

# VERS LA SIMULATION NUMERIQUE PAR AGENTS APPRENANTS

Segmentation, dynamisation  
et autres lois d'évolution des logiciels

DOSSIER D'HABILITATION A DIRIGER DES RECHERCHES

DE L'UNIVERSITE PIERRE ET MARIE CURIE

SPECIALITE : INFORMATIQUE

présenté par

BERTRAND BRAUNSCHWEIG

Institut Français du Pétrole

Soutenu le ... 2002 devant le jury composé de :

Michel GONDRAN, Conseiller à la Direction des Etudes et Recherches d'EDF	rapporteur
Jean-François PERROT, Professeur au Laboratoire d'Informatique de Paris 6	rapporteur
Marc SCHOENAUER, Directeur de recherche à l'INRIA, projet FRACTALES	rapporteur
Alain BAMBERGER, Directeur de la Recherche de l'ENS Ponts et Chaussées	examineur
Michel COMBARNOUS, Professeur à l'Université de Bordeaux I	examineur
Wolfgang MARQUARDT, Professeur à l'Université d'Aix-la-Chapelle	examineur
Michel PONS, ingénieur TotalFinaElf	examineur

## Résumé

Je m'intéresse à la segmentation et à la dynamisation du logiciel. Les logiciels, en particulier dans le domaine scientifique, sont parmi les objets artificiels les plus complexes créés par l'homme. La segmentation (division en modules intéropérants de plus en plus fins) et la dynamisation (introduction de fonctions d'adaptation) permettent de réaliser des logiciels disposant de fonctions sophistiquées, capables de traiter les défis de simulation et de contrôle posés par l'évolution de notre société. Mon travail sur la segmentation et la dynamisation des logiciels est un exemple d'application de deux principes fondamentaux de TRIZ, théorie de la résolution inventive de problèmes. TRIZ a été développée par le chercheur russe Altshuller au milieu du siècle dernier, afin de donner un cadre formel aux processus d'innovation dans les systèmes techniques (physiques, chimiques, mécaniques etc.). Je montre comment quelques travaux de recherche que j'ai menés peuvent être analysés comme contribuant à la segmentation du logiciel et à sa dynamisation. Je propose, en prolongement de cette expérience, un projet de développement de simulation de procédés au moyens d'agents intelligents utilisant le standard CAPE-OPEN. En complément, j'examine quelques autres principes et lois qui s'appliquent également au logiciel, à son architecture, à son développement et à son mode d'utilisation.

## Abstract

I study segmentation and dynamisation of software artifacts. Scientific software systems are among the most complex artificial systems created by mankind. Segmentation (division into smaller interoperable components) and dynamization (introduction of adaptive behaviour) allow to develop increasingly sophisticated software systems, able to tackle the technical challenges demanded by our society, such as simulation and control of very complex objects. My research on segmentation and dynamisation can be put in relation with TRIZ, the Theory of Inventive Problem Solving. TRIZ, was developed by the Russian researcher Altshuller in the nineteen forties and fifties. With this theory, Altshuller tried to provide a formal framework for innovation processes in technical systems (mechanical, physical, chemical systems etc.). I show that some of the research work that I developed and conducted over the years can be positioned as contributions to software segmentation and software dynamisation. Using these results, I propose to develop an agent-based approach to process simulation based on the CAPE-OPEN standard. I complement this by looking at how a few selected TRIZ principles and patterns of evolution can be utilized in the domain of software architecture and engineering.



# Table des matières

<b>GLOSSAIRE</b> .....	<b>1</b>
ABRÉVIATIONS TECHNIQUES, DE LOGICIELS ET DE PROJETS .....	1
NOMS D'ORGANISMES ET DE SOCIÉTÉS .....	2
<b>1. INTRODUCTION</b> .....	<b>5</b>
<b>2. UN RÉSUMÉ DE MES RECHERCHES, ET LEUR RAPPORT AVEC TRIZ</b> .....	<b>9</b>
2.1 TRIZ, THEORIE DE LA RESOLUTION INVENTIVE DE PROBLEMES .....	9
2.2 SEGMENTATION ET DYNAMISATION .....	11
2.3 MON PARCOURS PROFESSIONNEL .....	12
2.4 TABLEAU HISTORIQUE ET THÉMATIQUE .....	21
<b>3. SEGMENTATION: CAPE-OPEN, APPROCHES À COMPOSANTS</b> .....	<b>25</b>
3.1 LES LOGICIELS POUR L'INGÉNIERIE DE PROCÉDÉS ET LES BESOINS EN MATIÈRE D'INTÉROPÉRABILITÉ .....	27
3.2 CAPE-OPEN ET GLOBAL CAPE-OPEN .....	31
<b>4. DYNAMISATION: APPRENTISSAGE, ÉVOLUTION ARTIFICIELLE, AGENTS INTELLIGENTS</b> .....	<b>45</b>
4.1 DES RÉFÉRENCES ESSENTIELLES .....	46
4.2 LA DYNAMISATION: COMMENT? .....	48
<b>5. UNE APPLICATION/UN PROJET: LES COGENTS, AGENTS LOGICIELS INTÉROPÉRABLES POUR LA SIMULATION DE PROCÉDÉS</b> .....	<b>59</b>
5.1 MOTIVATIONS ET IDÉES DIRECTRICES DE COGENTS .....	59
5.2 LE PROJET "COGENTS" .....	62
5.3 L'ADAPTATION DANS COGENTS .....	68
5.4 PERSPECTIVES, DE COGENTS À E-WOK .....	71
<b>6. CONCLUSION</b> .....	<b>73</b>
<b>7. ANNEXES</b> .....	<b>75</b>
<b>ANNEXE 1: TRIZ, SES PRINCIPES ET LOIS D'ÉVOLUTION, ET L'INGÉNIERIE DU LOGICIEL</b> .....	<b>77</b>
LES 40 PRINCIPES DE TRIZ (D'APRÈS [ALTSHULLER 00]) .....	79
TRANSPOSITION DE QUELQUES PRINCIPES .....	82
TRANSPOSITION DES LOIS D'ÉVOLUTION .....	89
<b>ANNEXE 2: LE STANDARD CAPE-OPEN</b> .....	<b>97</b>
<b>ANNEXE 3: CURRICULUM VITAE</b> .....	<b>108</b>
<b>8. RÉFÉRENCES</b> .....	<b>110</b>
8.1 THÈSES ET SÉJOURS POSTDOCTORAUX .....	110
8.2 PUBLICATIONS PERSONNELLES .....	112
8.3 RÉFÉRENCES CITÉES DANS CE MÉMOIRE .....	121

## Table des illustrations

FIGURE 1: LE SCHÉMA «SEGMENTATION» .....	11
FIGURE 2: HISTORIQUE .....	23
FIGURE 3: COIN INFÉRIEUR DROIT DE LA MATRICE DES CONTRADICTIONS .....	26
FIGURE 4: UN ENVIRONNEMENT DE MODÉLISATION DE PROCÉDÉS .....	32
FIGURE 5: ARCHITECTURE TYPIQUE D'UN SIMULATEUR SÉQUENTIEL-MODULAIRE .....	33
FIGURE 6: ARCHITECTURE TYPIQUE DES OUTILS ORIENTÉS ÉQUATIONS .....	34
FIGURE 7: VISION D'UNE SIMULATION TYPIQUE "CAPE-OPEN" .....	34
FIGURE 8: ARCHITECTURE "CONVENTIONNELLE" À BASE DE COMPOSANTS POUR LA MODÉLISATION DE PROCÉDÉS .....	60
FIGURE 9: UNE ARCHITECTURE DÉCENTRALISÉE POUR LA MODÉLISATION DE PROCÉDÉS .....	60
FIGURE 10: LES TROIS NIVEAUX DE COGENTS .....	64
FIGURE 11: ARCHITECTURE CONCEPTUELLE DE COGENTS .....	65
FIGURE 12: ARCHITECTURE DE COGENTS EN MODE CONFIGURATION .....	66
FIGURE 13: ARCHITECTURE TECHNIQUE DE COGENTS EN MODE SIMULATION .....	68
FIGURE 14: UN MODÈLE COMPLET, VU COMME RÉSEAU DE COGENTS .....	70
FIGURE 15: FEEDBACK DANS LE SISMONAUTE .....	85
FIGURE 16: ARCHITECTURE CORBA .....	86
FIGURE 17: INTÉGRATION FLUENT/ASPEN PLUS .....	92
FIGURE 18: ÉVOLUTION DES LOGICIELS .....	93
FIGURE 19: CAPE-OPEN INTERFACES FOR PROCESS MODELLING COMPONENTS .....	99
FIGURE 20: CAPE-OPEN INTERFACES FOR SIMULATION EXECUTIVES (OR PROCESS MODELLING ENVIRONMENTS) ...	102
FIGURE 21: CAPE-OPEN COMMON SERVICES INTERFACES .....	103

## Remerciements

Mes premiers remerciements vont à **Jean-François Perrot** qui a bien voulu être le promoteur de ce dossier d'habilitation auprès de l'université Pierre et Marie Curie. Bien que nous connaissant depuis plusieurs années par l'AFIA, ce n'est que récemment, tout d'abord au sein du groupe OFTA "Architectures Logicielles et Réutilisation de Composants", puis dans le cadre de ce mémoire, que nous avons réellement collaboré. Je te remercie, Jean-François, de m'avoir accordé ta confiance, dont je sais qu'elle s'accompagne d'une exigence de qualité et de correction, et j'espère avoir répondu à tes attentes en la matière.

Mon confrère et ami **Marc Schoenauer** ne se doutait pas, lorsque nous devisions nonchalamment au bord de la piscine du Dolphin Hotel à Disneyworld lors d'une conférence IEEE sur l'*intelligence computationnelle*, qu'il aurait quelques années plus tard à juger de mon mémoire d'habilitation à diriger des recherches, et qu'il prendrait ma succession à la présidence de l'AFIA! A côté d'excellents moments passés ensemble dans moult conférences sur l'évolution artificielle, je suis redevable à Marc de ses conseils éclairés en optimisation non paramétrique dans le cadre de plusieurs projets de recherche liant l'IFP au Centre de Mathématiques Appliquées de l'Ecole Polytechnique. Je sais que notre collaboration continuera dans le nouveau cadre du projet Fractales de l'INRIA, que Marc a rejoint récemment.

**Michel Gondran** avait dirigé mes travaux de thèse de 1994 à 1998. J'avais écrit pour le remercier, à l'époque: "Michel possède une intelligence encyclopédique de tout ce qui touche aux méthodes formelles pour les mathématiques et l'informatique, et j'ai apprécié son souci de clarté, de formalisation, et sa constance dans l'approfondissement du détail qui peut faire tout l'intérêt scientifique et technique d'une recherche". Je maintiens! Ces dernières années, nous nous sommes retrouvés sur le sujet des composants logiciels, et cette nouvelle rencontre a été, je crois, une des clés du succès du groupe de travail OFTA que j'ai cité plus haut. Merci encore, Michel, de continuer à accompagner mon parcours académique, dans lequel tu as été omniprésent.

Mon parcours professionnel récent à l'IFP s'est fait sous la direction d'**Alain Bamberger**. C'est grâce à toi, Alain, que ce chemin s'est accompagné d'une démarche de reconnaissance académique que tu as constamment soutenue, pour ne pas dire plus! Les ingénieurs de la Division Informatique Scientifique Mathématiques Appliquées de l'IFP savent à quel point tu es soucieux d'une telle reconnaissance, puisque plusieurs d'entre eux ont effectué la même démarche de doctorat ou d'habilitation, avec une certaine réussite. C'est maintenant la recherche de l'ENSPC qui va bénéficier de tes compétences et de ton énergie. Je te souhaite, bien entendu, tout le succès que tu mérites dans cette nouvelle aventure, et je sais aussi que nous nous retrouverons sur des actions communes.

Lorsque j'ai parlé pour la première fois à **Michel Combarrous** de mon projet personnel, il m'a immédiatement encouragé dans cette voie, acceptant de consacrer un peu de son temps précieux à me fournir des pistes et des conseils que j'ai bien entendu utilisés du mieux que j'ai pu. Michel Combarrous a constamment soutenu les recherches en informatique que j'ai proposées, dans son rôle de président puis de membre du conseil

scientifique de l'IFP, donc c'est aussi un peu grâce à lui que ce mémoire a pu voir le jour. Michel, je suis très honoré que tu aies accepté de faire partie de ce jury, je t'en remercie tout particulièrement.

Tous ceux qui ont eu l'occasion de travailler avec **Wolfgang Marquardt** ont été impressionnés par sa très grande intelligence et son expertise dans les domaines de la modélisation de procédés, de l'informatique et des mathématiques appliquées (entre autres...). En quelques années, Wolfgang a créé, au sein de l'université d'Aix-la-Chapelle, un groupe de recherche en modélisation de procédés de réputation internationale, et lui a donné une vision et des technologies qui placent clairement ce groupe à la pointe des recherches mondiales en "*Computer-Aided Process Engineering*". J'ai été très chanceux de pouvoir travailler avec Wolfgang et son groupe dans les projets CAPE-OPEN et GCO, cette collaboration se poursuit aujourd'hui avec bonheur dans COGents, elle renforce une amitié qui dépasse le cadre professionnel.

**Michel Pons** a été mon compagnon de route CAPE-OPEN ces dernières années. Je prends un risque calculé en l'invitant dans ce jury. Je sais que pas une erreur, pas une ambiguïté, pas une faute d'orthographe, pas une phrase par trop évasive, n'échapperont à son proverbial souci de perfection. Merci Michel (encore un Michell!), de m'avoir aidé à réaliser de grandes choses dans nos projets communs, et à bientôt, pour d'autres aventures de CAPE et d'épée...☺

**Jean-Yves Jaffray**, président de la commission des thèses de l'Ecole Doctorale en Informatique de Paris 6, a immédiatement accepté le principe de ce mémoire d'habilitation lorsqu'il lui a été proposé par Jean-François Perrot. Je le remercie pour cet acte fondateur, et j'espère que mon mémoire figurera parmi les "succès de librairie" de cette commission!

**Marcel Dubourg** et **Marie-Thérèse Feld** m'ont offert l'hospitalité pour me permettre de préparer ce document dans le cadre reposant de leur "home" béarnais avec vue sur le Pic du Midi d'Ossau. Merci à vous deux, et merci au temps typique de la région paloise au printemps (pluie incessante...) qui m'a aidé à me concentrer sur le sujet, en me cachant les Pyrénées!

**Carole Guénaud** et **Catherine Lefebvre-Dubin**, nos deux secrétaires, m'ont bien aidé dans la préparation de ce document, l'ensemble du personnel de la DISMA a de la chance de pouvoir bénéficier de leur efficacité et de leur bonne humeur.

Je remercie **Jean-Marc David**, camarade de l'AFIA et d'INAO, pour l'enthousiasme qu'il a démontré lorsque je lui ai présenté les idées de ce mémoire.

Enfin, bien entendu, mon amour va à **Anita Giraud** et à **Sonia Braunschweig**, cette dernière continuant à me faire aveuglément confiance dans mon métier de chercheur en IS&MA - tout en me remettant à ma place dans mon rôle de père...

J'aurais aimé que **Bernard Papaz** assiste à ma soutenance. Je me souviens de son élégance, de son constant souci de qualité, de l'attention qu'il portait envers ses proches, et des *single malt* que nous aimions partager en écoutant du bon jazz.

Les cartes mentales de ce mémoire ont été réalisées avec le logiciel **MindManager 2002 Business Edition**.





## Glossaire

Ce mémoire utilise de nombreuses abréviations. J'ai essayé de les définir dans la plupart des cas lors de leur première utilisation, sauf pour les plus courantes, supposées connues du lecteur.

### Abréviations techniques, de logiciels et de projets

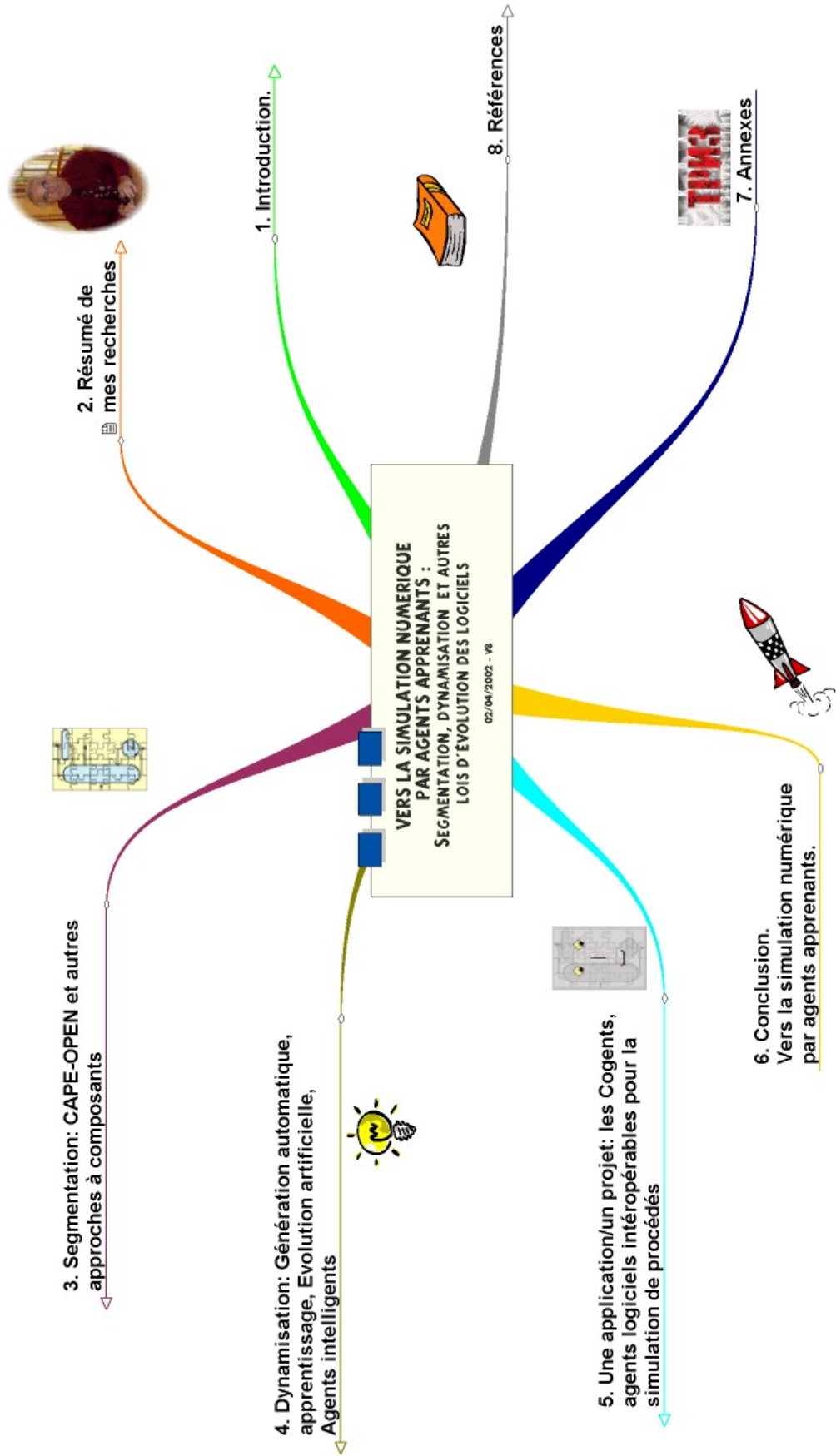
3G	Géologie, Géophysique, Gisement
ACL	Agent Communication Language (de FIPA)
AI	(IA en français) Artificial Intelligence
API	Application Programming Interface
ARIZ	(en russe) algorithme de résolution inventive de problèmes
ASP	Application Service Provision
ATN	Augmented Transition Network
CAO	Conception Assistée par Ordinateur
CAPE	(IPAO en français) Computer Aided Process Engineering
CBR	(RàPC en français) Case-Based Reasoning
CFD	Computational Fluid Dynamics
CHEM	projet européen de développement d'une architecture de supervision de procédés
CMP	Composant de Modélisation de Procédés
COGents	CAPE-OPEN Agents
COM	Component Object Model (intergiciel de Microsoft)
CORBA	Common Object Request Broker Architecture (intergiciel de l'OMG)
DAE	Differential Algebraic Equations
DAML+OIL	Darpa Agent Markup Language + Ontology Interchange Language
DIMA	plate-forme multi-agents du groupe OASIS
DTD	Document Type Definition (pour XML)
EA	Evolution Artificielle
EAI	Evolution Artificielle d'Individus Immortels
EIAO	Environnement Interactif d'Apprentissage avec Ordinateur
EJB	Enterprise Java Beans
EMP	Environnement de Modélisation de Procédés
ESO	(pas d'équivalent français) Equation Set Object, Objet Ensemble d'Equations
e-WOK	Energy, Web, Ontologies, Knowledge
IA	Intelligence Artificielle
IDEFO	standard graphique de représentation de diagrammes d'activité
INAO	Innovation Assistée par Ordinateur
KADS	méthode européenne d'acquisition de connaissances
LAE	Linear Algebraic Equations
MAS	(SMA en français) Multi-Agent System
MILASI	MIcro LAngage de SIMulation
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
MO	(pas d'équivalent français) Material Object, Objet de Matière
MOF	Meta Object Facility
NLAE	Non Linear Algebraic Equations
OntoCAPE	Ontology for CAPE
PDAE	Partial Differential Algebraic Equations
PERT	Program Evaluation and Review Technique
PMC	(CMP en français) Process Modelling Component
PME	(EMP en français) Process Modelling Environment
RàPC	Raisonnement à Partir de Cas
SMA	Système Multi-Agents

SPIKE	Software Platform for Interactive (chemical) Kinetic Experiments
SSH	Secure SHell
ST	Système Technique
TAXI	The Adjustment Companion for SIMulators
TOPSI	The Overall Project SIMulator
TRIZ	(en russe) théorie de la résolution inventive de problèmes
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
UO	(pas d'équivalent français) Unit Operation, Opération Unitaire
WSDL	Web Services Description Language
WSFL	Web Services Flow Language (déjà en voie de disparition...)
XMI	XML Metadata Interchange
XML	eXtensible Markup Language

## Noms d'organismes et de sociétés

AAAI	Artificial Intelligence Application Institute, Université d'Edimbourg
AFIA	Association Française pour l'Intelligence Artificielle
AI Petro	Artificial Intelligence in the Petroleum Industry
ASTI	Association des Sciences et Technologies de l'Information
Brite-EuRam	Une direction de la Commission Européenne
CAIPEP	Conference on Artificial Intelligence in Petroleum Exploration and Production
CO	CAPE-OPEN
CO-LaN	CAPE-OPEN Laboratories Network
CRIN	Coordination Recherche-INDustrie
DISMA	Division Informatique Scientifique Mathématiques Appliquées de l'IFP
ENSPC	Ecole Nationale Supérieure de Physique Chimie
EuroCAIPEP	European CAIPEP (voir CAIPEP)
FIPA	Foundation for Intelligent Physical Agents
GAIA	Groupe d'Application de l'Intelligence Artificielle (du groupe Elf)
GCO	Global CAPE-OPEN
GROWTH	Une direction de la Commission Européenne
IEEE	Association américaine d'informatique, électronique, électrotechnique, électrique
IFAC	International Federation of Automatic Control
IFP	Institut Français du Pétrole
IK-CAPE	Industrial Cooperation on CAPE (organisme allemand)
IMS	Intelligent Manufacturing Systems (organisme de coopération internationale)
INRIA	Institut National de la Recherche en Informatique et Automatique
IST	Information Society Technologies (une direction de la Commission Européenne)
ITMI	Industrie et Technologie de la Machine Intelligente
LIP6	Laboratoire d'Informatique de l'Université Paris 6
NITRD	Networking and Information Technology Research and Development
OASIS	Objet et Agents pour les Systèmes d'Information et de Simulation, groupe du LIP6
OFTA	Observatoire Français des Technologies Avancées
OMG	Object Management Group
RNTL	Réseau National des Technologies Logicielles





# 1. Introduction

Ce mémoire s'appuie sur vingt-cinq ans (!) de recherches et développements effectués dans l'industrie pétrolière, dont une petite douzaine au sein du groupe Elf Aquitaine, suivis par une grande douzaine à l'Institut Français du Pétrole. Au cours de ces années, j'ai eu la chance de travailler sur des sujets divers, toujours passionnants, souvent difficiles, essentiellement liés à la réalisation de logiciels dans le domaine de la simulation de systèmes scientifiques et techniques complexes.

Les travaux que je présente ici ont été réalisés dans trois équipes; tout d'abord, dans le Groupe Dynamique des Systèmes d'Elf Aquitaine, que dirigeait Michel Karsky<sup>1</sup>, où j'ai pu aborder de nombreux problèmes de simulation dynamique de systèmes socio-technico-économiques, tout en concevant et en réalisant des outils génériques permettant de programmer et d'utiliser plus facilement ces simulateurs. J'ai retiré de cette période un intérêt pour la simulation des évolutions temporelles des systèmes, mais aussi une certaine frustration de devoir coder des comportements d'organisations sous forme de lois numériques, parfois un peu "dures" et laissant peu de place au libre arbitre...

Mon séjour dans la deuxième équipe a duré un peu moins de trois ans, mais il m'a permis de me familiariser avec les méthodes et outils de l'IA, en tant que membre du "GAIA", le Groupe d'Application de l'Intelligence Artificielle de la branche Exploration-Production d'Elf Aquitaine. Après une formation initiale à ces techniques par immersion dans une société spécialisée, j'ai continué à apprendre l'IA en compagnie de Michel Bonnel et Dominique Giard, auteurs des premiers programmes d'IA du groupe Elf. Cette période m'a montré l'importance de la modélisation des connaissances (on parlait alors de systèmes experts), notamment sous forme qualitative et symbolique; j'y ai également expérimenté des systèmes d'apprentissage automatique (classification symbolique, réseaux neuronaux), et j'y ai découvert les langages orientés objet.

Enfin, le troisième temps, le plus long, et qui se poursuit, est celui de la Division Informatique Scientifique et Mathématiques Appliquées de l'IFP, la "DISMA", où Alain Bamberger m'avait fait venir pour créer et animer l'activité en Intelligence Artificielle. Cette période a été très riche sur le plan scientifique, puisque - on le verra par la suite - j'ai pu y conduire des recherches sur des sujets et des techniques particulièrement intéressantes et variées, allant de la modélisation des connaissances de supervision de procédés de raffinage à l'identification de modèles géologiques du sous-sol, ou de l'optimisation de paramètres de simulation moléculaire de Monte Carlo à l'intéropérabilité de composants logiciels de simulation, ou encore de l'utilisation de TRIZ pour l'innovation technique, etc. C'est également pendant cette période que j'ai pu occuper quelques responsabilités d'animation de la communauté IA - notamment plus de quatre ans de présidence de l'AFIA - et quelques fonctions de coordination de projet - notamment dans les consortiums CAPE-OPEN et Global CAPE-OPEN. Je ne vais pas résumer ici ce que j'ai retiré de cette période, d'une part parce qu'elle est encore

---

<sup>1</sup> toujours très actif, puisqu'il vient de publier un ouvrage de référence sur la systémique et son application concrète à des sujets industriels et sociaux [Donnadieu 02].

en cours, mais aussi et surtout parce que c'est le sujet central de ce mémoire, et je vais en développer le contenu maintenant!

Le **deuxième chapitre** présente un résumé un peu plus étendu de mes recherches - en donnant plus de poids aux travaux les plus récents - et fait le lien entre mes idées sur l'évolution du logiciel et les éléments correspondants de la théorie TRIZ, les principes (ou lois d'évolution) de segmentation et de dynamisation. Il renvoie à l'annexe 1 qui donne plus de détails sur TRIZ et qui présente une analyse, non systématique, mais sélective, de cette théorie et de sa transposition au domaine du logiciel.

Au **troisième chapitre**, je présente de manière synthétique un exemple concret de **segmentation** du logiciel de simulation numérique, l'exemple du standard CAPE-OPEN pour l'interopérabilité des composants de modélisation de procédés (CAPE = Computer-Aided Process Engineering). Ayant coordonné les projets européens et internationaux de définition de ce standard de 1996 à 2002, j'en connais bien les aspects techniques, industriels et commerciaux; la présentation que j'en fais est volontairement assez générale, l'annexe 2 donnant plus de détail. Ceci étant, ces deux textes - le chapitre 3 et l'annexe 2 - ne résument que partiellement les centaines de pages de spécifications techniques et d'explications scientifiques sur le standard, que le lecteur intéressé trouvera sans difficulté dans l'ouvrage que j'édite cette année avec mon collègue Rafiqul Gani "Software Architectures and Tools for Computer-Aided Process Engineering", ou dans la liste de publications que je donne en référence, ou encore sur le site web du CAPE-OPEN Laboratories Network, [www.colan.org](http://www.colan.org).

Le **quatrième chapitre** est consacré à la **dynamisation**. Après une introduction générale sur les moyens de donner un comportement dynamique ou adaptatif à des logiciels, et après avoir positionné ces comportements dans la mouvance actuelle des technologies de l'information, je donne plusieurs exemples concrets de dynamisation pris parmi des études et recherches développées pour la plupart dans le groupe IA de l'IFP. Les cinq techniques que je mets en valeur dans ce chapitre sont l'apprentissage automatique, le raisonnement par cas, l'évolution artificielle, la métamodélisation, et les systèmes multi-agents. Ce ne sont pas les seules manières de dynamiser un logiciel, mais ce sont celles sur lesquelles nous avons le plus travaillé sur des applications "pétrolières".

Enfin, au **cinquième chapitre**, je propose une recherche intégrant les deux aspects que sont segmentation et dynamisation, dans le cadre des agents CAPE-OPEN que j'appelle "**COGents**". L'idée centrale du projet européen COGents est de confier à des agents logiciels le soin de sélectionner et de configurer automatiquement les composants logiciels nécessaires à la simulation d'un procédé, ceci au moyen de connaissances de modélisation spécifiques du domaine de l'ingénierie de procédés et du standard CAPE-OPEN. Le sujet COGents nous permettra d'examiner comment la définition d'une ontologie de domaine peut contribuer à la mise au point de meilleurs codes de simulation; dans un deuxième temps, il nous donnera également l'occasion d'expérimenter quelques aspects de l'apprentissage automatique pour améliorer la performance du modèle. Je conclus ce chapitre en offrant une perspective supplémentaire, celle de l'intégration de tels systèmes au sein du Web Sémantique.

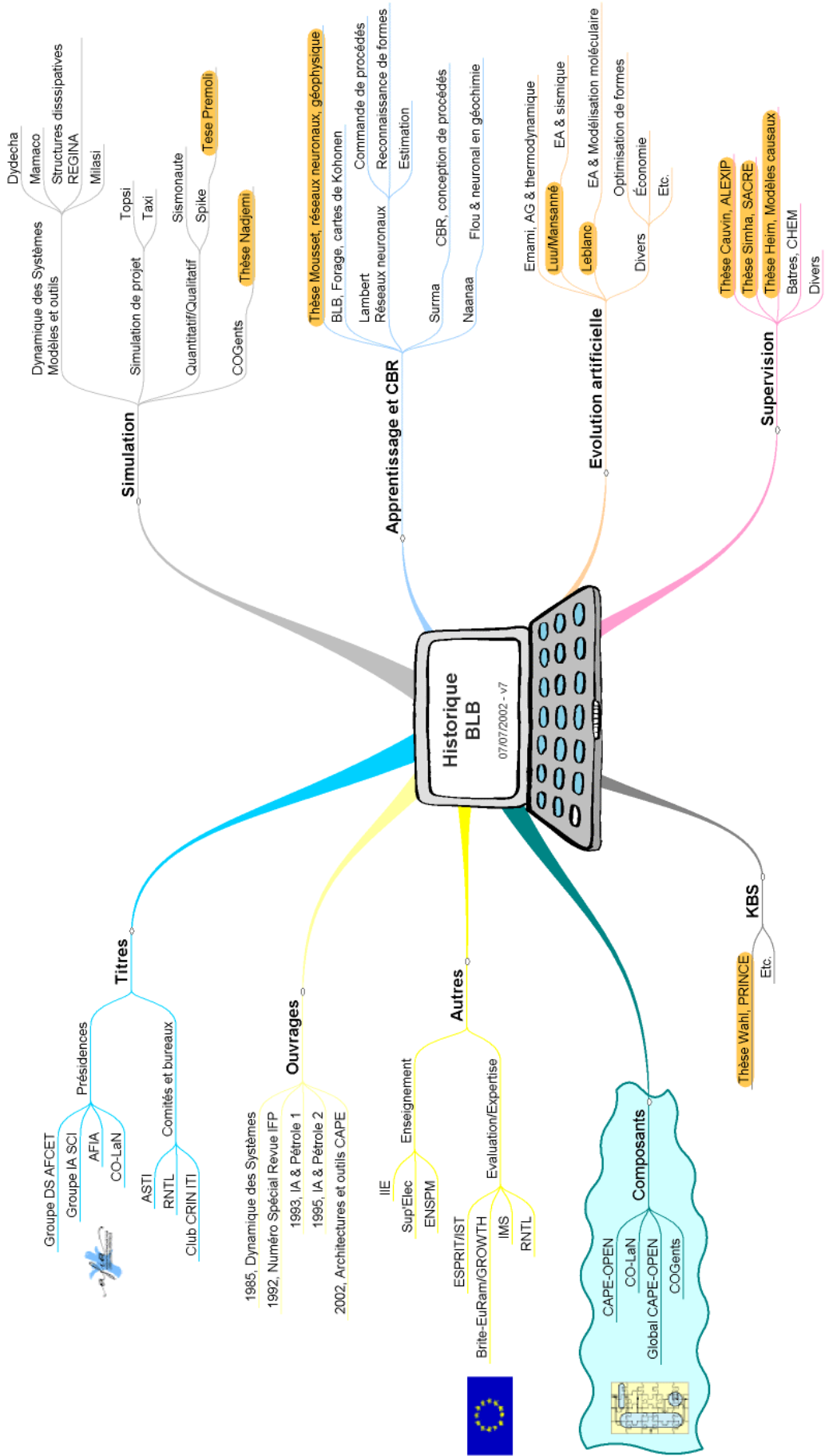
Ma conclusion, au chapitre 6, est que les perspectives qu'offrent segmentation et dynamisation du logiciel sont immenses, mais que la quantité de travail à réaliser pour cela est à la hauteur de l'objectif!

Note au lecteur:

Pour lever les ambiguïtés dans mes références bibliographiques personnelles, l'année de chaque référence est suivie d'une lettre indiquant son type: t (thèse), p (post-doc), L (livre), r (article de revue), C (conférence), m (mémoire), o (cape-open), D (divers). En cas d'occurrences multiples, les références sont complétées par a,b,c,d etc.

Ainsi, [Braunschweig 010a] est une référence d'une publication cape-open de 2001.

Ce système s'applique uniquement aux deux premières parties de la section 8.





## 2. Un résumé de mes recherches, et leur rapport avec TRIZ

La carte mentale ci-contre, reprise plus loin sur la figure 2, est un résumé de 25 ans de carrière professionnelle de chercheur depuis 1977. Il serait exagéré de dire que j'ai identifié dès le début les thèmes de la segmentation et de la dynamisation comme objets centraux de mes recherches. Disons plutôt que ces thèmes apparaissent comme un fond sur lequel un certain nombre de réalisations ont pu s'appuyer. La majorité de ces travaux a pour objet d'intérêt l'informatique scientifique et technique, et plus particulièrement la simulation numérique sous toutes ses formes. J'y reviendrai.

La présentation de mes travaux fait référence aux principes de segmentation et de dynamisation de TRIZ. Je commence donc par une introduction à TRIZ et à ces deux principes pour faciliter la compréhension de la suite, qui commence en page 12.

### 2.1 TRIZ, théorie de la résolution inventive de problèmes

TRIZ, théorie développée par le russe Genrich Altshuller [Altshuller 54, 96, 00] [Salamatov 99], repose sur une analyse des processus d'innovation dans les domaines techniques (physique, chimie, mécanique, technologie etc.). Altshuller a mis au point une palette d'outils : les 40 principes, les tendances d'évolution des technologies, les tables de résolution de contradictions, etc.

L'hypothèse de TRIZ est que les innovations reposent sur des principes universels d'invention, et que ces principes, une fois identifiés et formalisés, permettent de mieux maîtriser les processus d'invention. Au cours des 50 dernières années, Altshuller et ses disciples ont examiné plus de 2 millions de brevets, les ont classés selon le niveau d'innovation apporté, et les ont analysés pour rechercher des principes.

Ces principes sont aujourd'hui mis en œuvre dans des logiciels commerciaux proposés par quelques sociétés, dont par exemple Invention Machine avec TechOptimizer, ou Ideation avec Ideation WorkBench (IWB).

