Informatique et automatique industriels (C, assembleur, Matlab, grafcet, compilateurs **ANTLR**, multitâche)

Étude des besoins, architecture logicielle, tests, réalisation de documentations, formation, support

Automatique linéaire continue et discrète, Electrotechnique, Electronique, Mécanique des matériaux

**Anglais** (langue de travail, rédaction d'articles scientifiques, présentations)

## DIVERS

Titulaire du permis de conduire automobile (catégorie B) - Possède un véhicule

Pratique du piano

Pratique des randonnées pédestres, cyclistes et équestres

Journaux et Chapitres de Livres:

- [1] Baar, T.; Markovic, S.; Fondement, F.; Strohmeier, A.: *Definition and Correct Refinement of Operation Specifications*. To appear in B. Meyer, A. Schiper, J. Kohlas, editors, Dependable Systems, LNCS 4028, 2006, Pages 127 - 144.
- [2] Muller, P.-A.; Studer, P.; Fondement, F.; Bézivin, J.: *Platform Independent Web Application Modeling and Development with Netsilon*. Journal on Software and Systems Modeling ([SoSyM](#)), Vol. 4 Nr. 4, Best papers of the <<UML>>'03 conference, Springer, November 2005, Pages 424 - 442.

Conférences Internationales:

- [3] Muller, P.-A.; Fleurey, F.; Fondement, F.; Hassenforder, M. ; Schnekenburger, R.; Gérard, S.; Jézéquel, J.-M.: *Model-Driven Analysis and Synthesis of Concrete Syntax*. Accepted for publication to Proceedings of Model Driven Engineering Languages and Systems, 9th International Conference, MoDELS 2006, Genoa, Italy, October 2006. Lecture Notes in Computer Science, 2006. An improved version is submitted to the Journal on Software and Systems Modeling ([SoSyM](#)) after invitation.
- [4] Fondement, F.; Baar, T.: *Making Metamodels Aware of Concrete Syntax*, Proceedings of the European Conference on Model Driven Architecture - Foundations and Applications ([ECMDA-FA](#)), Nuremberg, Germany, November 2005. Lecture Notes in Computer Science, Volume 3748, October 2005, Pages 190 - 204.
- [5] Silaghi, R.; Fondement, F.; Strohmeier, A.: *Towards an MDA-Oriented UML Profile for Distribution*. Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference, [EDOC](#), Monterey, CA, USA, September 20-24, 2004. IEEE Computer Society, 2004.

Ateliers Internationaux avec Comité de Lecture:

- [6] Muller P.-A., Fleurey F.; Vojtisek D. ; Drey Z.; Pollet D.; Fondement F.; Studer P.; Jézéquel J.-M.: *On Executable Meta-Languages applied to Model Transformations*. Model Transformation In Practice ([MTIP](#)) workshop held in conjunction with MoDELS/UML 2005 conference, Montego Bay, Jamaica, October 2005.
- [7] Silaghi, R.; Fondement, F.; Strohmeier, A.: *"Weaving" MTL Model Transformations*. Proceedings of the 2nd International Workshop on Model Driven Architecture, Foundations and Applications, [MDAFA](#), Linköping University, Sweden, June 10-11, 2004. Lecture Notes in Computer Science, Volume 3599, August 2005, Pages 123 - 138.
- [8] Muller, P.-A.; Dumoulin, C.; Fondement, F.; Hassenforder, M.: *The TopModL Initiative*, 3rd Workshop in Software Model Engineering ([WISME @ UML 2004](#)), Proceedings of the UML Satellite Activities 2004, Lisbon, Portugal, Lecture Notes in Computer Science, Volume 3297, Feb 2005, Pages 242 - 245 .
- [9] Fondement, F.; Silaghi, R.: *Defining Model Driven Engineering Processes*, 3rd Workshop in Software Model Engineering ([WISME @ UML 2004](#)), UML 2004, Lisbon, Portugal, October 2003.
- [10] Ziadi, T.; Jézéquel, J.-M.; Fondement, F.: *Product line derivation with UML*. Proceedings Software Variability Management Workshop, University of Groningen Departement of Mathematics and Computing Science, February 2003.

