Projet ELO

Redacteurs: Christophe Dony, Bernard Carré, Roland Ducournau,
Michel Habib, Marianne Huchard, Amedeo Napoli, Mourad Oussalah et Jean-Claude Royer

1 Description du projet

La sémantique et les techniques d'implémentation de langages comme *Smalltalk*, *Eiffel* ou *CLOS* sont relativement stabilisées - certains de ces langages sont même en voie de normalisation - et permettent leur mise en œuvre quasi industrielle.
Cependant le succès des langages à objets a contraint les concepteurs à en figer trop tôt certains aspects. Certains résultats de recherches n'ont pas (encore) été intégrés dans les langages les plus utilisés industriellement, certains sujets ont été peu ou incomplètement traités, enfin certaines applications font apparaître de nouveaux problèmes nécessitant des modifications dans les modèles utilisés.

Le projet ELO regroupe des équipes de recherches académiques mais aussi industrielles travaillant sur l'utilisation, la compréhension, la sémantique ainsi que diverses évolutions des modèles et des langages de programmation à objets. Les sujets traités sont répartis en quatre grand thèmes:
- la sémantique de l'héritage et la manipulation des hiérarchies,
- la sémantique et l'évolution des différents modèles d'objets (avec ou sans classes par exemple) et la construction réflexive des langages,
- la représentation de connaissances complexes et évolutives dans les langages de programmation à objets
- l'analyse et la conception formelles de programmes à objets.

1.1 Responsable

Christophe DONY
LIRMM (Laboratoire d'Informatique, de Robotique et Micro-electronique de Montpellier)
161 rue Ada, 34392 Montpellier Cedex 5, France
email: dony@lirmm.fr;
tel: (33) 67 41 85 33;
fax: (33) 67 41 85 00.

1.2 Equipes

- CNET - Centre National d'Etude des Télécommunications (Issy-les-Moulineaux):
  M. Dao (Ingénieur), M. Shteto (doctorante).

- EMN - Ecole des Mines de Nantes:
  P. Cointe (professeur), P. Mulet (doctorant EMN).

- IRIN (institut de recherche en informatique de Nantes), Equipe Génie logiciel, Méthodes et Spécifications Formelles:
  J.C. Royer (mdc), P. André (ater), P. Poizat (thésard)

- LIFL - Laboratoire d'Informatique Fondamentale de Lille:
  B. Carré (mdc), L. Dekker, G. Vanwormhoudt (doctorant).

- LIRMM - Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier:
  C. Dony (mdc), R. Ducournau (Professeur), M. Habib (professeur), M. Huchard (mdc), T. Libourel (mdc), Lhouari Nourine (Mdc), J. Quinqueton (DR INRIA), P. Vismara (PostDoc), D. Bardou (doctorant), S. Pelenc (Doctorant).

- LGI2P [1] (Nîmes, Alès):
  Mourad Oussalah (professeur), Martine Magnan (mdc), Veronique Puig, Christelle Urtado.

- CRIN - Centre de Recherche en Informatique de Nancy: A. Napoli (CR CNRS)

1.3 Thème et objectif scientifiques

Ce projet a pour thème général l'évolution de la programmation par objets (langages, modèles et techniques). L'objectif scientifique est l'étude de certains aspects mal définis ou nouveaux des langages et modèles à objets. Cette étude doit conduire à la réalisation de nouveaux algorithmes ou à l'implantation de nouveau traits dans des langages existants voire si nécessaire à la définition de nouveaux langages. Plus précisément, les thèmes d'étude sont les suivants.

- La sémantique et l'algorithmique de l'héritage: multiplicité, exceptions, codage, réorganisation, classification (section 2.1).

- La compréhension des langages à objets sans classes et plus particulièrement du mécanisme de délégation, leur intérêt dans l'étude de la réflexivité et ses connections avec la représentation de connaissances. (section 2.2).

- La représentation de connaissances complexes et les outils qui lui sont nécessaires. Programmer dans un langage à objet c'est bien sûr représenter des connaissances, les techniques en la matière sont en pleine évolution: représentation de points de vues, de versions, d'objets composites, de contraintes (section 2.3).

- La spécification formelle de programmes à objets. Ces dix dernières années ont vu émerger de nombreuses méthodes d'analyse et de conception dites à objets. L'intégration de spécifications formelles à ces méthodes est un enjeu important en génie logiciel.

Un objectif important du projet est également de faire se rencontrer et collaborer différentes équipes travaillant sur des aspects très voisins des modèles à objets mais issues respectivement des sphères "langages de programmation" et "représentation des connaissances".

1.4 Exemples de résultats obtenus

En ce qui concerne la sémantique et l'algorithmique de l'héritage, les résultats sur la linéarisation (algorithme pour l'héritage multiple avec exception produisant une linéarisation monotone) correspondent à un sous-projet antérieur aux autres et sont les plus aboutis. Ils ont été mis en oeuvre dans le langage Y3 et sont utilisés dans des applications d'envergure menées en collaboration avec des industriels (cf. par exemple sections 3.2 et 3.1).

Pour le thème langages à prototypes et réflexion, les travaux publiés ont mis à jour à la fois l'intérêt et les problèmes que posent le mécanisme de délégation et permettent d'avancer dans la compréhension des langages. Une plate-forme de simulation des langages sans classe et/ou avec délégation (*Prototalk*) est à la disposition de la communauté. Pour ce qui est des langages à objets réflexifs, l'utilisation de langage sans classes a permis des avancées dans la compréhension de l'approche avec méta-objets. Un langage à objet sans classes et réflexif (*Moostrap*) a été développé.

1. Le LGI2P est un laboratoire commun de l'EERIE (Ecole pour les Etudes et la Recherche en Informatique et Electronique de Nîmes) et de l'EMA (Ecole de Mines d'Alès)

En représentation des connaissances, les modèles de points de vues et d'objets composites proposés et développés par les équipes du groupe sont utilisés dans des applications industrielles (voir par exemple la section 3.3).

Enfin, les formalismes de *classes formelles* et de *types abstraits graphiques*, destinés à la spécification formelle de système à objets ont, été développés. Ils sont soutenus par des outils: éditeurs de TAG(type abstrait graphique) et de CF(classes formelles), traducteur de CF en Smalltalk et en Eiffel et plate-forme générale permettant le support de ces concepts.