Les principaux outils de la théorie TRIZ sont:

- les 40 principes; ces principes indiquent des règles générales à suivre pour résoudre des problèmes techniques. J'en donne la liste en annexe, et je reviendrai sur une douzaine d'entre eux au chapitre 4, appliqués au logiciel.
- la matrice 39X39 de résolution des contradictions techniques; cette matrice, dont les 39 lignes représentent des paramètres des systèmes techniques que l'on cherche à améliorer sans en détériorer d'autres représentés dans les 39 colonnes, indique, à chaque intersection d'une ligne et d'une colonne, ceux des 40 principes qui sont statistiquement les plus utilisés pour résoudre les contradictions associées. Ainsi, si l'on cherche à améliorer la résistance d'un objet sans en augmenter le poids, la matrice indique que les principes 1 (segmentation) , 8 (contreponds), 15 (dynamisation) ou 40 (matériau composite) sont couramment employés.
- les huit lois d'évolution des systèmes techniques; ces lois sont des indications très générales et à long terme sur le devenir de systèmes techniques, comme la *loi de*

*coordination du rythme des parties* - qui indique que les systèmes évoluent vers une plus grande coordination, la *loi de transition vers le supersystème* - qui indique que lorsqu'un système ne peut plus évoluer, c'est son supersystème, qui l'englobe, qui prend le relais, ou la *loi d'augmentation du dynamisme et de la contrôlabilité* - qui indique que les systèmes deviennent mobiles, adaptatifs, tout en gagnant en contrôlabilité.

- la représentation en modèles substances-champs (vépoles) et les 76 standards; cette représentation sous forme de graphes liant des *substances* ou des *champs*<sup>2</sup> ("vépole" en étant phonétiquement l'abréviation en russe) permet à Altshuller de développer une analyse fine des formes typiques d'interactions entre ces éléments dans les systèmes techniques, et d'en dégager des lois sur les configurations opérantes.
- l'algorithme ARIZ d'application de la théorie TRIZ, dont plusieurs versions successives existent; cet algorithme instrumente une démarche systématique d'innovation en enchaînant plusieurs éléments de TRIZ et en les complétant par des étapes de questions servant à formuler les objectifs recherchés et les problèmes à résoudre. Deux versions anciennes d'ARIZ sont longuement décrites dans [Altshuller 00], et une version plus récente peut être consultée dans [Cavalucci 99].
- enfin, selon la société Invention Machine et ses concurrents, les bases d'effets scientifiques comme celle du logiciel TechOptimizer peuvent être considérées comme faisant partie de la théorie. Ces bases d'effets - 10 000 dans la version 4 de techOptimizer - sont en quelque sorte une encyclopédie scientifique structurée pour les besoins de l'innovation. Les bases d'effets contiennent aussi des animations, des références bibliographiques et des exemples de brevets.

Mon propos n'est pas ici de présenter tous ces outils, pas plus que de présenter la théorie TRIZ en tant que telle. Elle est très bien décrite dans les ouvrages d'Altshuller et de ses disciples, dans les versions originales en russe ou dans leurs traductions anglaises; la thèse de Denis Cavalucci [Cavalucci 99] est sans doute le meilleur document en français actuellement disponible sur le sujet. Mon propos est d'examiner si TRIZ peut être appliquée, au moins dans son principe, au domaine du génie logiciel, domaine qui n'avait pas été initialement abordé par Altshuller, ne serait-ce que parce que ce domaine n'existait pas au moment où les bases de la théorie ont été établies.

En dehors de la transposition de TRIZ à des domaines scientifiques initialement non couverts par Altshuller (biologie, chimie par exemple), des tentatives d'application dans des domaines non techniques ont été réalisées par quelques auteurs [TRIZ 02]. Je citerai les domaines de l'organisation, de la gestion personnelle, ou même de la politique. J'ai moi-même expérimenté l'algorithme de simplification des schémas fonctionnels sur le problème de l'organisation de la production de logiciels dans le secteur "3G<sup>3</sup>" de l'IFP. Les résultats ont été tellement iconoclastes que je n'ai pas souhaité les présenter aux principaux acteurs de ce secteur...et vous ne les verrez pas non plus dans ce mémoire. Je me contente ici de prendre comme centre d'intérêt l'objet technique qu'est le logiciel lui-même, ainsi que le génie logiciel, mais non l'organisation de son processus de production.

---

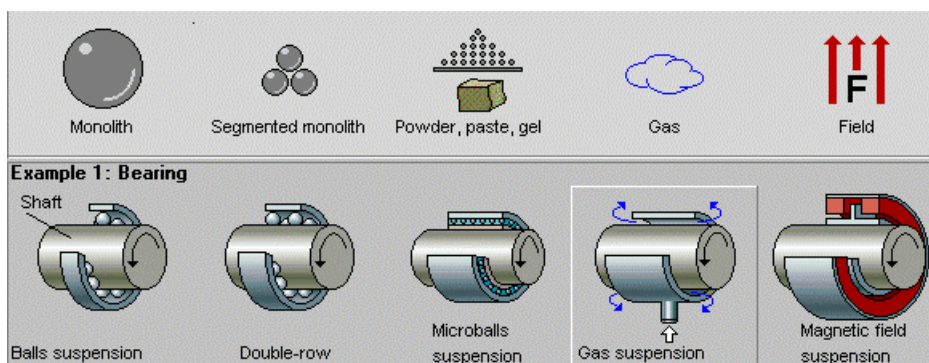
<sup>2</sup> substance et champ sont pris au sens courant de la physique

<sup>3</sup> Géologie, Géophysique, Gisement

## 2.2 Segmentation et dynamisation

Les deux tendances d'évolution "Segmentation" et "Dynamisation" s'appliquent au moins partiellement à l'évolution des logiciels.

La tendance "Segmentation" décrit l'évolution des systèmes monolithiques vers des systèmes segmentés, puis en poudre ou liquide, puis gazeux et finalement à base de champs. La figure 1, tirée du logiciel TechOptimizer, illustre ce schéma.



**Figure 1: le schéma «Segmentation»**

Appliqué au logiciel, le schéma se traduit immédiatement. La première étape est la réalisation des grands systèmes monolithiques. C'est ce que les informaticiens ont développé pendant des années, aussi bien pour la gestion que pour les codes scientifiques.

La deuxième étape est le découpage de ces codes en petits morceaux que l'on assemble. C'est la situation actuelle, l'industrie du logiciel est en train de vivre cette transition. La modularité, les objets, les composants, relèvent de cette phase. Cette évolution s'est faite progressivement. On a d'abord séparé les données des traitements grâce aux bases de données dans les années 1970-1980. Les données pouvaient alors évoluer indépendamment des traitements. Puis, on a écrit les interfaces pour les rendre indépendantes des traitements et des données. Cette évolution s'est concrétisée par la multiplication des architectures « *n-tier* » au sens anglo-saxon de « tier », « niveau » ou « étage », dont les plus courantes sont les architectures « *3-tier* », données/application/interface.

Les étapes suivantes donnent lieu à des changements de substance. La matière des logiciels évolue jusqu'à les faire disparaître dans un « fluide » ou un « champ logiciel » dont la nature reste encore mystérieuse à nos yeux. A ce niveau, j'introduis la tendance de dynamisation:

- 1 Permettre aux caractéristiques d'un objet, d'un environnement, ou d'un processus, de changer pour s'optimiser ou trouver une condition opératoire optimale  
Exemple: des feux de circulation dont la période s'adapte au trafic ou au temps.
2. Diviser un objet en parties en mouvement les unes par rapport aux autres.  
Exemple: Saab a conçu un cylindre fait de deux parties en mouvement relatif, permettant de modifier le volume de la chambre de combustion.
3. Si un objet ou un processus est rigide ou inflexible, le rendre souple ou flexible.  
Exemple: un 4x4 dont la garde au sol change en fonction de la route

Elle ajoute la notion centrale d'auto-adaptation, que l'on voit effectivement apparaître en ce moment dans les logiciels. Ceci provient notamment des besoins de personnalisation de services sur le web ou sur les systèmes mobiles : téléphones, cartes à puces, organiseurs de poche, par exemple doivent pouvoir reconfigurer leurs logiciels en fonction de leur environnement. Les technologies les plus récentes (EJB, JINI) visent à procurer cette flexibilité par l'intégration et la configuration de composants en temps réel sur une application en fonctionnement, de même que l'on peut brancher « à chaud » un périphérique sur un PC ou un Mac.

En complément du support middleware, qui peut être vu comme fournissant des capacités d'auto-adaptation grâce aux possibilités d'intégration dynamiques de composants, la recherche en logiciel poursuit deux pistes principales de dynamisation, qui sont d'ailleurs complémentaires: celle de la génération automatique de composants, d'une part, celle des agents logiciels et de l'évolution artificielle, d'autre part. L'action « disappearing computer » et le thème « ambient intelligence » du cinquième Programme Cadre de la Commission Européenne, ou la création d'une division «Pervasive Computing» (informatique diffuse) chez IBM illustrent également cette tendance au changement de phase<sup>4</sup> des logiciels. La machine à millions de neurones reconfigurables d'Hugo de Garis [de Garis 02] est un exemple de réalisation matérielle/logicielle que l'on pourrait commencer à qualifier de « composants en poudre ». Enfin, bien entendu, les approches de vie artificielle entrent également dans le cadre de la « pulvérisation » des logiciels. Mais on est encore loin de pouvoir imaginer ce que seraient les étages ultérieures de l'évolution, je laisse au lecteur la réflexion sur les « logiciels en phase gazeuse » et les « champs de logiciels » !

Les chapitres 3 et 4 de ce mémoire donnent des exemples de segmentation et de dynamisation de logiciels. Je ne m'étendrai donc pas plus sur ce sujet ici.

## 2.3 Mon parcours professionnel

Mon intérêt pour l'évolution des logiciels scientifiques est indéniablement lié à mon parcours professionnel. J'en donne ci-après un résumé "historique", positionnant un certain nombre de ces travaux comme relevant des processus de segmentation (**S**) et de dynamisation (**D**) des logiciels de simulation numérique:

### 2.3.1 Période "Dynamique des Systèmes"

Mes premières années de carrière ont été consacrées à la réalisation de modèles de systèmes dynamiques, ainsi qu'à la conception et au développement d'outils logiciels destinés à faciliter le processus de modélisation. Utilisant en particulier la Dynamique des Systèmes [Forrester 61,68,71], [Donnadieu 02], branche "opératoire" de la systémique qui s'appuie sur des modèles de simulation numérique des systèmes à étudier, mes travaux ont porté initialement sur l'optimisation de profils de production de champs pétroliers, combinant la recherche statique d'un optimum économique de

---

<sup>4</sup> pris au sens de phase solide, phase liquide etc.

production en univers incertain, à la dynamique de l'évolution des connaissances sur un champ pétrolier au fur et à mesure de son exploitation.

L'extension naturelle de ces travaux, à savoir l'optimisation globale de la production d'un bassin pétrolier, m'a conduit à traiter le problème d'allocation optimale de ressources d'exploration pétrolière (sismique et forage). A cette époque, nous travaillons étroitement avec l'équipe du Pr. Prigogine à l'Université Libre de Bruxelles, qui appliquait la théorie des structures dissipatives, initialement consacrée aux systèmes chimiques autocatalytiques [Glansdorff 71], à des systèmes socio-économiques à feedback [Schieve 82]. J'avais alors montré [Braunschweig 85r] que le comportement d'une mission d'exploration pétrolière pouvait être bifurcatoire selon un facteur décisionnel traduisant la réactivité des géologues par rapport aux informations en leur possession.

Prolongeant ces recherches sur les bifurcations dans les systèmes économiques, nous avons alors appliqué les mêmes principes, sur la base d'une fonction d'attractivité, à des modèles de marchés de produits, en l'occurrence le marché des matériaux composites (projet "MAMACO") qui intéressait la branche Chimie du groupe Elf Aquitaine.

Les programmes de simulation que nous développiions fonctionnaient initialement sur des *mainframes* en *batch*. Nous avons développé des fonctions interactives au-dessus de ces outils, mais elles restaient relativement limitées et réservées aux quelques spécialistes du langage utilisé. Avec l'arrivée des micro-ordinateurs, il devint possible de porter les programmes de simulation sur PC, leur confiant ainsi une bien plus grande interactivité. C'est dans ce contexte que j'ai réalisé le système MILASI (MIcro LAngage de SIMulation).

**D MILASI (1980-1983):** conception d'un langage de simulation dynamique par génération automatique du code simulé à partir d'un modèle déclaratif; utilisation de fonctions spéciales autorisant la modification interactive du programme pendant son exécution; malheureusement toute trace écrite de ce travail a disparu au cours de mes déménagements successifs.

Mon ouvrage sur la dynamique des systèmes [Braunschweig 85], qui avait pour objectif d'expliquer comment réaliser des programmes de simulation dynamique sur PC, s'appuyait sur une version très simplifiée de MILASI.

#### Note Technique

Dans MILASI (Micro Langage de Simulation), le modèle dynamique était écrit dans un langage proche du langage le plus utilisé en Dynamique des Systèmes, DYNAMO. Le système générait à partir du modèle un programme exécutable en SBASIC, langage disponible sur les micro-ordinateurs Goupil. La partie "dynamisation" concernait les fonctions d'édition interactives du programme, permettant par exemple de modifier le modèle et ses paramètres en cours de calcul, ou de changer dynamiquement les sorties graphiques. Ceci était obtenu par une utilisation intensive de l'instruction "Execute", analogue à l'instruction "eval()" de Lisp, et plus généralement disponible dans les langages interprétés. "Execute" permettait d'exécuter à la volée une chaîne de caractères, composée en fonction des besoins de la simulation.

### 2.3.2 Période "Topsi"

Cette deuxième période a été essentiellement consacrée à l'aide à la décision pour les projets grâce à la simulation dynamique. Nous nous sommes appuyés sur l'expérience de collègues nord-américains sur le système PMMS - Program Management Modelling System [Cooper 80], pour appliquer ces concepts et réaliser un environnement de simulation dédié à la prévision des coûts et des délais sur les grands projets *offshore*, un sujet qui intéressait au plus haut point la branche Exploration-Production d'Elf Aquitaine. Cet environnement de simulation, TOPSI (The Overall Project Simulator), partait des informations conventionnelles de planification gérés par les systèmes PERT [Chauveau 91], mais y ajoutait des comportements dynamiques, fonctions de divers facteurs organisationnels et humains, pour prédire dynamiquement la productivité et la qualité pour chaque "macro-tâche" du projet. Les idées directrices de TOPSI sont présentées dans [Braunschweig 88r, 87r, 90ra]. Après deux applications du logiciel en conseil sur les développements des champs pétroliers de North Alwyn et d' East Frigg, récompensées par une prime aux innovateurs du groupe Elf Aquitaine en 1987, nous avons entrepris d'appliquer les mêmes idées aux développements de projets informatiques [Abdel-Hamid 89]. Cette nouvelle utilisation était plus risquée, en raison de la différence de taille - de plusieurs ordre de grandeur - entre les projets de construction de plates-formes en mer (plusieurs milliers d'années-homme, plusieurs milliards de F.), et les projets de développement de logiciels (quelques dizaines d'années-hommes, quelques millions de F.). Il était plus délicat d'appliquer des lois statistique de productivité et de qualité, à des objets d'intérêt de plus petite taille.

- ◆ **D** TOPSI/TAXI (1981-1987): conception du système de simulation dynamique de grands projets, TOPSI, d'une interface permettant de générer semi-automatiquement le code de simulation à partir d'éléments fournis par l'utilisateur, et du système expert TAXI de diagnostic de simulation, permettant d'expliquer le comportement du système au cours du temps. [Braunschweig Papaz 87r, 88r, 90ra] [de Saint Etienne 87, 88C];

#### Note Technique

Les idées de MILASI ont été reprises dans TOPSI, mais implémentées très différemment. Le code de simulation Topsi était un grand programme écrit en Fortran, mais son interface d'utilisation avait été réalisée avec l'outil SYMPHONY de Lotus. Cet outil, précurseur de Microsoft Excel, permettait à l'utilisateur de paramétrer le code, ainsi que de visualiser les résultats. Le système expert TAXI effectuait un diagnostic de la simulation pour déterminer les éventuelles erreurs d'entrée des données. TAXI, écrit en LELISP, n'était toutefois pas connecté directement au simulateur, il se contentait de dialoguer avec l'utilisateur qui décrivait les problèmes rencontrés.

C'est à la suite des premiers essais d'application des concepts de TOPSI aux projets de développements logiciels que j'ai rejoint les équipes d'intelligence artificielle d'Elf Aquitaine, qui commençaient s'intéresser également à ce sujet, mais sous une autre forme.

Après une petite dizaine d'années consacrée à la simulation dynamique dans des domaines non conventionnels, j'étais convaincu de l'intérêt de cette approche pour l'aide à la décision. Nous avons prouvé, sur plusieurs sujets, que la dynamique des systèmes pouvait apporter un éclairage nouveau sur des systèmes industriels/organisationnels complexes, notamment de par sa capacité à simuler de

nombreux scénarios. J'étais par contre moins convaincu par la représentation de comportements sociaux ou humains par des lois numériques (équations ou relations tabulées), étape nécessaire à la construction des équations différentielles des modèles, mais qui forçait à chiffrer "en dur" des connaissances "molles" sur les objets d'intérêt de la modélisation. C'est pour cette raison que la période suivante, au GAIA, fut très enrichissante.

### 2.3.3 Période GAIA

Après une formation technique initiale à la conduite de projets de R&D en intelligence artificielle [Farreny 87], mon séjour au sein du Groupe d'Application de l'Intelligence Artificielle fut principalement consacré à des problèmes de planification automatique ou semi-automatique. Les techniques de planification automatique commençaient à produire des résultats, comme le fameux STRIPS [Fikes 93], ou les projets que l'Artificial Intelligence Application Institute de l'Université d'Edimbourg réalisait pour la NASA [Currie 91]. La collaboration avec l'AI AI sur le sujet de la planification des projets informatiques n'a pas donné de résultats exploitables; par contre, j'ai été impliqué dans une collaboration avec la société ITMI sur le problème de planification de campagnes de forage en mer. Le programme NAVAL utilisait les concepts mis au point par Yannick Descottes sur la résolution de systèmes de contraintes [Gama 90] pour produire des plans optimisant l'utilisation des appareils de forage, compte tenu de contraintes techniques. L'algorithme cherchait un plan minimisant le nombre de contraintes violées - on parlait plutôt de "conseils" que de contraintes, chaque conseil étant doté d'une valeur indiquant son caractère, obligatoire ou simplement souhaitable.

Le séjour au GAIA, les contacts avec mes collègues, et plus généralement la connaissance de l'ensemble des projets et activités du groupe, furent extrêmement formateurs. En plus des projets de développement de systèmes à base de connaissances, utilisant des outils du commerce ou des moteurs d'inférences développés en interne, nous étions également *branchés* sur les recherches en IA, ayant la possibilité d'expérimenter rapidement les nouvelles techniques dont nous entendions parler. C'est ainsi que nous avons pu tester les idées balbutiantes de la physique qualitative [de Kleer 84][Forbus 90 91][Hayes 85][Kuipers 94][Weld 90], programmer les premiers réseaux de neurones sur des problèmes jouets [Kosko 88], ou développer des systèmes de classification automatique inspirés de l'algorithme ID3 [Quinlan 93].

A la fin de ce séjour au GAIA, j'avais acquis la conviction que le domaine de la simulation dynamique pouvait bénéficier des technologies de représentation des connaissances développées en IA, et qu'il était donc possible de concevoir des systèmes de simulation combinant une modélisation numérique par équations différentielles, telle que pratiquée en Dynamique des Systèmes, à une représentation symbolique à base de connaissances, telle qu'issue de la physique qualitative ou des systèmes experts. Ce sont les idées que je présentais dans ma communication intitulée "Pour une intelligence des systèmes dynamiques", au Congrès Européen de Systémique [Braunschweig 91ra]:

"Cet article présente un cahier des charges pour un logiciel d'intelligence des systèmes dynamiques. Un tel système doit reposer sur une description structurale hiérarchisée, basée sur une représentation objet. Il doit fournir plusieurs options pour la modélisation de relations entre variables : des équations quantitatives lorsque cela est possible, des règles de production ou des équations qualitatives lorsque seule une connaissance symbolique est disponible, des réseaux de

neurones - utilisés comme mémoires associatives - lorsque l'on dispose de banques d'exemples. Il doit pouvoir être utilisé à des fins de simulation, de diagnostic, de contrôle, de compréhension. Il doit être capable d'expliquer son raisonnement. Il permettra de créer des modèles lisibles et maintenables."

### 2.3.4 Période initiale IFP

Rejoignant la Division Informatique Mathématiques Appliquées de l'IFP en 1989, ma première mission, préparatoire à la constitution d'un groupe de compétences en intelligence artificielle, fut de faire une étude sur les applications de l'IA dans le monde pétrolier, me donnant ainsi une vision incomparable des réalisations de l'industrie pétrolière dans le domaine à la fin des années 80. A cette époque, la majorité des recherches se portait sur des systèmes experts [Feigenbaum 88], plus ou moins sophistiqués, complétés par les premiers essais industriels de réseaux neuronaux. Le résultat de cette étude a été publié sous plusieurs formes, dont [Braunschweig 90rb, 91rb, 92r]. Il a servi de support à l'animation des conférences sur l'IA dans l'industrie pétrolière, initialement appelées "CAIPEP (Conference on AI in Petroleum Exploration and Production)", puis "EuroCAIPEP" pour la version européenne dont j'avais pris l'initiative, et finalement "AI Petro" pour sa dernière édition à Lillehammer en 1995.

Forts de cette connaissance, des intérêts et limites de l'IA dans le domaine pétrolier, nous avons pu lancer plusieurs études et recherches pour divers secteurs (ou "objectifs" selon la terminologie officielle) de l'IFP, en collaboration avec Michel Gondran [Gondran 93]. Touchant tous les secteurs d'application de l'IFP ("amont": exploration et production; "aval": raffinage et pétrochimie; "énergie-moteurs"), et mettant en œuvre des techniques variées : systèmes à base de connaissances, réseaux neuronaux, algorithmes d'évolution artificielle, raisonnement qualitatif, logique floue, raisonnement par cas, etc., ces études et recherches ont été présentées de manière synthétique dans les numéros 3, 10, et 19 du bulletin de l'AFIA [voir section correspondante de la bibliographie]. Je retiens ici un certain nombre de travaux relevant des processus de segmentation et de dynamisation.

- ◆ **S PRINCE** (1990-1995, thèse de François Wahl): conception d'un système d'aide à la préparation de propositions industrielles pour des procédés de raffinage, découpant le processus en un ensemble de tâches élémentaires, et enchaînant ces tâches de manière opportuniste en fonction des informations et données disponibles, en interaction avec l'utilisateur [Wahl 93+] [Wahl 95C];

#### Note Technique

Prince, développé par François Wahl, collaborateur du groupe Intelligence Artificielle de l'IFP, a été développé avec l'outil Smeci d'Ilog, et les interfaces Aïda et Masai de la même société, le tout au-dessus du langage LeLisp. L'idée essentielle de Prince était de s'intéresser aux tâches élémentaires de conception (calcul d'une propriété, détermination d'une configuration, accès à des données etc.) et d'enchaîner dynamiquement ces tâches en fonction des besoins. Pour ce faire, un moteur d'inférence spécifique orienté tâche avait été réalisé, dans la lignée des travaux de Chandrasekaran [Chandrasekaran 93], ou de ceux de Jutta Willamowski à l'INRIA, sous la direction de François Rechenmann [Willamowski 94]. La notion de segmentation provient ici du découpage d'un processus vu auparavant comme monolithique, en un ensemble variable d'actions élémentaires combinables. Ce système a reçu le prix du meilleur système d'IA appliqué à un procédé, prix attribué par le jury de la Société de Chimie Industrielle.



- ◆ D Thèse de E. Mousset, recherches de E. Emami et travaux de J.M. Lambert (1989-1993). Etude et appropriation des techniques d'apprentissage et d'optimisation: réseaux neuronaux, algorithmes génétiques, sur divers problèmes intéressant l'IFP: filtrage des signaux sismiques, commande de procédés, modélisation thermodynamique des fluides de gisement, caractérisation de réservoir etc. [Mousset 94+] [Braunschweig 91Cb] [Lambert 91] [Braunschweig 95Ca];

#### Note Technique

Parmi ces travaux, je souligne ceux de E. Emami, sous ma direction, consacrés à l'ajustement de modèles thermodynamiques; et ceux que j'ai réalisés en collaboration avec D. Pavone sur la caractérisation de signaux de forage. Le premier cité est la première application des algorithmes d'évolution artificielle (EA) dans l'IFP. A l'époque, ne disposant pas des bibliothèques publiques d'EA qui existent aujourd'hui [Schoenauer 95] [Keijzer 01] [Lutton 01] [Collet 00], nous avons dû développer l'ensemble du code, en C++. L'ajustement de la représentation de la partie lourde d'un gaz à condensats consistait à rechercher une combinaison en masse de trois molécules ou "pseudo-composants" pris parmi un ensemble de molécules candidates. Le problème comportait donc une partie discrète, le choix des pseudo-composants, et une partie continue, leur répartition en pourcentage de masse. Les résultats obtenus avec cette approche étaient particulièrement convaincants [Braunschweig 91Ca]. Le deuxième travail concerne l'utilisation des cartes topologiques [Kohonen 93 97], une forme particulière de réseaux neuronaux, pour la classification automatique non supervisée de signaux de forage [Braunschweig 95Ca]. Les morceaux de signal obtenus avec l'outil Trafor, prétraités et transformés en fréquentiel, servaient de données d'apprentissage à une carte auto-organisatrice de Kohonen - en général 10x10 - destinée à classer le signal en plusieurs catégories (normal, collage de l'outil de forage au fond, rebond, etc...); j'avais mis au point un traitement de la carte résultant de l'apprentissage, pour produire automatiquement quelques macro-classes regroupant en sous-ensembles homogènes les 100 classes "élémentaires" d'une carte 10x10, par l'analyse des pics et des vallées de la distribution des "individus" morceaux de signal sur la carte. L'ensemble était programmé en SN-Lisp, langage proposé par la société Neuristique.

- ◆ D SPIKE (1995-1998, stage *post-Laurea* de P. Arosio puis *Tese di Laurea* de G. Premoli). Réalisation d'un système d'aide à l'interprétation des simulations dynamiques de cinétique chimique, comportant la génération automatique de la simulation et de son interface, des outils d'analyse de simulation et de visualisation des cycles dominants [Arosio 95a, 95b, 96C, 97] [Premoli 97+]. Ce travail a été longuement décrit dans ma thèse [Braunschweig 98m] dont il constitue une partie importante.

#### Note Technique

La partie de ce travail dont je suis particulièrement satisfait concerne l'identification et la visualisation dynamique des cycles dominants, épisode par épisode. Pour résumer, le détail étant dans ma thèse, une fois effectué le découpage en épisodes de la simulation, sur la base des sens de variation des principales grandeurs et de leurs dérivées, SPIKE analysait automatiquement les cycles dominants au moyen d'un calcul de gains de cycles, et utilisait cette information pour modifier dynamiquement le graphe 2D représentant le modèle cinétique à l'écran: ce graphe se déformait en fonction de l'intensité des cycles, rapprochant espèces et réactions en forte interaction, et éloignant espèces et réactions contribuant peu à l'évolution du système. L'utilisateur pouvait ainsi immédiatement visualiser les mécanismes principaux de l'évolution du système chimique. Les outils de développement de SPIKE étaient C++ pour la saisie des mécanismes et la visualisation, et Matlab pour la simulation.

- ◆ **D et S** Thèse de Benoît Leblanc (1998-2002); utilisation d'algorithmes d'évolution artificielle pour optimiser les fréquences de mouvements de Monte Carlo en simulation moléculaire. La partie "dynamique" de ce travail est dans l'optimisation en ligne des paramètres de la simulation en fonction de son déroulement; la partie "segmentation" réside dans la distribution de la simulation sur une grappe de PC et dans l'application de l'algorithme EAI (Evolution Artificielle d'Individus Immortels) sous une forme client de simulation -serveur d'individus [Leblanc 01Ca, 01Cb, 02Ca, 02Cb].

Note essentiellement non technique

Plutôt que d'expliquer les aspects segmentation et dynamisation de ce travail, que j'ai abordés brièvement au paragraphe ci-dessus, je préfère reproduire les commentaires émis par H. Toulhoat, responsable de la modélisation moléculaire à l'IFP, qui donnent une idée de l'intérêt des résultats obtenus:

"Je voudrais souligner en quelques mots les principaux apports originaux de la thèse, de mon point de vue, dans le domaine de la simulation moléculaire: Conception et comparaison de plusieurs critères de mesure de la thermalisation d'un polymère dense amorphe, dont le critère multi-échelles de lacunarité (Chapitre 3); Démonstration du concept d'optimisation "à la volée" des fréquences et paramètres des mouvements Monte Carlo, le temps calcul consacré à l'échantillonnage étant intégralement préservé, et finalement sensiblement diminué à iso convergence. (Chapitre 4); Ouverture d'une voie de comparaison automatique de l'efficacité des différents mouvements Monte Carlo en compétition. (Chapitre 4); Ouverture d'une voie d'optimisation automatique en thermalisation parallèle (Chapitre 7). Je ne doute pas que ces nouveaux outils entreront rapidement dans la pratique générale en simulation moléculaire."

- ◆ Je n'inclus pas dans ces catégories **S** et **D** un certain nombre de recherches liées seulement indirectement au sujet de ce mémoire, comme tous les travaux autour de la supervision de procédés - essentiellement pilotés par Sylvie Cauvin, qui se développent actuellement dans le projet international CHEM que j'ai initié [Cauvin 95t] et [nombreuses références dans ma bibliographie personnelle]. J'aurais également pu mentionner l'utilisation de raisonnement à base de cas pour la conception (séjour post-doctoral de J. Surma) [Surma 96p] [Surma 96C], l'interprétation de simulations de propagation d'ondes par raisonnement qualitatif et à base de modèles (séjour post-doctoral de U. Junker) [Junker 94p] [Junker 94], et le travail de W. Naanaa lors de son séjour post-doctoral sur l'interprétation de résultats de pyrolyses par réseaux neuronaux et logique floue [Lafargue 98d].

### 2.3.5 Période CAPE-OPEN/COGents

Quelques-uns des travaux sur les applications de l'intelligence artificielle dans l'IFP me menèrent à m'intéresser de près à la représentation des connaissances sur les procédés de raffinage et de pétrochimie. Plusieurs études conduites à l'IFP traitaient de ce domaine:

- Prince, présenté ci-dessus;
- Sacre, système d'aide au contrôle de résultats expérimentaux, appliqué au procédé de réformage catalytique, un des procédés importants du raffinage [Ferraz 93t];

- Repro, système d'aide à la conception de procédés, utilisant le raisonnement par cas [Surma 94p];
- Alexip, environnement à base de connaissances pour la supervision de procédés [Cauvin 95t], et ses divers prolongements, dont le projet européen CHEM [CHEM 02].
- etc.

Ces expériences m'ont également conduit à participer, puis à assurer la présidence du Club Systèmes Experts de la Société de Chimie Industrielle, club d'échanges entre industriels français de la chimie s'intéressant aux applications de l'IA, et qui organisait, entre autres, le concours annuel du meilleur système expert appliqué à un procédé. Ce club était étroitement associé au groupe Informatique et Procédés de la même société. Par extension, lorsque l'IFP a été invité à participer à un groupe de réflexion français sur la représentation par objets pour le génie chimique [Bourseau 95a, 95b], j'ai été impliqué dans les travaux de ce groupe, qui ont donné aux participants la conviction que l'approche objet était le futur des logiciels dans ce domaine. Les travaux de ce groupe reposent sur les projets IFP cités plus haut, sur les recherches de Patrick Bourseau sur la représentation des connaissances de l'ingénieur de procédé décrites dans sa thèse d'état [Bourseau 93t], et celles de Laurent Jourda sur la simulation de procédés "100% objet", décrites dans sa thèse de doctorat [Jourda 93, 96, 96t].

D'autres groupes Européens et internationaux, principalement au sein de laboratoires de "CAPE" (*Computer-Aided Process Engineering*, Ingénierie de Procédés Assistée par Ordinateur), étudiaient ce même sujet. L'équipe Ecosse de Jack Ponton à l'Université d'Edimbourg, développait *Épée*, Ecosse Process Engineering Environment, environnement orienté objet de modélisation [Fraga 94]; l'équipe d'Art Westerberg à Carnegie-Mellon University, développait ASCEND, un environnement de simulation par équations, [Piela 91] [Westerberg 94]; George Stephanopoulos, au MIT, avait proposé plusieurs versions de l'environnement Model.La et du système Design-Kit [Stephanopoulos 87, 90]; le nouveau Lehrstuhl Für ProzessTechnik de Wolfgang Marquardt, à l'Université d'Aix-la-Chapelle, développait également des outils utilisant la représentation objet [Marquardt 92, 96] [Jarke 95][Bogusch 01]; des recherches identiques avaient lieu au Japon, au Tokyo Institute of Technology, sous la direction du Pr. Naka [Batres 99].

De manière complémentaire, le besoin d'interopérabilité des logiciels de modélisation de procédés devenait de plus en plus présent, en raison de la complexité croissante des applications de modélisation, et de l'incompatibilité des principaux logiciels de modélisation utilisés dans l'industrie. Au chapitre 3, je présente plus en détail ce besoin, qui était également décrit dans l'article fondateur de Herb Britt et Costas Pantelides à la conférence FOCAPD 94 [Pantelides 94].

La réunion du besoin d'interopérabilité et du potentiel de la programmation objet ainsi que des technologies de *middleware* (intergiciel) apparues au cours des années 90, aboutit au lancement par l'industrie de procédés européenne d'une action stratégique visant à définir une architecture ouverte de modélisation, devant aboutir à un standard appelé CAPE-OPEN. Après quelques études préliminaires en 1995 [Brice 95], j'ai pris la responsabilité de cette action, notamment comme coordonnateur de deux projets européens et internationaux successifs, CAPE-OPEN puis Global CAPE-OPEN. Cette

responsabilité m'a aussi conduit à présider le "CO-LaN" , CAPE-OPEN Laboratories Network, depuis début 2001, et à lancer le projet IST COGents.

- ♦ **S** et **D** CAPE-OPEN, GLOBAL CAPE-OPEN, CO-LAN, COGENTS (1996-). Approche de la simulation de procédés chimiques par composants logiciels intéropérables, définition de standards d'intéropérabilité, extension vers les agents logiciels de simulation numérique. Ces projets font l'objet des chapitres 3 et 5 de ce mémoire. [nombreuses références dans la section "CAPE-OPEN" de ma bibliographie personnelle];

Pas de note technique, voir les chapitres 3 et 5

Cette période CAPE-OPEN, qui n'est pas terminée, a été particulièrement enrichissante, tant sur le plan technique, que sur celui de la collaboration avec les leaders de la recherche mondiale en ingénierie de procédés assistée par ordinateur. Techniquement, nous avons utilisé les outils et méthodes les plus à jour en matière d'ingénierie de logiciel: approche objet, UML, middleware, modélisation XML etc., le tout dans un domaine technique particulièrement exigeant.

Après les projets de définition du standard CAPE-OPEN, qui est désormais en exploitation commerciale, je reviens aujourd'hui sur la représentation des connaissances de modélisation de procédés, dans le cadre du projet COGents dont je parle en détail au chapitre 5. Pour le moment, je me contente de souligner les bases techniques, particulièrement actuelles, de COGents: approche multi-agents et utilisation du standard FIPA, développement d'une ontologie de la modélisation de procédés en DAML+OIL pour réaliser l'intéropérabilité au niveau sémantique, le tout au-dessus du standard CAPE-OPEN...

### 2.3.6 Doctorants et postdoctorants dont j'ai eu la responsabilité

#### Doctorants

- Claudia Ferraz-Simha. Thèse soutenue en 1993, collaboration avec Gilles Muratet, Laboratoire d'Ingénierie des Matériaux et Hautes Pressions: SACRE, Système d'aide au contrôle de résultats expérimentaux [Ferraz 93t]
- François Wahl. Thèse soutenue en 1993, collaboration avec Michel Gondran, ENS Ponts et Chaussées. Un environnement d'aide aux ingénieurs basé sur une architecture en tâches et sur un module de visualisation de courbes. Application à la conception de procédés de raffinage, [Wahl 93t]
- Eric Mousset. Thèse soutenue en 1994, collaboration avec Patrick Gallinari, LIP6. L'apport des réseaux connexionnistes à la géophysique. Application au traitement des signaux sismiques. [Mousset 94t]
- Sylvie Cauvin. Thèse soutenue en 1995, collaboration avec Philippe Façon<sup>†</sup>, IIE. Un environnement générique à base de connaissances pour la supervision de procédés de raffinage et de pétrochimie. [Cauvin 95t]

- Giuseppe Premoli. Tesi di Laurea soutenue en 1997, collaboration avec Stefania Bandini, Université de Milan. Contributo allo Studio dei Modelli di Cinetica Chimica attraverso Metodi Simbolici e Numerici. [Premoli 97t]
- Benoît Leblanc. Thèse soutenue en 2002, collaboration avec Hervé Toulhoat, Groupe Modélisation Moléculaire, IFP, et Evelyne Lutton, INRIA. Simulations Moléculaires de Monte Carlo : amélioration de l'efficacité statistique de l'échantillonnage grâce aux algorithmes d'évolution artificielle. [Leblanc 02t].