Rapports Techniques:

- [11] Fondement, F.; Schnekenburger, R. ; Gérard, S. ; Muller, P.-A.: *Metamodel-Aware Textual Concrete Syntax Specification*, Technical Report LGL-REPORT-2006-005, Swiss Federal Institute of Technology in Lausanne, Switzerland, December 2006.

Mémoires et Rapports de Stage:

- [12] Fondement, F.: *Concrete syntax definition for modeling languages*. PhD. Thesis 3927, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2007.
- [13] Fondement, F.: *Création d'un Langage d'Action UML pour un Logiciel MDA*. Mémoire de DRT, ESSAIM en partenariat avec ObjeXion Software S.A., 2001.
- [14] Fondement, F.: *Développement de Composants pour une Machine Virtuelle UML*. Mémoire de stage ingénieur, ESSAIM en partenariat avec ObjeXion Software S.A., 2000.

[15] Fondement, F.: *Développement de Gestionnaires de PARTS sous Navigator*. Mémoire de stage DUT, IUT GEII Belfort-Montbéliard en partenariat avec Peugeot S.A., 1997.

Participation aux comités de programme des ateliers [Ingénierie Dirigée par les Modèles \(IDM\) 2006](#) et [IDM 2005](#).

Relecteur pour :

- Livre: [Model-Driven Software Development](#)
- Journaux: [HSoftware and Systems Modeling\(SoSyM\)](#), [Science of Computer Programming](#), [WWW Journal](#)
- Conférences: [Models/UML 2005](#), [VL-HCC 2005](#), UML 2003, [UML 2001](#), [UML 2000](#).

## TACHES D'ENSEIGNEMENT

A l'EPFL :

Suivis de 14 projets de Génie Logiciel (partie conception - 71 étudiants) au niveau bachelor sur 2 années pour l'apprentissage et l'application de la méthode de développement des systèmes réactifs Fondue (10 crédits sur 60).

Participation à la correction des épreuves du cours magistral.

Suivis de 2 projets de Semestre (312 heures de travail par étudiant – 12 crédits sur 60) :

[1] Hong, F.: *Provide Behaviour to XML/SVG*. 2005.

[2] Helg, F.; Rohrer, F.: *Synchronization between display objects and representation templates in graphical language construction*. 2006.

A l'ENSISA :

Préparation et suivi des travaux pratiques d'ingénierie des modèles et d'ingénierie dirigée par les modèles (niveau M1) – 120h TP.

## EXPERIENCES PROFESSIONNELLES

### PREPARATION DU DOCTORAT

**Assistant d'Enseignement et de Recherches - Ecole Polytechnique Fédérale de Lausanne (EPFL)**

Lausanne (Suisse/Vaud), IC/ISIM/LGL (Laboratoire de Génie Logiciel),

Egalement à l'ENSISA Mulhouse (68), MIAM/LSI (Logiciel et Systèmes Intelligents), après accord signé en février 2006.

Supervisé par Dr. Thomas Baar (EPFL), co-encadré par Pierre-Alain Muller (ENSISA).

#### **Octobre 2003 à Février 2005 :**

**Enseignement du Génie Logiciel.** Il s'agit du suivi de projets d'étudiants dans le but de leur faire expérimenter de façon pratique la méthode de développement logicielle Fondue, alliant la notation UML à la méthode Fusion (voir <http://fondue.epfl.ch>). Les étudiants, par groupe de 5, doivent créer un logiciel simple en suivant toutes les phases de la méthode : élaboration des besoins, analyse, développement et implémentation. Mon travail a consisté à les suivre de manière hebdomadaire pour la phase d'analyse pour contrôler le travail réalisé et répondre aux questions. Parfois, quelques compléments / explications de cours sont nécessaires. S'ensuit la correction des rapports (environ 50 pages chacun), avec une vérification de la cohérence de l'analyse, ainsi que son adéquation avec les phases précédentes.  
*Environnement* : Fondue, UML, OCL