1.5 Animation scientifique

Outre les réunions de pôle des journées du GDR évoquées précédemment. Le groupe ELO participe très activement à l'organisation pratique et scientifique de la conférence LMO (Langages et Modèles à Objets) tenue à Grenoble en Octobre 1994 et à Nancy en octobre 1995[LMO95] et qui traite directement des thèmes de recherche du groupe et plus généralement du pôle. Il y avait à Nancy plus de 80 inscrits et plus d'une centaine d'auditeurs.

Il a également participé, sous la direction d'Amédéo Napoli et de Jean-François Perrot, à la réalisation d'un numéro spécial sur les "systèmes à objets" de la revue Technique et Science Informatique à paraître.

2 Description des sous-projets

La description des différents sous-projets à été rédigée en collaboration avec leurs responsables: Bernard Carré, Roland Ducournau, Michel Habib, Amedeo Napoli, Mourad Oussalah et Jean-Claude Royer.

2.1 Sous projet: Sémantique et algorithmique de l'héritage

Ce sous-projet est plus important que les autres de par le nombre de travaux distincts qu'il regroupe. Ces travaux portent directement ou ont des applications directes sur les hiérarchies ou les mécanismes d'héritage en programmation par objets. Les relations entre ces quatre projets sont étroites mais il est plus simple de les présenter séparément.

2.1.1 Héritage multiple

Participants: Michel Habib (LIRMM), Roland Ducournau (LIRMM), Marianne Huchard (LIRMM), Marie-Laure Mugnier (LIRMM), Amédéo Napoli (CRIN-CNRS).

Thèmes et objectifs: Le thème de ces recherche, entreprises depuis 1987 par M. Habib (LIRMM) et R. Ducournau (Sema Group, puis LIRMM), est la compréhension et la modélisation formelle de l'héritage multiple et des algorithmes qui permettent de l'utiliser.

Etat d'avancement: Les travaux réalisés font l'objet d'un article de synthèse parut dans TSI [DHH95]. Les résultats présentés sont:

- une nouvelle modélisation de l'héritage, avec la distinction entre *héritage de nom* et *héritage de valeur* [DHH95] qui permet de distinguer les conflits syntaxiques de nom des conflits "sémantiques" de valeur.
- la construction d'une linéarisation *monotone* [DHH+94], dont la propriété de monotonie, annoncée dans nos travaux précédents (OOPSLA'92), assure que la linéarisation d'une sous-classe "contient" la linéarisation de ses super-classes, ce qui est particulièrement utile pour la combinaison des méthodes. Cette nouvelle linéarisation étend les linéarisations précédemment étudiées de LOOPS et de CLOS dans la mesure où elle leur est identique quand ces deux linéarisations le sont.

Les techniques développées ne sont pas sans applications: les bonnes propriétés de cette nouvelle linéarisation en font un bon candidat pour remplacer celle de CLOS et l'une des perspectives de notre travail consiste à étudier des applications de CLOS existantes, pour voir l'impact qu'aurait l'adoption par ce langage de cette nouvelle linéarisation. Nous sommes actuellement en contact avec Chris Riesbeck (NorthWestern University, Chicago) sur ce point. De plus, la technique d'héritage cumulatif non-monotone décrite dans [DHH95] est effectivement utilisée dans le projet *RESYN* (cf. section 3.1).

2.1.2 Classification

Participants: B. Carré (LIFL), L. Dekker (LIFL), R. Ducournau (LIRMM), A. Napoli (CRIN CNRS – Inria Lorraine), J. Quinqueton (LIRMM), P. Vismara(LIRMM).

Les travaux effectués dans le cadre du thème “Classification et objets” essaient de mettre en valeur le rôle de la classification dans le contexte d'un environnement à base d'objets, en tant que processus d'aide à la conception de hiérarchies et en tant que processus de raisonnement. Ils ont également pour but de mettre en valeur une sémantique dénotationelle et/ou opérationnelle des hiérarchies d'objets et d'étudier la combinaison programmation – raisonnement par classification dans l'univers des systèmes de représentation de connaissances à base d'objets — notés systèmes de RCO dans la suite — étude qui correspond en quelque sorte à une version moderne de la controverse déclaratif – procédural.

La difficulté spécifique de la programmation dans les langages à objets tient à la construction de bonnes hiérarchies d'héritage. À ce titre, étudier la sémantique de représentation des objets et comparer langages de représentation et langages de programmation, en particulier dans le but de leur intégration, est nécessaire [CDE+95]. Le cadre des *systèmes classificatoires* est une première proposition en ce sens [Duc95]. De ce point de vue, il est intéressant d'apparenter la classification à une *inférence de type*, et il est envisagé des études cherchant à évaluer l'intérêt de cette vision de la classification et son intégration dans des langages de programmation plus traditionnels.

D'autres directions de recherches sont également explorées, qui concernent notamment les rapports entre les systèmes de RCO et les systèmes logiques à base d'objets comme les logiques terminologiques (ou logiques de descriptions). Les techniques de programmation et de représentation évoluent en empruntant l'une à l'autre. L'étude de ces interactions a été un des thèmes majeurs de l'atelier de travail *Object-based Representation Systems* organisé lors de la conférence mondiale IJCAI'93 à Chambéry, en août 1993 [DN94]. Si les systèmes de RCO intègrent de façon naturelle la raisonnement par classification, dont la technologie a été essentiellement développée au sein des logiques terminologiques, la mise en œuvre d'un environnement minimal commun aux systèmes de RCO et aux logiques terminologiques est toujours d'actualité [Nap94] [Nap95].

Pour l'heure, de nombreuses directions de recherches restent ouvertes, et notamment : (1) dans quelle mesure la programmation doit-elle être autorisée dans le contexte des systèmes de RCO, sous quelle forme et avec quelle sémantique ? (2) comment représenter sous forme “déclarative” des composants procéduraux ou fonctionnels, qui deviennent alors des objets pouvant être traités par le processus de classification ? Ces travaux, dont une première ébauche se trouve dans [CDE+95] sont au premier plan d'une vision moderne de la programmation par objets et méritent de fait de plus amples développements.

2.1.3 Réorganisation de hiérarchies

Participants: H. Dicky, C. Dony, M. Huchard, T. Libourel (LIRMM).