## Post-doctorants

- Ulrich Junker, séjour post-doctoral en 94-95 sur le "SISMONAUTE", outil d'aide au suivi des fronts d'ondes dans les simulations sismiques, collaboration avec Laurence Nicoletis, Division Géophysique, IFP [Junker 95p] ;
- Jerzy Surma, séjour post-doctoral en 95-96, sur "REPRO", système de raisonnement par cas pour la réutilisation de schémas de procédés, collaboration avec Y. Charlez, Direction Industrielle, IFP [Surma 95p] ;
- Wady Naanaa, séjour post-doctoral en 97, utilisation de réseaux neuronaux combinés à la logique floue pour l'interprétation de pyrolyses pétrolières, co-encadrement avec Eric Lafargue, Division Géologie-Géochimie, IFP, [Lafargue et coll. 98d]
- Rafael Batres, séjour post-doctoral en 2001-2002, contribution à l'architecture et au modèle d'intégration dans le projet européen CHEM, co-encadrement avec Sylvie Cauvin, Division Informatique Scientifique Mathématiques Appliquées, IFP [Batres 02p][CHEM 02]

## 2.4 Tableau historique et thématique

Le tableau est présenté selon une liste de thèmes de recherche, les 6 premières lignes de l'axe vertical, au cours du temps, sur l'axe horizontal. Les trois dernières lignes sont des éléments de c.v. complémentaires donnant quelques responsabilités que j'ai occupées, des publications plus importantes que les autres (ouvrages, thèse), et des activités d'enseignement et d'évaluation. L'axe du temps n'est pas linéaire, afin de donner plus de place à quelques activités. Les travaux que j'ai réalisés moi-même sont en jaune. Les thèses que j'ai encadrées sont en marron foncé, et les thèses que j'ai co-encadrées sont en marron clair. Les séjours post-doctoraux que j'ai encadrés sont en bleu clair, et l'activité "CAPE-OPEN" est représentée en vert.

Le découpage est effectué comme suit:

- Simulation: je regroupe sous ce thème plusieurs projets de simulation numérique ou qualitative de systèmes physiques, techniques, économiques, organisationnels ou sociaux, ainsi que la conception et la réalisation des outils logiciels sous-jacents. Les méthodes et techniques employées pour cela sont la dynamique des systèmes, l'étude des bifurcations, la simulation de projet, la physique qualitative, la cinétique chimique [Eisenberg 91], et plus récemment la simulation multi-agents [Briot 01] [Guessoum 02].

- Apprentissage et raisonnement par cas: je rassemble ici, d'une part, des études utilisant divers modèles de réseaux neuronaux : réseaux perceptrons multicouches [Davallo 89], réseaux dynamiques récurrents [Hérault 94], cartes topologiques auto-organisatrices de Kohonen [Kohonen 89, 93], etc.; d'autre part, les applications à base de raisonnement par cas [Aamodt 94] [Kolodner 93].
- Evolution Artificielle: dans cette rubrique figurent toutes les recherches, essentiellement faites à l'IFP, sur les applications des algorithmes génétiques [Goldberg 89][Schoenauer 95] à divers problèmes intéressant l'industrie pétrolière;
- Supervision : le groupe IA de l'IFP a développé des compétences de pointe en systèmes d'aide au diagnostic et à la supervision de procédés, principalement dans le domaine du raffinage et de la pétrochimie, ainsi que, dans une moindre mesure, appliquées à la supervision de forages et d'essais moteurs; les premiers travaux sont décrits dans la thèse de Sylvie Cauvin [Cauvin 95t], dans celle de Claudia Ferraz-Simha [Ferraz 93t]; plusieurs autres études ont été conduites par la suite. Ces travaux sont en relation avec les principales recherches actuelles sur les méthodes de supervision de procédés, telles que présentées dans [Dague 97], ou dans les travaux du groupe IMALAI, auquel nous participons [Travé Massuyès 02].
- KBS: je regroupe ici les recherches sur les systèmes à base de connaissances [Feigenbaum 88] [Shrobe 88], assez diverses, dont la principale à l'IFP a été le développement de Prince
- Composants: cette partie est introduite au chapitre suivant, il s'agit de CAPE-OPEN, Global CAPE-OPEN etc.







### 3. Segmentation: CAPE-OPEN, approches à composants

Je démontre l'évolution des logiciels vers les composants intéropérables en donnant comme exemple les projets CAPE-OPEN de standards d'interopérabilité pour les codes de simulation de procédés, activité internationale que j'ai coordonnée pendant six ans.

Je prends CAPE-OPEN comme exemple, puisque j'ai principalement travaillé sur les approches à composants dans ce cadre. Ces approches sont aujourd'hui couramment employées aussi bien au sein de l'IFP que dans d'autres secteurs. Je citerai par exemple Open Spirit dans le domaine des applications en amont pétrolier (applications en géologie, géophysique, ingénierie de réservoir), ou SALOME, plate-forme RNTL de simulation numérique multi-physiques.

De nombreux auteurs, dont moi-même, ont écrit pourquoi les approches à composants sont importantes pour le génie logiciel. Je préfère, dans ce mémoire, en référer à TRIZ, en examinant certaines cases de la matrice de contradictions d'Altshuller. Je rappelle ici que cette matrice, dont les lignes représentent des paramètres que l'on cherche à améliorer, sans en détériorer d'autres représentés dans les colonnes, indique, à chaque intersection, ceux des 40 principes qui sont les plus utilisés pour résoudre les contradictions associées.

	27	28	29	30	31	32	33	34	35	36	37	38	39
27 Reliability		32,3 11,23	11,32 1	27,35 2,4	35,2 40,26	-	27,17 40	1,11	13,35 8,24	13,35 1	27,4 28	11,13 27	1,35 29,38
28 Accuracy of measurement	5,11 1,23		-	28,24 22,26	3,33 39,1	6,35 25,18	1,13 17,34	1,32 13,11	13,35 2	27,35 10,34	26,24 32,28	28,2 10,34	10,34 28,32
29 Accuracy of manufacturing	11,32 1	-		26,28 10,36	4,17 34,26	-	1,32 35,23	25,1	-	26,2 18	-	26,28 18,23	10,18 32,39
30 Harmful factors acting on an object from outside	27,24 2,4	28,33 23,26	26,28 10,18		-	24,35 2	2,25 28,39	35,1 2	35,11 22,31	22,19 29,4	22,19 29,4	33,3 34	22,35 13,24
31 Harmful factors developed by an object	24,2 40,39	3,33 26	4,17 34,26	-		-	-	-	-	19,1 31	2,21 27,1	2	22,35 18,39
32 Manufacturability	-	1,35 12,18	-	24,2	-		2,5 13,16	35,1 11,9	2,13 15	27,26 1	6,28 11,1	8,28 1	35,1 10,28
33 Convenience of use	17,27 8,4	25,13 2,34	1,32 35,23	2,25 28,39	-	2,5 12		12,26 1,32	15,34 1,16	32,26 12,17	-	1,34 12,3	15,1 28
34 Repairability	11,1 1,16	10,2 13	25,1	35,1 2,16	-	1,35 11,1	1,12 26,15		7,1 4,16	35,1 13,11	-	34,35 7,13	1,32 10
35 Adaptability	35,13 8,24	35,5 1,1	-	35,11 32,31	-	1,13 31	15,34 1,16	1,16 7,4		15,29 37,28	1	27,34 35	35,28 6,37
36 Complexity of a device	13,35 1	2,26 10,34	26,24 32	22,19 29,4	19,1	27,26 1,13	27,9 26,24	1,13	29,15 28,37		15,1 37,28	15,1 24	12,17 28
37 Complexity of control	27,4 28,8	26,24 32,28	-	22,19 29,28	2,21	5,28 11,29	2,5	12,26	1,15	15,1 37,28		34,21	35,18
38 Level of automation	11,27 32	28,26 10,34	28,26 18,23	2,33	2	1,26 13	1,12 34,3	1,35 13	27,4 1,35	15,24 10	34,27 25		5,12 35,26
39 Capacity/Productivity	1,35 10,38	1,1 34,28	18,1 32,1	22,35 13,24	35,22 18,39	35,28 2,24	1,28 7,19	1,32 10,25	1,35 28,37	12,17 28,24	35,18 27,2	5,12 35,26	

**Figure 3: Coin inférieur droit de la matrice des contradictions**

Les 25 premiers des 39 paramètres de la matrice des contradictions sont très "physiques": poids, longueur, surface, vitesse, pression, force etc. Je préfère utiliser les derniers paramètres, qu'Altshuller a voulu plus génériques: fiabilité, souplesse d'utilisation, adaptabilité, réparabilité, complexité etc.

Le lecteur constatera aisément avec moi l'omniprésence du principe 1, segmentation, dans cette partie de la matrice. Ainsi, par exemple, un examen de la ligne 33, "convenience of use", que je traduis par "souplesse" plutôt que par "confort" d'utilisation, montre que la segmentation permet de rendre l'utilisation plus souple sans pour autant nuire à la précision de fabrication (29), à la réparabilité (34), à l'adaptabilité (35), au niveau d'automatisation (38), à la productivité (39). Sachant que l'approche par composants est, pour le logiciel, l'équivalent de la segmentation pour les objets techniques, cette ligne 33 peut être reformulée de la manière suivante:

*Les approches à composants permettent de rendre plus souple l'utilisation des logiciels, sans nuire à leur qualité (29), à leur maintenance (34), à leur adaptabilité (35), ni à leur coût de développement (39)<sup>5</sup>.*

Le lecteur pourra passer des moments agréables à chercher d'autres analogies dans la table ou dans l'extrait que je publie. J'en cite simplement une deuxième, pour la ligne 27, fiabilité:

*Les approches à composants permettent de réaliser des logiciels plus fiables, comportant moins de bogues, sans nuire à leur qualité (!), à leur maintenabilité, à leur coût de développement, ni les rendre plus complexes.*

Altshuller était sans le savoir un apôtre des composants logiciels!

Il est temps de revenir au sujet central de ce chapitre, CAPE-OPEN. Je reprends essentiellement le texte sur CAPE-OPEN que j'avais rédigé pour l'ouvrage collectif "OFTA Arago 28" [Braunschweig 000a], en le complétant par la mention de développements plus récents.

### 3.1 les logiciels pour l'ingénierie de procédés et les besoins en matière d'interopérabilité

L'utilisation toujours plus intensive des modèles numériques dans toutes les phases du développement de procédés a été l'objet de plusieurs publications récentes. Cet accroissement de la demande en simulation a eu comme contrepartie un accroissement équivalent du nombre de logiciels de modélisation, de plus en plus sophistiqués, et de sources nombreuses et variées. Ainsi, des programmes de modélisation de procédés ont été réalisés par des fournisseurs de logiciel ou de systèmes de contrôle-commande, des laboratoires universitaires et des groupes de modélisation internes aux sociétés opératrices de procédés.

Certains de ces logiciels réalisent des fonctions précises et bien définies comme par exemple le calcul de propriétés physiques, la simulation d'une opération unitaire<sup>6</sup>

---

<sup>5</sup> Je n'ai rien trouvé de frappant pour "level of automation"

<sup>6</sup> Une opération unitaire peut être définie comme un équipement réalisant un traitement particulier, comme par exemple un réacteur, une colonne de distillation ou un échangeur de chaleur. Un schéma de procédés contient plusieurs telles opérations unitaires reliées par des flux de matière, d'énergie et d'information.

particulière, ou la solution d'un type de problème numérique que l'on peut rencontrer en modélisation. D'autres logiciels sont des environnements généraux destinés à faciliter la création de modèles, soit à partir de principes fondamentaux, soit à partir de bibliothèques de modèles existants, soit les deux. Ils permettent ensuite d'effectuer un ensemble de tâches autour du modèle créé, comme la simulation et l'optimisation. Cette deuxième catégorie de logiciels incorpore ou fait appel, en général, à plusieurs programmes de la première catégorie. La distinction entre ces deux catégories, qui n'est cependant pas toujours aussi nette, est importante pour comprendre la suite de ce chapitre. J'appelle la première catégorie "*Composants de Modélisation de Procédés*" (CMP) et la deuxième "*Environnements de Modélisation de Procédés*" (EMP<sup>7</sup>).

Cette offre large en logiciels d'aide à l'ingénierie de procédés (en anglais, *CAPE, Computer-Aided Process Engineering*), est bien entendu un facteur positif pour l'industrie de procédés, et a déjà conduit à des bénéfices importants pour les utilisateurs. L'expérience acquise autour de ces outils a permis de faire un certain nombre de constatations, que je présente ci-après.

♦ Le besoin de modélisation intégrée:

On sait aujourd'hui qu'il est nécessaire d'adopter une démarche intégrée d'ingénierie de procédés, qui puisse prendre en compte toutes les interactions souhaitables. Les composants (CMPs) isolés sont d'utilisation limitée. Ainsi, si un réacteur particulier est au cœur d'une installation, et si on dispose d'un modèle détaillé de ce réacteur, il faut pouvoir coupler le modèle avec la partie simulant les étapes de séparation des fluides, afin de permettre une analyse fine de l'inévitable compromis à faire entre la qualité de la conversion et la sélectivité dans le réacteur, et le coût de la séparation qui suit. Ceci indique que le CMP réacteur devrait pouvoir être incorporé dans un EMP général facilitant le couplage en question.

♦ L'adoption des EMP génériques.

Dans le passé, avec le développement de l'informatique scientifique, la plupart des sociétés de procédés ont développé en interne leur propre EMP, en général sous la forme d'un programme de simulation statique de schémas de procédés modulaires<sup>8</sup>. Cette utilisation de programmes internes, qui a pris naissance dans les années 60 et 70, s'est poursuivie dans les années 80 et, pour certains, même dans la dernière décennie. Il est toutefois devenu de plus en plus difficile à ces sociétés de supporter les importants coûts de maintenance et de mise à jour de ces logiciels, dans un contexte d'évolution des techniques de simulation, des techniques de résolution des grands systèmes numériques que sont les modèles de procédés, et bien entendu des méthodes et outils informatiques associés. La tendance récente a été d'interrompre ces développements "maison" et de les remplacer par des logiciels commerciaux provenant de fournisseurs spécialisés. Cependant, souvent, une partie d'un modèle maison, petite

---

<sup>7</sup> J'utilise ces abréviations françaises, qui sont l'équivalent des termes consacrés en anglais, PMC (Process Modelling Component) et PME (Process Modelling Environment)

<sup>8</sup> Simulation statique = de l'état stationnaire. La distinction entre simulation modulaire et simulation par équations est explicitée plus loin

par sa taille mais très importante par son contenu, par exemple un modèle d'opération unitaire particulier, ou le calcul d'une propriété physique, est susceptible de donner un avantage compétitif à la société qui l'a développée. Dans de nombreux cas, il est nécessaire d'introduire cette partie dans l'EMP externe choisi pour la modélisation.

◆ La fourniture limitée des EMP

Développer un EMP demande des compétences techniques de plus en plus grandes et des ressources très importantes, en raison de la complexité et de la quantité de fonctionnalités demandées à ce genre d'outils. Pour cette raison, le nombre de sociétés de logiciel "CAPE" capables de créer et de maintenir un tel environnement est resté relativement limité. En fait, le marché est pratiquement dominé par les produits de trois fournisseurs mondiaux : Aspen Technology, Hyprotech et Simulation Sciences<sup>9</sup>. Des fournisseurs plus petits présentent des offres couvrant une partie du domaine ; par exemple, en Europe, Process Systems Enterprise, ProSim SA, RSI ou BelSim.

◆ La diversité croissante des utilisations des modèles

Les modèles de procédé ont été traditionnellement utilisés pour les applications principales que sont la simulation statique, la simulation dynamique, et l'optimisation<sup>10</sup>. On reconnaît aujourd'hui le potentiel d'utilisation de ces modèles pour d'autres activités, comme par exemple la validation de données d'ateliers industriels, l'analyse de contrôlabilité, les études de sécurité, la détection des pannes etc.

Chacun de ces quatre facteurs relève d'une certaine logique et traduit le passage d'un domaine technologique naissant vers un état de maturité. Leur combinaison conduit cependant à l'existence de problèmes, qui doivent absolument être résolus faute de quoi les industriels de procédés ne pourront pas bénéficier des extraordinaires progrès techniques réalisés dans le domaine.

1. Il y a abondance de CMPs de bonne qualité. Mais ces CMPs ne seront vraiment utilisés par l'industrie que s'il est facile de les incorporer aux quelques EMPs commerciaux qui dominent le marché.
2. Il y a une forte motivation à réaliser des nouvelles applications autour des modèles de procédés. En général, le principal investissement pour une application est le développement et la validation du modèle lui-même. Nombre de ces modèles existent déjà aujourd'hui, mais sont "enfouis" à l'intérieur d'un EMP et sont par conséquent difficilement accessibles et utilisables pour ces nouvelles applications.

Autrement dit, il est indispensable de disposer de moyens pour (a) incorporer des nouveaux CMPs dans les EMPs existants, et (b) de rendre les modèles incorporés dans les EMPs accessibles à d'autres applications.

---

<sup>9</sup> Depuis la publication de ce chapitre, Aspen Technology a acheté Hyprotech, constituant ainsi un pôle qui représente plus de la moitié du marché mondial.

<sup>10</sup> Simulation dynamique: simulation du comportement du procédé au cours du temps; l'optimisation consiste à rechercher les meilleures conditions opératoires pour un critère (économique, technique, d'environnement) donné.

L'approche conventionnelle utilisée pour le premier problème (a) a été depuis toujours, que chaque vendeur de logiciel fournissait des possibilités d'introduire des CMPs externes dans son EMP, en définissant la nature des liens externes autorisés, son propre protocole d'intégration (une API<sup>11</sup> spécifique), et son propre format de données. Ces fonctionnalités étaient la plupart du temps réalisées par des appels de sous-programmes externes, et des transferts de données par fichiers. Plus récemment, des vendeurs ont adopté les standards *middleware* que sont CORBA [Orfali 96][Siegel 96] [OMG 97] et COM [Microsoft 99] [Rosen 98]. On peut qualifier ces approches de semi-ouvertes dans la mesure où elles permettent de satisfaire au besoin (a), mais de manière spécifique pour chaque fournisseur. Un créateur de modèle doit toutefois réaliser de nouvelles interfaces dès qu'il souhaite faire communiquer son modèle avec un nouvel EMP. Ceci est pénalisant, à la fois en raison du coût important pour développer et maintenir des implémentations multiples d'un CMP, et parce que les fournisseurs d'EMP ne rendent pas nécessairement publiques leurs solutions d'incorporation de logiciels externes.

Le deuxième problème (b) n'est pas correctement résolu non plus à ce jour. S'il est vrai que les principaux fournisseurs d'EMP offrent des facilités pour faire communiquer leurs environnements avec d'autres applications ou programmes - comme par exemple Microsoft Excel ou Visual Basic, il n'est pas possible en général d'extraire les modèles de ces EMPs pour les utiliser dans d'autres EMPs ou d'autres applications. De plus, les mécanismes de communication présentent le même inconvénient que ceux d'intégration des CMPs, à savoir qu'une implémentation différente doit être réalisée pour chaque EMP. Cette difficulté technique a des conséquences importantes sur les aspects industriels et commerciaux associés, puisqu'elle empêche, une fois de plus, de réaliser simplement des nouvelles applications à forte valeur ajoutée.

Il a été donc proposé, et c'est l'objet du reste de ce chapitre, de développer une approche alternative, celle des standards ouverts et de l'interopérabilité. Cette approche vise à identifier les principales catégories de CMPs, et de définir des standards d'interfaces pour chaque catégorie. Dans cette approche: (i) les nouveaux CMPs développés adhéreront directement aux standards proposés, tandis que les CMPs existants seront "encapsulés" afin de présenter ces mêmes interfaces; (ii) les EMPs seront adaptés de manière à pouvoir accueillir des CMPs conformes aux standards. Grâce à ces développements, les ingénieurs de procédés pourront faire cohabiter des logiciels de sources hétérogènes au sein de leurs applications, répondant ainsi aux exigences de modélisation actuelles que j'ai présentées en introduction.

La définition de ces standards ouverts a été l'objectif du projet européen *CAPE-OPEN* et de sa suite internationale *Global CAPE-OPEN*, coopération majeure entre les acteurs principaux du domaine sur plus de cinq ans. Les projets, aujourd'hui terminés, ont produit la version 1.0 des standards d'interfaces appelés "standard CAPE-OPEN" ([Co-LaN 02]. L'annexe 2 de ce mémoire présente en anglais cette version 1.0 du standard.

Dans les sections suivantes, je présente les objectifs de ces projets, ainsi que les principaux concepts architecturaux développés. Cette présentation est suivie par une présentation plus technique des principales interfaces, du processus de développement,

---

<sup>11</sup> API: Application Programming Interface

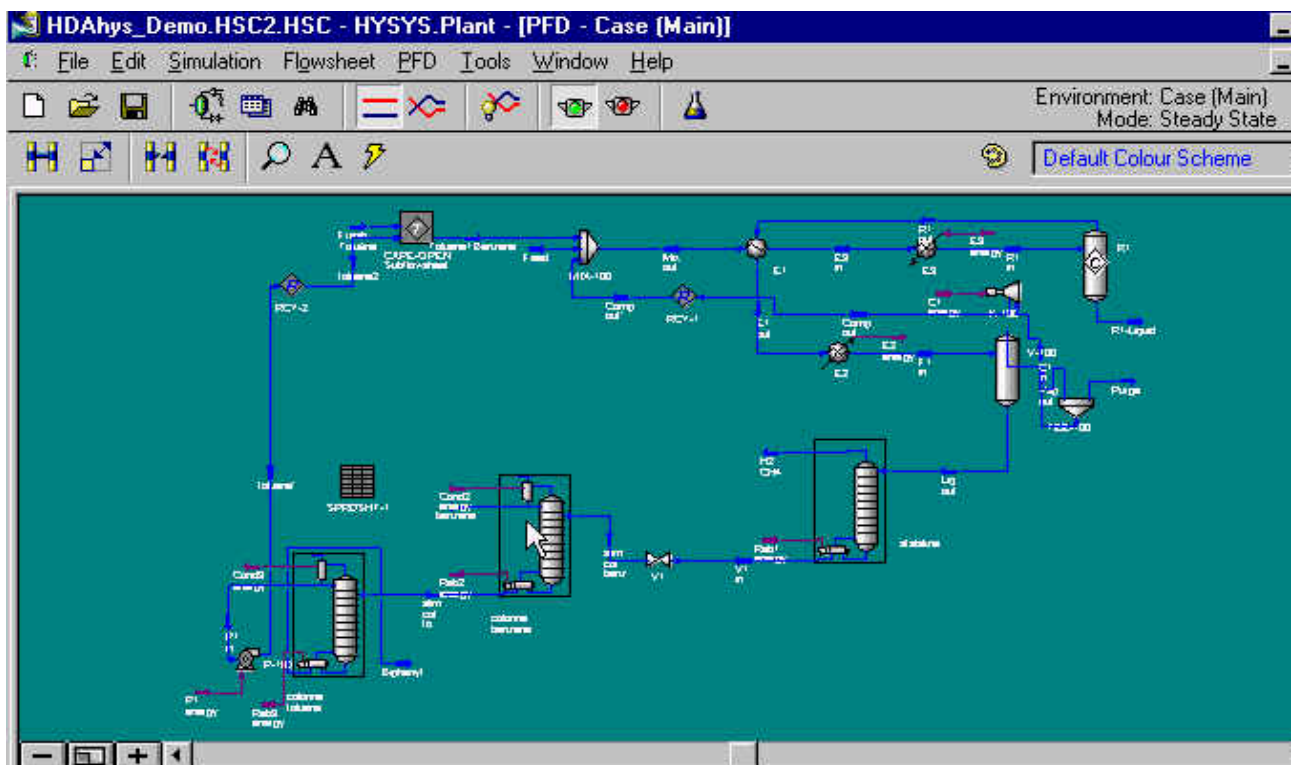
et de l'implémentation au-dessus des *middleware* COM et CORBA. Les aspects organisationnels sont abordés dans la conclusion en section 6.

### 3.2 CAPE-OPEN et Global CAPE-OPEN

Le projet CAPE-OPEN (CO) a réuni pendant deux ans et demi un ensemble de sociétés opératrices et bailleurs de procédés (BASF, BAYER, BP, DUPONT, ELF, ICI, IFP), de laboratoires universitaires (Imperial College, INP Toulouse, RWTH Aachen) et de fournisseurs de logiciels de simulation de procédés (Aspen Technology, AEAT Hyprotech, Simulation Sciences, QuantiSci). Son budget d'environ 3.25Meuros a fait l'objet d'un soutien partiel de la Communauté Européenne dans son programme Brite-EuRam. Il a démarré en 1997 et s'est conclu mi-1999 par la publication de la première version des standards CO. Ses objectifs étaient les suivants:

1. identifier les principales catégories de CMPs, et définir des interfaces pour chacune de ces catégories;
2. réaliser et tester des prototypes logiciels implémentant les interfaces afin de démontrer leur utilisation et d'estimer les bénéfices que l'on peut en attendre;
3. disséminer le plus largement possible les résultats du projet afin de créer les conditions d'acceptation de ces standards dans la communauté CAPE.

### 3.2.1 Domaine traité par le projet CAPE-OPEN

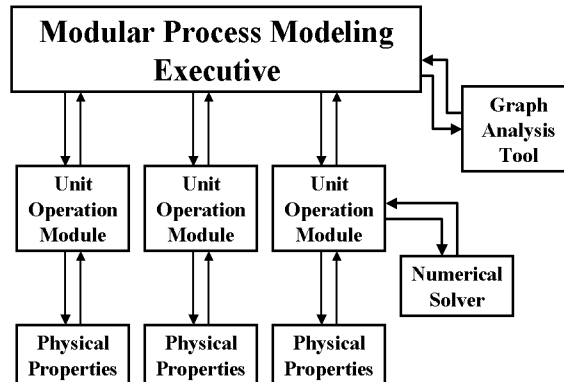


**Figure 4: Un environnement de modélisation de procédés**

Le domaine spécifique traité par CO est celui des outils généraux de modélisation de procédés, et, en particulier, leur utilisation en simulation statique et dynamique. Dans ce domaine, il est couramment admis que les outils peuvent mettre en œuvre l'une ou



l'autre des deux méthodes usuelles de simulation; *séquentielle-modulaire*, et *par équations*. Les architectures typiques correspondant à ces deux méthodes sont différentes et méritent d'être examinées, c'est un préalable indispensable à l'identification des catégories de CMPs dont j'ai parlé. La structure générale d'un outil *séquentiel-modulaire* est présentée sur la figure 5.



**Figure 5: architecture typique d'un simulateur séquentiel-modulaire**

Dans un tel environnement, la responsabilité générale de la construction d'un modèle de procédés, et la gestion des simulations et calculs divers faits avec ce modèle, est confiée à un "Exécutif de Simulation (*process modelling executive*). Ce programme interagit avec des modules décrivant les opérations unitaires (UO) présentes au sein du procédé. Celles-ci, à leur tour, font appel à des modules de calcul de propriétés physiques et thermodynamiques des fluides traités par les opérations unitaires. De plus, chaque opération unitaire doit être capable de résoudre, de manière autonome, les équations du modèle mathématique qui la représente. Cette résolution autonome est le plus souvent codée en interne, elle peut aussi faire appel à un solveur externe, comme représenté sur la figure 2 dans le cas de la troisième UO.

Un aspect important des outils *modulaires-séquentiels* est que l'exécutif de simulation doit organiser et coordonner les calculs effectués par les opérations unitaires. Ceci implique généralement qu'il faut faire une analyse du schéma du procédés afin d'identifier des partitions qui peuvent être résolues de manière indépendante, ou des ensembles de flux qui doivent être "coupés" afin d'éliminer les dépendances cycliques - la méthode de résolution numérique procédant alors par une recherche de convergence aux limites de ces flux coupés. Des outils de séquençement et de partitionnement, basés sur la théorie des graphes, sont utilisés pour ce faire.

L'architecture typique d'un environnement *par équations* est présentée sur la figure 6. La structure générale est assez proche de celle des environnements *modulaires-séquentiels*, mais toutefois avec une différence fondamentale: les modules d'opérations unitaires ne sont plus supposés résoudre leurs propres équations. Ils transmettent ces équations à l'exécutif de simulation qui les réunit dans un système d'équations, en général de grande taille. L'exécutif applique ensuite un ou plusieurs solveurs numériques à cet ensemble d'équations.

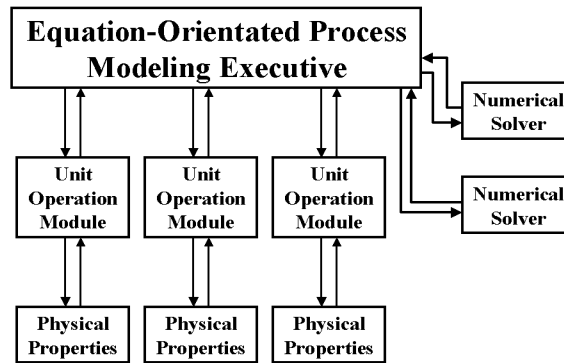


Figure 6: architecture typique des outils orientés équations

Il ressort de cette analyse que les principales catégories de *CMPs* pour lesquelles il faut définir des interfaces standards sont:

1. Les calculs de propriétés physiques et thermodynamiques;
2. Les modules d'opérations unitaires;
3. Les solveurs numériques;
4. Les outils de séquençement et de partitionnement de schémas de procédés;

La vision à long terme de *CAPE-OPEN* est de permettre aux utilisateurs de réaliser des tâches complexes de modélisation de procédés, grâce à l'interopérabilité de tels composants, pouvant provenir de sources diverses, et pouvant s'exécuter sur un réseau hétérogène. Un exemple est présenté sur la figure 7. Dans cet exemple, l'EMP, c'est-à-dire l'exécutif de simulation et son interface utilisateur, sont fournis par un vendeur, alors que certains *CMPs* (une ou plusieurs opérations unitaires, un serveur de calculs de propriétés physiques, un algorithme de résolution numérique, une équation d'état), proviennent d'autres sources. Par extension, les composants mentionnés devraient être capables de communiquer de manière standardisée avec d'autres environnements, par exemple des systèmes de contrôle commande et de supervision, ou des applications de calculs économiques.

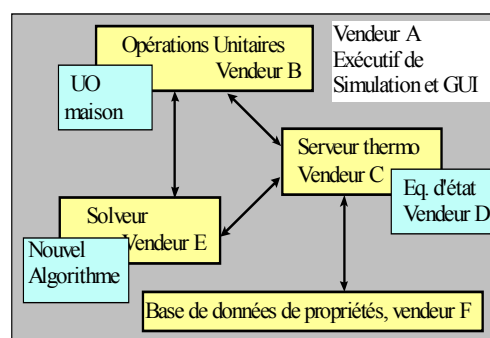


Figure 7: Vision d'une simulation typique "CAPE-OPEN"

### 3.2.2 Les interfaces proposées par CAPE-OPEN

Au regard de la diversité des matériaux et des fluides utilisés dans l'industrie de procédés, de la diversité correspondante des opérations unitaires, et de la diversité également grande des outils et des techniques appliqués à ces objets, il est logique de décomposer les grandes catégories proposées ci-dessus en autant de sous-classes. Les ressources disponibles dans le projet CAPE-OPEN n'ont pas permis de traiter toutes ces classes de composants, mais les principales ont fait l'objet de spécifications d'interfaces. Ce sont les grandes lignes de ces spécifications que je vais présenter dans cette section.

#### ◆ Propriétés physiques et thermodynamiques

CO a défini des interfaces pour les systèmes de calcul de propriétés physiques de fluides homogènes, mélanges de composants purs ou de pseudo composants<sup>12</sup>, et dont la composition molaire est définie. Des interfaces standards ont été produites pour les méthodes de calcul des équilibres liquide-vapeur, ainsi que pour la détermination d'autres propriétés thermodynamiques ou de transport couramment utilisées.

Un important concept introduit est celui de l'Objet de Matière (*Material Object*). Chaque fluide présent dans un procédé sera caractérisé par un objet de ce type. Une opération unitaire communique avec un ou plusieurs *Material Objects*. Ainsi, une unité séparant un fluide en une sortie liquide et une sortie gazeuse, communiquera avec quatre MOs représentant les flux d'entrée et de sortie. Dans la pratique, ces MOs sont reliés à des modules de calcul qui peuvent être configurés au moyen de serveurs de modèles thermodynamiques.

Les interfaces standard comportent l'accès aux *Material Objects*, aux serveurs de calculs thermodynamiques et aux calculs d'équilibre. Elles ont été définies en prenant en considération les contraintes de performance qui sont particulièrement importantes pour des modules qui sont appelés des dizaines de milliers de fois dans une seule simulation. Elles ont également été conçues pour être aisément étendues, car la liste des propriétés calculables pour un fluide donné peut grandement varier suivant les applications.

#### ◆ Opérations Unitaires

Les interfaces d'opérations unitaires sont définies dans le cas des EMPs *modulaires-séquentiels*. Une opération unitaire a un certain nombre de ports qui permettent de la connecter à d'autres modules et d'échanger de la matière, de l'énergie et de l'information avec eux. Dans le cas de l'échange de matière, le port est associé à un MO. Les ports ont un sens : entrée, sortie, entrée-sortie.

Les opérations unitaires ont des paramètres, qui représentent l'information qui ne passe pas par les ports, mais que le module doit pouvoir présenter à ses clients logiciels

---

<sup>12</sup> Utilisés pour représenter les fractions lourdes des charges pétrolières, pour lesquelles on ne dispose pas de la composition détaillée

ou utilisateurs. Des exemples sont les paramètres de conception, comme la géométrie d'un réacteur, ou des quantités calculées par l'UO : rendement, sélectivité, etc.

Une opération unitaire peut avoir sa propre interface graphique qui permet à l'utilisateur de la configurer. Cette fonctionnalité est importante car, même si les interfaces standard prévoient que l'exécutif de simulation est capable de générer un écran d'après la liste des paramètres, il est en général préférable de présenter les données de manière organisée afin d'en faciliter la saisie. Un fournisseur d'opération unitaire compatible CO peut concevoir une interface graphique sophistiquée et mettre cette interface à disposition de ses clients à l'intérieur d'un EMP compatible.

Les autres interfaces (définies Global CAPE-OPEN) s'intéressent à la persistance des UOs, à la production de rapports de calcul, à la vérification de cohérence etc.

Si les interfaces ont été définies dans le cadre *modulaire-séquentiel*, elles s'appliquent pour l'essentiel dans le cas des EMP *par équations*. Une importante contrainte supplémentaire est de pouvoir communiquer l'ensemble des équations au solveur externe. Pour ce faire, nous avons défini des objets "systèmes d'équations" qui sont présentés ci-dessous.

#### ♦ Solveurs

CAPE-OPEN a défini des interfaces pour les solveurs numériques utilisés dans les simulations statiques et dynamiques de procédés. Ceci concerne spécialement les algorithmes de résolution de grandes matrices creuses représentant des systèmes d'équations algébriques et algébro-différentielles ordinaires non linéaires. Ces algorithmes utilisent en interne des résolutions de systèmes creux linéaires, pour lesquels des interfaces ont également été définies.

La difficulté est de bien représenter la grande quantité d'information qui constitue la définition du système à résoudre. Cette quantité augmente, de plus, avec la sophistication croissante des méthodes de résolution. Ainsi, les outils modernes exploitent l'information sur la structure et le remplissage de la matrice, et sont capables de calculer les résidus et les dérivées partielles des équations. On peut aussi avoir besoin de matérialiser les discontinuités du système et les conditions logiques qui donnent lieu à ces discontinuités.

Le projet a introduit le concept d'"objet système d'équations (ESO, ou *Equation Set Object*)", représentation logicielle abstraite d'un ensemble d'équations algébro-différentielles ordinaires non linéaires. L'interface vers un ESO permet d'accéder à la structure du système, c'est-à-dire son nombre de variables, son nombre d'équations, et son remplissage; elle donne également accès aux variables elles-mêmes: nom, valeur courante, bornes inférieure et supérieure. Cette interface permet également aux clients de l'ESO de modifier ces valeurs, leurs dérivées, et de demander les résidus et dérivées partielles (Jacobien) correspondants.

Le concept des ESO a été développé pour les besoins d'interfaçage avec des solveurs non linéaires compatibles CAPE-OPEN. Il constitue en outre un mécanisme générique qui permet à un EMP de communiquer la structure mathématique des modèles qu'il contient, et donc de jouer le rôle de serveur de modèles, permettant le développement d'applications à base de modèles au delà des EMPs.

Pour rendre compte des discontinuités pouvant exister, notamment dans le traitement des procédés "batch" et "hybrides", les interfaces vers les ESO sont encapsulées dans des structures plus générales qui contiennent également un ou plusieurs *State Transition Networks* indiquant les changements possibles d'états. Je ne détaillerai pas ces concepts ici.

Enfin, dans le cadre séquentiel-modulaire, des interfaces ont été définies vers les modules de partitionnement et de séquençement des calculs, exploitant la théorie des graphes. L'interface permet de construire les graphes, de les découper en partitions, d'ordonner la séquence de calculs. Je ne détaille pas non plus ces concepts ici, l'information étant disponible sur le site [www.colan.org](http://www.colan.org).

### 3.2.3 Résultats

Le premier résultat de CAPE-OPEN a été de réunir à la même table les acteurs centraux de la simulation de procédés : des industriels représentatifs, les leaders du marché logiciel, des universités de renom, et de faire collaborer ces acteurs aux intérêts divers et parfois concurrents. Cette collaboration est allée jusqu'au consensus sur les besoins, les priorités, les principaux concepts d'architecture des logiciels, et bien entendu jusqu'aux spécifications techniques d'interfaces pour les objets principaux de la simulation.