#### **Octobre 2003 à Novembre 2005 :**

**Développement d'un atelier pour la méthode fondue.** La méthode Fondue allie la notation UML à la méthode de développement logicielle Fusion (voir <http://fondue.epfl.ch>). Cependant, certaines différences existent entre la notation UML standard (voir <http://www.omg.org/uml>) et la notation utilisée en pratique par Fondue. Ce projet consiste à implémenter un environnement de modélisation pour Fondue en étendant l'atelier Borland Together Architect pour les aspects d'édition, et la métamodélisation conjuguée

à un dépositaire de modèle pour les aspects d'utilisation externe. L'architecture centrée dépositaire a d'ailleurs permis une collaboration simplifiée avec d'autres équipes de recherche / développements, et la collaboration entre plusieurs outils.

- Conception du métamodèle de la notation utilisée par Fondue
- Implémentation de l'export de modèle tel que dessiné dans Together
- Conception de transformation de modèle pour la vérification de la validité du modèle
- Participation à l'automatisation du déploiement (scripts de compilation, auto-installateur)
- Mise en place d'outils d'aide au développement (viewcvs, statcvs)
- Mise en place d'un site internet type WIKI pour la diffusion de l'outil (application Netsilon ciblant PHP/MySQL – voir <http://fondue.epfl.ch>)
- Maintenance du serveur accueillant le site internet
- Réalisation et présentation de tutoriel pour l'utilisation de l'outil.

*Environnement* : Fondue, Together, Eclipse, Java, Rose, métamodélisation MOF, JMI MDR, transformation de modèles MTL, CVS, ANT, NSIS, apache, PHP, MySQL, viewcvs, statcvs.

### **Depuis Octobre 2003:**

**Recherche sur l'ingénierie pilotée par les modèles.** L'ingénierie pilotée par les modèles est un sujet majeur de recherche en génie logiciel. Elle propose d'utiliser des raffinements de modèles et de la génération de code pour la construction de logiciels. De nombreuses recherches tendent à rendre cette technique utilisable par le monde industriel. Ma problématique est de résoudre la définition de syntaxes concrètes, soit textuelles soit graphique, pour obtenir des langages de modélisation (voir <http://fondement.free.fr/lgl/docs/RP.pdf>). La définition de syntaxes graphiques fut réalisée grâce à des patrons écrits en SVG étendus par des éléments de dynamisme réalisés avec DOM, et des instructions KerMeta pour la manipulation de modèle en dépositaire. La définition de syntaxes textuelles fut résolue par un nouveau langage de spécification de grammaires piloté par le métamodèle du langage de modélisation. Des recherches sur la réutilisation de ces définitions et des transformations de modèles pour les appliquer à des modèles différents ont également été réalisées.

- Rédaction de publications scientifiques dans un contexte international
- Evaluation d'outils et de structures de départ
- Implémentations en langage Java et MTL
- Pilotage et coordination de projets d'étudiants pour l'implémentation des composants nécessaires
- Collaborations scientifiques avec le Commissariat à l'Energie Atomique (CEA), l'Institut National de Recherche en Informatique (INRIA) et l'université de Mulhouse
- Mise en place d'outils d'aide au développement (CVSNT, viewcvs, statcvs)

*Environnement* : CVS, CVSNT, Eclipse, Rose, dopidom, DOM, XML, SVG, Batik, MTL, viewcvs, statcvs, KerMeta, Word, PowerPoint, FrameMaker, LATEX.

## **RECHERCHE ET DEVELOPPEMENT A L'INRIA**

### **Ingénieur Expert de Recherche et Développements – Institut National de Recherche en Informatique et Automatique (INRIA)**

Rennes (35), IRISA /Triskell.

Supervisé par Prof. Jean-Marc Jézéquel.