Thèmes et objectifs: Les hiérarchies d'héritages utilisées en programmation et en représentation de connaissances par objets ainsi que dans les bases de données du même nom sont le plus souvent construites ”à la main” par un concepteur ou un groupe de concepteurs selon des règles difficilement modélisables formellement. Lorsque la complexité et la quantité des concepts à structurer devient trop importante, ou lorsque qu'une hiérarchie a beaucoup évolué depuis sa conception initiale, il devient intéressant d'automatiser au moins en partie sa (re)structuration.

Etat d'avancement: Un des premiers critères pris en compte par les recherches dans ce domaine a été celui de la factorisation maximale de propriétés. Nous avons ainsi proposé [DDH94] un algorithme de réorganisation de hiérarchies d'héritage basé sur le treillis de Galois de la relation liant classes et attributs. Le second problème essentiel sur lequel nous avons travaillé est celui de la prise en compte de la surcharge des noms de propriétés. Nous avons ainsi proposé une extension de notre premier algorithme[DDH95] avec une première prise en compte de la surcharge supposant connues les relations entre les différentes propriétés de même nom. Une seconde extension plus ambitieuse, n'utilisant pas ces relations est en cours de définition[DDH95b].

Relations nationales et internationales: Nous sommes en contact avec les collègues travaillant sur le sujet et en particulier avec R.Godin de l'université de Montréal et K.Lieberherr de *NorthEastern University.*

2.1.4 Représentations efficaces de hiérarchies

Responsable: Michel Habib

Chercheurs impliqués: Michel Habib, Marianne Huchard, Lhouari Nourine - LIRMM

Thème et objectif scientifiques: Dans de nombreuses applications, il est important de pouvoir stocker et manipuler efficacement les hiérarchies d'héritage qui sont des graphes sans circuit ou des treillis. Ce problème est un cas particulier de celui du codage d'un ordre.

Un codage ou plongement d'un ordre est une application f d'un ordre $P = (X, \leq)$ dans un ordre $Q = (Y, \leq)$ telle que $f : X \to Y$ et vérifie: $x \leq_P y$ si et seulement si $f(x) \leq_Q f(y)$. Un tel codage est intéressant lorsque la relation d'ordre sur Q est plus facile à manipuler que celle sur P, où lorsque la place mémoire utilisée est réduite. L'opération couramment utilisée est du type test de comparabilité $x \leq y$?, mais dans le cas des treillis, il est naturel de vouloir calculer efficacement des bornes supérieures ou inférieures.

Les codages les plus fréquents sont ceux où Q est un treillis booléen, on obtient alors des codages par vecteurs de bits. Les problèmes d'optimisation sous-jacents sont NP-difficiles et la détermination de l'optimum semble être un problème combinatoire extrêmement difficile, mais aucun argument théorique n'interdit de construire des codages optimaux sur des classes particulières d'ordres ou de concevoir des heuristiques très efficaces pour le cas général. Les principales heuristiques actuelles ont été proposées par Aït-Kaci, Caseau, Ellis et Habib-Huchard-Nourine.

L'objectif est de produire des codages plus efficaces que ceux actuellement proposés, tout en restant calculables efficacement. Un deuxième problème voisin est celui de l'organisation (indexation) de bases de données de graphes (par exemple de graphes de molécules) qui permette la recherche efficace d'appariements. Par exemple si l'on veut rechercher à partir d'une molécule donnée toutes les molécules de la base qui sont contenues (morphisme injectif) dans la molécule donnée.

Etat d'avancement: Nous avons formalisé dans [HHN95] ce problème de codage comme celui du calcul d'une nouvelle dimension dite **dimension de codage**, qui est le plongement le plus efficace d'un ordre dans un produit de chaînes. Nous avons montré que la 2-dimension ou dimension booléenne (plongement dans un produit de chaînes à 2 éléments) était une bonne approximation. En outre nous avons proposé une heuristique de codage basée sur une décomposition coatomique des treillis.
Dans [HN95] nous avons proposé une autre heuristique de calcul de la 2-dimension basée sur un théorème de décomposition.

Relations nationales et internationales: Nous sommes en liaison étroite avec les principaux chercheurs du domaine (Aït-Kaci et son étudiant A. Fall de Vancouver (Canada), G. Ellis de Melbourne (Australie), Y. Caseau de l'ENS).

2.2 Sous projet Langages à prototypes et réflexion

2.2.1 Chercheurs impliqués

Pierre Cointe, Philippe Mulet (EMN), Christophe Dony, Daniel Bardou (LIRMM). Ce projet est mené en collaboration directe avec l'équipe de Jacques Malenfant à l'université de Montréal.

2.2.2 Thème et objectif scientifiques

Les systèmes à prototypes sont utilisés aussi bien en représentation des connaissances, dans les langages de *frames*, qu'en programmation avec les langages à prototypes apparus vers la fin des années 80. La programmation par prototype est caractérisée par un mécanisme d'héritage différent (la délégation) de celui des langages à classes et par l'absence possible des dites classes.
Initié dans l'équipe RXF/LITP par Pierre Cointe, ce projet est maintenant mené en collaboration entre l'EMN, le LIRMM et l'équipe de Jacques Malenfant à l'Université de Montréal.

Il a pour objectifs: (1) de mieux comprendre la programmation sans classes, (2) de fournir une (ou des) sémantique(s) précise(s) au mécanisme de délégation et de ses bonnes utilisations, savoir s'il est caractéristiques des objets sans classes ou s'il peut s'intégrer, et si oui comment, à n'importe quel sorte de langage à objets.

(3) d'expliciter, à partir de cette étude, les différences fondamentales entre les différents paradigmes de programmation par objets que sont la programmation par classes, par *frame* et par prototypes (comme par exemple Garnet, Kevo, Self ou Yafool), auxquels on peut ajouter la programmation par acteurs qui présente des similitudes avec la programmation par prototype si on fait abstraction de l'aspect parallélisme.

4) d'utiliser les langages sans classes (plus simple) comme support à l'étude des langages à objets réflexifs.