Le deuxième résultat notable est l'évaluation et l'application, dans un nouveau domaine, de méthodes et outils soutenant ce processus de spécification. Initialement, nombre de partenaires ne connaissaient que la programmation procédurale en FORTRAN et avaient du mal à penser en objets. Finalement, et après un gros investissement de formation, on peut dire que le choix d'une architecture moderne à base de composants a été un facteur déterminant pour la réussite du projet. Son implémentation a été rendue plus complexe en raison de la présence à l'époque de deux *middleware* concurrents, COM et CORBA. Au bout du compte, les standards ont été réalisés pour les deux systèmes, et en quelque sorte unifiés par la représentation UML de plus haut niveau d'abstraction.

La faisabilité et le potentiel industriel de CAPE-OPEN ont été illustrées par deux démonstrations. La première, sur la base de COM, a été développée par les fournisseurs Aspen Technology et AEAT-Hyprotech; la démo CORBA, par l'université d'Aix-la-Chapelle et d'autres, allait plus loin et montrait la voie pour le futur des environnements hétérogènes de simulation.

La démonstration COM fait coopérer des composants opérations unitaires et calculs thermodynamiques des deux fournisseurs. Elle montre la motivation de ces deux vendeurs à intégrer le standard dans leur offre. Elle a été suivie d'annonces commerciales d'Aspen Technology et de AEAT Hyprotech qui ont aujourd'hui intégré ces interfaces à leurs produits phares, AspenPlus 10.2 et Hysys.Cape-Open - voir les zones "communiqués de presse" sur les sites web [www.aspentec.com](http://www.aspentec.com) et [www.hyprotech.com](http://www.hyprotech.com).

La démonstration CORBA intégrait des outils industriels et de recherche développés dans plusieurs pays, sur un réseau hétérogène, tant machines (PC et Sun) que systèmes d'exploitation (Windows NT et Solaris). Précisément, les éléments en sont: (i) des modèles d'opérations unitaires de l'IFP en France; (ii) le solveur de l'EMP orienté

équations gPROMS d'Imperial College, Londres; (iii) la bibliothèque thermodynamique IK-CAPE [IK-CAPE 95] développée conjointement par les principaux chimistes allemands; (iv) un outil de partitionnement et de séquençement réalisé par l'INPT à Toulouse; (v) le tout intégré dans la plate-forme CHEOPS de RWTH Aachen, en Allemagne. Ces composants sont écrits dans des langages différents, qui vont de FORTRAN à Java en passant par C et C++. Cette démonstration montre le potentiel des interfaces et les innovations qu'elles permettent, comme l'intégration de parties orientées équations et séquentielles-modulaires au sein d'une seule application. Accessoirement, elle démontre également le savoir-faire de la recherche européenne.

Nous sommes allés beaucoup plus loin dans le projet Global CAPE-OPEN, qui fait l'objet de la section suivante.

### 3.2.4 Le projet Global CAPE-OPEN et ses résultats

Le projet CAPE-OPEN s'est formellement conclu le 30 Juin 1999, en fait pratiquement le 1<sup>er</sup> Octobre de la même année, avec la publication de la première version des spécifications. Global CAPE-OPEN (GCO) en a pris la suite. Démarré en juillet 1999, et d'une durée de trente trois mois, GCO a été à la fois soutenu par la commission européenne dans son action Brite-EuRam, et endossé par le programme IMS<sup>13</sup>.

GCO, dont le budget était d'environ 8 millions d'Euros, toutes régions confondues, bénéficia en Europe d'un financement de 2.5 millions d'Euros de la CE. Le consortium européen regroupait seize partenaires industriels, fournisseurs, et universitaires. Le consortium international y ajoute trois groupes aux USA, Japon et Canada. Toutes ces organisations se sont accordées sur les objectifs suivants:

1. Développement de standards pour des nouveaux domaines de la modélisation de procédés, prolongeant et étendant ceux de CAPE-OPEN: fluides complexes, serveurs de modèles cinétiques, algorithmes d'optimisation, d'estimation de paramètres et de validation de données, systèmes distribués spatialement.
2. Migration de composants logiciels: de nombreux partenaires des trois classes ont porté leurs logiciels au standard, soit pour leur utilisation interne, soit à des fins de recherche, soit en préparation de leur commercialisation.
3. Intégration de l'approche de simulation ouverte dans le processus de production: GCO a pris des cas d'étude industriels concrets, et examinera comment se fait l'introduction de la technologie CAPE-OPEN sur ces exemples, sur des aspects de sélection de composants, de validation, de qualité.
4. Recherche sur les besoins de nouveaux standards dans des domaines d'applications nouveaux et plus pointus, au-delà de la simulation: systèmes "online", systèmes discrets ou hybrides (continus/discrets), planification et séquençement; gestion du cycle de vie des modèles.

---

<sup>13</sup> Intelligent Manufacturing Systems, programme de coopération internationale entre sept régions de la planète.

5. Dissémination des standards pour faciliter leur adoption dans l'industrie.
6. Création du "CAPE-OPEN Laboratories Network" (CO-LaN), organisme à but non lucratif qui a pris la responsabilité à long terme des standards.

Ces objectifs ont été atteints. Des nouvelles spécifications ont été publiées; des composants logiciels ont été migrés; des prototypes de sélection/configuration ont été réalisés; les domaines nouveaux que sont les systèmes de production hybrides, la planification, la commande, ont été abordés et des démonstrations convaincantes ont été réalisées; le CO-LaN a été créé. Les principaux résultats de GCO sont présentés dans l'annexe 2.

### 3.2.5 Ce que l'on peut faire avec CAPE-OPEN aujourd'hui

Depuis la conclusion des projets CAPE-OPEN et Global CAPE-OPEN, l'intéropérabilité entre composants de modélisation de procédés est démontrée et appliquée dans les principaux environnements commerciaux. Elle offre des opportunités de développement de solutions à haute valeur ajoutée, permettant aux industriels de mieux concevoir, opérer, optimiser et superviser leurs procédés.

Les partenaires du projet Global CAPE-OPEN ont pu constater les possibilités offertes par le standard. En voici quelques illustrations. Dans cette section, des noms de produits sont donnés, qui sont bien entendu propriétés de leurs auteurs respectifs. De plus, Aspen+ et Hysys sont considérés comme des logiciels différents, même si l'acquisition attendue de Hyprotech par Aspen Technology devrait conduire à terme à un rapprochement des deux environnements. Mais cela ne change rien au principe.

- ◆ Un composant de calcul de propriétés physiques et thermodynamiques développé par une société, peut être utilisé de la même manière au sein de divers EMPs. Par exemple, Multiflash, produit de la PME britannique Infochem, peut être utilisé de manière identique pour des simulations effectuées avec Aspen+, gPROMS ou Hysys. Ainsi, l'utilisateur économise le temps passé à la configuration des calculs de propriétés pour ces trois environnements, et a de plus une garantie d'homogénéité des méthodes et des résultats. Ceci est obtenu par simple encapsulation du serveur thermodynamique au standard CAPE-OPEN.
- ◆ Un EMP compatible CAPE-OPEN peut faire appel à plusieurs serveurs de calculs de propriétés thermodynamiques de manière quasi transparente. Par exemple, la version compatible CAPE-OPEN de Hysys peut être configurée pour utiliser soit la thermodynamique propriétaire de Hyprotech, COM Thermo, soit le serveur Properties Plus d'Aspen Technology, soit Multiflash d'Infochem, soit le serveur "maison" SPIP de l'IFP. Cette configuration peut se faire sur la base du remplacement immédiat du serveur unique pour le procédé concerné, ou même de la combinaison de plusieurs serveurs, chacun traitant un sous-ensemble du schéma<sup>14</sup>. Ainsi, le concepteur d'un modèle de procédés peut expérimenter diverses méthodes

---

<sup>14</sup> moyennant des précautions sur la base enthalpique, précautions, non inhérentes au standard, à prendre dès que des modèles thermodynamiques différents sont utilisés

de calculs de propriétés et choisir la meilleure pour le cas à traiter, sans passer un temps important en interfaçage ou en saisie de données. Le concepteur peut également faire appel au meilleur module thermodynamique pour chaque partie de son procédé, une souplesse apportée naturellement par le portage de l'EMP au standard, c'est-à-dire, par l'introduction de l'API CAPE-OPEN "Thermo" dans l'EMP.

- ◆ Un modèle d'opération unitaire comme, par exemple, un réacteur chimique propriétaire, développé par un opérateur ou par un bailleur de procédés, peut être utilisé de manière transparente par les environnements compatibles CAPE-OPEN. Ainsi, le modèle générique de réacteur "FIBER" (FIXed BEd Reactor) de l'IFP fonctionne de la même manière dans la plupart des environnements commerciaux de modélisation, sans qu'il soit besoin d'une quelconque intervention de codage ou même de recompilation. Le bailleur peut ainsi servir sans effort divers clients qui chacun imposent l'utilisation d'un logiciel de simulation spécifique dans leur organisation. Ceci provient simplement de la mise en conformité des appels aux principales fonctions du modèle d'opération unitaire : inclusion dans un schéma, connexion des ports d'entrée-sortie, spécification des paramètres du modèle, vérification de la calculabilité du modèle, appel au calcul, et publication des résultats.
- ◆ Réciproquement, et ceci est un des principaux intérêts du standard, un développeur de modèle, utilisant un EMP disposant des interfaces d'opérations unitaires CAPE-OPEN, peut inclure des opérations unitaires externes de manière extrêmement aisée, par sélection dans la liste des opérations unitaires compatibles disponibles. Ainsi, lors du choix d'un équipement spécifique (pompe, compresseur, échangeur etc..), le concepteur du modèle peut tester rapidement l'adéquation de l'équipement au besoin du procédé, et ainsi choisir rapidement une solution technique optimale. Aujourd'hui, avec la version actuelle du standard, cette fonction nécessite que les modèles d'opérations unitaires soient physiquement installés sur la machine du modélisateur. Demain, grâce au développement des services d'identification de composants sur l'internet, il sera possible de balayer un catalogue de modèles disponibles sur le réseau, de télécharger les composants potentiellement intéressants, de les introduire "à chaud" dans la simulation, et donc d'obtenir très rapidement une évaluation. Les équipementiers de l'industrie chimique et de procédés ne sont pas encore suffisamment informés de cette possibilité, qui est réelle, et qui peut apporter des bénéfices extrêmement importants à ceux qui sauront l'utiliser.
- ◆ Grâce au nouveau module commercial GO:CAPE-OPEN, un modèle d'opération unitaire "orienté équations" écrit en gPROMS de Process Systems Enterprise Ltd., quelque complexe qu'il soit, peut être automatiquement inséré comme objet "GO:CAPE-OPEN" dans un EMP compatible, sans le moindre effort. Tout développeur de modèles d'opérations unitaires avec gPROMS peut bénéficier de cette fonctionnalité.
- ◆ L'introduction d'une interface CAPE-OPEN autour d'un module de mécanique des fluides tel que Fluent, par exemple, fonctionnant sur station de travail, permet d'utiliser le code de CFD à l'intérieur d'une simulation de procédé fonctionnant sur PC. Des exemples ont été implémentés et présentés aux partenaires de Global



CAPE-OPEN. Attention, bien évidemment, dans ce cas, les temps de calcul de chaque section du procédé sont à prendre en considération. Avec les performances actuelles, le lien entre le schéma et le réacteur CFD doit se faire avec une période d'échantillonnage supérieure au pas de résolution de la simulation. Si le module CFD est installé sur un supercalculateur très performant, ou une grappe de PC, cette limitation peut tomber, mais nous n'en n'avons pas encore l'expérience.

- ◆ Les exemples précédents reposent sur les interfaces CAPE-OPEN pour les opérations unitaires et les serveurs de calculs de propriétés physiques et thermodynamiques. En fait, la version 1.0 du standard, mars 2002 [CO-LAN 02], comporte une dizaine d'autres interfaces pour plusieurs catégories de composants: bases de données de propriétés physiques, schémas cinétiques, pseudo-composants et fractions pétrolières, solveurs numériques, optimisation linéaire et non linéaire, estimation de paramètres et validation de données, disposent également de spécifications. Les exemples illustratifs déclinés ci-dessus peuvent être transposés pour la plupart des composants faisant l'objet de spécifications: utilisation de banques de données de corps purs différentes dans une simulation, ou utilisation de la même banque de données dans plusieurs environnements, réutilisation de schémas cinétiques, choix du meilleur solveur pour une application etc.
- ◆ Une première liste, non exhaustive, d'EMP et de CMP compatibles CAPE-OPEN a été publiée dans la lettre d'information CAPE-OPEN Update [Braunschweig 02]. Cette lettre étant disponible au téléchargement sur le site internet [www.colan.org](http://www.colan.org), je ne la reproduis pas ici.

### 3.2.6 Le "CO-LaN"

Les projets CAPE-OPEN et Global CAPE-OPEN ont un début et une fin. La durée de vie des standards doit dépasser celle de ces projets, et il est donc nécessaire de mettre en place une instance chargée de leur gestion, de leur diffusion, de leur évolution, et de la certification de composants.

C'est pourquoi les partenaires ont décidé de créer un organisme à but non lucratif, le "CAPE-OPEN Laboratories Network", "CO-LaN", dont c'est la responsabilité. L'organisme a été légalement établi début 2001 sous forme d'une association loi de 1901, j'en suis le premier président depuis cette date. Ce premier laboratoire, qui a mis au point ses outils et modes de fonctionnement, sera probablement soutenu à terme par deux antennes nord-américaine et japonaise. Une étude préalable a défini les objectifs et les modalités opératoires du CO-LaN, tant sur le plan de l'organisation interne et des responsabilités, que sur le plan technique, notamment parce que le laboratoire utilise l'internet comme outil privilégié pour toutes ses opérations.

Les missions confiées au CO-LaN sont:

- ◆ Dissémination: organiser la distribution des informations sur le standard;
- ◆ Gestion du cycle de vie des spécifications: notamment en matière de maintenance, d'évolution, et de définition de nouveaux standards;

- ◆ Support au développement: fourniture de logiciels de tests de conformité, et organisation de sessions de tests d'interopérabilité;
- ◆ Formation: s'assurer que suffisamment de formations au standard sont offertes par les partenaires, et soutenir ces actions de formation par la publication d'informations techniques destinées aux organismes de formation.

L'activité du CO-LaN est visible sur son site web, [www.colan.org](http://www.colan.org).

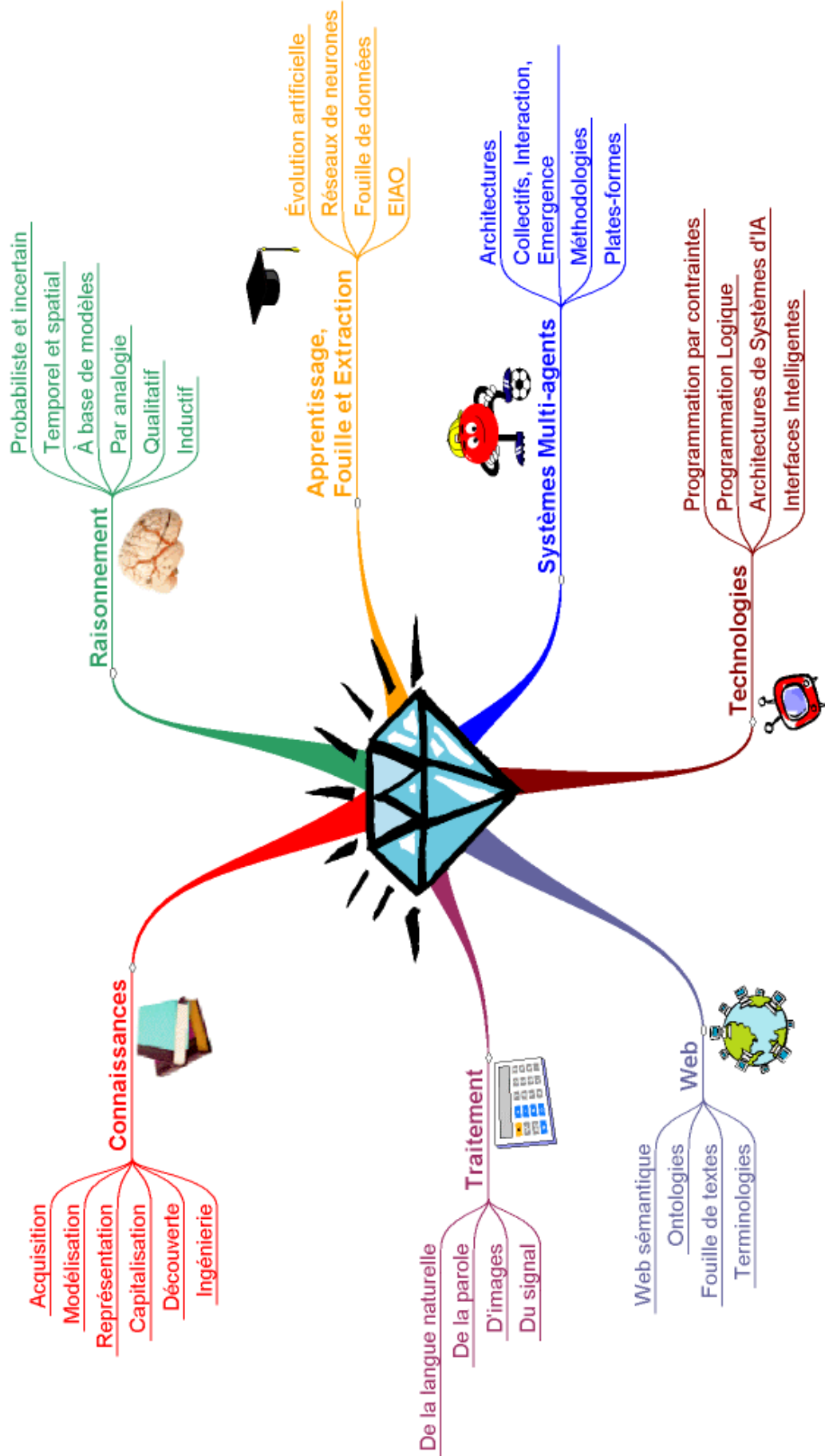
### 3.2.7 Conclusion: nouveaux débouchés envisageables

Avant d'aborder l'évolution vers la simulation multi-agents, au chapitre 6, je présente quelques idées sur les débouchés envisagés pour le standard CAPE-OPEN, qui peut avoir pour conséquence une nouvelle approche de la conception, du développement, de la commercialisation, de la distribution et de l'utilisation des logiciels CAPE.

Un scénario réaliste est que les vendeurs principaux offriront désormais des combinaisons d'EMPs et de CMPs provenant de sources diverses, devenant ainsi également des intégrateurs de systèmes. Les bénéfices que l'on peut en attendre sont :

1. Des meilleurs composants logiciels :calculs thermodynamiques plus précis, modèles de procédés plus fidèles, solveurs plus performants;
2. De nouveaux types d'utilisation mettant à profit ces meilleurs composants : optimisation dynamique en temps réel; diagnostic à base de modèles, ...
3. Ces composants seront à la base du *eBusiness/eWork* dans le domaine. Par exemple, les acheteurs de pompes, échangeurs, etc. sur le web, pourront également télécharger ou opérer à distance les simulateurs de ces équipements en les intégrant directement dans leurs modèles de procédés.
4. Il sera possible de développer des « compagnons logiciels intelligents », capables d'observer le trafic aux bornes des composants, et de faire des diagnostics sur la performance du système afin d'aider l'utilisateur. De telles aides existent dans les environnements propriétaires (aide à la sélection des modèles thermodynamiques, conseils sur la convergence), il sera désormais possible de proposer des compagnons génériques utilisables dans les environnements hétérogènes.
5. L'utilisation des interfaces dans les applications de diagnostic et de supervision de procédés offre également de nombreuses possibilités ; j'avais publié un article à ce sujet lors d'un workshop IFAC sur le diagnostic et la commande de procédés, en 1998 [Braunschweig 98o];
6. Enfin, le travail coopératif sera rendu possible, entre collègues d'une même société, entre clients et fournisseurs, entre partenaires sur un projet, par la distribution d'un modèle en réseau.





## 4. Dynamisation: apprentissage, évolution artificielle, agents intelligents

Je n'ai pas pu ni voulu résister à la tentation d'associer l'AFIA, Association Française pour l'Intelligence Artificielle, à ce chapitre. Au cours de mes quatre années et demi de présidence de cette association, j'ai parlé et écrit à de nombreuses reprises, sur le potentiel des technologies de l'IA pour améliorer les logiciels qui nous entourent. C'est pourquoi la carte mentale qui me semble la plus *adaptée* à illustrer ce chapitre est celle des thèmes de l'IA, qui est souvent utilisée pour présenter les activités et les domaines d'intérêt de l'AFIA.

On retrouve sur cette carte la plupart des techniques qui ont donné lieu à des travaux de recherche du groupe de compétences en intelligence artificielle de l'IFP, groupe que j'ai dirigé pendant la dernière décennie du vingtième siècle.

Dans les sept paragraphes suivants, je souligne les techniques pratiquées par le groupe IA, mais dont je ne parlerai pas plus, parce que cela sort du cadre de ce mémoire, et j'encadre celles que je développe plus loin dans le corps de ce chapitre et du suivant, parce qu'elles contribuent à la dynamisation du logiciel.

- ◆ D'abord, on trouve les diverses facettes de la mécanisation du raisonnement humain: raisonnement probabiliste et incertain, temporel et spatial, à base de modèles, par analogie, qualitatif, inductif.
- ◆ Ensuite, les techniques de l'apprentissage, de la fouille de données et de l'extraction de connaissances, parmi lesquelles les algorithmes d'évolution artificielle et leurs nombreuses déclinaisons, ou les diverses architectures de réseaux neuronaux. Ces techniques d'apprentissage sont l'objet d'intérêt d'un collège de l'AFIA, le Collège d'Apprentissage, Fouille et Extraction (CAFE). L'EIAO (environnements interactifs d'apprentissage avec ordinateur) est rangée un peu *artificiellement* dans cette catégorie, pour des raisons plus graphiques qu'épistémologiques.
- ◆ Un volet très important, qui constitue le deuxième collège de l'association, est celui des systèmes multi-agents, de leurs aspects architecturaux, collectifs, notamment sur la prise en compte des interactions et sur la (trop fameuse) émergence<sup>15</sup>, méthodologiques, et techniques - sur les plates-formes multi-agents.
- ◆ Les technologies sont également identifiées comme une thématique de recherche, en particulier la programmation par contraintes et sa cousine la programmation logique, les interfaces intelligentes et les architectures de systèmes d'IA.
- ◆ J'ai regroupé quelques techniques autour du web, plus comme sujet d'intérêt et cible d'applications, que comme sous-domaine scientifique de l'IA. Il faut dire que le web a été un facteur dynamisant de l'IA ces dernières années, et qu'il le sera de

---

<sup>15</sup> qui est plus, pour moi, dans la tête de l'observateur que dans le système multi-agent observé!

plus en plus avec l'avènement espéré du web sémantique, alimenté par des ontologies, elles-mêmes statiques ou évolutives grâce aux techniques de fouille de textes. La terminologie en est un sous-ensemble, elle est identifiée dans la carte notamment en raison de l'existence d'un groupe de travail sur le sujet.

- ♦ La rubrique du "traitement" est également plus un regroupement applicatif qu'un domaine scientifique en soi: traitement de la langue naturelle, de la parole, du signal, et de l'image, sont autant de thèmes d'applications importants, et font appel aux techniques présentées dans les paragraphes précédents.
- ♦ Enfin, il me semblait crucial d'identifier clairement le sujet de la connaissance comme un domaine à part entière, tant les activités autour des connaissances (acquisition, découverte, représentation, modélisation, ingénierie, capitalisation), ont été au coeur des recherches en IA depuis plus de vingt ans.

## 4.1 Des références essentielles

Je veux souligner que les idées exprimées dans ce mémoire sont en phase avec ce que des organismes ou organisations importantes ont dit ou écrit sur le sujet de l'évolution des architectures logicielles. Je pense en particulier à la Commission Européenne, dans son programme IST; au National Science and Technology Council américain, dans son projet Network and Information Technology Research and Development (NITRD); et à IBM, qui a publié son manifeste "Autonomic Computing" à l'automne 2001. Je cite des extraits des documents en question. Je ne traduis pas les textes afin de ne pas alourdir le mémoire.

### 4.1.1 Commission Européenne

Le programme de travail IST 2002 [IST 02] est très disert sur les notions de segmentation et de dynamisation, notamment dans le texte de son action-clé 4:

#### **IV.2.3: Networks and technologies for distributed services and applications**

...To develop and assess models, technologies and tools for sharing and interactive use of applications and resources in geographically dispersed locations, in the context of heterogeneous components and architectures. The focus of the work is on development environments to support application service provisioning, peer-to-peer computing and other distributed applications.

#### **IV.3.1 Composability and dynamic adaptability in software, systems and services**

... Integration and/or development of distributed components, including (intelligent) software agents, and associated co-operation mechanisms. This includes self-description of databases, components and agents and composition of heterogeneous software and application service elements with predictable impact on the properties of the resulting system.

Dynamically re-configurable and self repairing systems and systems that adapt themselves to growth, inclusion or withdrawal of components and devices and to other events that may occur during its lifetime.

L'accent est mis sur la configuration dynamique d'applications réparties par combinaison de composants, plus que sur l'adaptation des composants eux-mêmes. Je pourrais toutefois en citer d'autres extraits: le document "IST 2002" contient une demi-douzaines d'occurrences du terme "adapt" dans le contexte de la dynamisation du logiciel. Notez qu'on ne dit pas comment! Les programmes IST sont exprimés en termes de buts, non de moyens ou de technologies. J'y viendrai dans la deuxième partie de ce chapitre.

#### 4.1.2 National Science and Technology Council (USA)

Le document de programme "NITRD", qui fixait les budgets de recherche en technologies de l'information pour 2002 [NITRD 02], comportait nombre de références à la dynamisation du logiciel: j'en cite quelques extraits.

Critical Federal missions and industry needs both call for new scientific and technical paradigms that significantly raise high-end computational speeds, provide adaptable and reconfigurable computing environments, and reduce the size, cost, and power requirements of high-performance computing and data storage equipment...

**NIST:** Research in techniques for self-integration of manufacturing system components to enable full or partial interoperability in a world where standards are changing quickly; automated or partially automated creation of software using combinations of formal methods, machine learning, and knowledge-based techniques

...Methods for putting together software "components" to reduce development time and increase reliability, including technologies for developing distributed, autonomous and/or embedded software; software development automation...

Le programme US, tout en reprenant à son compte les thèmes IST, ajoute le thème de l'apprentissage, du "self" et de l'autonomie. On voit ici la dynamisation non seulement par la combinaison de services logiciels, mais encore par la reconfiguration interne des composants, de par eux-mêmes, face à un environnement changeant.

#### 4.1.3 IBM (USA)

IBM va beaucoup plus loin. Les thème de la dynamisation est au cœur du document "Autonomic Computing" publié en 2001 [IBM 01]. Il m'est impossible de citer tous les extraits pertinents sur ce sujet, puisque c'est l'ensemble du document qui est consacré aux techniques permettant de rendre les logiciels autonomes et adaptatifs. Je cite les meilleurs passages, dont le "leitmotiv", ligne directrice du manifeste.

**It's time** to design and build computing systems capable of running themselves, adjusting to varying circumstances, and preparing their resources to handle most efficiently the workloads we put upon them. These autonomic systems must anticipate

needs and allow users to concentrate on what they want to accomplish rather than figuring how to rig the computing systems to get them there... The underlying technologies to enable greater automation of complex systems management are ripe for innovation. The emergence of XML and a host of new standards give us a glimpse of the glue we'll need to bind such self-governing systems, and advances in workload management and software agents promise possible incremental paths to autonomic computing.

... An autonomic computing system must configure and reconfigure itself under varying and unpredictable conditions....

...An autonomic computing system must perform something akin to healing — it must be able to recover from routine and extraordinary events that might cause some of its parts to malfunction....

... Many established fields of scientific study will contribute to autonomic computing. What we've learned in artificial intelligence, control theory, complex adaptive systems and catastrophe theory, as well as some of the early work done in cybernetics, will give us a variety of approaches to explore...

Si on oublie quelques instants le caractère commercial du manifeste - un des objectifs clairement avoués est de vendre les nouveaux serveurs de calcul reconfigurables - on peut reconnaître qu'IBM a clairement pris position pour la dynamisation du logiciel et de son support matériel, une attitude qui était prévisible depuis la création d'une division "pervasive systems" au sein de la société. Dépassant les notions d'auto-adaptation de NITRD, IBM décrit un monde de composants logiciels en reconfiguration permanente, apprenant, se régénérant, grâce à leur interaction continue avec le monde alentour. IBM, de plus, dit *comment*: multi-agents, intelligence artificielle, apprentissage, théorie du contrôle, des systèmes complexes sont mentionnés à de nombreuses reprises, pour apprendre les préférences d'un utilisateur, générer des règles de reconfiguration, contrôler le système par feedback.

## 4.2 La dynamisation: comment?

Je présente, dans les cinq sections suivantes, quelques approches de l'adaptation et de la dynamisation. Notez bien que les exemples sont pris dans mon vécu à l'IFP, et non dans des expériences visant directement à la dynamisation de logiciels. Il s'agit donc plutôt de passer en revue quelques techniques. Je tâche de présenter une vue synthétique des travaux réalisés, les articles complets étant cités en référence. De plus, je ne présente pas une énième fois les techniques concernées, la littérature étant remplie d'articles et d'ouvrages explicatifs sur ce que sont les réseaux neuronaux, les algorithmes d'apprentissage symbolique, l'évolution artificielle, le raisonnement par cas etc. D'ailleurs, j'ai expliqué la plupart des techniques en question dans les articles introductifs des deux ouvrages de la série « Artificial Intelligence in the Petroleum Industry », [Braunschweig 95La] et [Braunschweig 96La], pour les lecteurs de l'industrie pétrolière, non spécialistes de ces sujets.



#### 4.2.1 Apprentissage, réseaux neuronaux

Nous avons expérimenté les réseaux neuronaux en raison de leur capacité d'apprentissage de relations non linéaires, sous plusieurs formes. Nous avons traité aussi bien des grandeurs numériques que symboliques.

- ♦ *Classification d'expériences de forages.* Dans cette expérience menée avec mon collègue Didier Pavone [Braunschweig 95Ca], nous avons utilisé l'algorithme des cartes topologiques auto-organisatrices de Kohonen pour effectuer une classification non supervisée de signaux de forages obtenus avec l'outil d'acquisition en fond de puits TRAFOR. Les cartes obtenues ont été ensuite traitées par un algorithme de notre cru, afin de repérer les pics et les vallées de la distribution ainsi obtenue, donnant lieu à la création de macro-classes. Le but de cette approche était, comme nous l'avons écrit dans un rapport interne IFP :

*« D'après les premiers résultats obtenus, il est possible de se faire une idée d'un système de classification des modes de fonctionnement du forage utilisable en ligne sur l'outil Trafor: les cartes topologiques une fois constituées hors-ligne, sur la base des prétraitements numériques les plus appropriés, peuvent aisément être mises en exploitation en ligne, et donner au foreur une représentation simple et intuitive du mode de fonctionnement du forage basée sur la position et la tendance sur une carte des modes opératoires de l'outil, permettant à l'opérateur de diagnostiquer la situation et d'intervenir le cas échéant. »*

Dans une telle approche, la dynamisation consiste à relancer périodiquement une étape d'apprentissage sur les données obtenues en ligne, afin de recalibrer le logiciel de suivi par rapport aux évolutions possibles de l'outil et du milieu rencontré.

- ♦ *Interprétation quantitative de diagraphies.* Ces deux travaux, complémentaires, ont été effectués en collaboration avec la société Western Geophysical, pour le premier [Draelants 92] et avec Elf Aquitaine, pour le deuxième [Abbassi 92]. Dans les deux cas, nous avons cherché à obtenir une meilleure prédiction de propriétés de roches pétrolières traversées par des forages, au moyen d'enregistrements multiségnaux appelés « diagraphies ». Les logiciels commerciaux utilisent la régression linéaire ou une méthode d'optimisation de gradient pour prédire des propriétés des roches à partir de signaux enregistrés. Dans le deuxième cas, l'idée était d'améliorer les performances du logiciel OPTIM utilisé couramment par le groupe Elf. Nous écrivions:

*« les premiers résultats que nous avons obtenus fournissent une motivation suffisante pour poursuivre cette expérimentation dont l'aboutissement pourrait être l'intégration d'un module neuronal dans la chaîne de traitement des informations de diagraphies »*

Dans cette étude, reposant sur des classiques architectures de Perceptron multicouches, nous avons consacré beaucoup d'efforts à analyser la répartition des exemples d'apprentissage sur les fonctions de transfert sigmoïdales afin de déterminer l'importance relative des phénomènes linéaires et non linéaires. Sur le plan applicatif, nous n'avons pas poursuivi ce travail, pour des raisons indépendantes de la technique employée, et dont je ne peux faire état.

- ♦ *Interprétation de caractéristiques pétrolières de sédiments géologiques.* Toujours dans le domaine pétrolier « amont » (Exploration-Production), nous avons travaillé à

inclure des fonctions d'interprétation avancées dans le logiciel de traitement de résultats de pyrolyses d'échantillons de roches pétrolifères, ROCKINT. Ce travail, entamé au moyen d'un stage d'ingénieur, puis prolongé par le stage post-doctoral de Wady Naanaa, a donné lieu au dépôt d'un brevet [Lafargue et coll. 98]. L'idée directrice mise en œuvre dans le système d'interprétation est de combiner réseaux neuronaux et traitement flou. Deux réseaux de neurones fournissent les paramètres de caractérisation de la matière organique d'un échantillon de roche, et la caractérisation obtenue est affinée à l'aide de la composante floue, afin d'une part, d'interpréter les réponses des réseaux lorsqu'ils ne produisent pas un diagnostic précis, de nuancer ces diagnostics pour les rendre plus proches de la réalité, et de les exprimer dans un langage compréhensible par un utilisateur non familier du four ROCK-EVAL<sup>16</sup>.

- ◆ *Induction de règles de production en catalyse et en forage.* Il s'agit de notre seule expérience en génération de règles de production à partir de données. Ce travail, qui avait constitué le stage d'ingénieur de Sylvain Sarda [Sarda 90], visait à expérimenter un outil commercial utilisant l'algorithme de référence ID3, algorithme capable de générer des arbres de décision à partir d'exemples, en apprentissage supervisé. Dans la première application, on cherchait à déterminer comment des caractéristiques de zéolites (catalyseurs de raffinages) dépendaient de leurs conditions de préparation : température et durée et concentration d'attaque acide. Dans la deuxième application, on cherchait à établir des relations symboliques entre certains paramètres descriptifs de roches rencontrées en cours de forage, et la résistance au forage (énergie spécifique), qui indique la difficulté de forer et donc les risques de dommages sur l'outil. Dans les deux cas, l'algorithme ID3 a fourni des résultats assez corrects, validés par les spécialistes du domaine ; nous avons dû ajouter une étape de post-traitement pour éliminer des redondances dans les règles générées.
- ◆ Je cite également, dans cette rubrique, trois applications des réseaux neuronaux à des problèmes divers :
  - (i) le *neural network controller project* de Jean-Michel Lambert [Lambert 91], en collaboration avec R. Hecht-Nielsen, UCSD, visait à développer un système de contrôle neuronal en ligne pour une colonne de distillation. Cette application, utilisant les réseaux récurrents, était capable d'apprendre en ligne la régulation d'un procédé fortement non linéaire - moyennant des temps de calcul assez importants... Elle préfigurait des développements faits par d'autres par la suite, que l'on peut trouver par exemple dans l'offre commerciale de Pavilion Technologies, spécialiste de l'application des réseaux neuronaux à la supervision de procédés [Keeler 95], ou dans celle de Gensym avec leur outil NeurOnLine [Gensym 02].
  - (ii) La thèse d'Eric Mousset [Mousset 94+] portait sur l'utilisation de réseaux neuronaux en couches pour le filtrage des "spikes" dans les signaux sismiques. C'est une assez belle application des réseaux

---

<sup>16</sup> Extraits du brevet d'invention correspondant

neuronaux, combinant les perceptrons multi-couches à des prétraitements en ondelettes, et faisant le lien théorique avec des techniques non neuronales de filtrage, notamment le filtre médian. Je la cite pour mémoire, nous n'avions pas implémenté de réapprentissage en ligne, d'une part en raison du très important volume de données à traiter, et d'autre part, de manière assez surprenante, en raison de la simple constatation que le réseau, entraîné sur un premier type de données provenant d'une campagne sismique en Algérie, fonctionnait parfaitement sur les données d'une campagne faite en Russie, pourtant fort différente [Mousset 91]!