### **Décembre 2002 à Septembre 2003 :**

**Etude et développement d'un langage de transformation de modèles.** Il s'agit de développer un nouveau langage permettant la transformation de modèles, quel que soit leur nombre et quel que soit leur métamodèle. Cette réalisation doit cadrer autant que possible avec la future norme MOF « Query/View/Transformation » de l'OMG (<http://www.omg.org/>). La transformation de modèle est indispensable à la méthodologie MDA (<http://www.omg.org/mda/>), et aucun langage standard (indépendant de l'outil dépositaire de modèle – soit une transformation indépendante de la plate-forme PIT) n'existe à ce jour. Nous avons opté pour une approche la plus indépendante possible des normes puisque ces dernières (Infrastructure UML 2.0, MOF 2.0, MOF QVT 2.0, JMI 2.0,...) sont en cours de finalisation ; il est donc nécessaire de savoir s'adapter rapidement (<http://modelware.inria.fr>).

- Conception d'une API d'interrogation des dépositaires de modèle compatible IDL pour rester indépendant du langage d'implémentation de celui-ci
- Développement d'un driver pour JMI MDR

- Mise en place d'une approche de Vue/Adaptateur pour métamodèle de façon à rester indépendant des métamodèles des modèles manipulés.
- Conception d'une syntaxe textuelle de haut niveau
- Conception d'un arbre abstrait de haut niveau
- Conception d'un arbre abstrait de bas niveau
- Conception d'une syntaxe textuelle de bas niveau (pour de besoins d'utilisation immédiate)
- Participation à la réponse RFP MOF QVT

*Environnement* : IBM Eclipse, Softeam Objecteering, Rational Rose, plug-in Eclipse Objecteering, module de travail collaboratif Objecteering, CVS, Log4J

### **Décembre 2002 à Septembre 2003 :**

**Réalisation de livrables.** Le contexte est le projet de recherche européen ITEA CAFE

(<http://www.esi.es/Cafe>) qui traite des lignes de produits. Il s'agit de présenter sous forme de transparents largement annotés du travail de l'INRIA au sein de ce projet. Les tâches de l'INRIA portaient spécialement sur :

- Les plate-formes et l'approche MDA (<http://www.omg.org/mda>), et notamment les contraintes entre variants.
- Développement des aspects et notamment la modélisation des aspects dynamiques dans l'approche ligne de produits
- La dérivation de produits
- La modélisation des tests
- La validation

## **PREPARATION DU DIPLOME DE RECHERCHE TECHNOLOGIQUE**

### **Ingénieur Logiciel – Objexion Software / Université de Haute Alsace (UHA)**

Vieux-Thann (68).

Supervisé par Dr. Pierre-Alain Muller (Objexion/UHA), Philippe Studer (Objexion, désormais UHA), Prof. Michel Hassenforder (UHA).

### **Juin 2000 à Octobre 2000 :**

« **Développement d'un langage d'action pour UML** ». Le travail portait sur un nouveau logiciel de création et de génération d'application internet complète, quel que soit le langage de programmation et la base de données pour les informations persistantes. Le modèle métier de l'application était décrit par un diagramme de classes UML. Cependant, UML ne permet pas de décrire un corps de méthode. Mon travail a donc consisté à concevoir un langage de programmation décrivant les actions, comme celles réalisées par les méthodes, capable de naviguer dans tout le modèle ainsi que de la modifier en tout point, donc autosuffisant, et demeurant traduisible en un langage existant (ceux qui furent choisis étaient PHP et Java). J'ai choisi de me baser sur OCL pour la partie navigation dans le modèle, les types de base et les opérations prédéfinies, et sur java pour la partie présentation (action, blocs, déclarations, boucles...). Ce langage s'est appelé Xion. Le document de recherche correspondant est disponible sur <http://fondement.free.fr/drt/rapport2.pdf>.