2.2.3 Etat d'avancement

Compréhension du mécanisme de délégation

Nous avions proposé une première analyse couplée à une taxinomie des langages à prototypes (OOPSLA'92). Suite à cette première analyse une plate-forme logicielle de simulation de langages avec délégation à été réalisée en Smalltalk. Elle permet d'évaluer opérationellement ces langages ou de tester aisément de nouveaux traits. ELle est accessible sur le serveurs ftp anonyme du LIRMM.

Une seconde analyse de ces langages, basée sur le nombre de sortes d'objets et le nombre de liens et complétée par la présentation d'une sémantique dénotationelle est proposée dans [Ma95a]. Ces analyses font apparaître un certain nombre de problème sémantique liés à la délégation[Mal] dont celui, fondamental, de perte de l'identité des objets. Une solution à ce problème est proposée dans [BD95, Ma95b] sous la forme d'un modèle d'objet morcelés, qui continue à être étudié et développé, notamment pour son intérêt dans la mise en oeuvre de mécanismes de points de vues.

Par ailleurs, il a été établi une analogie entre les concepts d'environnements Scheme et de prototypes Self. Cette analogie nous a conduit à proposer un outil pédagogique de simulation de prototypes en Scheme [MM94].

Prototypes et Réflexion

Notre second thème d'intérêt est l'étude et la mise en oeuvre de modèles de prototypes réflexif. Les résultats obtenus dans ce domaine sont les suivants:

- Formalisation d'une sémantique d'un modèle à objets réflexifs dans lequel a été mise en œuvre une réflexion de comportement permettant de paramétrer au niveau de chaque objet la sémantique de l'envoi de messages. Cette sémantique est formulée par système de ré-écriture [MDC95]. Cette formalisation démontre "la convergence" du processus de recherche de méthodes par envoi de message en présence de méta-objets.

- Implémentation en Scheme du langage à prototypes Moostrap, [MC94].

  Dans la tradition des micro-interprètes à la MicroPlasma, ObjVlisp, ClassTalk ou Actalk, Moostrap est construit à partir d'un noyau primitif définissant une sémantique "minimale" mais paramétrable de manière générique. Ce noyau a pour vocation d'être étendu pour étudier et modéliser des variations sur la sémantique de résolution des envois de messages Cette sémantique repose sur un protocole de méta-objets (MOP) permettant d'en contrôler dynamiquement la teneur.

  Dans la continuité, nous avons étudié les problèmes de méthodologie d'utilisation de notre langage réflexif. Les méta-objets posent des problèmes quand il devient nécessaire de les composer. Nous avons proposé des mécanismes permettant de composer les sémantiques de différents méta-objets, et ce de manière incrémentale. Dans [MMC95], nous posons les jalons de l'étude d'un tel mécanisme, lui-même réflexif, qui garantit de manière automatique la monotonie du système.

  Le langage Moostrap et les travaux qui s'y rapportent sont décrits dans la thèse de Philippe Mulet[Mul95].

2.3 Sous-projet Représentation de connaissances complexes et évolutives: Points de vues, Objets composites, Evolution.

Ce sous-projet regroupe des recherches sur la représentation et la manipulation de connaissances complexes utilisant les langages à objets.

Le succès des langages à objets provient tout autant des qualités des logiciels obtenus: modularité, extensibilité, réutilisabilité que du fait que les objets (informatiques) ont une capacité “naturelle” de représentation (des objets) du “monde”. Cette capacité est à l'origine de la représentation des connaissances par objets, autant que des méthodes d'analyse et de conception.

2.3.1 Responsables

Bernard Carré, Mourad Oussalah

2.3.2 Chercheurs impliqués

Bernard Carre, Gilles Vanwormhoudt, Laurent Debrauwer - LIFL.
Mourad Oussalah, Martine Magnan, Veronique Puig, Christelle Urtado - (LGI2P)

2.3.3 Thèmes et Objectifs Scientifiques

Un premier thème traité au LIFL concerne la représentation et la manipulation de **points de vue**. Il s'agit de permettre la programmation ou la représentation de connaissances dans des mondes ou les objets manipulés peuvent être vus selon différentes perspectives. Les systèmes de gestion des points de vue permettent d'associer plusieurs représentations (intensions) a une même entité (le référent, d'extension unique dans le monde réel). Il existe essentiellement deux approches du sujet : l'une reposant sur la composition d'objets (agrégation de perspectives), l'autre considérant qu'il s'agit à la base d'un problème d'héritage multiple. Nous nous intéressons plus particulièrement a l'étude de cette dernière approche autour du langage ROME réalisé par B.Carré qui propose une stratégie d'héritage par points de vue.

Un second thème, traité au LGI2P, s'intéresse aux différentes techniques d'évolution d'objets complexes sous contraintes. Ces techniques s'appuient d'une part sur la modélisation d'objets (simples et composites) et d'autre part sur des mécanismes de propagation basés sur la conversion et le versionnement. Par ailleurs, pour maintenir la cohérence de ces objets complexes, l'équipe développe un modèle de contraintes inter objets.

2.3.4 Etat d'avancement

Points de vues:
A la suite du langage Rome décrit dans des publications antérieures, le langage de représentation de connaissances à base de *frames* "FROME" [Dek93, Dek94], a été développé. Il adapte et enrichit ROME pour les structures d'objets plus complexes que sont les *frames*. Ce travail, son application au mécanisme de classification et la mise en oeuvre du langage ont donné lieu à la thèse de Lenneke Dekker[Dek94].

Un projet plus récent, CROME (Contextes en ROME), tente de systématiser l'approche par points de vue pour permettre et structurer les descriptions fonctionnelles multiples d'objets. Cette approche s'adresse plus particulièrement à la conception par objets d'architectures multi-applications[Car94] telles qu'on les rencontre en CAO ou dans les plate-forme de travail coopératif, elle fait l'objet du rapport de DEA de Gilles Vanwormhoudt et Philippe Rio (**CROME, Contextes en ROME**).
Une description de ces travaux se trouve à l'URL http://www.lifl.fr/LIFL/EQUIPE/goal.html.

Objets composites:
Au LGI2P a été défini un modèle d'objet composites [Mag94, MO93] permettant de définir explicitement les différents types de relations de composition , leurs propriétés sémantiques et leurs modèles de partage d'informations. Voici différents résultats relatifs à ce modèle.