- (iii) Le mémoire d'ingénieur de Franck Jamin [Jamin 91] sur l'estimation du cliquetis dans les moteurs à allumage commandé. Le but de ce travail était de modéliser le phénomène (non linéaire) de l'apparition du cliquetis dans un moteur, en fonction des paramètres de fonctionnement: vitesse de rotation, pressions, températures, délais d'allumage, fraction d'hydrocarbures brûlés etc. Le cliquetis est un sérieux problème des moteurs 4 temps, lié à l'efficacité de la combustion, et peut causer d'importants dommages. Le phénomène étant particulièrement bruyé, la prédiction de cliquetis sur un cycle moteur n'était pas possible; l'approche qui a été finalement prise, et qui donnait des bons résultats, consistait à réaliser une prédiction statistique du phénomène sur un ensemble de cycles: à partir d'une représentation statistique des données d'entrées, déterminer un histogramme prévisionnel d'intensité de cliquetis sur ces cycles. Par la suite, il aurait été possible de mettre en place un système de prédiction en ligne du cliquetis sur un moteur, apprenant continuellement et produisant des informations utiles à la prévention du phénomène, mais l'IFP a arrêté de travailler avec le constructeur automobile concerné, qui était celui pour lequel le problème était vraiment crucial (je ne cite pas de marques...☺).

#### 4.2.2 Raisonnement par cas

Une manière efficace et relativement simple de donner des fonctions d'adaptation aux logiciels est de les faire raisonner sur des cas.

Le raisonnement par cas, ou Case-Based Reasoning (CBR) [Aamodt 94] permet de réaliser des systèmes d'aide à la conception qui utilisent les expériences passées pour résoudre de nouveaux problèmes. Le principe de base du CBR : « pour résoudre un problème, rappelez-vous un problème similaire que vous avez résolu dans le passé et adaptez sa solution au nouveau problème », est bien adapté pour assister l'activité de conception lorsque l'on dispose d'une bibliothèque de cas résolus. Lors du séjour post-doctoral de Jerzy Surma, nous avons réalisé un système raisonnant à partir de cas [Surma 96p] pour assister un ingénieur dans les premières phases de conception de procédés de raffinage ou de pétrochimie, en s'appuyant sur la représentation au sein de schémas de procédés. Lors de la phase de conception préliminaire, des cas existants (i.e. des études de procédé) sont sélectionnés, leur similarité avec le problème de conception à résoudre est évaluée, et les études les plus similaires servent de base pour la résolution du problème courant.

Le choix du raisonnement par cas à cette étape est pertinent pour deux raisons : d'une part, il modélise bien la façon dont l'ingénieur travaille pendant la phase de conception; d'autre part, son utilisation est moins aisée dans les phases ultérieures de spécification qui font appel à des techniques numériques et informatiques établies, complétées par le savoir-faire du concepteur, et qu'il est plus difficile de modéliser au sein d'un système informatique.

Techniquement, la sélection des cas reposait sur une procédure qui utilisait alternativement la discrimination par une hiérarchie de procédés, le calcul d'une similarité de surface utilisant simplement la liste des équipements de procédés, et la comparaison profonde de graphes typés, utilisant un algorithme de recherche d'isomorphisme de graphes.

Une telle architecture est un support technique intéressant pour la mise au point de mémoires d'études de procédés se prêtant à la réutilisation. Nous nous sommes contentés, en première approche, de développer une architecture de sélection et de stockage des études. La phase d'adaptation ne nous paraissait pas vraiment nécessaire.

Cette approche peut rentrer dans le cadre général de l'apprentissage et de la dynamisation, puisqu'elle se base sur une mémoire des cas passés, elle-même en évolution permanente, car stockant les nouveaux cas traités afin qu'ils puissent être réutilisés. L'utilisation d'algorithmes d'induction est un plus utile mais non indispensable, puisque la compétence du logiciel s'accroît simplement au fil de son utilisation. Elle devient nécessaire lorsque les bases de cas sont trop grandes pour pouvoir être directement consultées, ou lorsque les connaissances génériques obtenues peuvent être exploitées, en particulier dans la phase d'adaptation des cas sélectionnés.

D'autres applications de raisonnement par cas ont été réalisées à l'IFP, notamment dans le domaine du diagnostic de corrosion d'équipements et dans le domaine du diagnostic d'essais moteurs.

### 4.2.3 Evolution artificielle

Je crois pouvoir affirmer que l'IFP a été un des premiers industriels français à s'intéresser aux algorithmes d'évolution artificielle, puisque nos premiers essais datent de 1989, époque où les algorithmes génétiques n'étaient encore connus que d'un petit milieu de spécialistes. Après avoir entendu parler pour la première fois de ces algorithmes, j'avais programmé, à titre de démonstration, l'algorithme génétique de base de [Goldberg 89] sur un Apple Macintosh, au moyen du logiciel d'hypertexte HyperCard, en faisant ainsi l'algorithme génétique le plus lent jamais programmé ! Très rapidement, j'ai été fasciné par les capacités de ces algorithmes, malgré leur apparente naïveté, à l'image de nombre de chercheurs qui ont depuis bâti leur carrière universitaire sur l'utilisation de ces étranges systèmes.

Après une phase initiale de quelques années, au cours de laquelle les principaux essais ont concerné l'ajustement de modèles thermodynamiques [Emami 93C], ou l'inversion sismique, l'utilisation de l'évolution artificielle s'est répandue sur plusieurs domaines, avec notamment les deux thèses de Frédéric Mansanné [Mansanné 00t] et de Benoît Leblanc [Leblanc 02t], complétées par quelques autres études. Ces travaux se sont appuyés sur des collaborations toujours fructueuses avec Marc Schoenauer et Evelyne

Lutton, deux « personnalités » de l'EA française, tout d'abord indépendamment, puis ensemble, depuis leur réunion dans le projet FRACTALES de l'INRIA.

- ♦ *Ajustement de modèles thermodynamiques.* Notre premier essai « en vraie grandeur » s'intéressait à la représentation de la fraction lourde des gaz à condensats, pour les besoins des modèles thermodynamiques utilisés en ingénierie de gisement et en transport. Les techniques analytiques permettent de déterminer la partie légère de la composition d'un fluide de gisement, en général jusqu'aux espèces en  $C_{10}$ . Par contre, il n'est pas possible de mesurer précisément la composition de la partie la plus lourde du fluide, et on ne dispose que de mesures expérimentales d'équilibres liquide vapeur : pression, densité à quelques températures, et points singuliers. Le principe adopté est de représenter cette fraction lourde par un ou plusieurs « pseudo-composants » dont la nature et la répartition sont des inconnues à déterminer. Si on dispose d'une hypothèse sur ces pseudo-composants et leur fraction massique, il est possible de faire tourner le code thermodynamique, et donc d'observer la correspondance entre les résultats obtenus et les données expérimentales, ce qui fournit donc un critère d'ajustement assez objectif. Nous avons longtemps travaillé avec nos collègues thermodynamiciens, d'abord pour réaliser un système expert d'aide à l'ajustement de modèles [Nabec 89], puis pour effectuer cet ajustement à l'aide d'algorithmes génétiques. Cette dernière application utilisait initialement l'algorithme de référence de Goldberg, a finalement combiné de manière « Lamarckienne » un algorithme génétique travaillant sur la partie symbolique du problème - choisir trois pseudo-composants parmi une cinquantaine de composants potentiels - à une optimisation numérique par descente de gradient pour trouver la répartition en masse des pseudo-composants dans le fluide. La fonction d'adaptation (*fitness*) était simplement une fonction de moindres carrés sur une courbe d'apparition de liquide dans le fluide en fonction de la pression, pondérée selon l'importance et la précision des points expérimentaux de référence. Suite à ce travail, mes collègues thermodynamiciens sont partis sur d'autres approches utilisant des représentations continues de la partie lourde, à la place de représentations discrètes par pseudo-composants, sans toutefois obtenir d'aussi bons résultats de calage que par l'approche évolutionnaire - à ma connaissance, qui date de quelques années [Sportisse 97].
- ♦ *Identification de modèles géologiques.* Je n'ai pas directement encadré la thèse sur de Frédéric Mansanné [Mansanné 00+] sur ce sujet, même si j'en ai défini le contenu, si j'ai encadré les travaux préliminaires, et si je suis intervenu en conseil au long de la thèse, qui, se déroulant à Pau, a été prise en charge par mes collègues du département de Mathématiques Appliquées. Cette thèse portait sur l'utilisation des algorithmes génétiques pour la détermination de vitesses en imagerie sismique. Elle faisait appel à une population de représentations du sous-sol par des diagrammes de Voronoï, sur lesquelles étaient effectuées les opérations génétiques de sélection, croisement, mutation. Le détail de ces opérations est complètement décrit dans la thèse. Les modèles de sous-sol étaient ensuite transformés en maillages cartésiens, utilisables par les algorithmes de migration sismique, qui produisaient, après un temps de calcul non négligeable, une image à comparer à l'image de référence pour l'optimisation. Ma contribution à ce travail a été principalement de proposer des combinaisons de fonctions directes à utiliser pour la génération d'images, mêlant migration et propagation des ondes, en raison de la complémentarité des informations que ces deux techniques produisent. D'ailleurs, l'approche en simple

migration conduit à des problèmes de faux optima très difficiles à traiter, puisque des modèles très différents de la solution recherchée produisent des résultats aussi bons, voire meilleurs. Ce travail a notamment eu le mérite d'identifier ces questions de faux optima, qui étaient méconnus de nos collègues géophysiciens.

- ◆ *Optimisation de structures flexibles.* Dans cette petite étude conduite avec un collègue du département de Mécanique Appliquée, nous avons étudié l'utilisation des algorithmes d'évolution artificielle pour l'optimisation de conduites flexibles sous-marines, afin d'obtenir des propriétés mécaniques satisfaisantes. Il ne m'est pas possible d'en dire plus sur ce travail, pour des raisons de confidentialité, bien que nous étions heureux, à cette occasion, de trouver une fonction *fitness* rapide à calculer, contrairement à la plupart des cas proposés par l'IFP !
- ◆ *Amélioration de l'efficacité statistique des simulations moléculaires de Monte Carlo.* La thèse de Benoît Leblanc [Leblanc 02†] est sans doute le travail le plus achevé utilisant les algorithmes d'évolution artificielle à l'IFP. Nous avons beaucoup écrit et publié à son sujet, et je vais donc citer des textes existants plutôt que de les paraphraser! Permettez-moi de citer d'abord le petit mot que j'ai écrit, au nom du directeur de la Division Informatique Scientifique Mathématiques Appliquées, au sujet de la thèse :
  - « Le travail de Benoît Leblanc s'est inscrit dans les axes de recherche et développement de la DISMA sur les méthodes d'optimisation globales non paramétriques et a apporté des résultats significatifs et particulièrement originaux en la matière; il a contribué de manière très positive à nos relations scientifiques avec l'INRIA, avec le projet FRACTALES de Rocquencourt qui a hébergé la thèse, et avec le projet APACHE de l'INRIA Rhône Alpes sur les grappes de PC pour le calcul intensif.»

Je cite également la description du travail lui-même, telle que résumée par Benoît Leblanc (je surligne le passage sur l'adaptation):

- « Nous nous sommes confrontés dans cette thèse au problème de l'efficacité statistique des simulations moléculaires de Monte Carlo, qui se pose essentiellement pour des molécules complexes, au nombre desquelles les polymères amorphes en état dense. Dans ce cas l'efficacité désigne l'aptitude à fournir en un temps de calcul fini, un échantillon statistiquement valide afin qu'il puisse servir à donner des estimations fiables des propriétés du système physico-chimique simulé. Nous avons vu que cela revenait à fournir une suite de configurations qui soient les plus décorréelées possible et que ce concept, intuitivement simple, n'est pas évident à mesurer explicitement. Nous avons donc déterminé des critères numériques représentatifs de cette efficacité, applicables à la simulation des polymères. Il nous a encore fallu identifier des paramètres libres de ces simulations, c'est-à-dire qui ne soient pas imposés par les conditions de simulations et dont le changement occasionnel en cours de simulation n'invalide pas celle-ci. Nous avons voulu transformer le choix de ces paramètres en atout plutôt qu'en embarras. Ayant alors des paramètres à optimiser et des critères d'optimisation nous avons pu appliquer les algorithmes d'EA, ayant rappelé par ailleurs que si leur application est rarement triviale, ils offrent souvent la possibilité de venir à bout de

problèmes d'optimisation pour lesquels on ne dispose pas d'algorithme simple et efficace...Nous insisterons d'abord sur la manière dont nous avons combiné des simulations parallèles avec un « serveur » génétique afin d'effectuer un échantillonnage tout en optimisant simultanément les fréquences des mouvements de Monte Carlo. Ainsi, il n'y a pas de temps perdu pour l'optimisation proprement dite, car elle se fait en cours de simulation même. **C'est à ce titre que l'on peut qualifier notre algorithme d'outil de contrôle adaptatif**, car il a pour but d'améliorer les performances d'un système en cours de simulation et non de trouver un optimum global a posteriori. »

D'autres études utilisant l'évolution artificielle ont été faites à l'IFP, j'ai cité celles dans lesquelles j'ai été le plus impliqué.

#### 4.2.4 Génération automatique, métamodélisation

Dans le chapitre introductif du rapport OFTA sur les architectures logicielles et la réutilisation de composants, nous écrivions, avec Michel Gondran :

« La montée vers les niveaux de modélisation et de métamodélisation est un facteur clé de la réutilisation. Les changements de représentation sont la méthodologie permettant ce passage à niveaux d'abstraction plus élevé. UML et les métamodèles de l'OMG (MOF) sont les outils incontournables. XML et ses dérivés - notamment XMI - vont jouer un rôle prépondérant dans l'expression des modèles et métamodèles. La flexibilité des applications proviendra de ces approches d'abstraction par modélisation et métamodélisation, puis par génération de codes. »

Notre expérience pratique de la montée vers la métamodélisation est encore limitée, à comparer par exemple à celle du LIP6 avec MétaGen [Perrot 95], et surtout n'est pas basée sur les niveaux d'abstraction de l'OMG, qui sont apparus postérieurement ; je dirai simplement ici que nous avons utilisé des techniques de génération automatique essentiellement dans le domaine de la supervision de procédés, afin de permettre à nos systèmes, devant fonctionner au plus près des installations, d'être facilement modifiés lorsque des modifications physiques sont faites sur celles-ci. Dans cette optique, l'utilisation de techniques de génération de codes - essentiellement pour le diagnostic et le filtrage d'alarmes - permet de réaliser automatiquement des systèmes qui s'adaptent relativement aisément aux variations de l'environnement. Ces travaux ont été décrits dans la thèse de Sylvie Cauvin [Cauvin 95t] et dans des publications plus récentes comme celles de Bruno Heim [Heim 00]. Une idée directrice omniprésente dans ces travaux est de générer le système de diagnostic à partir de représentations graphiques « sachantes » du procédé, et à partir de connaissances génériques sur des structures récurrentes de diagnostic.

#### 4.2.5 Multi-agents

Exception au préambule de cette section, je donne ci-après ma vision synthétique de ce que sont les systèmes multi-agents, puisque leur utilisation va être développée au

chapitre suivant. Il faut également dire que cette utilisation ne fait que débiter à l'IFP : je n'ai pas d'exemple historique à présenter.

Un agent est un programme autonome qui peut communiquer et coopérer avec d'autres agents; il se situe dans un environnement, il a une identité, un comportement individuel et social.

Un système multi-agents est composé de plusieurs agents qui sont en relation. Cela semble évident, mais il ne faut pas confondre les agents autonomes, qui représentent par exemple un utilisateur pour faire des transactions, avec les systèmes multi-agents qui mettent vraiment en œuvre les phénomènes collectifs. On parle ainsi d'agents de recherche sur l'internet, mais cela implique rarement une collectivité d'agents. Les spécialistes des SMA (Systèmes Multi-Agents) s'intéressent aux dialogues entre agents, aux groupes d'agents et aux rôles que peuvent jouer les agents dans ces groupes, à l'émergence de propriétés et de comportements, et plus généralement à tout ce qui fait l'intérêt et la difficulté de mettre ensemble une population de tels agents.

Dans le cadre de cette recherche sur les collectivités d'agents, la recherche sur les comportements individuels est nécessaire, puisque c'est à la base de ces comportements individuels que le travail de groupe va s'opérer. On distingue ainsi les agents réactifs des agents cognitifs. Les premiers ont des fonctionnements simples de réponses à des stimuli. La richesse du multi-agents proviendra alors de l'établissement de mécanismes de collectivité. La métaphore la plus utilisée est celle de l'agent fourmi, qui se déplace dans un environnement en ramassant de la nourriture et en déposant des phéromones. Les agents cognitifs ont en eux des mécanismes de perception, de décision, et souvent d'apprentissage. La richesse du multi-agents provient alors de la combinaison variable de ces mécanismes complexes en relation. La métaphore est plutôt humaine, elle passe par des rôles, des structures, des groupes, et de la négociation.

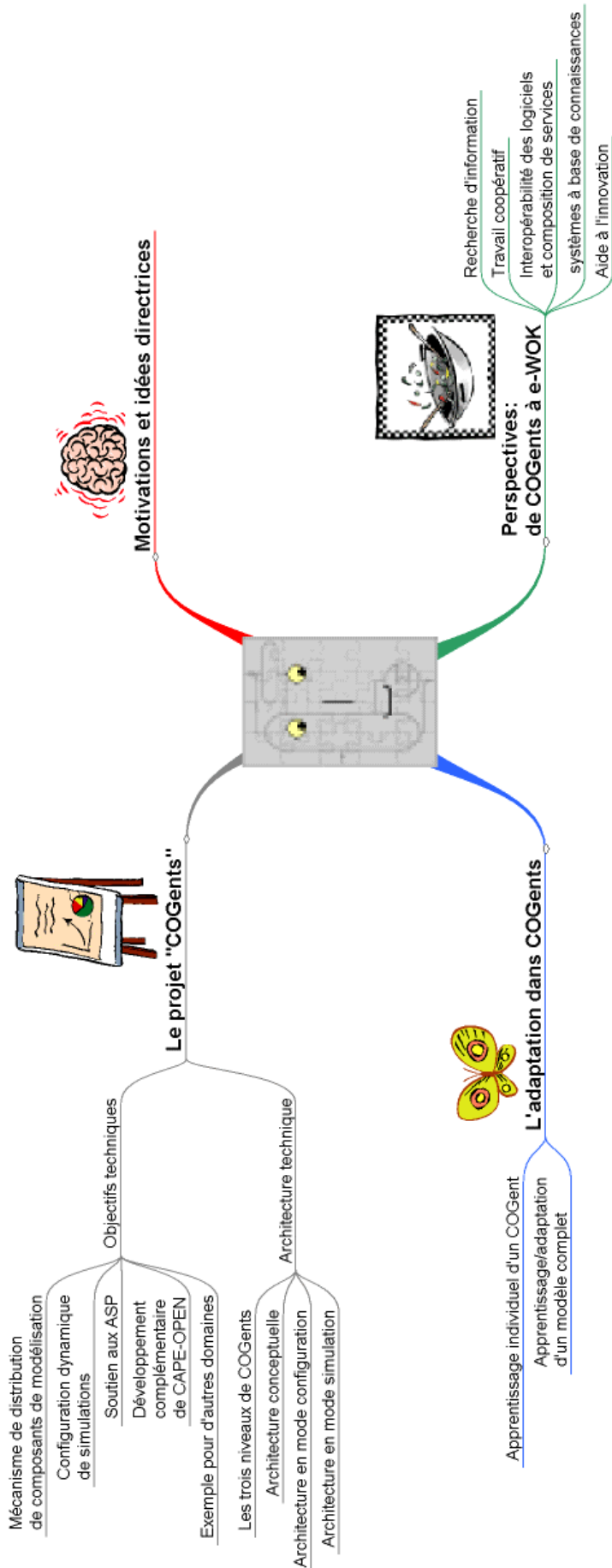
Pour permettre l'exécution de ces systèmes, des plates-formes multi-agents sont développées. Ce sont des outils sophistiqués, qui permettent de créer des agents logiciels, de les mettre en service, de les connecter à un environnement, de les faire communiquer entre eux par messages, et d'observer le système. La communication à l'intérieur d'une plate-forme est définie par son créateur et n'a pas besoin de correspondre à un standard particulier. Mais de plus en plus, on cherche à réaliser des applications mettant en jeu plusieurs plates-formes distribuées, par exemple sur l'internet. Dans ce cas, il faut que les messages échangés soient compréhensibles par tous. C'est pour cela que se développent des standards de messagerie entre agents. En retour, ces standards peuvent aussi être utilisés pour la communication à l'intérieur d'une plate-forme, cela simplifie le travail.

Technologiquement, les SMAs utilisent les outils informatiques les plus récents, notamment en raison de leur utilisation sur l'internet. Les plates-formes et les applications sont en Java, C++ ou Smalltalk, la communication utilise les middleware CORBA, JavaBeans ou COM, XML est pris comme langage de description de données et de messages, les méthodologies reposent sur UML. On est en univers familier et au goût du jour.

Je voudrais terminer en illustrant le principe du SMA avec un tout petit exemple jouet, les agents "PingPong", un des exemples de la plate-forme MADKIT [Gutknecht 00]. Un tel agent ne sait faire que peu de choses, chercher des partenaires et émettre des



messages "Balle". On crée un agent dans la plate-forme. Seul, il ne fait rien, il est au repos et attend des partenaires. Dès qu'un deuxième agent est créé, les deux se reconnaissent, et chacun émet le message "balle" en réponse à l'autre. Au bout d'un nombre déterminé de messages, le jeu s'arrête. Cet exemple illustre - très simplement, j'en conviens - l'émergence d'une propriété nouvelle, ici, la partie de ping-pong, dans un SMA. Imaginez maintenant un agent "réacteur", un agent "distillation", un agent "thermodynamique" et un agent "solveur", capables de se reconnaître les uns les autres et de composer une simulation de procédé. Voilà, c'est un petit bout de la puissance des SMA. L'exemple n'est pas innocent, c'est celui que je développe au chapitre suivant: allons-y!



## 5. Une application/un projet: les COGents, agents logiciels intéropérables pour la simulation de procédés

Ce chapitre est consacré à mon nouveau projet, COGents - concaténation de "CO" pour CAPE-OPEN et de "Gents" pour agents. Le mot "cogent", en anglais, signifie "convaincant" pour un argument. J'espère que je saurai convaincre le lecteur de l'intérêt de cette approche. Je donne également la nouvelle définition du terme, définition que nous utilisons pour le projet. Je les donne en anglais d'abord, puis en français.

**co.gent** (-jənt): 1. *adj.* [L. *cogens*, prp. of *cogere*, to collect <*co-*, together + *agere*, to drive] forceful and to the point, as a reason or argument; convincing - **SYN.** See VALID - **co'gent-ly** *adv.*

2. *n. pl.* [**CSG.** <*CO-*, abbrev. of CAPE-OPEN, + -*Gents*, **IST.** agents], new generation of simulation software based on cognitive and distributed heterogeneous software components.

**co.gent** (-jənt): 1. *adj.* [L. *cogens*, prp. de *cogere*, collecter <*co-*, ensemble + *agere*, mener] qui fait force, pour une raison ou un argument; convaincant; - **SYN.** voir VALIDE - **co'gent-iment** *adv.*

2. *n. pl.* [**CCD.** <*CO-*, abrégé. de CAPE-OPEN, + -*Gents*, **SIC.** agents], nouvelle génération de codes de simulation basés sur des composants logiciels hétérogènes, cognitifs et répartis.

COGents est à la fois un objectif, le sujet de thèse d'Othmane Nadjemi à l'IFP, et un petit<sup>17</sup> projet de recherche et développement financé par la Commission Européenne dans le programme IST (Information Society Technologies). Je présente les motivations et idées directrices de ce projet, les perspectives qu'il offre, et les liens avec les lourdes tendances actuelles en matière de logiciel.

### 5.1 Motivations et idées directrices de COGents

Avec l'avènement des architectures d'interopérabilité de composants en simulation de procédés, il est possible à un utilisateur d'assembler une simulation en combinant des composants aux interfaces standardisées, au sein d'un environnement « EMP »

---

<sup>17</sup> Budget d'environ 1,2 M€

compatible. Cet environnement est appelé « COSE », CAPE-OPEN Simulation Executive, sur la figure 8, qui illustre cette nouvelle manière de réaliser des simulateurs, dans laquelle le COSE a une place centrale.

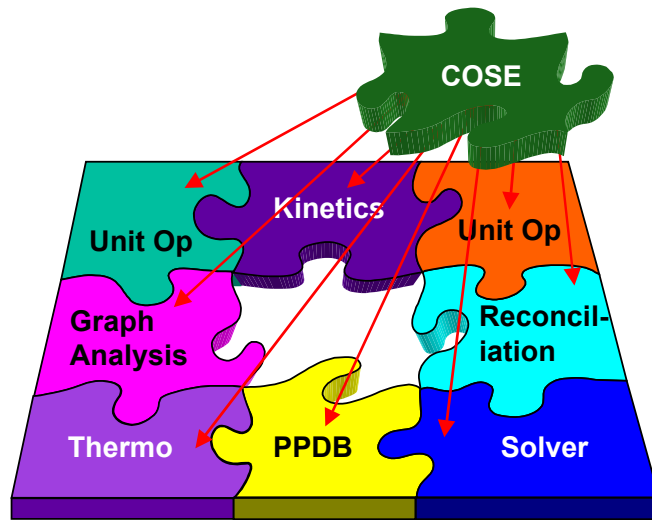


Figure 8: Architecture "conventionnelle" à base de composants pour la modélisation de procédés

Je propose maintenant de dépasser cette organisation en donnant à chacune des pièces de cet assemblage un comportement autonome d'agent logiciel, capables de se coordonner pour résoudre un problème de simulation posé par un utilisateur. L'illustration suivante, inspirée d'Alexis Drogoul, LIP6, groupe OASIS [Drogoul 02], donne une idée du but que je cherche à atteindre:

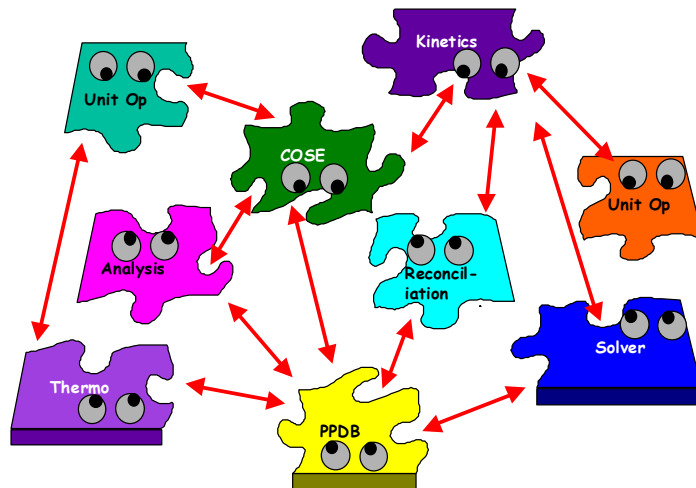


Figure 9: Une architecture décentralisée pour la modélisation de procédés

Au moment où se termine la série des deux projets CAPE-OPEN et Global CAPE-OPEN, nous avons à notre disposition les résultats suivants :

- ◆ un standard d'interopérabilité dans le domaine de la simulation de procédés, valable aussi bien pour les CMP que pour les EMP ;

- ◆ un certain nombre de composants et d'environnements ayant été rendus conformes au standard, en particulier par leur « encapsulation » avec une couche logicielle leur conférant les interfaces nécessaires ;
- ◆ des outils de tests produits par l'organisme CO-LaN, capables de tester la conformité des CMPs et EMPs au standard ;
- ◆ un catalogue de composants compatibles, destiné à rendre publique sur l'internet la disponibilité des composants certifiés, y compris le description sommaire de leurs principales fonctionnalités;
- ◆ les prémices d'un langage de représentation des procédés, OntoCAPE, indépendant des langages d'implémentation fournis par les EMPs;
- ◆ enfin, des principes et des architectures permettant de concevoir et réaliser des systèmes multi-agents, et qui sont basés sur des standards logiciels proches de ceux utilisés dans les spécifications CAPE-OPEN.

Il s'agit maintenant d'utiliser l'ensemble de ces briques de base pour concevoir le système multi-agents, que je décris de la manière suivante :

- ◆ Un utilisateur décrit son problème de simulation en utilisant le langage général OntoCAPE qui permet de définir l'objet de la simulation et les données connues ;
- ◆ Le problème est transmis à un EMP capable de dialoguer avec le catalogue de composants compatibles ;
- ◆ L'EMP transmet le problème au catalogue qui entame à son tour un dialogue avec les 'cogents' (composants agentifiés) susceptibles d'être utilisés dans la solution informatique ;
- ◆ Il se forme ainsi un ou plusieurs réseaux de 'cogents' qui vont tenter de simuler le procédé ;
- ◆ Les simulations sont éventuellement confrontées aux données expérimentales disponibles, et une sélection s'effectue sur la base de la qualité de restitution de ces données ; cette sélection peut elle-même faire intervenir d'autres agents d'observation et d'optimisation ;
- ◆ Le modèle de procédé est le résultat de cette sélection.

Ce scénario pose plusieurs problèmes de recherche: conception du langage OntoCAPE pour les besoins de l'application; introduction dans un EMP des fonctionnalités de dialogues avec les catalogues de composants compatibles ; introduction des fonctions agent dans les composants, le catalogue, et l'EMP ; sélection sur données expérimentales; adaptation éventuelle des réseaux et des agents par apprentissage, etc. Il demande également une réflexion et une conception architecturale pour bénéficier au mieux de l'important existant en matière d'intéropérabilité et de technologies multi-agents. Je vais développer quelques idées ci-dessous.

Les technologies pour réaliser ce scénario existent et ne demandent qu'à être utilisées:

- ◆ Langages description d'ontologies de domaine pour représenter simplement les concepts manipulés : le langage DAML+OIL, utilisant XML, s'impose actuellement comme standard pour les ontologies dont a besoin le web sémantique;
- ◆ Langages de descriptions de services logiciels : WSDL (Web Services Description Language), WSFL (Web Services Flow Language), eux aussi utilisant XML, sont de bons candidats pour l'échange de fonctionnalités sur le web;
- ◆ Catalogues de services : le système d'enregistrement et de découvertes de services UDDI (Universal Description, Discovery and Integration) permet de réaliser de tels catalogues de services interrogeables par les clients logiciels potentiels;
- ◆ Plates-formes multi-agents : ces plates-formes, et notamment celles qui sont compatibles avec les standards FIPA (Foundation for Intelligent Physical Agents), permettent à des agents logiciels intelligents de communiquer entre eux indépendamment de leur implémentation et de leur localisation, à condition qu'ils partagent la même ontologie de domaine, pour se comprendre;
- ◆ Business rules : les COGents doivent conduire des raisonnements à plusieurs niveaux, sur la constitution du besoin de modélisation, sur l'appariement entre le besoin et les services proposés, comme dans [Sycara 99], sur la composition dynamique de services. Pour cela, les environnements à base de règles tels que CLIPS ou JESS permettent de doter les agents de comportements "raisonnés" - à défaut de pouvoir être réellement qualifiés d'intelligents;
- ◆ Apprentissage : c'est par l'introduction de fonctions adaptatives que le système peut accroître sa capacité de résolution de problèmes: adaptation des composants eux-mêmes par optimisation de leurs paramètres sur un ensemble de problèmes donnés ; adaptation des modèles par optimisation de l'ensemble des composants utilisés pour l'implémentation, par exemple en utilisant des algorithmes d'évolution artificielle sélectionnant et recombinaient des populations de modèles potentiels pour un problème donné.
- ◆ Et enfin, bien entendu, le standard d'interopérabilité sans lequel rien ne serait possible...

## 5.2 Le projet "COGents"

COGents, projet "IST 2001", financé par la Commission Européenne, a débuté en avril 2002 pour deux ans.

Ce que le projet propose est une approche complètement nouvelle de la simulation numérique à l'aide de la technologie multi-agents, des composants logiciels et de l'internet, avec un exemple dans le domaine la modélisation de procédés et du standard CAPE-OPEN. En d'autres termes, nous démontrerons l'utilisation des agents cognitifs pour l'interopérabilité dynamique et opportuniste de composants CAPE-OPEN distribués sur l'internet, facilitant ainsi la fourniture de services d'application (*ASP: application services provision*) en modélisation de procédés, dans un cadre générique qui peut être appliqué à d'autres domaines de la simulation numérique.

Le consortium COGents est composé:

- de l'IFP en tant que coordonnateur technique du projet, et de sa filiale, RSI, fournisseur de logiciels de simulation, pour développer quelques cogents;
- d'Hyprotech, fournisseur d'environnements de simulation, qui a participé aux projets CAPE-OPEN et qui a une offre commerciale basée sur cette technologie; Hyprotech adaptera son environnement Hysys au cadre proposé et développera quelques cogents;
- de trois universités, RWTH.LPT, UCL et LIP6, spécialistes de modélisation de procédés (les deux premiers) et de technologies multi-agents (LIP6); les deux premières universités développent des cogents spécialisés et adaptent leur environnement de modélisation; le LIP6 apporte la technologie multi-agents et les logiciels correspondants, en particulier la plate-forme DIMA [Guessoum 96].

### 5.2.1 Les objectifs techniques du projet

Lorsque je présente le projet COGents, je mentionne en général les cinq objectifs techniques suivants :

Nouveau mécanisme de distribution de composants de modélisation: Les principaux fournisseurs de logiciel de simulation de procédés offrent les interfaces CAPE-OPEN dans leurs environnements, facilitant ainsi l'interopérabilité avec des composants externes. Mais leur mode de distribution n'a pas changé, il est basé sur des licences logicielles conventionnelles. Nous cherchons à expérimenter un autre mode, en permettant à ces composants de simulation d'être distribués référencés sur l'internet et sur les intranets, dans le but de fournir des services ASP en modélisation.

Configuration dynamique de simulations, basée sur la connaissance: pour permettre cette interopérabilité dynamique et opportuniste, nous définissons des représentations des connaissances de modélisation et de composition de services. Ces représentations seront sous la forme d'une ontologie de la modélisation de procédé [von Wedel 00], OntoCAPE, au-dessus de XML. Ces représentations fourniront un complément sémantique au standard CAPE-OPEN qui est purement syntaxique, qui ne définit que la syntaxe des entrées-sorties des composants. Cette ontologie sera à la base d'un processus d'appariement entre les besoins des utilisateurs et les services disponibles.

Soutien aux ASP grâce à des agents middleware cognitifs: au-dessus de ces représentations, nous développerons des agents cognitifs personnalisés, en encapsulant des composants par des agents, leur déléguant ainsi la représentation du composants dans le processus d'assemblage. Ces agents cognitifs contiendront des connaissances explicites de simulation, leur permettant ainsi de négocier pour établir la configuration de simulation la plus apte à traiter un problème posé par un utilisateur.

Développement complémentaire du standard CAPE-OPEN: le développement du standard est sous la responsabilité du CO-LaN, organisme de maintien et de diffusion, dont j'ai parlé au chapitre 4. Dans COGents, nous en développons une dimension différente, celle de la sémantique et de la connaissance. Une fois que nous aurons démontré le concept, nous espérons que le CO-LaN reprendra l'idée à son compte pour l'appliquer à l'ensemble des interfaces, selon le modèle que nous aurons défini.

Exemple pour d'autres domaines de simulation: le projet COGents est intéressant pour d'autres domaines, de deux manières. Premièrement, une architecture d'ASP basée sur des agents logiciels personnalisés peut servir d'exemple concret pour d'autres. C'est vrai dans plusieurs domaines d'application en simulation numérique, par exemple la CAO, la mécanique des fluides, ou la simulation environnementale. Deuxièmement, certains des éléments techniques du projet peuvent être réutilisés dans d'autres secteurs: la combinaison des interfaces standard exprimées en COM et CORBA, d'une ontologie comme modèle de connaissances, et des systèmes et plates-formes multi-agents.

## 5.2.2 Architecture technique de COGents

Je décris l'architecture technique telle que définie au début du projet. Au moment où j'écris ce chapitre, des petites maquettes de faisabilité ont été développées, mais l'ensemble n'a pas encore été testé, l'essentiel du travail est devant nous!