- Réalisation de la grammaire de Xion à partir de la grammaire précédemment réalisée d'OCL et celle disponible sur Java.
- Adaptation des types et des opérations prédéfinies d'OCL ; étude et mise en place d'un système efficace et évolutif de stockage des types et opérations prédéfinis, éventuellement construits au vol, notamment pour les collections.
- Etude et mise en place d'un système de stockage des classes, attributs, liens, opérations et méthodes du modèle métier plus exploitable et juste à coup sûr du modèle métier que le métamodèle UML.
- Réalisation des analyses lexicales syntaxiques et sémantiques ainsi que du contrôle des types (à partir du modèle métier et des types prédéfinis du langage), générant un arbre concret comportant toutes les informations nécessaires à la traduction vers le bon langage cible.
- Description complète de l'arbre abstrait et fourniture d'un visiteur, la partie de génération spécifique au langage cible n'étant pas de mon ressort.
- Adaptation du compilateur ANTLR afin qu'il génère des messages d'erreur dans le langage de l'utilisateur, le logiciel étant multilingue, qu'il retienne les lignes erronées ainsi que

les types de retour de chaque nœud, et, le cas échéant, le type statique ou le champ (attribut, lien ou méthode) mis en jeu.

- Documentation du langage en langue anglaise au format HTML (<http://fondement.free.fr/objx/netsilon/xion/index.html>).
- Documentation des types prédéfinis et opérations prédéfinies par génération à partir d'un diagramme de classes correspondant documenté ; la génération partait de Java, puis Javadoc et Shell script Unix (<http://fondement.free.fr/objx/netsilon/xion/predefined/index.html>).

*Environnement* : Borland JBuilder, Java 2, grammaire Java, norme UML (métamodèle, OCL), ANTLR 2.7.0, CVS, Javadoc, Microsoft FrontPage, Shell Unix, Windows 98, HTML, Microsoft FrontPage, Java Help.

### **Octobre 2000 à Novembre 2000 :**

**Développement d'un éditeur de code pour Xion.** Après avoir conçu ce langage objet, il était nécessaire de réaliser un éditeur de code simplifiant d'une part la tâche des développeurs, et d'autre part l'apprentissage du langage. Plus de détails sont donnés au chapitre 4 du document présent sur

<http://fondement.free.fr/drt/rapport2.pdf>.

Détermination des spécifications : fonctionnalités standards (couper, copier, coller, sélectionner tout, défaire, refaire, saut automatique de tabulations, choix des options de couleur, de tabulation, internationalisation des commandes...), coloration syntaxique (différentiation visuelle entre texte, mots-clés, valeurs littérales et commentaire), complétion sémantique à la manière de Visual Studio et JBuilder, et intégration d'un panel montrant les erreurs et surlignant les lignes incriminées.

- Intégration d'un éditeur de texte « open source » après avoir déterminé qu'il était inefficace d'utiliser les composants standards Java ; réalisation un analyseur rapide pour déterminer le type de chacun des mots du code entré par le programmeur.
- Utilisation de contrôle des types de la chaîne de compilation après mise en forme du texte afin de déterminer le type de l'expression en cours d'édition ; ceci permet alors de proposer une liste des opérations disponible sur l'expression en cours.
- Codage des autres spécifications déterminées.
- Documentation utilisateur en langue anglaise (<http://fondement.free.fr/objx/netsilon/ug/xioneditor.html>).

*Environnement* : Borland JBuilder, Java 2, Swing, Windows 98, NT, XP, Linux, MacOS X, HTML, Microsoft FrontPage, Java Help

### **Décembre 2000 et Janvier 2001 :**

**Création de l'interface du modèle de navigation de Netsilon.** Netsilon est un logiciel (basé sur Argo UML) de création et de génération d'application internet complète, quel que soit le langage de programmation et la base de données pour les informations persistantes. La description de l'agencement était faite par une extension au métamodèle UML. Il était donc impératif de fournir une interface utilisateur permettant d'entrer les informations de navigation.