- Exceptions dans les modèles d'objets composites:

  Afin de favoriser l'évolution d'objets complexes, nous avons introduit les notions d'exception dans les objets composites. Nous distinguons trois types d'exceptions: les exceptions à l'héritage (horizontal) d'une propriété que nous appelons exceptions classe-propriété [OM95, MO95] et les exceptions par annulation d'un lien de la relation de composition que nous appelons exceptions classe-classe ou exceptions instance-classe [Mag94].

– Contraintes dans les hiérarchies de composition:

L'étude que nous avons menée sur l'intégration des contraintes dans les objets composites nous a permis de définir un modèle de contraintes inter-objets [OP95, PO95]. Ce modèle permet d'une part la représentation et la pose de contraintes dans les hiérarchies de composition d'objets, et d'autre part, le contrôle et la résolution de ces contraintes.

Versions d'objets:
Dans le cadre de l'étude et du développement d'un modèle de gestion de versions d'objets simples et composites, une synthèse des solutions existantes en matière de versions et plus particulièrement de versions d'objets a été réalisée [TA94]. Parallèlement, un sustème de stockage de versions d'objets à été implanté (DEA de X.Vincent) et un modèle de propagation des opérations de création et de destruction de versions d'objets a été défini[TO93, OT93].

2.3.5 Relation Nationales et Internationales

Ces travaux sont utilisés dans diverses applications en collaboration avec des industriels
- **Presage**: (collaboration CNET(Paris)-LGI2P) "Un environnement orienté objet pour la modélisation de réseaux de télécommunication" que nous développons à la section 3.3.
- **Leopard**: (collaboration CNET(Robinson)-LGI2P "Extension de la base de donnée VERSANT à la prise en compte d'objets complexes évolutifs sous contraintes".

2.4 Sous-projet: Spécifications formelles

2.4.1 Responsable: Jean-Claude Royer

2.4.2 Chercheurs impliqués

Jean-Claude Royer, Pascal André, Pascal Poizat - IRIN.

2.4.3 Thème et objectif scientifiques

Ces recherches se situent à l'intersection des spécifications formelles et de l'approche à objets. Les spécifications formelles s'imposent progressivement comme une condition essentielle à un bon développement logiciel. Leurs avantages sont nombreux : rigueur, preuve, automatisation partielle etc. Par contre leur utilisation dans le domaine industriel reste marginale entre autre par ce qu'elles demandent une formation spécifique et qu'elles sont peu outillées.

L'intégration d'aspects formels (spécifications, modèles formels à objets, preuves, contrôles, transformations de programme, ...) dans le cadre de la programmation par objets, et particulièrement dans les méthodes d'analyse et de conception à objets, est un enjeu important en génie logiciel. En effet ces préoccupations sont à la base d'une bonne inter-opérabilité et de la spécification de composants réutilisables. Les objectifs immédiats sont d'obtenir des concepts, règles, algorithmes, outils que l'on pourra introduire dans les méthodes à objets actuelles : OMT, Fusion, OOD, Shlaer et Mellor, etc. A plus long terme, il s'agit de concevoir et de réaliser un environnement minimum supportant les concepts et les outils mais aussi d'interfacer vers d'autres environnement existants.

2.4.4 Etat d'avancement

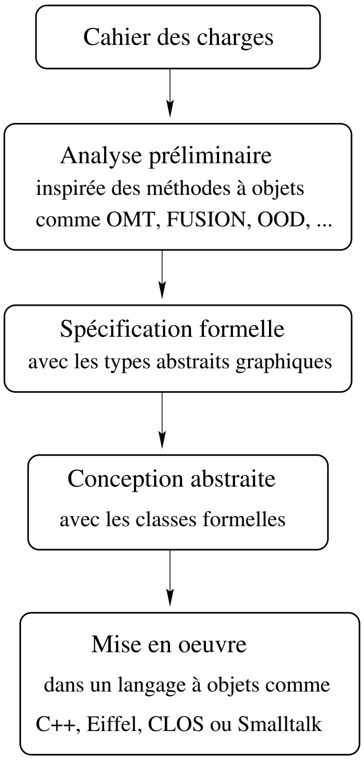
Une des premières tâches a été de définir un modèle abstrait et formel pour les langages à classes comme C++, Eiffel, Smalltalk ou CLOS. Ce formalisme, dit **des classes formelles**, est destiné à la conception plus abstraite qu'avec les langages de programmation courants. Le modèle est toujours en constante amélioration, plusieurs publications sur ses divers aspects ont été faites[AR94, ACR94, ACCR94]. Parmi ses points forts : nous disposons d'un modèle algébrique avec évaluation par réécriture de termes [Poi95] et d'un emploi assez strict mais conforme aux habitudes de programmation en objets. Récemment nous avons abordé le contrôle de type, nous avons défini un système de contrôle sûr et montré les diverses alternatives viables tout en gardant des règles naturelles pour le programmeur [CCR95].

La deuxième tâche est la mise au point d'un formalisme plus abstrait et mieux adapté à la spécification de haut niveau d'un système. Ce formalisme dit **des types abstraits graphiques** associe de façon cohérente le comportement dynamique (automate gardé) et une description fonctionnelle (type abstrait algébrique) du système. L'avantage est de disposer d'un formalisme visuel facilement compréhensible et d'outils facilitant l'écriture de la spécification algébrique à partir de l'automate. Cet aspect a été le sujet de thèse de Pascal André [And95], il a aussi montré comment on pouvait concevoir les types abstraits graphiques en classes formelles.

Un certain nombre d'outils ont été développés: éditeurs de TAG et de CF, traducteur de CF en Smalltalk et en Eiffel et une plate-forme générale permettant le support des précédent concepts [AR95, And95].
Le dernier point de notre travail concerne l'utilisation de ces concepts dans un cadre de développement plus réaliste. Le schéma général des différentes phases est décrit dans la figure 2.4.4:

Une première approche a été tenté en utilisant pour l'analyse préliminaire la méthode OMT dont le principal défaut est sûrement un manque de rigueur et de contrôle dans la démarche. Cette expérimentation a été faite en collaboration avec Franck Barbier spécialiste des méthodes à objets et de leur application dans un contexte industriel [ABR95].