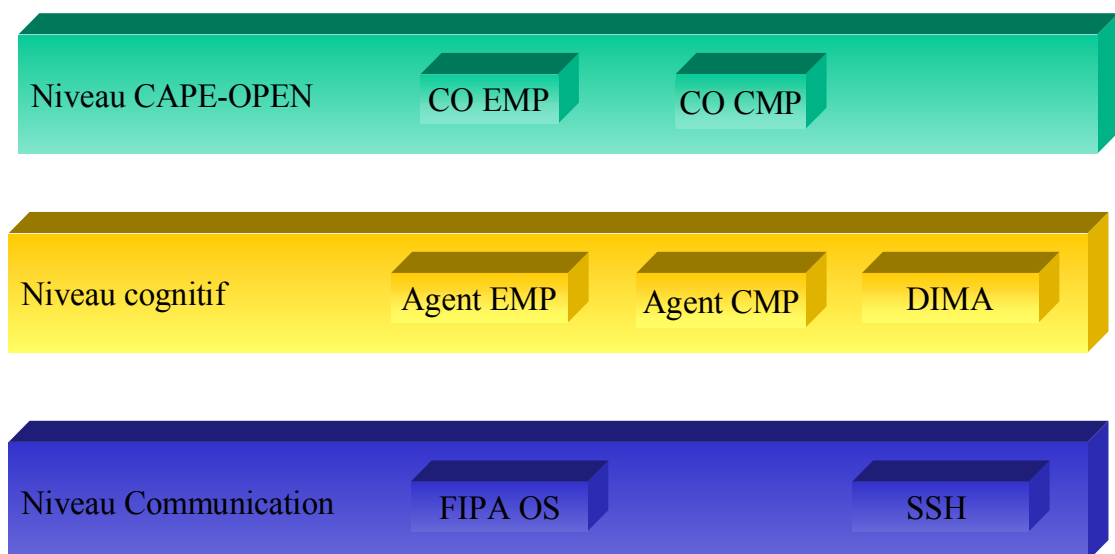


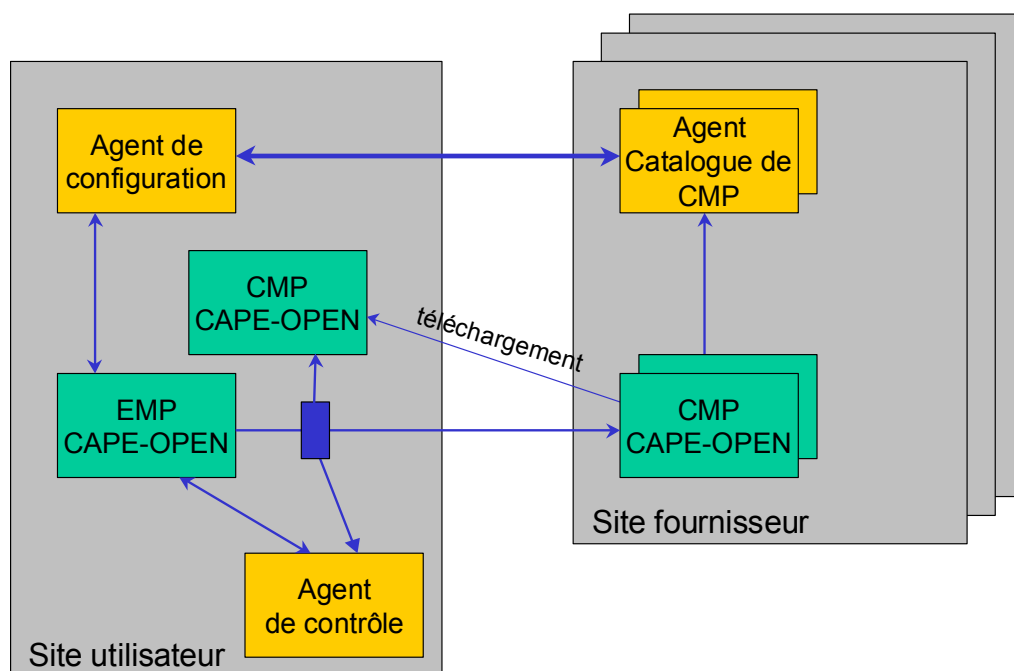
Figure 10: Les trois niveaux de COGents



COGents peut être vu comme la combinaison de trois niveaux:

- ◆ le niveau CAPE-OPEN, est le niveau technique de l'interopérabilité entre composants. On y retrouve les EMPs et les CMPs, assemblés pour réaliser la simulation;
- ◆ le niveau cognitif est celui où interagissent les "cogents" au moyen de la plate-forme multi-agents. On y retrouve les agents encapsulant les EMPs et les CMPs, et d'autres agents cognitifs qui négocient entre eux;
- ◆ le niveau de la communication, celui de FIPA-OS et du protocole de sécurité SSH, est celui qui assure les échanges entre agents dans le processus de configuration.

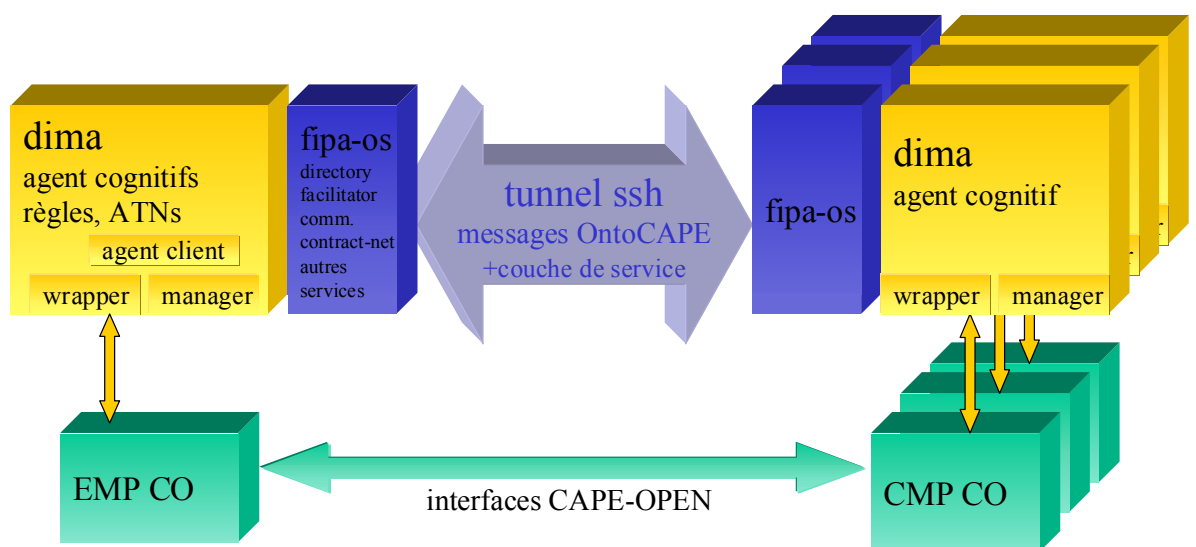
Les couleurs utilisées, jaune, bleu, vert, sont gardées dans les figures suivantes. Je présente maintenant une vision conceptuelle de COGents.



**Figure 11: architecture conceptuelle de COGents**

L'utilisateur est en relation avec un agent de configuration, lui-même représentant l'EMP ; l'agent de configuration dialogue avec des agents catalogues de CMPs, qui sont en contact avec les composants ; une fois la sélection effectuée, le composant choisi peut être téléchargé sur le site utilisateur ; le processus est répété jusqu'à obtention d'un simulateur complet. Pendant la simulation, un agent de contrôle observe les échanges aux bornes des composants et de l'EMP, et peut ainsi fournir un diagnostic en cas de problèmes - comme par exemple un problème numérique de convergence.

Je termine cette section par une vision technique plus détaillée, dans laquelle les outils et les méthodes sont précisées. C'est l'objet des deux figures suivantes, la première en mode «configuration» , la deuxième en mode «simulation».



**Figure 12: Architecture de COGents en mode configuration**

- ◆ Niveau cognitif : les agents sont implémentés à l'intérieur de la plate-forme multi-agents DIMA [Guessoum 96]; leur comportement cognitif est assuré essentiellement de deux manières : au moyen d'ATNs (Augmented Transition Networks), qui représentent les états possibles des agents : initialisé, en recherche de CMP, configuré, paramétré, en simulation etc..., et les conditions de passage entre états ; au moyen de règles de production représentant des connaissances du domaine<sup>18</sup>. Il y a trois types principaux d'agents : les agents clients, qui représentent les exigences de l'utilisateur ; les agents «manager», qui effectuent la sélection, gèrent les relations avec les catalogues etc. ; et les agents «wrapper», des agents techniques qui servent d'interface de communication avec les CMPs .

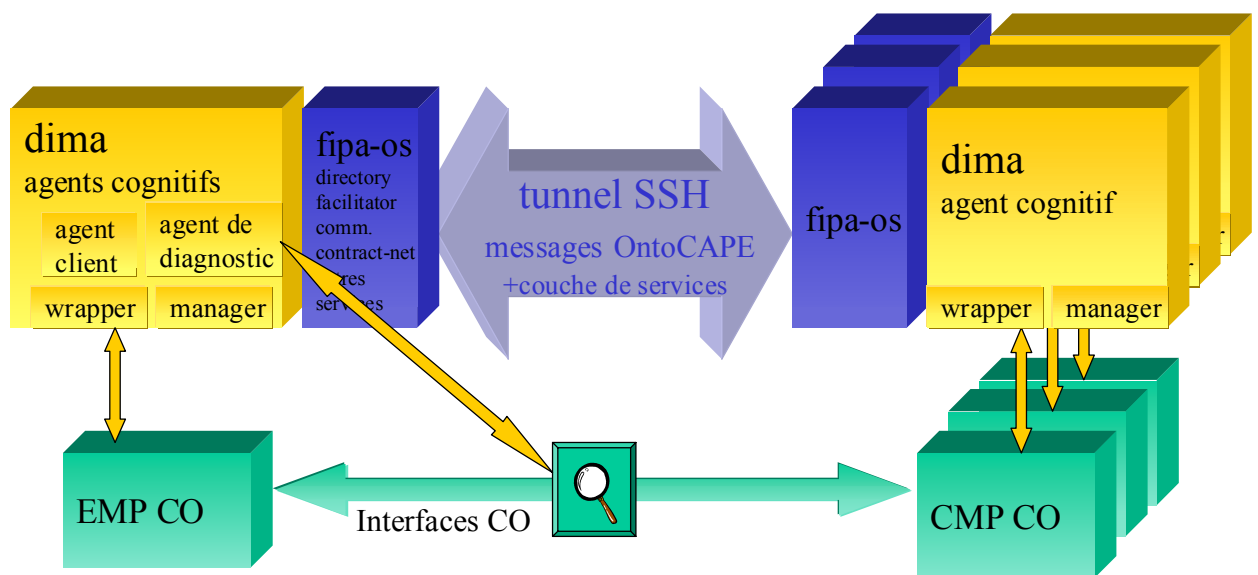
<sup>18</sup> Rien n'empêche d'introduire d'autres processus cognitifs, comme du raisonnement par cas, etc., dans les agents. J'ai d'ailleurs un projet en cours, en relation avec Loughborough University, dans ce sens. Je mentionne ici les processus les plus fréquemment utilisés dans les COGents.

- ◆ Niveau CAPE-OPEN : je n'ajoute rien par rapport à ce qui a été dit plus haut. C'est le niveau où se fait la simulation.
- ◆ Niveau communication : il est composé de plusieurs éléments. Les services de communications entre agents sont assurés par le FIPA-OS, un protocole de respectant le standard FIPA [FIPA 02], dans lequel on trouve un langage de communication entre agents, ACL, un service de catalogue, le protocole de négociation ContractNet, et d'autres services . Le transport des messages exprimés en ACL est effectué à l'intérieur d'un «tunnel SSH», système de cryptage qui assure la confidentialité des informations échangées . Enfin, le contenu des messages est double : les informations sémantiques du domaine<sup>19</sup> sont exprimées par référence à l'ontologie OntoCAPE , et les informations de service<sup>20</sup> sont exprimées au moyen d'un langage de service, en l'occurrence le langage «SL» qui fait également partie de la norme FIPA. Je ne donne pas plus de détails sur tous ces éléments, d'une part pour ne pas alourdir inutilement le mémoire, et d'autre part parce que tout est à faire ! Par contre, je complète par une présentation de la même architecture, en mode *run-time*.

---

<sup>19</sup> «je suis un modèle de réacteur adiabatique d'hydrogénation des C4, fonctionnant en stationnaire ou en dynamique, dans des conditions de température et de pressions telles que ... »

<sup>20</sup> « j'ai besoin d'un modèle d'opération unitaire », « utilise-moi » ...



**Figure 13: Architecture technique de COGents en mode simulation**

Par rapport au mode configuration, le mode simulation ajoute deux éléments : un observateur, qui capte les informations échangées aux bornes de l'EMP et des composants, et un ou plusieurs agents de diagnostic, qui utilisent ces informations pour étudier le comportement du système et l'améliorer.

L'observation des échanges aux bornes des composants impose que les composants soient légèrement modifiés pour fournir les données nécessaires au diagnostic. Le fait que les composants respectent la norme CAPE-OPEN standardise la production de ces données, notamment au moyen de l'interface «Report», qui fait partie de la norme.

### 5.3 L'adaptation dans COGents

Le projet européen COGents traite directement d'adaptation, de par son principe de sélection des composants logiciels les mieux adaptés à réaliser une simulation

particulière. La partie diagnostic en temps réel, qui peut conduire à des modifications de paramètres ou de structure, entre également dans ce cadre. Cependant, le système ne comporte pas explicitement de volet d'apprentissage. Je présente dans les quelques lignes qui suivent, des idées sur l'intégration d'un tel mécanisme dans l'architecture. Ces idées devraient être développées dans la deuxième partie de la thèse d'Othmane Nadjemi, et éventuellement prolongées dans un autre cadre qui reste à définir.

Le type d'adaptation que je présente ici est différent et complémentaire de celui récemment présenté par Wolfgang Marquardt [Marquardt 02], qui s'intéresse à l'adaptation des algorithmes numériques en modélisation de procédés. Dans ce travail, Marquardt examine essentiellement comment adapter le niveau de résolution numérique au problème rencontré, au moyen de raffinements de modélisation, d'approche multi-échelle, ou d'introduction de mécanismes de feedback.

J'imagine principalement deux types, ou plutôt deux objectifs, d'apprentissage : l'apprentissage individuel d'un *COGent*, qui cherche à s'améliorer ou à améliorer la connaissance qu'il a de lui-même ; l'apprentissage ou adaptation d'un modèle complet, composé d'un réseau de *COGents*, qui cherche à améliorer la qualité de la simulation du procédé. Sur ces deux sujets, ici encore, je vais faire court, dans la mesure où ce ne sont que des idées préliminaires. Je ne parle pas des techniques et des représentations, qui ne posent pas de problèmes particuliers; je m'intéresse plutôt au cadre possible pour ces apprentissages.

### 5.3.1 Apprentissage individuel d'un *COGent*

Un *COGent* est composé de deux parties : le composant de modélisation - opération unitaire, serveur thermodynamique, schéma cinétique, solveur, base de données etc. - et son agent «wrapper» qui le représente dans les discussions de configuration.

L'apprentissage ou l'adaptation du composant de modélisation ne possède pas de caractère particulièrement original. Il s'agit d'une pratique assez courante, soit en matière d'optimisation numérique de paramètres, comme pour les opérations unitaires, soit plus largement comme dans le cas de l'apprentissage de modèles thermodynamiques (voir au chapitre 5, évolution artificielle), ou dans l'optimisation de schémas cinétiques, généralement par des méthodes d'optimisation numériques conventionnelles, mes collègues de l'IFP ayant également expérimenté les algorithmes génétiques pour ce faire.

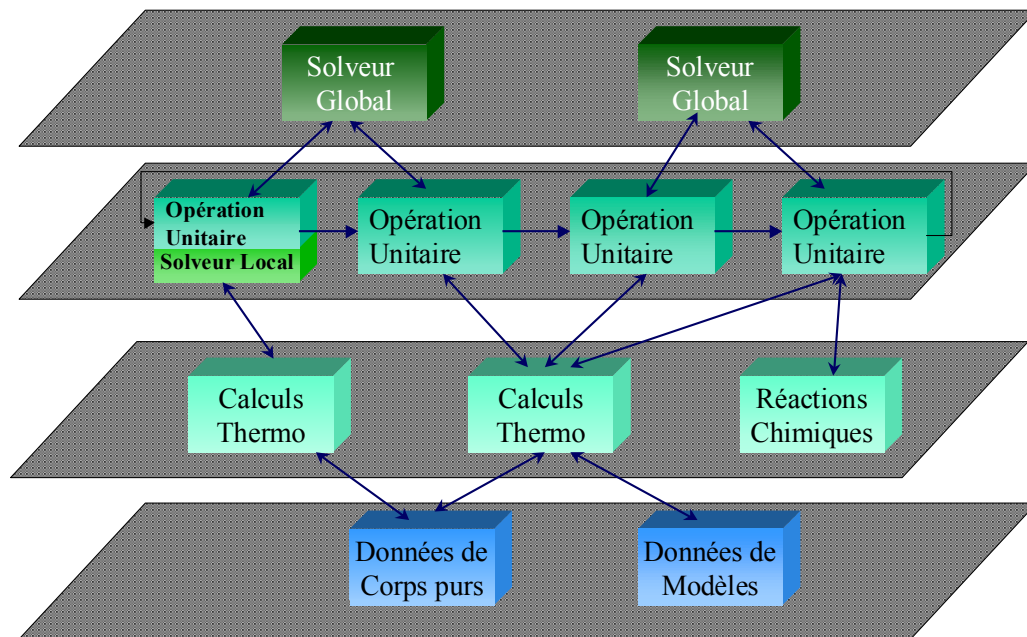
Une partie plus originale concerne l'apprentissage de l'agent wrapper, que je vois comme une évolution vers une plus grande connaissance de soi-même. Un agent cognitif représentant un *CMP* est initialement doté d'un ensemble de connaissances, fournies par son créateur, lui permettant de se décrire par rapport à un besoin exprimé. Cette description a un aspect taxinomique - à quel endroit de la taxinomie de composants de modélisation se situe-t-il - et un aspect utilitaire de modélisation - quelles sont ses capacités de modélisation.

L'apprentissage de taxinomies est un sujet traité largement par la communauté des chercheurs en intelligence artificielle. Cela suppose de disposer de grands nombres d'exemples - ici, d'un grand nombre de *COGents* - et sort donc du cadre de l'apprentissage individuel d'un *COGent*.

L'apprentissage de capacités de modélisation peut être vu comme la génération automatique de règles de production à partir d'exemples. Les exemples peuvent venir de simulations effectuées avec le COGents, en supposant les questions de confidentialité résolues, voir même de simulations «à blanc» pilotées par l'agent wrapper, afin de tester le composant dans des situations variées.

### 5.3.2 Apprentissage/adaptation d'un modèle complet

Je vois cette adaptation comme une deuxième phase de configuration d'une simulation, s'appuyant sur l'expérimentation. Pour cela j'ai besoin de représenter un modèle complet comme un réseau de COGents, ainsi que le montre la figure 14.



**Figure 14: un modèle complet, vu comme réseau de COGents**

Où l'on constate qu'un modèle peut être vu comme un graphe non planaire, reliant des composants de simulation par des flux - sur le plan des opérations unitaires, ou par des dépendances de calcul- les relations avec les trois autres plans : solveurs globaux calculant des sous-réseaux d'opérations unitaires, opérations unitaires et flux utilisant des serveurs de calculs thermodynamiques ou des systèmes de réactions chimiques, qui eux-mêmes utilisent des bases de données de modèles ou de corps purs, etc. - la figure n'étant qu'un exemple simplifié.

L'étape de configuration d'une simulation n'aboutit pas obligatoirement à une seule solution. Plusieurs COGents peuvent candidater pour une même fonction, le simulateur peut disposer potentiellement d'un grand nombre de réseaux concrets résultant de la combinatoire appliquée aux composants candidats. Vient alors assez naturellement l'idée de confronter ces réseaux à des données pour effectuer un apprentissage supervisé - les données étant des résultats de simulation désirés, par exemple obtenus par une simulation de référence, ou, bien mieux, par des expérimentations réelles sur le procédé à simuler ! Le problème d'apprentissage peut alors devenir un problème d'optimisation de configuration de graphe par rapport à un objectif, et pour cela, vue la nature

extrêmement non linéaire, non purement numérique, et non dérivable du problème, des algorithmes d'évolution artificielle me semblent particulièrement appropriés.

Une telle approche rejoint assez fortement ce que l'on appelle la synthèse de procédés : trouver une suite d'opérations unitaires et les conditions opératoires qui transforment optimalement une charge en un produit désiré, à partir d'une liste d'opérations unitaires disponibles (voir par exemple [Fraga 02] pour des approches génétiques ou MINLP ). La différence importante est que je ne me limite pas aux opérations unitaires : dans l'apprentissage de réseau de COGents, tous les composants sont mis en cause, aussi bien opérations unitaires que modèles thermodynamiques, systèmes de réactions chimiques etc. Ainsi, l'outil de synthèse de procédés Jacaranda [Fraga 00] peut lui-même être un COGent, comme nous le faisons actuellement dans le projet européen.

## 5.4 Perspectives, de COGents à e-WOK

Je conclus cette section par une brève introduction à ma proposition de projet intégré pour le prochain Programme Commun de Recherche et Développement de la Commission Européenne («FP6»). Cette proposition, et donc ce projet intégré, si son financement est accepté, va bien plus loin que COGents en matière d'ontologie et d'interopérabilité dynamique sur l'internet.

Dans la proposition « e-WOK<sup>21</sup> », mon idée est de s'appuyer sur les technologies de modélisation des données et des connaissances - ontologies, web sémantique, technologies d'interopérabilité, XML etc. - pour développer un référentiel de connaissances pour diverses applications du web sémantique, dans le domaine pétrolier et énergétique. Les applications sont listées de la plus simple à la plus complexe:

- ◆ recherche d'information: utilisation du référentiel de connaissances pour améliorer la recherche d'informations sur le web, en la basant sur une description métier ;
- ◆ travail coopératif : organiser le partage intelligent de documents et d'applications, et gérer le processus coopératif (*workflow*) en s'appuyant sur des ontologies de domaine ;
- ◆ interopérabilité des logiciels et composition de services : prolonger ce qui est fait dans COGents et l'élargir à de nouveaux types d'applications et de logiciels, dans un domaine plus vaste ;
- ◆ réalisation de systèmes à base de connaissances : utiliser les ontologies et modèles de connaissances pour réaliser des assistants logiciels d'aide à l'utilisateur ;
- ◆ aide à l'innovation : où l'on retrouve TRIZ ! baser les processus d'innovation assistée par ordinateur dans le domaine pétrolier et énergétique sur des ontologies sectorielles plus précises et plus complètes que celles que l'on trouve actuellement dans les outils commerciaux.

---

<sup>21</sup> energy, Web, Ontologies, Knowledge

Dans e-WOK, un projet pivot développera le référentiel et les outils permettant de l'exploiter ; des projets ciblés viseront les applications. Il s'agit d'un projet très ambitieux, développant les bases du « web sémantique de l'énergie », dont les résultats concrets sont à moyen et long terme. Alors, comme ce paragraphe est le dernier avant la conclusion, accordez-moi le privilège de raconter l'histoire du roi et du jardinier :

*Un roi, regardant son jardin, convoqua son jardinier. « Jardinier, j'ai envie de cerises. Demain matin, tu planteras un cerisier dans mon jardin ». Le jardinier répondit : « Mais, il faudra bien dix ans avant d'avoir des cerises ». Alors le roi lui dit : « Plante donc le cerisier cet après-midi ».*

Je n'attends pas que le web sémantique « pétrolier » arrive tout seul : je travaille pour le réaliser, et le plus tôt sera le mieux !



## 6. Conclusion

Ce mémoire présente quelques idées sur l'évolution des logiciels, et développe notamment les aspects de segmentation et de dynamisation, principalement dans les domaines des codes scientifiques. Les autres principes et lois de TRIZ y sont évoqués, essentiellement parce qu'ils poussent à la réflexion sur les méthodes et outils du logiciel, un domaine que le créateur de TRIZ n'avait pas imaginé lorsqu'il a conçu sa théorie, mais qui se révèle riche d'applications et intellectuellement stimulant.

Je suis bien conscient que ces idées sont loin des préoccupations quotidiennes des développeurs et utilisateurs des codes en question, qui héritent de décennies de développements conventionnels. Mais je suis convaincu du caractère inéluctable de l'évolution de ces codes vers plus de composants et plus de capacités d'apprentissage et d'adaptation.

L'observation des grands programmes de R&D nationaux et internationaux autour du logiciel (RNTL en France, EPSRC au Royaume-Uni, IT2/NITRD aux USA, IST en Europe, IMS dans le monde) montre que l'industrie du logiciel s'approprie les techniques de distribution de calculs et de services (les datagrids, les web services, les architectures distribuées, les couplages de codes, les middleware spécialisés etc.) , tout en accompagnant cette appropriation d'une recherche significative sur les méthodes d'apprentissage et d'optimisation permettant, on le suppose et on l'espère, de configurer au mieux, et automatiquement, les applications.

Issues du monde des systèmes d'information, ces techniques peuvent probablement, moyennant adaptation, s'appliquer au monde des logiciels scientifiques, dans lequel il ne s'agit pas seulement de combiner des services, mais il s'agit de le faire de manière correcte - au profit de la qualité scientifique des travaux - et performante - au profit de la productivité de la recherche. Faire appel à un service d'optimisation hybride non linéaire disponible sur l'internet, pour résoudre un problème de planification de production de raffinerie, ne pose pas tout à fait les mêmes questions que de combiner les offres d'une compagnie aérienne et d'une chaîne hôtelière pour composer un voyage! De même, apprendre automatiquement les préférences de lecture d'un utilisateur d'Amazon.com n'est pas tout à fait équivalent à apprendre quel modèle thermodynamique est le plus adapté pour la prédiction de formation d'hydrates dans les conduites sous-marines à partir des données de simulation et d'expérience disponibles.

J'ai toutefois tenté de montrer comment les techniques informatiques actuelles permettent d'aborder de tels sujets avec une certaine confiance. J'ai également présenté mes contributions en la matière, en distinguant les approches de segmentation, plutôt basées sur les technologies de génie logiciel, et les approches de dynamisation, plutôt à base d'outils et méthodes d'intelligence artificielle. J'ai conclu en présentant un projet de mise en œuvre de ces idées sur un cas concret, celui de la simulation de procédés par multi-agents, sujet actuellement poursuivi dans le cadre du projet européen IST "COGents" (2002-2004).

Enfin, je pense que le web sémantique (le "*web pour les machines*") et ses technologies (ontologies, XML) est le canal privilégié par lequel circulera la connaissance sur les logiciels scientifiques et leur composition dynamique opportuniste, grâce aux moyens

qu'il offre pour donner du sens aux composants logiciels disponibles en ligne. Ces agents logiciels pour la simulation numérique sont destinés à devenir des agents intelligents, capables de décider du bon module à utiliser pour la bonne application, offrant ainsi à une multitude d'utilisateurs non experts des outils de calcul réservés jusqu'ici aux *happy few*. C'est une aventure passionnante et un défi à relever: il me reste du travail pour un bon moment!

## 7. Annexes



## Annexe 1: TRIZ, ses principes et lois d'évolution, et l'ingénierie du logiciel

Que peut-on utiliser de TRIZ pour les logiciels? Peut-on simplement s'inspirer de la théorie, ou peut-on en utiliser les outils?

- les 40 principes; plusieurs d'entre eux peuvent être appliqués au logiciel, sous réserve d'un effort de transposition;
- la matrice 39X39 de résolution des contradictions techniques; elle peut être utilisée partiellement. D'une part, cette matrice utilise les 40 principes, et je viens de dire que seule une partie de ces principes est applicable. D'autre part, certains des paramètres physiques (surface d'un objet mobile, température, puissance etc.) qui constituent les lignes et colonnes de la matrice ne sont pas transposables immédiatement au logiciel, même si d'autres plus généraux (fiabilité, complexité, réparabilité etc.) le sont. Enfin, cette matrice n'est pas universelle. Elle a été établie sur la base d'observations, en grand nombre, de publications scientifiques et de brevets dans les domaines techniques. Un travail analogue devrait être fait pour les logiciels; je vais malgré tout montrer qu'en l'état, il est possible de s'inspirer de cette matrice pour définir des axes de progrès pour des logiciels;
- les huit lois d'évolution des systèmes techniques; ce sont des lois d'évolution très générales, elles peuvent être transposées moyennant un effort modeste. Mais ces lois ne donnent pas de solutions pratiques aux problèmes d'invention; elles indiquent des tendances d'évolution attendues;
- la représentation en modèles substances-champs (vépoles) et les 76 standards; la représentation elle-même offre peu d'intérêt dans le domaine qui m'intéresse, car le monde du génie logiciel est déjà bien fourni en matière de représentations graphiques, des diagrammes d'activité IDEFO ou SADT aux divers modèles de la notation UML, en passant par des approches systémiques comme SAGACE, des modèles de connaissances comme KADS [Schreiber 95], etc.
- l'algorithme ARIZ; son application est sans doute possible, mais je ne l'ai pas expérimentée.
- enfin, les bases d'effets scientifiques; premièrement, elles ne font pas réellement partie de la théorie, même si leur utilité est indiscutable; deuxièmement, leur transposition au domaine du logiciel n'a pas beaucoup de sens; il vaudrait mieux établir une base d'effets particulières dans le domaine du logiciel, ce qui reviendrait plus ou moins à établir une base d'architectures, de traitements ou d'algorithmes connus. Si, pour les systèmes d'information en général, une telle base existe (les schémas de conception ou "*design patterns*" [Gama 95]), on ne peut pas en dire autant pour le domaine des logiciels scientifiques. Voici un travail intéressant qui pourrait certainement faire l'objet d'une thèse en informatique....

Dans la suite de cette annexe, je transpose quelques principes de TRIZ au domaine du logiciel. A ma connaissance, seuls deux autres auteurs se sont essayés à cette transposition:

Kevin Rea [Rea 01a, 01b, 02] a publié une liste exhaustive des 40 principes et d'analogies pour le génie logiciel pour une trentaine d'entre eux; il déclare avoir pu déposer plusieurs brevets grâce à l'application de cette méthode; ses analogies sont intéressantes, plus près du code que les miennes;

Michael Schlueter [Schlueter 01] a utilisé des éléments de TRIZ pour résoudre un problème de programmation en PERL. Il n'utilise pas directement les 40 principes, il enchaîne plutôt les raisonnements à la manière de l'algorithme ARIZ, en y ajoutant des représentations substances-champs.

Il est possible que d'autres auteurs, notamment de l'école russe et biélorusse, fondatrice de TRIZ, aient entrepris des démarches similaires. Mes discussions avec quelques-uns d'entre eux n'ont pas permis d'identifier de référence claire sur le sujet. Si des publications ont été faites en russe, je n'ai pas pu en prendre connaissance. Je me baserai donc sur des transpositions personnelles, en comparant, lorsque cela a un sens, avec ce que propose Rea. Les deux principes qui m'intéressent le plus particulièrement sont 1, Segmentation et 15, Dynamisation, auxquels ce mémoire est consacré. Je regroupe les quelques autres ci-après. A la différence de Rea, je ne cherche pas l'exhaustivité, je veux simplement souligner et illustrer que ces principes, conçus pour des systèmes physiques, prennent également du sens dans le domaine du génie logiciel.

Le lecteur comprendra certainement que ce texte me permet également de présenter, de manière non conventionnelle, des connaissances que j'ai acquises au cours de recherches que j'ai effectuées dans le groupe Elf et à l'IFP, et qui sont évoquées plus précisément au chapitre 3.

# Les 40 principes de TRIZ (d'après [Altshuller 00])

## **Segmentation**

Divide an object into independent parts.

Make an object sectional (for easy assembly or disassembly).

Increase the degree of an object's segmentation.

## **Extraction**

(Extracting, Retrieving, Removing)

Extract the "disturbing" part or property from an object.

Extract only the necessary part or property from an object.

## **Local Quality**

Transition from homogeneous to heterogeneous structure of an object or outside environment (action).

Different parts of an object should carry out different functions.

Each part of an object should be placed under conditions that are most favorable for its operation.

## **Asymmetry**

Replace symmetrical form(s) with asymmetrical form(s).

If an object is already asymmetrical, increase its degree of asymmetry.

## **Consolidation**

Consolidate in space homogeneous objects, or objects destined for contiguous operations.

Consolidate in time homogeneous or contiguous operations.

## **Universality**

An object can perform several different functions ; therefore, other elements can be removed.

## **Nesting (Matrioshka)**

One object is placed inside another. That object is placed inside a third one. And so on...

An object passes through a cavity in another object.

## **Counterweight**

Compensate for the weight of an object by combining it with another object that provides a lifting force.

Compensate for the weight of an object with aerodynamic or hydrodynamic forces influenced by the outside environment.

## **Prior Counteraction**

Preload countertension to an object to compensate excessive and undesirable stress.

## **Prior Action**

Perform required changes to an object completely or partially in advance.

Place objects in advance so that they can go into action immediately from the most convenient location.

## **Cushion in Advance**

Compensate for the relatively low reliability of an object with emergency measures prepared in advance.

## **Equipotentiality**

Change the condition of the work in such a way that it will not require lifting or lowering an object.

## **Do It in Reverse**

Instead of the direct action dictated by a problem, implement an opposite action (i.e., cooling instead of heating).

Make the movable part of an object, or outside environment, stationary and stationary part moveable.

Turn an object upside-down.

## **Spheroidality**

Replace linear parts with curved parts, flat surfaces with spherical surfaces, and cube shapes with ball shapes.

Use rollers, balls, spirals.

Replace linear motion with rotational motion ; utilize centrifugal force.

## **Dynamicity**

Characteristics of an object or outside environment, must be altered to provide optimal performance at each stage of an operation.

If an object is immobile, make it mobile. Make it interchangeable.

Divide an object into elements capable of changing their position relative to each other.

## **Partial or Excessive Action**

If it is difficult to obtain 100% of a desired effect, achieve more or less of the desired effect.

### **Transition Into a New Dimension**

Transition one-dimensional movement, or placement, of objects into two- dimensional ; two-dimensional to three- dimensional, etc.

Utilize multi-level composition of objects.

Incline an object, or place it on its side.

Utilize the opposite side of a given surface.

Project optical lines onto neighboring areas, or onto the reverse side, of an object.

### **Mechanical Vibration**

Utilize oscillation.

If oscillation exists, increase its frequency to ultrasonic.

Use the frequency of resonance.

Replace mechanical vibrations with piezo-vibrations.

Use ultrasonic vibrations in conjunction with an electromagnetic field.

### **Periodic Action**

Replace a continuous action with a periodic one (impulse).

If the action is already periodic, change its frequency.

Use pauses between impulses to provide additional action.

### **Continuity of Useful Action**

Carry out an action without a break. All parts of the objects should constantly operated at full capacity.

Remove idle and intermediate motion.

Replace "back-and-forth" motion with a rotating one.

### **Rushing Through**

Perform harmful and hazardous operations at a very high speed.

### **Convert Harm Into Benefit**

Utilize harmful factors - especially environmental - to obtain a positive effect.

Remove one harmful factor by combining it with another harmful factor.

Increase the degree of harmful action to such an extent that it ceases to be harmful.

### **Feedback**

Introduce feedback

If feedback already exists, change it.

### **Mediator**

Use an intermediary object to transfer or carry out an action.

Temporarily connect the original object to one that is easily removed.

### **Self-service**

An object must service itself and carry-out supplementary and repair operations.

Make use of waste material and energy.

### **Copying**

A simplified and inexpensive copy should be used in place of a fragile original or an object that is inconvenient to operate.

If a visible optical copy is used, replace it with an infrared or ultraviolet copies.

Replace an object (or system of objects) with their optical image. The image can then be reduced or enlarged.

### **Dispose**

Replace an expensive object with a cheap one, compromising other properties (i.e., longevity).

### **Replacement of Mechanical System**

Replace a mechanical system with an optical, acoustical, thermal or olfactory system.

Use an electric, magnetic or electromagnetic field to interact with an object.

Replace fields that are Stationary with mobile

Fixed with changing in time.

Random with structured.

Use fields in conjunction with ferromagnetic particles.

### **Pneumatic or Hydraulic Constructions**

Replace solid parts of an object with a gas or liquid. These parts can now use air or water for inflation, or use pneumatic or hydrostatic cushions.

### **Flexible Membranes or Thin Films**

Replace customary constructions with flexible membranes or thin film.

Isolate an object from its outside environment with flexible membranes or thin films.

### **Porous Material**

Make an object porous, or use supplementary porous elements ( inserts, covers, etc.).



If an object is already porous, fill pored in advance with some substance.

#### **Changing the color**

Change the color of an object or its environment.

Change the degree of translucency of an object or its environment.

Use color additives to observe an object or process which is difficult to see.

If such additives are already used, employ luminescent traces or trace atoms.

#### **Homogeneity**

Objects interacting with the main object should be made out of the same material (or material with similar properties) as the main object.

#### **Rejecting and Regenerating Parts**

After completing its function, or becoming useless, an element of an object is rejected (discarded, dissolved, evaporated, etc.) or modified during its work process.

Used-up parts of an object should be restored during its work.

#### **Transformation of Properties**

Change the physical state of the system.

Change the concentration or density.

Change the degree of flexibility.

Change the temperature or volume.

#### **Phase Transition**

Using the phenomena of phase change (i.e., a change in volume, the liberation or absorption of heat, etc.).

#### **Thermal Expansion**

Use expansion or contraction of material by changing its temperature.

Use various materials with different coefficients of thermal expansion.

#### **Accelerated Oxidation**

Make transition from one level of oxidation to the next higher level :

Ambient air to oxygenated;

Oxygenated to oxygen.

Oxygen to ionized oxygen.

Ionised oxygen to ozoned oxygen.

Ozoned oxygen to ozone.

Ozone to singlet oxygen.

#### **Inert Environment**

Replace a normal environment with an inert one.

Introduce a neutral substance or additives into an object.

Carry out the process in a vacuum.

#### **Composite Materials**

Replace homogeneous materials with composite ones.

## Transposition de quelques principes

Segmentation et dynamisation ne sont pas les seuls principes applicables à l'ingénierie logicielle. J'en présente ici quelques autres, en ordre de numérotation croissant.