- Réalisation de la représentation du graphe d'agencement des pages internet
- Réalisation d'une interface spécifique de description et d'édition de chacune des parties de ce graphe (pages internet et chacune des sortes d'élément dynamique).
- Intégration au logiciel.
- Documentation utilisateur en langue anglaise (<http://fondement.free.fr/objx/netsilon/ug/navigationdiagram.html>).

*Environnement* : Borland JBuilder, Java, Swing, ArgoUML, CVS, Windows 98, NT, XP, Linux, MacOS X, HTML, Microsoft FrontPage, Java Help

### **Janvier 2001 à Juin 2001 :**

**Optimisation SQL.** Le travail portait sur un logiciel création et de génération d'application internet complète, quel que soit le langage de programmation et la base de données pour les informations persistantes. Le modèle métier de l'application était décrit par un diagramme de classes UML et les actions comme celles décrites par les méthodes des classes, étaient décrite en langage Xion. Chaque demande de valeur de champ sur un objet donnait lieu en l'état à un accès à la base de données, ce qui est inadmissible pour des raisons de performances, notamment sur des demandes de champs successives ou des demandes de champs sur une collection d'objet. Il m'a donc été donné de traduire du mieux possible des expressions de requête complexes en Xion en code SQL. Il est à noter que les objets pouvaient indifféremment être stockés en base Oracle, MySQL ou PostgreSQL. J'ai déterminé qu'il était idéal de modifier sur l'arbre concret qui référence tous les types et champs utilisés du modèle métier : il

suffisait dès lors d'écrire un visiteur d'arbre modifiant ce dernier. Une meilleure description se trouve au chapitre 3 du document disponible sur <http://fondement.free.fr/drt/rapport2.pdf>.

- Description des instructions SQL en un arbre abstrait avec différenciation des bases Oracle, MySQL et PostgreSQL.
- Détermination de la politique de transformation de l'arbre concret représentatif du code entré par l'utilisateur.
- Récupération des valeurs d'un champ d'un objet ou d'une sélection en base de données relationnelle à partir du mapping objet-relationnel.
- Codage des « patterns » de transformation, chaque instruction Xion possible (appel d'attribut, appel de lien, appel à certaines opérations prédéfinies) étant traduite par une modification apportée à l'arbre concret, ces transformations pouvant être différentes suivant le type de base.
- Codage du visiteur.

*Environnement* : Borland JBuilder, SQL'92, SQL Oracle MySQL et PostgreSQL, CVS, Windows 98, NT, Linux

### **Juin 2001 à Août 2001 : Ingénieur logiciel – ObjeXion Software**

**Découpage de projets Netsilon en modules.** Le travail portait sur un logiciel création et de génération d'application internet complète, quel que soit le langage de programmation et la base de données pour les informations persistantes. Dans un souci de réutilisation des composants, il m'a été demandé de concevoir et réaliser un système permettant de découper un projet d'application internet en sous-parties réutilisables, à la manière d'une bibliothèque statique. Toute la difficulté reposait en ceci que le projet est décrit non pas par un ensemble de fichiers, mais par un ensemble d'objets intimement liés.

- Conception en concertation avec les membres de l'équipe, du principe de modularisation.
- Import d'un sous-projet.
- Contrôle des liaisons entre modules et projet.
- Persistance des modifications d'import et de liaison entre modules et projet.
- Modification de la génération.
- Réalisation des interfaces utilisateur pour l'import et la détermination des liaisons entre modules et projet.
- Documentation utilisateur en langue anglaise  
(<http://fondement.free.fr/objx/netsilon/ug/modules.html>)

*Environnement* : Borland JBuilder, Java, Swing, CVS, HTML, Microsoft FrontPage, Java Help