Pour poursuivre ces expérimentations nous avons obtenu un contrat (financement: Plan Etat-Région-FEDER-Ville d'Angers-Département d'Angers) avec des laboratoires régionaux. Ce projet concerne la gestion de production et est piloté par le CRITT pays de Loire Productique. Notre travail consiste à développer un prototype du système en utilisant la technologie à objets. Ce projet vient de démarrer on ne pourra en tirer quelques conclusions que fin 1997.



Chaîne de développement

2.4.5 Relations nationales et internationales

Michel Augeraud,Laboratoire d'Informatique et d'Imagerie Industrielles, Université de La Rochelle.
Roger Rousseau, I3S, Université de Nice Sophia-Antipolis.

Ces travaux sont faits en collaboration avec l'équipe de Dan Chiorean du laboratoire d'informatique de l'Université de CLUJ (Roumanie).

3 Applications Industrielles

Certaines des recherches que nous venons de décrire sont utilisées dans des applications d'envergure menées en collaboration avec des industriels. Nous détaillons ici plus en détails les utilisations de: l'héritage non cumulatif non-monotone (cf. section 2.1.1 et les techniques de classification (cf. section 2.1.2) dans le projet *RESYN*, les implantations des techniques de représentation de connaissances (cf. section 2.3) et les contraintes à objets dans les projets *RODIN* et *PRESAGE*.

3.1 RESYN : applications de la programmation par objets en chimie organique

Cette application résulte de la coopération entre un industriel de la chimie (Roussel Uclaf), un industriel de l'intelligence artificielle (Framentec) et des groupes de recherches publics (LIRMM et CCIPE). Ces travaux se font dans le cadre du GDR 1093 du département Chimie du CNRS et du groupe ELO du GDR Programmation.

Les systèmes de synthèse assistée par ordinateur — systèmes de SAO dans la suite — existent depuis la fin des années soixante. Ils ont pour but d'aider les chimistes à élaborer un plan de synthèse pour une molécule donnée (un plan de synthèse n'est autre qu'un plan de construction de la molécule, où les outils de base sont des molécules accessibles et des réactions chimiques). RESYN (REtroSYNthèse) est un système de SAO, qui s'appuie sur un système de RCO où sont combinés programmation par objets, raisonnement par classification et raisonnement à partir de cas [NLD94] [NL94]. RESYN a été entièrement écrit avec le système à objets Y3, qui intègre les algorithmes d'héritage décrits au § 2.1.1. En particulier, le projet a nécessité la construction de bases hiérarchiques d'objets conséquentes et le développement de méthodes spécifiques pour la représentation d'objets complexes (que sont les molécules) [Vis95].

La philosophie de RESYN repose sur une combinaison des techniques de programmation par objets et de RCO [NLD94]. En ce sens, le système RESYN illustre parfaitement la notion de système hybride, où programmation et raisonnement se complètent l'un et l'autre. À l'heure actuelle, les travaux sur RESYN continuent et concernent également l'intégration de la programmation par contraintes et l'étude d'architectures distribuées pour les systèmes de SAO.

Le système RESYN a obtenu en 1994 le prix ADER – Banques Populaires *Entreprises – Innovation* qui récompense chaque année un travail de recherche alliant un laboratoire et une entreprise.

3.2 RODIN: Planification de réseaux

L'équipe de Michel Dao fait partie d'un groupement du CNET [2] qui développe des méthodes et des outils pour la planification de réseaux. Cette équipe a la responsabilité du projet RODIN qui a pour objectif d'étudier, au travers notamment de la réalisation de prototypes, l'apport des techniques à objets (représentation, programmation, conception, bases de données) pour le dimensionnement des réseaux, qui constitue un des aspects de la planification.

Le support commun aux travaux du projet RODIN est CONRAD-OBJETS [DL94, MD94] qui est un prototype d'outil de dimensionnement de réseaux téléphoniques commutés à acheminement adaptatif. Cet outil comporte une modélisation en classes du domaine concerné utilisant les traits de représentation des connaissances offerts par Y3 ;

CONRAD-OBJETS est développé avec CASSIS qui est un environnement de conception d'applications dédiées réseaux, développé au dessus de Y3.

Deux activités de ce projet intéressent le groupe ELO :

- **Contraintes et objets**
  CONRAD-OBJETS gère un grand nombre de dépendances complexes entre les différentes propriétés (principalement numériques) des objets. L'utilisation de mécanismes de propagation de contraintes sur les propriétés d'objets offre une possibilité de gestion automatique de ces dépendances d'une manière plus déclarative. Des contraintes fonctionnelles unidirectionnelles de type agrégat sur les attributs d'objets sont maintenues paresseusement par des réflexes attachés à ces attributs. Ces travaux ont été effectués dans le cadre du stage de DEA de Stéphane Pelenc (*Application de techniques de propagation de contraintes en langage à objets*) et se prolongent actuellement par une thèse. Un système de contraintes fonctionnelles a été développé dans Y3, et est en cours de mise en øuvre dans CONRAD-OBJETS.

- **Représentation par objets de comportements**
  L'implémentation d'algorithmes de dimensionnement de réseaux sur une représentation par objets de ceux-ci fait l'objet du travail de thèse de Milène Shteto au CNET. Ce travail consiste à concevoir deux hiérarchies de classes du domaine étudié : une pour les différentes architectures de réseaux, et l'autre pour les méthodes (aux sens mathématique et algorithmique) de dimensionnement, et à étudier la manière de mettre en relation ces deux hiérarchies, afin d'appliquer les comportements de la seconde sur la première. Les méthodes sont réifiées, et la composition est utilisée pour leur combinaison.

3.3 PRESAGE: Modélisation de réseaux de télécommunication

Le projet *Presage* de réalisation d'un environnement orienté objet pour la modélisation de réseaux de télécommunication fait intervenir les sociétés CNET(Paris) et ITESOFT(Nîmes), il est prévu pour la période

2. Centre National d'Etudes des Télécommunications, centre d'étude et de recherche de France Télécom.

sept. 90, octobre 96.

Ce projet porte sur l'étude et le développement d'un environnement de conception et d'exécution de logiciel de planification de réseaux, il s'intéresse essentiellement à la partie descriptive du processus de planification de réseaux TAR (trafic-acheminement-routage).

Dans ce projet, les bases théoriques sont fondées sur des modèles complexes utilisant des représentations d'objets composites. Pour ce faire le noyau du langage **Yafool** a été étendu pour y intégrer les mécanismes de représentation et de gestion d'objets composites ainsi que les mécanismes de gestion des versions décrits précédemment (cf. section 2.3.4).