### Local Quality/Architectures distribuées

#### 3. Local Quality

- a. Transition from homogeneous to heterogeneous structure of an object or outside environment.
- b. Different parts of an object should carry out different functions.
- c. Each part of an object should be placed under conditions that are most favorable for its operation.

Le principe 3 propose de rendre les systèmes hétérogènes, en identifiant les fonctions à réaliser, et à les affecter à diverses parties du système. Les architectures distribuées hétérogènes, client-serveur, n-tier, viennent directement à l'esprit. Avec ces architectures, et notamment avec CORBA, un système peut être distribué, hétérogène, fait de composants intéropérables écrits dans plusieurs langages, réalisant des fonctions complémentaires, et localisés sur des machines spécifiques en fonction des besoins.

Je prends un exemple en modélisation de procédés, relevant de la section 3 sur la segmentation dans CAPE-OPEN: un système de simulation peut être constitué d'une interface homme-machine écrite en Java, accessible sur un PC ou sur un assistant personnel; d'un environnement de modélisation développé en C++ et tournant sur PC; d'un module de mécanique des fluides écrit en FORTRAN et localisé sur une station de travail performante ou sur un calculateur parallèle; et d'une base de donnée de propriétés physiques et thermodynamiques située sur une machine départementale en raison de son utilisation par de nombreux clients. Dans cette architecture, les objets sont hétérogènes dans leur nature et dans leur langage; ils réalisent des fonctions séparées; et chacun est placé dans les conditions les plus favorables à son utilisation, à savoir sur la bonne machine.

L'analogie proposée par Rea sur ce principe est très différente et moins générique.

### Universality/généricité

#### 6. Universality

- a. An object can perform several different functions ; therefore, other elements can be removed.

Pour moi, la notion d'universalité des objets techniques trouve son pendant dans la généricité des objets informatiques. L'industrie du logiciel toute entière cherche cette généricité, afin notamment de faciliter la réutilisation et d'économiser en temps et coût de développement.

Cette notion d'universalité/généricité se retrouve dans au moins quatre aspects du génie logiciel. Je souligne quels sont les objets techniques éliminés dans chaque cas:

les notions d'héritage qui permettent de constituer des objets génériques et de les spécialiser en fonction des problèmes à traiter; l'héritage élimine le code redondant pour des programmes qui partagent des fonctions communes;

les bibliothèques de composants génériques, qui permettent de bâtir rapidement des applications par assemblage; les bibliothèques évitent de programmer et de reprogrammer plusieurs fois les mêmes fonctions;

la modélisation et la méta-modélisation, qui sont à la base de la génération automatique de programmes; la génération automatique de code élimine le code "trivial" qui peut être engendré automatiquement d'après des spécifications formelles;

les environnements génériques/multifonctionnels, qui fournissent un ensemble de services aussi bien dans des domaines scientifiques et techniques - p.ex. Matlab, SAS, etc. que de gestion personnelle - p.ex. MS-Office et ses équivalents; ces environnements éliminent le besoin de développer le "tout-venant" et permettent de se concentrer sur les fonctions à valeur ajoutée<sup>22</sup>.

## Nesting/récurtivité, sous-programmes

### 7. Nesting (Matrioshka)

a. One object is placed inside another. That object is placed inside a third one. And so on...

b. An object passes through a cavity in another object.

Le principe des poupées russes est bien connu des développeurs de logiciels. Les sous programmes, la récursivité, les boucles imbriquées, sont l'équivalent logiciel de la *matrioshka*. De manière moins évidente, un effet similaire est recherché et obtenu dans le domaine des algorithmes de résolution numérique, j'en prends deux exemples:

- celui de l'imbrication des solveurs et optimiseurs, comme dans le cas de la modélisation de procédés: les algorithmes de résolution de systèmes aux dérivées partielles reposent sur des outils de résolution de systèmes algèbro-différentiels qui eux-mêmes utilisent en interne des solveurs algébriques non linéaires, qui linéarisent le problème pour faire appel à des solveurs algébriques linéaires, qui font donc la majeure partie du travail. Les outils d'optimisation MINLP (*mixed-integer nonlinear programming*) font appel à des solveurs MILP (*mixed integer linear programming*) etc. C'est aussi d'une certaine manière, ce que fait le fameux algorithme "inside-out" [Boston 74], qui imbrique des modèles pseudo-linéaires les uns dans les autres.

- les simulations basées sur des maillages de domaines (1D, 2D, 3D) peuvent utiliser des méthodes de raffinement local de maillage, en fonction de la précision demandée, qui peuvent pousser le raffinement de manière récursive pour parvenir à une bonne solution.

Ces exemples concernent la partie a) du principe. La partie b), un objet passe par le trou d'un autre objet, peut se transposer dans les systèmes de temps partagé, lorsqu'un programme s'exécute pendant le temps mort d'un autre programme.

---

<sup>22</sup> Je ne parle pas ici, bien entendu, de l'élimination de la concurrence, qui est une motivation pour les fournisseurs de ces environnements, pas pour leurs utilisateurs ☺

Les analogies proposées par Rea concernent l'imbrication de classes et d'objets les uns dans les autres.

## Feedback

### 23. Feedback

- a. Introduce feedback
- b. If feedback already exists, change it.

La notion de feedback est essentielle dans de nombreux cas. J'ai passé mes premières années de carrière professionnelle à étudier les rétroactions dans les systèmes dynamiques. La dynamique des systèmes, par exemple, s'est donnée l'analyse et la maîtrise des boucles de rétroaction comme objet principal de son investigation. Le feedback est également essentiel en automatique et théorie de la commande, il en est la base. Je reprends ici quelques extraits [Braunschweig 98t].

Si ces théories s'intéressent à l'étude des boucles de rétroaction dans les systèmes dynamiques, c'est parce que c'est la présence de ces boucles qui engendre la complexité. Les rétroactions positives amplifient les non-linéarités, introduisent des retards, des régulations qui, combinées, rendent les évolutions des systèmes difficiles à prévoir en dehors des cas élémentaires et donc donnent tout leur intérêt à la simulation. Il est bien connu, par exemple, que les bifurcations peuvent apparaître dans des systèmes avec une composante stochastique intervenant au sein de phénomènes auto-catalytiques. La théorie du chaos explore les cycles dans les systèmes continus et les itérations dans les systèmes discrets, comme moteurs des comportements chaotiques.

Autre exemple, les théories sur les capacités d'apprentissage des réseaux neuronaux, ont été établies le plus souvent dans le cadre de réseaux en couches, tels que le perceptron multicouche, ou le réseau à base radiale, dans lesquels l'information se propage dans un sens sans jamais retourner en arrière. Les réseaux bouclés tels que les réseaux de Hopfield, les réseaux récurrents, présentent des capacités d'apprentissages de phénomènes sans doute plus complexes, mais sont encore moins bien maîtrisés.

Enfin, c'est dans le domaine des algorithmes d'évolution artificielle que l'effet des boucles de rétroactions est le plus reconnu et utilisé. Les systèmes de vie artificielle font intervenir des relations complexes au sein d'une population d'individus, et dans ce cadre les rétroactions sont innombrables, agissent en permanence, et gouvernent les mécanismes d'évolution, faisant émerger des comportements nouveaux de par les dynamiques engendrées par les interactions en tous sens. Il est d'ailleurs significatif que les recherches sur la complexité se positionnent aujourd'hui dans le cadre de ces mécanismes d'évolution artificielle, qui produisent de nouveaux défis en matière de compréhension de phénomènes dynamiques non linéaires.

Lorsqu'il n'existe pas à l'état "naturel", le feedback est exploité pour améliorer le comportement d'un processus ou d'un système. En automatique, la conception du système de contrôle-commande d'un processus consiste principalement à en concevoir le mécanisme de rétroaction ou à modifier les rétroactions existantes.

Je ne voudrais pas terminer cette section sans mentionner le SISMONAUTE [Junker 95p], également décrit dans ma thèse [Braunschweig 98t], système produisant des historiques de

fronts d'ondes dans les simulations sismiques: le programme interprète des séries d'instantanés résultant de simulations, et reconstruit les relations de cause à effet entre les événements. Il fabrique une base d'objets graphiques reliant les fronts d'ondes, les obstacles, et leurs manifestations dans les instantanés. Cette base de données peut être qualifiée d'interprétation de la simulation.

Un aspect important du SISMONAUTE concerne le retour sur la prédiction opéré d'après les résultats de l'interprétation. A chaque étape d'interprétation qui marque un événement intéressant est associée une étape de rétroaction, afin de réajuster la prédiction qualitative compte tenu des informations numériques obtenues. Cette étape joue un rôle essentiel dans le processus global, puisqu'elle assure que le niveau symbolique est toujours proche du niveau numérique, facilitant ainsi l'interprétation.

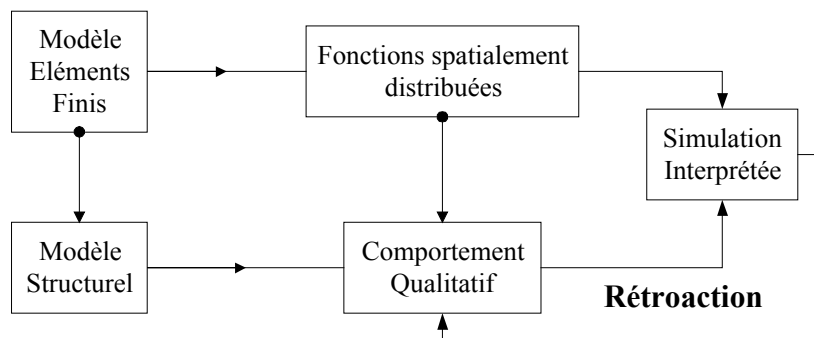


Figure 15: feedback dans le SISMONAUTE

Rea donne une déclinaison très technique du feedback, utilisé dans les réseaux de transmissions de données pour vérifier la qualité des informations. Cette déclinaison est compatible avec mon analyse, mais plus spécifique.

### Self-service: mises à jour à distance

#### 25. Self-service

- a. An object must service itself and carry-out supplementary and repair operations.
- b. Make use of waste material and energy.

Seule la partie a) du principe 25 m'intéresse. Avec le développement de l'internet et des intranets, les utilisateurs ont vu apparaître de plus en plus de logiciels capables de se mettre à jour ou de se configurer automatiquement, en allant chercher des composants sur le réseau. C'est le cas, par exemple, de produits courants comme *RealAudio* ou *Norton Utilities*. Les navigateurs web, tels que *Netscape*, *Explorer*, sont capables de diagnostiquer l'absence d'un composant nécessaire à l'exécution d'une application (un *plug-in*), et de le télécharger automatiquement - parfois moyennant l'envoi d'informations personnelles de l'utilisateur, exploitées à des fins de marketing!

Cette fonction est très générale, on la retrouve dans les *applets java*, également au cœur des navigateurs web.

## Mediator: intergiciels(*middleware*)

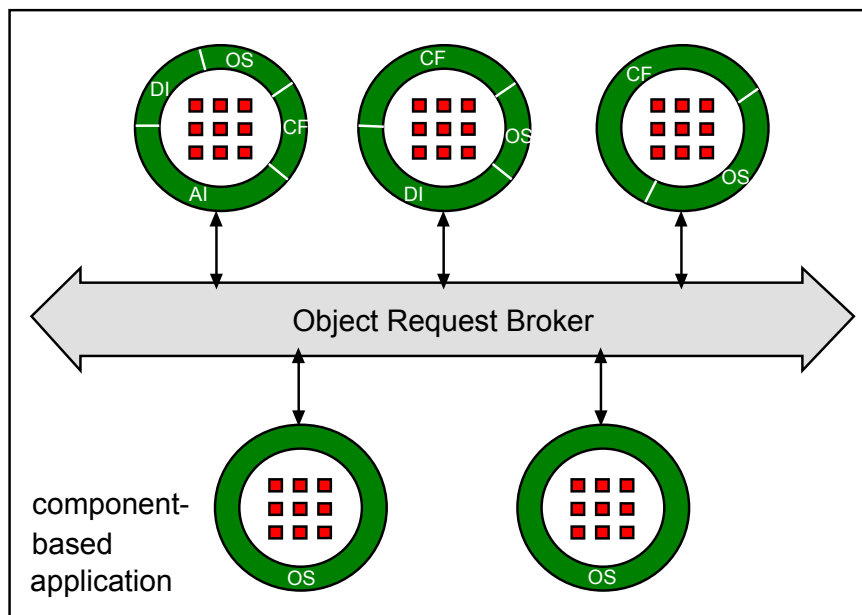
### 24. Mediator

- a. Use an intermediary object to transfer or carry out an action.
- b. Temporarily connect the original object to one that is easily removed.

C'est bien évidemment le très riche domaine du *middleware* ou intergiciel qui surgit lorsqu'on évoque le principe de médiation. Déjà présent dans le principe de qualité locale pour les architectures distribuées, l'intergiciel est la couche permettant de faire passer contrôle et information entre des systèmes.

On trouve cette notion dans toutes les architectures de *middleware* couramment utilisées. Dans les cas de COM, CORBA et SOAP, l'objet intermédiaire permet de transférer des actions entre des composants logiciels écrits dans des langages différents, fonctionnant sur des machines différentes, et sous des systèmes d'exploitation différents (p.ex. Visual Basic/PC/Windows NT et C++/Sun/Unix). Dans le cas des EJB (Enterprise Java Beans), l'objet intermédiaire permet à des composants écrits en Java de communiquer et d'échanger des fonctionnalités sans avoir besoin de connaître le détail de leur implémentation, et en bénéficiant de services additionnels.

Ainsi, le schéma classique de l'architecture CORBA donne à l'objet intermédiaire qu'est l'*Object Request Broker* (ORB) le rôle de médiateur entre composants, comme le montre la figure ci-dessous. Il est intéressant de noter que ce rôle médiateur est, en pratique, joué par plusieurs objets techniques qui composent l'architecture CORBA: *object adapter*, *server skeleton*, *client stub*. Je n'entrerai pas ici dans des considérations techniques détaillées hors de propos. L'important est de montrer le rôle de médiateur joué par l'ORB et par ses services.



AI ... Application Interfaces DI ... Domain Interfaces CF... Common Facilities OS ... Object Services

Figure 16: Architecture CORBA

Dans le système COM de Microsoft, l'objet intermédiaire est simplement le système d'exploitation lui-même: ce sont les services COM, présents au cœur de Windows, qui transmettent les fonctionnalités entre composants. Dans SOAP, initialement proposé par Microsoft puis adopté plus largement, ce sont des objets XML qui jouent ce rôle. Enfin, dans EJB, les "containers" - des objets d'un type particulier, développés en Java - non seulement font passer l'information entre composants Java, mais, au passage, fournissent des services tels que persistance, transactions, répartition de charge, etc.. comme le dit Jean-Marie Chauvet dans [Chauvet 00], "*Le conteneur EJB est en fait la pièce maîtresse du dispositif qui assure le contrat entre le composant EJB et le serveur d'applications*".

La partie b) du principe découle naturellement de la précédente: les services d'intergiciels permettent de connecter et de déconnecter "à chaud" des composants logiciels en fonction des besoins de l'application. On retrouve également ces notions, bien sûr, dans les architectures de Web Services, extension à l'internet des systèmes d'intergiciels<sup>23</sup>.

Kevin Rea donne l'exemple de XML comme médiateur, dans le cas particulier de la transmission de données suivant une DTD (*Document Type Definition*), le moyen de définir des structures de documents en XML.

## Copying

### 26. Copying

- a. A simplified and inexpensive copy should be used in place of a fragile original or an object that is inconvenient to operate.
- b. If a visible optical copy is used, replace it with an infrared or ultraviolet copies.
- c. Replace an object (or system of objects) with their optical image. The image can then be reduced or enlarged.

Seule la partie a) du principe m'intéresse, les autres étant trop liées à l'optique. Et je n'ai pas besoin de beaucoup d'explications. L'utilisation de copies à la place de l'objet original précieux est une pratique courant en informatique! Je ne pense pas seulement à la copie de fichiers, mais aussi à la copie interne de structures dans les programmes, ce qui permet au code de travailler librement sur la copie, la modifier, en pouvant toujours se ramener à la structure originale en cas de besoin: échec, demande d'annulation de transaction etc.

## Dispose: maquettes, prototypage

### 27. Dispose

- a. Replace an expensive object with a cheap one, compromising other properties (i-e., longevity).

---

<sup>23</sup> Ne considérez pas ceci comme une définition des web services, qui serait très réductrice, mais comme une illustration du propos.

Ce principe est bien connu dans l'industrie du logiciel, et dans la recherche. La plupart des projets de R&D comportent des phases de maquettage et de prototypage permettant de lever des incertitudes techniques et de recueillir rapidement des réactions d'utilisateurs futurs du système sur sa conception, son organisation, son ergonomie, ses performances. Ces logiciels "cheap" sont souvent abandonnés par la suite, une fois leur mission accomplie, pour laisser la place à des développements de plus longue haleine. Ceci constitue une parfaite illustration du principe 27, et c'est également celle que Rea met en exergue dans son analyse.

## Rejecting or regenerating parts: mémoire, versions

### 34. Rejecting and Regenerating Parts

- a. After completing its function, or becoming useless, an element of an object is rejected (discarded, dissolved, evaporated, etc.) or modified during its work process.
- b. Used-up parts of an object should be restored during its work.

Je fais deux transpositions pour ce principe. La première transposition, faite également par Rea, concerne la gestion de l'espace mémoire ou disque disponible. Les logiciels consacrent beaucoup d'effort à optimiser l'utilisation de cette précieuse ressource, qui a tendance à être systématiquement envahie par les programmes et les données, même encore aujourd'hui, alors que son coût unitaire a été réduit de plusieurs ordres de grandeur, suivant la fameuse loi de Moore. Qu'ils soient gérés par le système (libération de l'espace disque), par les applications, ou automatiques comme pour le *garbage collection* (algorithme de ramasse-miettes), les systèmes de gestion des ressources mémoire et disque sont apparus comme des composants essentiels des logiciels, suivant ainsi le principe 34 d'Altshuller.

La deuxième transposition que je fais concerne les systèmes de gestion de version, dont on peut dire qu'ils permettent, d'une certaine manière, de rejeter/regénérer des parties de programmes - seule la partie a) du principe est concernée ici.

## Résumé

J'ai montré que parmi les quarante principes de TRIZ, onze ont un intérêt direct pour le logiciel. Kevin Rea est allé plus loin en donnant des analogies pour trente-quatre d'entre eux, mais je ne suis pas particulièrement convaincu ni intéressé par ces analogies.



## Transposition des lois d'évolution

Je reproduis ci-après des sections de [Cavalucci 99] qui présente bien ces lois, et je complète par une réflexion sur leur transposition au domaine du logiciel. Les extraits de la thèse de Cavalucci sont encadrés comme ceci. Je ne cite que les passages essentiels à la compréhension du discours, utilisant ... pour indiquer les parties supprimées.

### Les lois statiques (1 à 3)

Tout Système Technique (ST) apparaît suite une synthèse de parties séparées en un tout. Mais une synthèse aléatoire de ces parties ne peut former un ST capable de fonctionner. Il existe trois lois dont les règles doivent être respectées pour que le ST soit opérationnel.

#### LOI 1 : Loi d'intégralité des parties du ST.

Une condition indispensable pour que le ST fonctionne réside en la capacité de travail des parties principales constituant ce ST. Chaque ST doit comprendre quatre parties principales à savoir un élément moteur, un organe de transmission, un organe de travail et un organe de contrôle. ... Pour synthétiser un ST, selon la loi 1, il est nécessaire :

que le ST soit composé des quatre parties principales ;

que ces parties aient respectivement une utilité minimale pour remplir les fonctions du ST car une partie qui fonctionne dans un ST peut s'avérer inutile dans un autre.

La loi 1 peut être définie ainsi : Un ST est capable de fonctionner uniquement si toutes ses parties ne sont pas jugées "médiocre" ... S'il y a au moins une partie jugée "médiocre", le ST ne fonctionnera pas même si les autres parties sont jugées "performantes". ...

Plusieurs analogies viennent à l'esprit. Une analogie simple, mais qui n'apporte pas grand-chose, est celle de l'implémentation d'un logiciel sur une machine:

moteur	processeur
transmission	système d'exploitation
organe de travail	logique du programme
organe de contrôle	interface

Il est clair que chacune de ces parties ne peut être médiocre, faute de quoi, le logiciel ne fonctionnera pas!

Une analogie plus intéressante est celle de l'architecture de systèmes à étages en "n-tiers", (voir section 2.1 de ce mémoire). La transposition de la loi 1 devient

moteur	base de données
transmission	middleware
organe de travail	logique du programme
organe de contrôle	interface

Les recherches en IA, notamment sur les systèmes à base de connaissances, ont fait admettre la séparation nécessaire entre moteur de raisonnement et représentation des connaissances. Dans cet esprit, l'analogie suivante est porteuse de sens:

moteur	moteur d'inférence
transmission	réseau RETE ou autre, représentation interne
organe de travail	connaissances/ faits
organe de contrôle	interface utilisateur ou programme

Enfin, je donne un exemple de transposition en simulation numérique, en prenant l'exemple de la simulation de procédés décrite au chapitre 4.

moteur	solveur
transmission	équations mathématiques
organe de travail	flowsheet
organe de contrôle	optimiseur

Dans tous ces exemples (et je laisse le lecteur en imaginer d'autres...) la loi 1 s'applique correctement, mais n'apporte comme seul enseignement que chaque partie doit fonctionner correctement si l'on veut que l'application soit de bonne qualité. Merci pour le conseil!

**LOI 2 : Loi de conductibilité énergétique du système.**

Une condition indispensable au fonctionnement d'un ST réside en un libre passage de l'énergie à travers toutes ses parties. De plus, tout ST peut être un convertisseur d'énergie. Par conséquent, il est nécessaire de transmettre l'énergie du moteur via la transmission, à l'organe de travail. La transmission de l'énergie d'une partie à une autre peut être matérielle (par exemple par un arbre, un pignon, un levier, etc.), magnétique (par exemple un champ magnétique) ou les deux en même temps (par exemple une transmission d'énergie par le flux des particules chargées). Beaucoup de problèmes d'innovation se ramènent à la sélection du type : transmission d'énergie le plus efficace dans des conditions données.

Cette propriété de la loi 2 est d'une grande importance : Pour qu'une partie d'un ST soit contrôlable, il est nécessaire d'assurer la conductibilité énergétique entre cette partie et les organes de contrôle.

La transposition, évidente, qui s'impose ici, est énergie  $\Leftrightarrow$  information. Je n'ai pas besoin d'expliquer plus avant que les logiciels traitent de l'information et que cette information doit bien circuler entre les composants: je préfère simplement réécrire la loi 2, en prenant la deuxième analogie précédente, celle du système n-tier:

**LOI 2 : Loi de conductibilité informationnelle du système.**

Une condition indispensable au fonctionnement d'un logiciel réside en un libre passage de l'information à travers toutes ses parties. De plus, tout logiciel peut être un convertisseur d'information. Par conséquent, il est nécessaire de transmettre l'information de la base de données via le middleware, à la logique de programme. La transmission de l'information d'une partie à une autre peut être directe (par exemple par une API), indirecte (par exemple par des

mécanismes de publish/subscribe) ou les deux en même temps (comme dans le *message-oriented middleware*). Beaucoup de problèmes d'innovation se ramènent à la sélection du type : transmission d'information la plus efficace dans des conditions données.

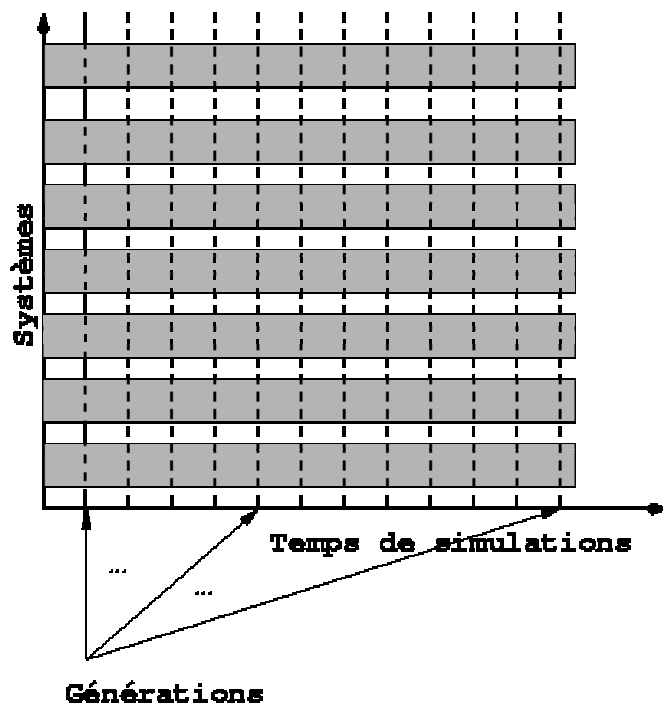
Cette propriété de la loi 2 est d'une grande importance : Pour qu'une partie d'un logiciel soit contrôlable, il est nécessaire d'assurer la conductibilité informationnelle entre cette partie et les organes de contrôle, c'est à dire souvent les utilisateurs.

**LOI 3 : Loi de coordination du rythme des parties.**

Une condition indispensable au fonctionnement d'un ST est la coordination du rythme (fréquence, vibrations, périodicité) de toutes ses parties. Dans l'analyse que les concepteurs font des systèmes techniques l'absence de coordination du rythme des parties limite la qualité du fonctionnement. L'évolution du système devient alors logique il s'agit d'accroître la coordination du rythme des parties qui ne sont pas en concordance.

Je transpose cette loi dans le domaine des logiciels fonctionnant en mode distribué, et de préférence asynchrones. Soit que le logiciel soit un assemblage de composants disposant chacun de son propre "rythme", soit qu'il soit distribué sur une grille de calcul ou une machine parallèle, une telle coordination est absolument nécessaire.

Ainsi, par exemple, dans la thèse de Benoît Leblanc [Leblanc 02+], un ensemble de simulations de Monte Carlo est conduit en parallèle sur une grappe de PCs. Cet ensemble de simulations sert à évaluer une population de distributions de fréquences de mouvements de Monte Carlo, population qui évolue de manière à maximiser un critère d'exploration d'un l'espace de configurations de molécules de polymères denses. La figure ci-contre, extraite de la thèse de B. Leblanc, illustre le



principe de ces simulations parallèles: la population est composée d'un ensemble de systèmes de Monte Carlo qui sont évalués en parallèle, et se retrouvent lors de rendez-vous appelés "Génération" pour échanger de l'information, en l'occurrence pour calculer une *fitness* et effectuer des opérations d'évolution artificielle: sélection, croisement, mutation. Dans cet exemple la coordination du rythme des parties est indispensable; elle est assurée, d'une part, par l'affectation d'un temps de simulation (temps de processus machine) égal quel que soit le nombre de mouvements effectués; d'autre part, par la

pondération des fréquences de mouvements autorisés en fonction du temps de calcul nécessaire à évaluer chaque mouvement (ces mouvements déplacent des monomères dans l'espace, et le calcul de la variation d'énergie du système est plus ou moins long selon le nombre de monomères déplacés et le type de déplacement).

Un deuxième exemple d'application de la loi de synchronisation des parties est celui de l'intégration d'une opération unitaire en mécanique des fluides au sein d'un simulateur de procédés "conventionnel" [Zitney 02]. Une telle intégration a été expérimentée dans le projet Global CAPE-OPEN, où un modèle de réacteur "Fluent" est simulé à l'intérieur d'un schéma de procédé évalué par un environnement conventionnel (Aspen Plus de Aspen Technology, ou Hysys de Hyprotech). Il faut comprendre que le temps de calcul de l'état d'un réacteur en mécanique des fluides est nettement supérieur à celui des autres opérations unitaires du procédé, puisque le réacteur en question est simulé en 3D, alors que les autres sont en général a-dimensionnels. Il est hors de question d'attendre que la simulation en CFD ait convergé pour chaque itération de calcul de l'ensemble. La coordination est, dans le cas présent, simplement assurée par un paramètre donnant le nombre maximal d'itérations CFD pour chaque itération de l'ensemble du procédé<sup>24</sup>. Il s'agit bien d'une coordination du rythme des deux parties que sont le réacteur et le reste du procédé. Cette tentative, à mon avis, est loin d'être satisfaisante, la troisième loi indique que le système évoluera vers une meilleure coordination (en accélérant le calcul CFD par des multiples processeurs, en ralentissant le système principal, ou autrement ...).

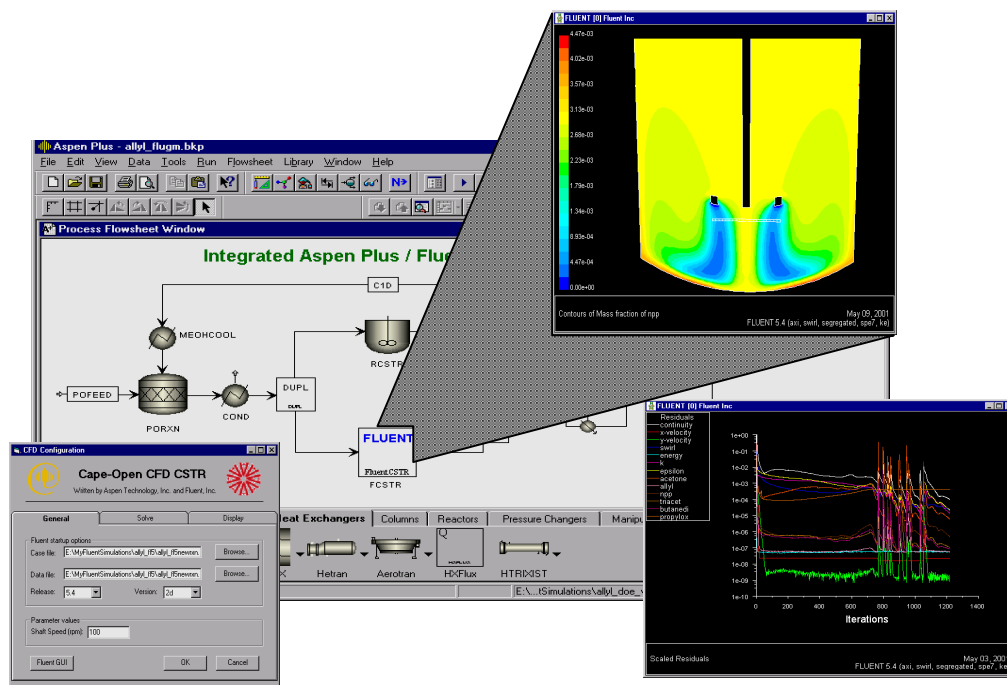


Figure 17: Intégration Fluent/Aspen Plus

Je ne voudrais pas terminer cette évocation de la troisième loi sans mentionner son application dans les systèmes d'information, et notamment dans tous les systèmes qui font appel à des transactions. Que ce soit dans les environnements traditionnels de l'informatique d'entreprise ou départementale, ou dans les environnement actuels de l'e-business, il est indiscutable que les systèmes transactionnels demandent une importante coordination rythmique, afin que les opérations puissent être faites et défaites simultanément (un débit sur un compte doit être coordonné avec le crédit sur un autre compte, avec une commande, une facturation, une opération de gestion de stock etc.). Cette coordination pose d'ailleurs de sérieux problèmes lorsque les

<sup>24</sup> Une autre manière de synchroniser serait de se baser sur une borne de d'erreur de résolution.

opérations sont effectuées par l'internet, d'où la mise au point de systèmes de communication transactionnels au-dessus d'XML pour ce seul besoin.

### Les lois Cinématiques (4 à 6)

La partie Cinématique comprend des lois qui définissent le développement d'un ST sans tenir compte des facteurs concrets techniques et physiques définissant ce développement.

#### LOI 4 : Loi d'augmentation du niveau de perfectionnement d'un ST.

Le développement de tout ST tend vers un niveau plus élevé de perfectionnement. Comme il est défini dans les notions fondamentales de TRIZ, un ST idéal est un système dont le poids, le volume, la surface, le coût, cherchent à atteindre zéro et dont la capacité de travail, les fonctionnalités, restent toujours identiques. Autrement dit, un ST idéal est un système qui n'existe pas mais qui conserve toutes ses fonctionnalités. Malgré l'aspect évident du concept de "ST idéal", il subsiste un paradoxe : les ST réels deviennent de plus en plus lourds et importants. Les dimensions et le poids des avions, des pétroliers, des automobiles, etc., augmentent. Ce paradoxe s'explique par le fait que les moyens dégagés pour le perfectionnement d'un ST servent principalement à augmenter ses dimensions et à élever, ce qui est le plus important, ses paramètres de travail. ...

La transposition au domaine des logiciels ne demande pas beaucoup d'imagination. Depuis plusieurs années, j'utilise l'illustration suivante dans mes présentations pour parler d'évolution des logiciels:

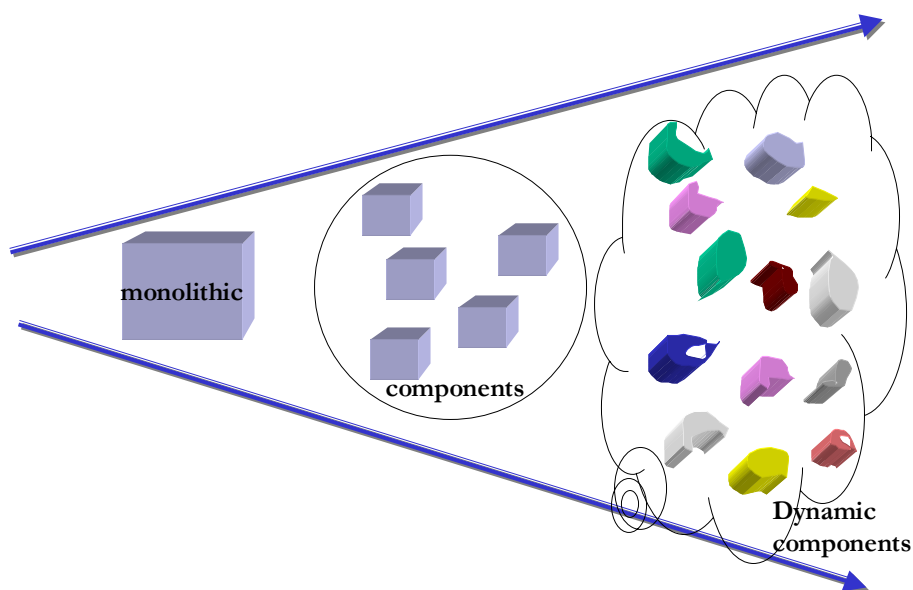


Figure 18: Evolution des logiciels

Si segmentation et dynamisation apparaissent évidemment dans cette figure, par la tendance Monolithic → Components → Dynamic Components, une autre dimension, celle de la complexité croissante, est indiquée par les deux axes temporels qui s'écartent. La simulation numérique continue à percer dans de nombreux domaines et à attaquer des problèmes "challenges" de plus en plus grands, à l'image du million de mailles en simulation de réservoir, ou des progrès de la modélisation moléculaire. Le volume de données en entrée et en sortie des modèles nécessite en accompagnement une gestion organisée des projets et des objets de simulation, comme le

démontre le grand projet américain "ASCI" sur la simulation nucléaire. En résumé, malgré l'augmentation constante des capacités de calcul et de stockage, résultant de la loi de Moore, programmes et données continuent à envahir tout l'espace mémoire et le temps de calcul disponibles, simplement parce que les problèmes traités sont de plus en plus grands, de plus en plus complexes, et demandent de plus en plus de précision.

Les logiciels se comportent, conformément à ce qu'indique Cavalucci, comme les avions ou les pétroliers: plus loin, plus vite, plus fort... les utilisateurs de MacOS et autres systèmes d'exploitation graphiques conviendront aisément de l'évolution du niveau de perfectionnement de leurs outils. Cette évolution ne se fait pas encore avec des programmes de coûts, le poids, et volumes nuls, mais une plus grande granularité dans la segmentation pourrait nous en rapprocher, tout en maintenant l'exigence de fonctionnalité et de perfectionnement.

#### **LOI 5 : Loi de développement inégal des parties d'un ST.**

Les parties d'un système se développent d'une manière inégale. Plus le système est complexe, plus le développement de ses parties est inégal. Le développement inégal des parties d'un système génère l'apparition de contradictions techniques et physiques et, par conséquent, suscite le besoin d'innover. Dans des conditions industrielles il arrive souvent que les parties d'un ST évoluent lentement. Les contradictions apparaissent alors graduellement, de façon parfois peu évidente. Tout se complique encore par le fait que certains ST réels contiennent beaucoup de sous-systèmes. Les contradictions s'accumulent alors dans ces sous systèmes pour après «éclater» et se répandent sur tout le système.

Tous ceux qui ont eu l'occasion de travailler sur des grands logiciels scientifiques ont rencontré ce problème - qui se rencontre également dans la gestion, mais c'est un domaine que je connais moins, l'ayant moins fréquenté.

Prenons comme exemples de système complexe un logiciel de simulation de réservoir pétrolier en ingénierie de gisement, un code d'interprétation géologique, un environnement de modélisation moléculaire, un logiciel de modélisation de procédés. Ces codes sont composés de centaines de milliers, voire de millions de lignes de code source. Ils sont (heureusement) modulaires, parfois sous forme d'assemblages de composants, plus souvent sous forme de modules communiquant par échange de fichiers, ou au moyen d'une base de données partagée. Ils contiennent fréquemment un préprocesseur, préparant les données pour le calcul; un processeur effectuant le calcul; un postprocesseur permettant d'accéder aux résultats.