### **Février 2001 et Avril 2002 :**

**Editeur des objets métier Netsilon.** Netsilon est un logiciel de création et de génération d'application internet complète, quel que soit le langage de programmation et la base de données pour les informations persistantes. Le modèle métier de l'application était décrit par un diagramme de classes UML. Une fois l'application générée, les objets décrivant les informations dynamiques sont stockés en base de donnée relationnelle, avec impossibilité pour le concepteur de les créer, les détruire ou les modifier directement, par exemple par des interfaces telles que PHPMyAdmin ou PHPOracleAdmin. Or, il est possible grâce à ce logiciel, en partant des informations du modèle métier, de générer automatiquement au sein de l'application internet, une partie dédiée à l'administration des données sous forme d'objet. Une meilleure description se trouve au chapitre 5 du document disponible sur <http://fondement.free.fr/drt/rapport2.pdf>.

- Génération sous forme de modèle de l'application internet d'administration des données dynamique sous forme d'objet à partir du modèle métier : présentation des types disponibles, des diagrammes de classe, de tous les objets d'une classe, éditeur pour un objet en particulier (attributs, liens, opérations), présentation des constructeurs, création et destruction d'un objet.
- Documentation utilisateur en langue anglaise  
(<http://fondement.free.fr/objx/netsilon/ug/contentmanagement.html>).

*Environnement* : Borland JBuilder, Java, HTML 3.2, Microsoft FrontPage, Java Help

### **Mai et Juin 2002 : Ingénieur logiciel – ObjeXion Software**

**Etude d'un système collaboratif pour Netsilon.** Netsilon est un logiciel de création et de génération d'application internet complète, quel que soit le langage de programmation et la base de données pour les informations persistantes. Le problème de base est que ce logiciel est mono-utilisateur : il est donc impossible de développer une application internet à plusieurs. Mon but était de trouver une solution de travail collaboratif.

- Etude d'une architecture réseau permettant l'échange de parties de projet par RMI.

- Etude d'un système d'export de partie de projet.
- Etude d'un système d'import de partie de projet.
- Etude d'un système de stockage et de mise à disposition sécurisée d'un projet.
- Etude d'un système d'administration du service par interface HTML à distance sécurisée.

*Environnement* : Borland JBuilder, Java, RMI, Windows 98, NT, XP, Linux, socks, SSL, Jetty, servlet, NAT

### **Divers travaux de développement**

**Transfert par FTP sélectif d'application internet.** Le travail portait sur un logiciel de développement d'application internet. Il était nécessaire de transférer le résultat de la génération par ftp des fichiers modifiés vers les sites de déploiement.

- Recherche d'une librairie de transfert de fichiers par ftp pour Java.
- Adaptation de la librairie trouvée au problème.
- Interface utilisateur.

*Environnement* : Borland JBuilder, Java, Swing, ftp.

**Sites Isostar et Solinest sous Netsilon.** Netsilon est un logiciel de création et de génération d'application internet. Il m'a été donné de l'utiliser pour réaliser un site dynamique en collaboration avec l'entreprise Boomerang Pharmaceutical Communications, chargée de fournir l'interface visuelle.

- Détermination des spécifications en collaboration avec Boomerang.
- Réalisation de la motorisation du site.
- Intégration de la partie interface.

*Environnement* : Netsilon, HTML, Apache, PHP, MySQL

**Déverminage logiciel.** Il s'agissait de déverminer un logiciel de génération d'application internet, par test et par remontée d'information de la part des utilisateurs. Il s'agissait également d'y modifier ou d'y intégrer de nouvelles fonctions et interfaces suivant les suggestions, quelle qu'elle soit.

*Environnement* : Borland JBuilder, Java, Argo UML, Swing, Apache, Tomcat, Resin, PHP, Servlet, Windows 98, NT, XP, Linux, MacOS X.

### **Septembre 2000 à Juin 2002 :**

**Présentation, formation et support.** Il s'agissait de présenter ou de dispenser une formation aux logiciels de l'entreprise (Model Prototyper et Netsilon), ainsi que de répondre aux questions des utilisateurs par mail ou téléphone, voire sur site. Il était parfois nécessaire des les aiguiller sur des solutions. Dans ce cadre, il m'a été donné de réaliser le guide de l'utilisateur du logiciel en langue anglaise (<http://fondement.free.fr/objx/netsilon/ug/index.html>). Une autre partie a consisté à la relecture et l'évaluation de publications scientifiques en anglais sur OCL.