Références

[ABR95] Pascal André, Franck Barbier, and Jean-Claude Royer. Une expérimentation de développement formel à objets. *Technique et Science Informatique*, 14(8):973–1005, octobre 1995.

[ACCR94] Pascal André, Dan Chiorean, Corina Cîrstea, and Jean-Claude Royer. Object-Oriented Design With Formal Classes. In *ConTI'94: International Conference on Technical Informatics*, 1994. 16-19 November, Timisoara, Romania.

[ACR94] Pascal André, Dan Chiorean, and Jean-Claude Royer. The Formal Class Model. In *Joint Modular Languages Conference, Modula, Oberon & friends*, Ulm, Germany, 1994. GI, SIG and BCS. September, 28-30.

[AR94] Pascal André and Jean-Claude Royer. La modélisation des listes en programmation par objets. In Pierre Cointe, Christian Quiennec, and Bernard Serpette, editors, *Journées Francophones des Langages Applicatifs (JFLA'94)*, number 11 in Collection Didactique, pages 259–285, Noirmoutier, 31 janvier - 1 février 1994. INRIA.

[AR95] Pascal André and Jean-Claude Royer. Formal Concepts and Tools for Object-Oriented Analysis and Design. In *5th ICC: Object-Oriented Technology: Methods and Applications*. Tata McGraw-Hill, January 1995. 9-11 January, Hyderabad, India.

[And95] Pascal André. *Méthodes formelles et à objets pour le développement du logiciel : Etudes et propositions*. Thèse de doctorat, IRISA, Université de Rennes I, 7 juillet 1995. préparée à l'IRIN - Nantes.

[BD95] Daniel Bardou, Christophe Dony Propositions pour un nouveau modèle d'objets dans les langages à prototypes. Actes de LMO'95 (Langages et Modèles à objets), pp. 93-109, Nancy, octobre 1995.

[CCR95] Dan Chiorean, Corina Cîrstea, and Jean-Claude Royer. Type Checking Formal Classes: Principles and Algorithms. Rapport de recherche 99, Institut de Recherche en Informatique de Nantes, October 1995.

[CDE+95] B. Carré, R. Ducournau, J. Euzenat, A. Napoli, and F. Rechenmann. Classification et objets : programmation ou représentation ? In *Actes des cinquièmes journées nationales PRC GDR Intelligence Artificielle, Nancy*, pages 213–237. Teknea, Toulouse, 1995.

[Car94] Bernard Carré et Lenneke Dekker. CSCW et partage d'objets. Actes des journees Ganymede sur le travail cooperatif et la communication avancee, Lille, Juin 1994. Publication interne LIFL no 159.

[DDH94] H. Dicky, C. Dony, M. Huchard, T. Libourel. ARES, un algorithme d'Ajout avec REStructuration dans les hiérarchies de classes. Actes de la conférence LMO'94 (Langages et Modèles à objets), pp. 125-136, Grenoble, octobre 1994.

[DDH95] H. Dicky, C. Dony, M. Huchard, T. Libourel. Ares, Adding a class and REStructuring Inheritance Hierarchies Dans Actes de BDA'95 (Bases de Données Avancées) pp.25-42, Nancy, 1995.

[DDH95b] H. Dicky, C. Dony, M. Huchard, T. Libourel. On automatic class insertion with overloading Rapport de recherche LIRMM no 95 054, 1995.

[DHH+94] R. Ducournau, M. Habib, M. Huchard, and M-L. Mugnier. Proposal for a monotonic multiple inheritance linearization. In *Proceedings of the Ninth ACM Conference on Object-Oriented Programming, Languages and Applications, OOPSLA'94*. ACM Press, 1994.

[DHH95] R. Ducournau, M. Habib, M. Huchard, M-L. Mugnier, and A. Napoli. Le point sur l'héritage multiple. *Technique et Science Informatiques*, 14(3):309–345, 1995.

[DL94] M. Dao and M. Lebourges. Integration of Object-Oriented Programming and Distributed Algorithms for Network Design. In *Networks94*, pages 161–166. Hungarian Scientific Society for Telecommunications, september 1994.

[DN94] L. Dekker and A. Napoli. Report on the IJCAI-93 Workshop on "Object-Based Representation Systems". *SIGART Bulletin*, 5(1):57–59, 1994.

[Dek93] Lenneke Dekker. La reification des filtres en FROME Actes de RPO'93, Représentations par Objets, EC2 Ed, La Grande Motte, 17-18 juin 1993.

[Dek94] Lenneke Dekker. FROME : Représentation multiple et classification d'objets avec points de vue Thèse de l'Universite des Sciences et Technologies de Lille. 13 Juin 1994.

[Duc95] R. Ducournau. Les systèmes classificatoires. Research Report 95-038, L.I.R.M.M., Montpellier, 1995.

[HHN95] M. Habib, M. Huchard, L. Nourine: Embedding partially ordered sets into chain-products, in International KRUSE Symposium: Knowledge Retrieval, Use and Storage for Efficiency, University of Santa-Cruz, Aout̂ 1995, 147-161.

[HN95] M.Habib, L. Nourine, A new dimensional parameter and a removal theorem for boolean dimension, 4TH Twente Workshop on graphs and combinatorial optimization, juin 1995, 127-132.

[HHN95] M. Habib, M. Huchard, L. Nourine: newblockEmbedding partially ordered sets into chain-products. newblock In International KRUSE Symposium: Knowledge Retrieval, Use and Storage for Efficiency, University of Santa-Cruz, Aout̂ 1995, pp.147-161.

[HN95] M.Habib, L. Nourine. newblockA new dimensional parameter and a removal theorem for boolean dimension newblock 4TH Twente Workshop on graphs and combinatorial optimization, juin 1995, 127-132.

[LMO95] Actes du colloque "Langages et Modèles à Objets" (LMO'95), Nancy. Publiés sous la direction de A. Napoli, Unité de recherche Inria Lorraine, Nancy, 1995.

[MC93] Philippe Mulet et Pierre Cointe. Définition d'un noyau réflexif pour un langage à prototypes. Dans les *Actes des Secondes Journées Représentation par Objets*, pages 101–116, La Grande Motte, France, Juin 1993. EC2-IA2.