## **PREPARATION DU DIPLOME D'INGENIEUR**

### **Stagiaire Ingénieur – ObjeXion Software / Ecole Supérieure des Science Appliquées pour l'Ingénieur de Mulhouse (ESSAIM – UHA – désormais ENSISA)**

Vieux-Thann (68).

Supervisé par Dr. Pierre-Alain Muller (Objexion/UHA), Philippe Studer (Objexion, désormais UHA), Prof. Michel Hassenforder (UHA).

### **28 février au 2 Septembre 2000 :**

« **Développement d'un interpréteur OCL pour une machine virtuelle UML** ». Le travail portait sur un logiciel de prototypage de diagrammes de classes UML, c'est à dire un logiciel capable d'instancier, modifier et détruire des objets persistants sans besoin de programmation, grâce à une interface graphique. Il s'agissait d'un projet en J++. Le mémoire de stage correspondant est disponible sur <http://fondement.free.fr/drt/rapport1.pdf>.

- Autoformation à OCL et étude précise des diagrammes de classe ; résolution des indéterminations de la norme définissant OCL.
- Codage des différents types et différentes opérations prédéfinies du langage OCL ; extraction des données du modèle UML et des données des objets stockés.
- Réalisation des analyses lexicales, syntaxiques et sémantiques ainsi que du contrôle des types (à partir des types prédéfinis et du diagramme de classes considéré), et d'un visiteur d'arbre mettant en jeu les types et opérations précédemment codées à l'aide d'un contexte.

- Intégration au logiciel existant : vérification d'une contrainte sur un objet en particulier par une interface graphique ; vérification d'une contrainte sur l'ensemble des objets existants par une seconde interface graphique ; évaluation d'un corps de méthode décrite par une expression OCL.
- Documentation utilisateur en langue anglaise (<http://fondement.free.fr/objx/modelprototyper/Model%20Prototyper.chm>, Reference Manual / FacSimile / Using OCL et Reference Manual / OCL Specification ).

*Environnement* : J++, Microsoft Visual Studio, Microsoft Visual SourceSafe, compilateur de compilateur ANTLR 2.7.0, Windows 98 et NT, Rational Rose, Objexion Model Prototyper et FacSimile, HTML, Microsoft FrontPage.

---

## DIVERS

### **9 Août au 10 Septembre 1999 : Technicien informatique – Sécurité Sociale CPAM**

Belfort (90), SOMI

**Installation** et paramétrage d'une quarantaine d'ordinateurs en vue du passage à l'an 2000.

- Déballage des nouvelles machines puis installation par clonage du disque dur.
- Mise en station : mise au rebut de l'ancien poste et paramétrage des données spécifiques au poste installé.
- Installation poste par poste de logiciels d'installation et de contrôle à distance.

*Environnement* : Windows 98.

### **14 Avril au 21 Juin 1997 : Stagiaire technicien IUT GEII Belfort – PSA Peugeot-Citroën**

Sochaux (25), service DETA/MSD/DCL/SUP

« **Développement de gestionnaires de parts sous Navigator** » : Développement de programmes en Shell script Unix permettant la gestion et les échanges des fichiers de graphisme 3D utilisés pour la conception automobile.

- Développement de Shell permettant de lier des parts entre elles, de renommer des parts, de changer de leur propriétaire dans un but de faire collaborer différents spécialistes, de les proposer au service de synthèse, de les sauvegarder, de les déverrouiller en cas de sortie prématurée du programme, et de les envoyer à différents intervenants.
- Intégration des programmes dans l'environnement graphique Navigator et le menu général de chaque métier

*Environnement* : Bourne Shell Unix, stations de travail HP (HT-UX) et Sun (Sun OS).