[MC94] Philippe Mulet et Pierre Cointe. Definition of a Reflective Kernel for a Prototype-Based Language. Edité par Shojiro Nishio et Akinori Yonezawa, *Lecture Notes in Computer Science vol.742: First International Symposium on Object Technologies for Advanced Software*, pages 128–144, Kanazawa, Japan, Novembre 1993. JSSST-JAIST, Springer-Verlag.

[MD94] Milène Shteto and Michel Dao. An Object-Oriented Tool for Partial Network Sizing. In Michel Dao, editor, *Workshop on the Use of Object-Oriented Technology for Network Design and Management*, pages 75–85. ECOOP'94, july 1994.

[MDC95] Jacques Malenfant, Christophe Dony et Pierre Cointe, A Semantic of Introspection in a Reflective Prototype-Based Language, LASC: Journal on Lisp And Functional Programming à paraître en 1996.

[MM94] Philippe Mulet and Marco Jacques. De la parenté entre les environnements de MIT Scheme et les prototypes de Self. Dans les *Actes de Langages et Modèles à Objets'94*, pages 167–178, Grenoble, France, Octobre 1994.

[MMC95] Philippe Mulet, Jacques Malenfant, and Pierre Cointe. Towards a Methodology for Explicit Composition of Meta-Objects. In *Proceedings of OOPSLA'95*, pages 316–330, Austin, Texas, October 1995. ACM Sigplan Notices.

[MO93] M. Magnan, C. Oussalah. Object Evolution In *Proceedings of the* 5^{th} *IEEE Intl Conf. on Software Engineering and Knowledge Engineering*, San Francisco, June

[MO94] lM. Magnan, C. Oussalah. Multiple Inheritance Systems with Exceptions In *Artificial Intelligence Review Vol. 6*, Kluwer Acad.Pub.1994, pp. 3-24.

[MO95] M. Magnan, C. Oussalah. Composition Graphs and Exception In *KBKS'95 Proceedings*, Avril 1995, Hollande, N.J.I. Mars-IOS Press, pp. 246-258.

[Ma95a] Jacques Malenfant. On the semantic Diversity of Delegation-Based Programming Languages In *Proceedings of OOPSLA'95*, pages 215-227, Austin, Texas, October 1995. ACM Sigplan Notices.

[Ma95b] Split Objects: Taming value sharing in Object-Oriented Languages Rapport de Recherche 968 - IRO - Univ. de Montréal, Avril 1995.

[Mag94] M. Magnan. Réutilisation de Composants : un modèle d'objets composites avec exceptions (Thèse soutenue à l'Université de Montpellier II, Septembre 94).

[Mal] Jacques Malenfant. Abstraction et Encapsulation en Programmation par Prototypes Technique et science informatique. A paraître, 1996.

[Mul95] Philippe Mulet. *Réflexion & Langages à prototypes.* PhD thesis, Université de Nantes, Ecole des Mines de Nantes, July 1995.

[NLD94] A. Napoli, C. Laurenço, and R. Ducournau. An object-based representation system for organic synthesis planning. *International Journal of Human-Computer Studies*, 41(1/2):5–32, 1994.

[NL94] A. Napoli and J. Lieber. A First Study on Case-Based Planning in Organic Synthesis. In S. Wess, K.-D. Althoff, and M.M. Richter, editors, *Topics in Case-Based Reasoning – First European Workshop (EWCBR'93), Kaiserslautern*, Lecture Notes in Artificial Intelligence 837, pages 458–469. Springer Verlag, Berlin, 1994.

[Nap94] A. Napoli. Studies about the Integration of Classification-Based Reasoning and Object-Oriented Programming. In F. Baader, M. Lenzerini, W. Nutt, and P.F. Patel-Schneider, editors, *Working Notes of the 1994 Description Logics Workshop*, pages 60–62. DFKI Saarbruecken, 1994.

[Nap95] A. Napoli. Objects, Classes, Specialization and Subsumption. In A. Borgida, M. Lenzerini, D. Nardi, and B. Nebel, editors, *Proceedings of the 1995 International Workshop on Description Logics, Universita di Roma, June 2–3 (Technical Report 07.95)*, pages 52–55, 1995.

[OM95] C. Oussalah, M. Magnan. Exceptions in Composition Links In *Proceedings of the Workshop on Monotonic Reasoning*, 29-31 Mars 1995, Utrecht- Hollande, pp 114-126.

[OP95] C. Oussalah, V. Puig. Combining Objects and Constraints In *IEEE (Software Engeneering, Knowledge Engeneering) SEKE'95 Proceedings*, Juin 1995, Rockville Maryland-USA, pp. 133-145.

[OT93] C. Oussalah, G Talens. Concepts and Methods for version Modeling In *Proceedings of the* 3^{rd} *IEEE/ACM EuroDAC*, Hamburg, Germany, Septembre 1993, pp. 332-338.

[PO95] V. Puig, C. Oussalah. An Object Constraint Model In *Scandinave Conference on AI, SCAI'95 Proceedings*, Mai 1995, Trondheim, Norvège, pp. 310-322. (1993), pp. 546-554. (nominated as one of the best five papers of

[Poi95] Pascal Poizat. Applications de la réécriture aux modèles à objets. Mémoire de dea d'informatique, Institut de Recherche en Informatique de Nantes, 2, rue de Houssinière 44072 Nantes Cedex 03, 14 juin 1995.

[TA94] G. Talens. Gestion de Versions d'Objets Simples et Composites (Thèse soutenue à l'Université de Montpellier II, Février 94).

[TO93] G Talens, C. Oussalah. Versions of Simple and Composite Objects In *Proceedings of the 19th Intl. Conf. on Very Large Data Bases*, Dublin, Ireland, Aug. (1993), pp. 62-72.

[VQ95] Philippe Vismara and Joel Quinqueton. Design of planification strategies in organic chemistry. In *Proceedings of IASTED'95 (Int. Association of Science and Technology for Development Conference*, Innsbruck, 1995.

[Vis95] P. Vismara. Reconnaissance et représentation d'éléments structuraux pour la description d'objets complexes. Application à l'élaboration de straégies de synthèse en chimie organique. Thèse de l'Université des Sciences et Techniques du Languedoc, Montpellier, 1995.