Il est rare que tous les composants de tels systèmes soient au même niveau de développement. Bien au contraire, il est fréquent que l'un des composants soit un goulot d'étranglement, qui pénalise les autres, limitant ainsi la fonctionnalité globale du système. Dans certains cas la base de données n'est pas assez rapide; dans d'autres cas le calcul est trop long par rapport aux capacités d'affichage des résultats en 3D; dans d'autres cas encore, ce sont les outils de visualisation qui ne permettent pas de rendre compte de la richesse des informations contenues dans les données d'entrées et dans les résultats. On a déjà vu des grands codes scientifiques "s'écrouler" en raison de la défection d'un composant n'ayant pas subi d'évolution depuis trop longtemps.

Mais, comme le dit Cavalucci, ces inégalités donnent lieu à des innovations, que ce soit en matière d'architectures de calcul (calcul parallèle, *grid computing*), de bases de données réparties, de visualisations avancées (3D, 4D, environnements de fouille de données visuels).

#### **LOI 6 : Loi de la transition vers le supersystème.**

Après avoir épuisé ses possibilités de développement, un système se rattache à un supersystème en tant qu'une de ses parties. Alors, son développement ultérieur se poursuit via le supersystème. Nous retrouvons ici, une notion similaire au cycle de vie, où un système, à la fin de l'exploitation des ressources de son développement, laisse la place à un autre système pour repartir sur une nouvelle courbe de croissance.

Cette loi décrit très bien les processus de développements de logiciels à base de composants, ou de bibliothèques de modules réutilisables. Les modules de base, développés depuis longtemps, fournissent les accès au système, au réseau, aux données; ils n'évoluent plus une fois le système mis au point. Au-dessus viennent se greffer des modules utilitaires: tris, gestion de structures, de graphes, de listes, etc.; encore au-dessus, on peut trouver des bibliothèques de services génériques tels que des algorithmes de résolution numérique, ou des bibliothèques graphiques; le niveau suivant sera celui des services applicatifs, des composants métier; enfin, l'application elle-même mettra en œuvre ces ensembles de composants métier.

Chacun de ces niveaux peut être considéré comme un système qui est sujet à un développement initial jusqu'à sa finalisation, et qui se rattache ensuite au supersystème qui l'exploite. Les routines de manipulation de structures de graphes sont des sous-systèmes des bibliothèques de visualisation qui gèrent des graphes dont les nœuds sont des menus, des fenêtres, des objets graphiques; les routines numériques de base (résolution d'une équation linéaire par exemple), une fois développées et finalisées, sont exploitées par des routines numériques plus complexes (résolution d'un système d'équations, ou de systèmes non linéaires), qui elles-mêmes, une fois établies, sont mises en œuvre au sein d'applications.

Le même processus se déroule pour les environnements de logiciel libre, dont Linux est l'exemple le plus connu. Le noyau de Linux n'évolue plus, sauf en cas de révision majeure. Ce sont les compléments de Linux, et les compléments à ces compléments, qui font l'objet de nouveaux développements, et ainsi de suite.

#### **Les lois Dynamiques (7 et 8)**

La partie dynamique comprend des lois qui expriment le développement des ST modernes sous l'effet de facteurs techniques et physiques concrets. Les lois statiques" et "cinématiques" sont universelles, c'est à dire qu'elles sont toujours vraies, non seulement pour les ST mais aussi pour les systèmes en général (biologiques, sociaux, etc.). Le terme "Dynamique" exprime les tendances principales de développement des ST actuels.

Ces deux lois sont essentiellement une réécriture des principes 1, segmentation, et 15, dynamisation (incluant aussi le principe de feedback). Le fait de les positionner au niveau des lois d'évolution leur donne un caractère inéluctable, alors que les principes constituent essentiellement des outils pour résoudre des contradictions sur des systèmes techniques, c'est à dire des outils de résolution de problèmes. Ayant largement présenté des idées sur segmentation et dynamisation des logiciels dans le chapitre 2 de ce mémoire, je me contente de citer ces lois dynamiques sans commentaires.

**LOI 7 : Loi de transition du macroniveau vers le microniveau.**

Le développement des organes de travail du système passe d'abord par le macroniveau puis ensuite évolue vers le microniveau. Dans la plupart des ST modernes, les organes de travail sont des objets physiques solides. Leur développement n'est possible que dans les limites du macroniveau. Mais tout objet voit ses performances évoluer et le moment de voir un développement ultérieur saturé au macroniveau, arrive inévitablement. Tout en gardant ses fonctions, le système doit alors changer radicalement. Son organe de travail commence à fonctionner au microniveau. A la place des "objets" on trouve des molécules, des atomes, des ions, des électrons, etc. qui effectuent le travail.

**LOI 8 : Loi d'augmentation du dynamisme et de contrôlabilité**

Le développement des ST contrôlables tend toujours vers un niveau plus élevé de contrôlabilité, avec notamment :

des systèmes non contrôlables qui cherchent à devenir contrôlables  
dans des systèmes contrôlables, un développement qui suit une transition de champs mécaniques vers des champs électromagnétiques;  
dans des systèmes contrôlables, des développements qui cherchent à établir des liens entre les éléments;  
dans des systèmes contrôlables, un développement qui tend vers la compatibilité des éléments.



## Annexe 2: le Standard CAPE-OPEN<sup>25</sup>

The CAPE-OPEN 1.0 standard, a product of several years of collaborative work between partners and contractors of the CAPE-OPEN and Global CAPE-OPEN projects has been recently released and is available for download from the [www.colan.org](http://www.colan.org) CO-LaN website.

This article aims at giving an overview of the technical elements of the standard - the set of open standard interface specification documents which describe the main software interfaces that process modelling components and environments must provide in order to be CAPE-OPEN compliant. It must be understood that there is no requirement for a piece of software to implement all interfaces specified by CAPE-OPEN. On the contrary, the interface specifications are organised in such a way that each process modelling component (PMC) or process modelling environment (PME) only needs to implement a limited and focused set of interfaces to allow it to interoperate with other software. As an example, a Unit Operation software component (e.g. a model of a distillation column, of a reactor, of a heat exchanger, or a pump etc.) is only required to implement Unit Operation "plug" interfaces, plus a small number of Common Services interfaces, in order to be used with a CAPE-OPEN compliant PME. Should the Unit Operation require direct use of a Physical Properties or of a Chemical Reactions component for internal calculation, then it would have to implement physical properties or chemical reactions "sockets" interfaces in order to gain access to these services.

Note that we have been using the terms *socket* and *plug* to refer to the calling and called components respectively: where a software component **A** requires service **S** from component **B** through a published CAPE-OPEN interface specification of service **S**, **A** is said to implement an **S** *socket*, and **B** is said to implement an **S** *plug*.

In the following, we will look at the architecture of CAPE-OPEN 1.0 interfaces, we will briefly address performance issues, and we will give a list of currently available implementations of CAPE-OPEN interfaces in both commercial and research PMCs and PMEs. Some of the text is borrowed from J.-P. Belaud and M. Pons, " *Open Software Architecture For Process Simulation: The Current Status of CAPE-OPEN Standard*", ESCAPE-12 conference, The Hague, May 2002, with permission from the authors.

### Technical elements of CO 1.0

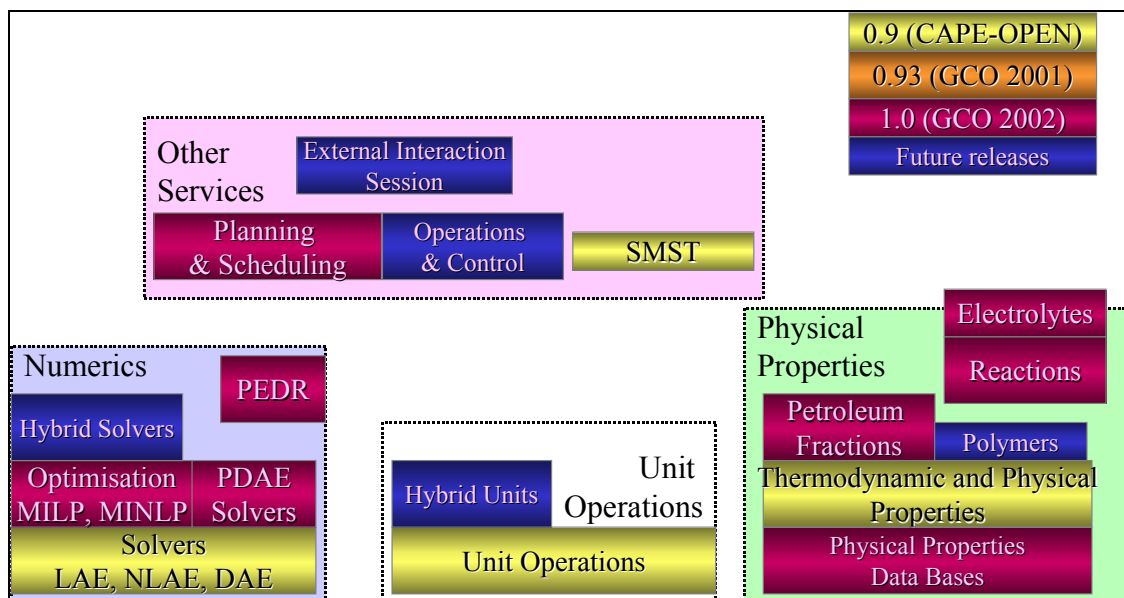
The CAPE-OPEN interfaces are split in three groups corresponding to three blocks of functionality: business interfaces (also called PMC interfaces), COSE interfaces (also called PME interfaces), and common services. They were introduced by J.-P. Belaud in Vol.1 of this newsletter:

---

<sup>25</sup> Article paru dans "CAPE-OPEN Update" Vol. 2, Avril 2002, "**The CAPE-OPEN 1.0 Standard**", by Bertrand Braunschweig

- ◆ *Business interfaces* or *PMC Interfaces* are domain-specific interfaces for the CAPE application domain. They define services provided by CAPE-OPEN compliant process modelling components involved in a CO modelling application. Examples are Unit Operations, Thermodynamic and Physical Properties, Solvers, Physical Properties Data Bases, etc.
- ◆ *COSE (CAPE-OPEN Simulator Executive) Interfaces* or *PME Interfaces* are horizontal interface specifications. They define services provided by CAPE-OPEN compliant PME. Services of general use are defined, such as *diagnostics* and *material template system* in order to be called by any CO PMC using a callback pattern.
- ◆ *Common Services Interfaces* define services that may be required by any *Business* and *COSE interfaces*. They support basic functions, such as identification, error handling, collections of objects, parameters, etc.

The first set, *business/PMC interfaces*, is a comprehensive set of functionalities of process modelling application elements. Let's look at it in detail.



**Figure 19: CAPE-OPEN Interfaces for Process Modelling Components**

Business interfaces can be roughly grouped in four categories, as shown above: numerics, unit operations, thermodynamic and physical properties, and others. Although it can be debated if chemical reactions/electrolytes interfaces should be kept together with physical properties, we will keep them as such, since it is merely a question of presentation - there are no technical implications.

Version 1.0 updates previously published interfaces (known as "CAPE-OPEN 0.93") and delivers brand new interfaces. These new interfaces have not been as extensively tested as the ones previously published, therefore will be subject to updates after subsequent implementation in operational software. They are identified by the **New!** sign.

Several interfaces have been extensively tested and debugged by the Interoperability Task Force of Global CAPE-OPEN, a group of specialists led by Peter Banks and Malcolm Woodman of BP. They are identified by the "ITF-ed!" sign.

The *Numerics* class groups services related to numeric processing of process models.

- ◆ The base interface in this class is *Solvers*: it focuses on the solution algorithms that are necessary for carrying out steady state and dynamic simulation of lumped systems. In particular, this includes algorithms for the solution of large, sparse systems of non-linear algebraic equations (NLAEs) and mixed (ordinary) differential and algebraic equations (DAEs). Algorithms for the solution of the large sparse systems of linear algebraic equations (LAEs) that often arise as sub-problems in the solution of NLAEs and DAEs are also considered. The CO standard introduces new concepts, such as models and the equation set object (ESO), which is a software abstraction of a set of non-linear algebraic or mixed (ordinary) differential and algebraic equations.
- ◆ **New!** The *PDAE (Partial Differential Algebraic Equations)* interface defines, on top of the *Solvers* specification, numerical services for systems with some variables distributed along one or several dimensions. In PDAEs the dependent model variables depend on one or more independent variables. Independent variables are for instance spatial co-ordinates, particulate co-ordinates (in case of population balance models) or time (in case of dynamic models). Thus, models of computational fluid dynamics are also included in this class of problems.
- ◆ **New!** The *Optimisation* interfaces define access to Mathematical Programming optimisation services. They are also based on the *Solvers* architecture. Mathematical programming (IP / LP / NLP / MILP / MINLP) problems involve the minimisation or maximisation a linear / nonlinear objective function subject to linear / nonlinear constraints. The optimisation may involve both continuous and discrete (integer-valued) decision variables. Mathematical programming optimisation problems arise in many process engineering applications, including process synthesis, process design, product design and others.
- ◆ The *Hybrid Solvers* interface is not part of CAPE-OPEN 1.0 release and is not addressed here.
- ◆ **New!** The *Parameter Estimation and Data Reconciliation* interface (PEDR), at its name clearly states, defines interface to (i) parameter estimation algorithms where the value of a model parameter must be adjusted in order to meet constraints such as experimental data; and (ii) data reconciliation packages which eliminate noisy factors from raw measurements of process variables; The DR and PE are very similar problems in the sense that both are constrained optimization problems. Since a PEDR module may require using external optimisation services, the PEDR module may call an optimisation solver module through a CO-compliant interface.

The *Unit Operation* class groups two services related to unit operations.

- ◆ **ITF-ed!** Base *Unit Operation* component: CAPE-OPEN defines a comprehensive set of standard interfaces for unit operation modules being used within modular and steady-state PMEs. A unit operation module may have several ports that allow it to

be connected to other modules and to exchange material, energy or information with them. In the material case (which is also the most common), the port is associated with a Material Object. Ports are given directions (input, output, or input-output). Unit operation modules also have sets of parameters. These represent information that is not associated with the ports, but that the modules wish to expose to their clients. Typical examples include equipment design parameters (e.g. the geometry of a reactor) and important quantities computed by the module (e.g. the capital and operating cost of a reactor).

- ◆ *Hybrid Unit Operations* extend the base UO interface towards the processing of batch and hybrid UO's. This interface is not part of CAPE-OPEN 1.0 release and is not addressed here.

The *Physical Properties* class groups services related to obtaining or calculating thermodynamic and physical properties of matter, and to handling chemical reactions.

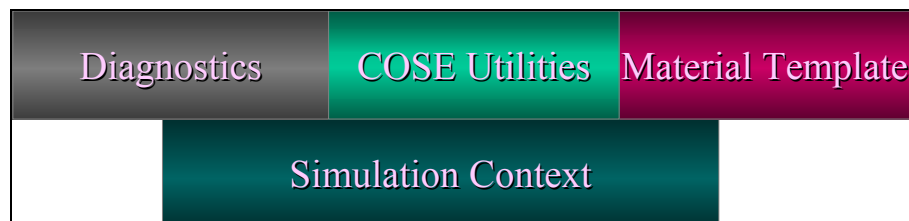
- ◆ **ITF-ed!** *Thermodynamic and Physical Properties* base interface: CAPE-OPEN focuses on uniform fluids that are mixtures of pure components or pseudo-components, and whose quality can be described in terms of molar composition. The physical properties operations that have been provided with standardised interfaces are those required for the calculation of vapour-liquid or liquid-solid equilibria or subsets thereof, as well as other commonly used thermodynamic and transport properties. A key concept is that of a Material Object. Typically, each distinct material appearing in a process (in streams flowing between unit operations, as well as within individual unit operations) is characterised by one such object. To support the implementation of the above framework, the CO standard defines interfaces for Material Objects, as well as for thermodynamic property packages, calculation routines and equilibrium servers.
- ◆ **New!** *Physical Properties Data Bases* (PPDB) interfaces define a CAPE-OPEN compliant standard interface for connecting a data base with recorded physical property values and model parameters to flowsheeting and other engineering programs. This interface deals with measured, correlated or estimated values of physical property data at discrete values of the variables of state (temperature, pressure, composition).
- ◆ **New!** *Petroleum Fractions* (PetroFrac) interfaces extend the standard Material Object for use in the modelling of hydrocarbon fluids processed in refining, petrochemicals and offshore production facilities. They supply additional access to petroleum-specific properties (e.g. RON, MON, cetane index, TBP curves, etc.), and allow characterising parameters of the mixtures. They also introduce a small change in the Unit Operation interfaces in order to distinguish Unit Operations handling petroleum fractions from others.
- ◆ *Polymers* interfaces extend the base Thermo interface towards the processing of polymers. This interface is not part of CAPE-OPEN 1.0 release and is not addressed here.
- ◆ **New!** *Chemical Reactions/Electrolytes* interfaces support the management and processing of kinetic, equilibrium and electrolytes reaction systems in process models. These interfaces support any reaction model, they are clients to formulate

reaction equations, and they support reaction model parameter estimation. Initially planned as two different specifications, Chemical Reactions and Electrolytes were finally merged into one consistent set<sup>26</sup>.

The *Other Services* class groups various services which do not belong to the three main classes. Only two interfaces are part of CAPE-OPEN 1.0, the others (*Operation and Control* and *External Interaction Session*) are published as drafts, but not officially released. They are not presented here

- ◆ *Sequential Modular Specific Tools* (SMST) interface. A key part of the operation of sequential modular simulation systems is the analysis of the process flowsheet in order to determine a suitable sequence of calculation of the unit operation modules. This task is typically carried out using a set of tools that operate on the directed graph representation of the flowsheet. The SMST specification defines standard interfaces for the construction of these directed graphs, and for carrying out partitioning, ordering, tearing and sequencing operations on them.
- ◆ **New!** *Planning and Scheduling* interface defines interfaces to components delivering procedures and processes for allocating equipment over time to execute the chemical and physical-processing tasks required for manufacturing chemical products, generally in batches. These interfaces deal with managing requirements, production resources, recipes, planning and scheduling problems and their solutions.

*COSE/PME* interfaces define general services that can be requested from CAPE-OPEN compliant process modelling environments such as Hyprotech's HYSYS, AspenTech's AspenPlus, PSE's gPROMS, or RSI's INDISS. All these specifications are **New!** in CO 1.0. The *COSE/PME* interface is a small and simple interface with only half a dozen methods. It might be extended in the future.



**Figure 20: CAPE-OPEN Interfaces for Simulation Executives (or Process Modelling Environments)**

- ◆ *Simulation Context* interface specification is the base specification for COSE services. It gathers three functionalities, Diagnostic, Material Template System and Utilities. That results in three interfaces, one for conveying verbose information to the PME, one for allowing the unit to choose between all the Thermo

---

<sup>26</sup> At the same time, we publish future draft releases (1.1) of Thermodynamic and Physical Properties, and of Reactions; anyone intending to implement these versions should contact CO-LaN.

Material factories supported by the COSE, and one for requesting diverse values from the PME.

- ◆ *Diagnostic* interface allows communication of verbose information from the PMC to the PME (and hence to the user). PMCs should be able to log or display information to the user while executing a flowsheet.
- ◆ *Material Template* interface provides the mechanisms for accessing CAPE-OPEN Property Packages managed by the COSE, in order to allow PMCs to directly choose and configure material objects as needed.
- ◆ *COSE Utilities* provide a small list of other useful functions, in particular the FORTRAN Channel selection which prevents FORTRAN-based PMCs from sending output to channels already used by the COSE.

*Common Services* interfaces define base services that CAPE-OPEN components can either use or provide, independently of the nature of the component. Some of these specifications were published in release 0.93 last year, some are new. They are gathered in one of the two "Methods and Tools" documents which were developed by the Methods and Tools group of Global CAPE-OPEN: the Common Interfaces Reference Manual.

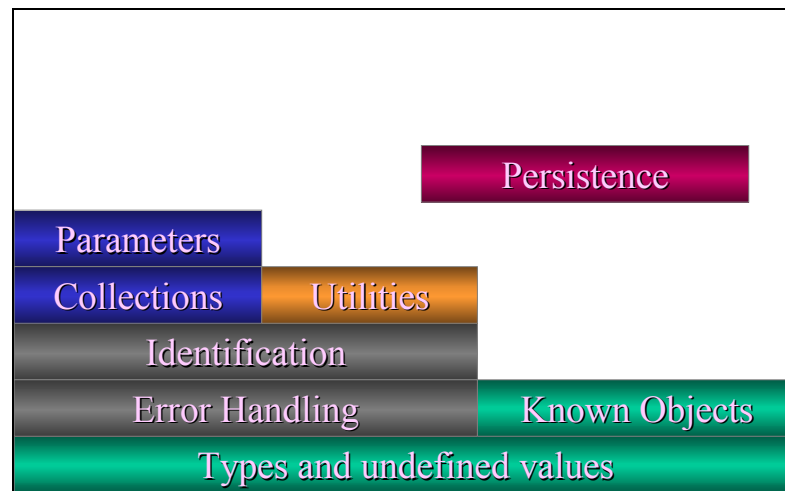


Figure 21: CAPE-OPEN Common Services Interfaces

- ◆ **ITF-ed!** *Types and undefined values* are the most basic elements of the CAPE-OPEN 1.0 specification. They define the standard types of all elements passed through CO interfaces: integer values, real values, arrays, character strings, etc. as well as "undefined" values. All CO interfaces rely on the standard types.
- ◆ **New!** *Known objects* gather a large number of standard identifiers used in interfaces, such as Thermo, PPDB, and petroleum fractions. These are lists of properties names, names of methods; identification of phases, etc., which form a consistent set of names used throughout these interfaces.

- ◆ **ITF-ed!** Error Handling defines how to manage execution errors (abnormal terminations). When a request is made, if this request is successful it raises no error, otherwise it raises an error. When an error occurs, the execution is immediately aborted. This Error Handling interface gives a classification and a hierarchy of potential errors occurring in CO compliant components. All CO components must implement it.
- ◆ **ITF-ed!** Identification interface provides the means for all CO components to be identified by name and textual description. All CO components must implement it.
- ◆ **New!** Collections interface defines a standard way of managing collections of things. The aim of the Collection interface is to give a CO component the possibility to expose a list of objects to any client of the component. The client will not be able to modify the collection, i.e. removing, replacing or adding elements. However, since the client will have access to any CO interface exposed by the items of the collection, it will be able to modify the state of any element.
- ◆ **ITF-ed!** Parameters interface defines a standard access to component parameters. This specification will be used by CO components wishing to expose some of their internal data to their clients. The interface is made up of two different parts, each corresponding to a different client need: the first part is a fixed, static aspect that describes the Parameter, such as a type, name, description, dimensionality etc. The second part deals with value of the parameter itself. It is expected that the parameter values will change quite frequently both within and outside of the component that needs it.
- ◆ **ITF-ed! New!** Utilities interface gathers a number of useful functionalities that can be requested from process modelling components. In version 1.0, this interface provides the means to set the simulation context, to collect component parameters, to manage lifecycle of components (creation and termination) and to edit, that is, to open an edit GUI for the component.
- ◆ **ITF-ed! New!** Persistence interface defines how models and model elements are stored and retrieved. Most simulation environments allow the possibility to store at any moment the state of a simulation case, in order to be able to restore it at any time in the future. In the CAPE-OPEN distributed environment, where different pieces of the simulation may be implemented by different vendors, the Persistence interface proposes a standard mechanism to provide this feature. This interface is different from all others, as it does not define any new method. Instead, it explains how to use standard persistence mechanisms provided by middleware (COM and CORBA) for this purpose.

### **Performance Issues**

The Interoperability Task Force of Global CAPE-OPEN spent considerable effort on performance issues in the last months of 2001. The ITF was charged with facilitating the delivery of practical, plug and play interoperability within commercial flowsheet simulators, via industrially-relevant test scenarios and the critical review of the CAPE-OPEN interface specifications for ambiguities and omissions. This capability within mainstream commercial flowsheet simulators is a prerequisite for acceptance of open simulation in the work place.



Good performance is critical especially for thermodynamic and physical properties calculations which are called thousands of times during a process simulation, so that the overhead due to standard interfaces should be minimal. This issue is also of importance for other components but the focus has been on physical properties and thermodynamics since the initial performance degradation was huge in some implementations.

The result of the ITF work is that there is almost no degradation of performance if the interfaces are correctly and efficiently implemented. Revisions of internal implementation of physical properties components by vendors and others (without any revision of the interface specifications) led to orders of magnitudes of improvement, with final values of around 15-20% overhead in calculation time due to CAPE-OPEN interfaces being possible. Such an overhead is not significant and is very quickly compensated by the consequences of Moore's Law, which makes computing speed higher and higher and computing power cheaper and cheaper.

In conclusion, you should not worry about performance of CAPE-OPEN interfaces in general. You should only worry about efficient implementation of CAPE-OPEN compliant physical properties and thermodynamic services: the development of physical and thermodynamic properties plugs or sockets must take serious care of performance, whether the CAPE-OPEN or any other standard is used.

#### **CO-compliant Process Modelling Components and Process Modelling Environments**

The following table lists a number of commercial or research PMCs and PMEs providing CAPE-OPEN interfaces. This table cannot be exhaustive since we cannot be aware of all the current developments and migration of existing software packages to CO compliance. If you are interested in specific software package shown in this list, please contact its supplier. If you are interested in any software package **not shown** in the list, please contact its supplier and ask for CO compliance!

A catalogue of CO-compliant PMCs and PMEs will be published and regularly updated on [www.colan.org](http://www.colan.org).

Supplier	Software	Interfaces	Technology
AspenTech <a href="http://www.aspentech.com">www.aspentech.com</a>	Aspen Plus 11.1	Thermodynamic and physical properties socket Unit operations socket	COM
AspenTech	Aspen Properties 11.1	Thermodynamic and physical properties plug	COM
Hyprotech <a href="http://www.hyprotech.com">www.hyprotech.com</a>	HYSYS.Plant 2.4	Thermodynamic and physical properties socket Unit operations socket	COM
Hyprotech	Distil	Thermodynamic and physical properties socket	COM
Hyprotech	COMThermo 1.1	Thermodynamic and physical properties plug	COM
Process Systems Enterprise (PSE) <a href="http://www.psenterprise.com">www.psenterprise.com</a>	gPROMS	Thermodynamic and physical properties socket Unit plug	COM CORBA
Process Systems Enterprise (PSE)		Numerical solvers sockets (linear algebraic, nonlinear algebraic, differential-algebraic)	CORBA
Process Systems Enterprise (PSE)	gO:CAPE-OPEN	Equation Set Object (unit) plug	COM
Belsim <a href="http://www.belsim.com">www.belsim.com</a>	VALI III	Thermodynamic and physical properties socket	COM
Prosim S.A. <a href="http://www.prosim.net">www.prosim.net</a>	ATOM	Thermodynamic and physical properties plug	COM
Prosim S.A.	Odysseo	Dynamic flash unit plug	COM
Infochem <a href="http://www.infochemuk.com">www.infochemuk.com</a>	Multiflash 3.1	Thermodynamic and physical properties plug	COM
RSI <a href="http://www.rsi-France.com">www.rsi-France.com</a>	INDISS	Thermodynamic and physical properties plug and socket Unit Operation plug and socket	COM COM
IFP <a href="http://www.ifp.fr">www.ifp.fr</a>	SPIP	Thermodynamic and physical properties plug	
IFP	FIBER	Unit Operation plug	
INP Toulouse-LGC-CNRS <a href="http://www.inp-toulouse.fr/lgc">www.inp-toulouse.fr/lgc</a>	Numerical Services Provider and Continuous Model Builder M&S	Numerical Solvers plug and socket  Unit socket	CORBA COM
INP Toulouse-LGC-CNRS	Flowsheet Server	Sequential Modular Specific Tools plug	CORBA
DECHEMA <a href="http://www.dechema.de">www.dechema.de</a>	DETERM	Physical Properties Data Bank Plug	COM
RWTH.LPT <a href="http://www.lfpt.rwth-aachen.de">www.lfpt.rwth-aachen.de</a>		Numerical solvers plug	CORBA
RWTH.I5 <a href="http://www-i5.informatik.rwth-aachen.de">www-i5.informatik.rwth-aachen.de</a>	COM-CORBA Bridge Java Unit Skeleton Java Material Object Skeleton	Bridge Unit Operation plug Material Object and Material Template	COM, CORBA CORBA CORBA
CO-LaN <a href="http://www.colan.org">www.colan.org</a>	Tester Suite (1)	Thermodynamic and physical properties plug and socket Unit Operation plug and socket	COM, CORBA through bridging
CO-LaN	Tester Suite (1)	MINLP socket PPDB socket SMST socket & plug	COM
NORSK HYDRO <a href="http://www.hydro.com">www.hydro.com</a>		Heating Tank Unit Operation Fluent Wrap Unit Operation CASE test socket	CORBA
UPC <a href="http://www.upc.es/eq/">www.upc.es/eq/</a>	MOPEDR MOPP	PEDR Prototype Planning and Scheduling Package	CORBA CORBA

(1) The CO-LaN tester suite is a comprehensive set of public-domain tools aiming at helping organisations develop CAPE-OPEN components through detailed compliance checking mechanisms. This is one of the methods that CO-LaN uses to facilitate the migration to CAPE-OPEN, together with additional how-to documents (the *migration cookbook*) and software *wizards*. The Unit Operation wizard is presented in this volume. The tester suite will be presented in Vol. 3. of this newsletter.

### **Conclusion**

CAPE-OPEN interfaces match the interoperability needs of the process modelling community; they are debugged; they are efficient; they are publicly available on the web site for anyone to use; they have been implemented in many software packages; in short, they work!

Open process simulation is not a futuristic thing. It is here and now. If you are a supplier, migrate your software to CO compliance; if you are a user, ask for CO compliance from your suppliers.

### **Acknowledgements**

This text summarises many other documents authored by contributors of Global CAPE-OPEN from several organisations. It is better not to give any names, since the list would be too long. Many thanks to all.

In addition, this article was reviewed and commented by Peter Banks, Jean-Pierre Belaud, Kerry Irons, Knut Mathisen, Costas Pantelides, Michel Pons, Luis Puigjaner, and Pascal Roux.

### **References**

The CAPE-OPEN 1.0 standard is downloadable from [www.colan.org](http://www.colan.org).

## Annexe 3: Curriculum vitae

### ◆ **Etat civil**

Bertrand Lionel Braunschweig

Né le 20 Avril 1954 à Bagneux (92), nationalité française

Domicile : 14 rue Constant Pape 92140 Clamart

Voix : +33 1 46 42 70 41 , Mobile +33 6 74 29 28 61

Marié, un enfant

### ◆ **Fonction**

Directeur Expert, Institut Français du Pétrole, Division Informatique Scientifique  
Mathématiques Appliquées

### ◆ **Formation**

- Ingénieur Institut d'Informatique d'Entreprise, Paris, 1977
- Formation longue, chef de projet en Intelligence Artificielle, Grenoble, 1987
- Docteur en Intelligence Artificielle, Université Paris-Dauphine, avril 1998

### ◆ **Carrière professionnelle à l'IFP**

#### Titres

- De 1989 à 1991 : Ingénieur de recherche détaché de Elf Aquitaine à l'IFP, responsable de l'activité Intelligence Artificielle
- De 1991 à 1998 : Ingénieur de recherche principal, responsable du groupe Intelligence Artificielle.
- De 1999 à 2000 : Ingénieur de recherche en chef, responsable du groupe IA et Statistique.
- Depuis 2000 : Directeur Expert rattaché au Directeur Informatique Scientifique Mathématiques Appliquées. Co-directeur par intérim de la Division Informatique Scientifique Mathématiques Appliquées (avril 2002 -).

#### Projets

- De 1990 à 2002: chef du projet exploratoire "Intelligence Artificielle";
- Depuis 1995 : coordonnateur des projets européen CAPE-OPEN (1996-99) et européen/international GLOBAL CAPE-OPEN (1999-2001) et de la mesure d'accompagnement GCO-Support. Président du « CAPE-OPEN Laboratories Network». Coordonnateur du projet IST "COGents".
- Depuis 2000: responsable du projet européen "BF-CHARISMA" (2000-2003) . Coordonnateur de la proposition de projet intégré FP6 "e-WOK"

### ◆ **Carrière professionnelle en dehors de l'IFP**

- 1977-1989, groupe Elf : ingénieur pour la direction centrale de recherche du groupe Elf ; conception et réalisation de modèles de simulation; puis chef de projet pour la Direction des Systèmes d'Information ; puis chef de projet dans le Département Recherche Informatique Avancée, Groupe d'Application de l'Intelligence Artificielle.

### ◆ **Activités d'enseignement, de représentation, électives, honorifiques, etc. en dehors de l'IFP**

- Président de l'AFIA, Association Française pour l'Intelligence Artificielle (1998-2002). Membre du bureau depuis 1994, vice-président en 1997.
- Evalueur/auditeur/expert ESPRIT, Brite-EuRam, IST, IMS pour la Commission Européenne.
- Membre du Comité d'Evaluation du Réseau National des Technologies Logicielles.
- Président du club IA et Procédés de la Société de Chimie Industrielle (trois ans).
- Président du groupe « Dynamique des Systèmes » de l'AF CET (quatre ans) .
- Membre de conseils scientifiques : ERIC, Equipe de Recherche en Ingénierie des Connaissances, Université Lumière Lyon 2 ; ESIEA Recherche ; Centre de Recherche en Informatique de Paris 5.
- Membre du bureau du club CRIN "Ingénierie du Traitement de l'Information".
- Membre du conseil d'administration de l'ASTI (2001 -).
- Lauréat d'une prime aux innovateurs du groupe Elf en 1987.
- Prix de la meilleure présentation orale de la conférence "ESCAPE-12", mai 2002.
- Membre du Management Board du réseau d'excellence "EvoNet", réseau européen sur l'évolution artificielle.
- Membre du Management Board du réseau « ERNST », réseau de projets européens Brite-EuRam.
- Enseignement en dynamique des systèmes et en intelligence artificielle à l'IIE, Sup'Elec, ENSPM.
- Président et membre de comités de nombreuses conférences.

◆ **Rédaction d'ouvrages - Principales publications**

- Auteur de l'ouvrage « La simulation sur micro-ordinateur », Eyrolles, 1985.
- Editeur de la série d'ouvrages collectifs « Artificial Intelligence in the Petroleum Industry : Symbolic and Computational Applications », Editions Technip. 2 volumes parus, le premier avec R. Day, le second avec B. Bremdal
- Editeur de l'ouvrage « Software Architectures and Tools for Computer-Aided Process Engineering », en collaboration avec R. Gani, à paraître chez Elsevier en septembre 2002.
- 80+ publications, rapports sous référence IFP.

## 8. Références

Cette section est composée de trois parties.

La première partie présente les thèses et séjours post-doctoraux qui se sont déroulés sous ma direction, et les jurys de thèses auxquels j'ai participé.

La deuxième partie présente la plupart de mes publications, regroupées selon leur nature (ouvrages, articles, actes de conférences, rapports) et comporte une section spéciale "CAPE-OPEN".

La troisième partie regroupe les références citées dans ce mémoire.

Pour lever les ambiguïtés dans les deux premières parties, l'année de chaque référence est suivie d'une lettre indiquant son type: t (thèse), p (post-doc), L (livre), r (article de revue), C (conférence), m (mémoire), o (cape-open), D (divers).

Lorsque ce système ne suffit pas, les références sont nommées a,b,c,d etc.

### 8.1 Thèses et séjours postdoctoraux

- Thèses encadrées
- [Ferraz 93t] Ferraz-Simha C. (1993) SACRE. Système d'aide au contrôle de résultats expérimentaux, Thèse de doctorat de l'Université Paris XIII
- [Wahl 93t] Wahl F. (1993) Un environnement d'aide aux ingénieurs basé sur une architecture en tâches et sur un module de visualisation de courbes. Application à la conception de procédés de raffinage, Thèse de l'Ecole Nationale Des Ponts Et Chaussées, Janvier 1993
- [Mousset 94t] Mousset E., (1994) L'apport des réseaux connexionnistes à la géophysique. Application au traitement des signaux sismiques. Thèse de doctorat de l'université Pierre et Marie Curie, Février 1994
- [Cauvin 95t] Cauvin S. (1995) : Un environnement générique à base de connaissances pour la supervision de procédés de raffinage et de pétrochimie. Thèse de Doctorat du Conservatoire National des Arts et Métiers, Novembre 1995
- [Premoli 97t] Premoli G. (1997) Contributo allo Studio dei Modelli di Cinetica Chimica attraverso Metodi Simbolici e Numerici, Tesi di Laurea, Università degli Studi di Milano, Corso di Laurea in Scienze dell'Informazione, Février 1997
- [Mansanné 00t] Mansanné F. (2000) Analyse d'algorithmes d'évolution artificielle appliqués au domaine pétrolier: inversion sismique et approximation de fonctions. Thèse de l'Université de Pau et des Pays de l'Adour, Octobre 2000

- [Leblanc 02t] Leblanc B. (2002) Simulations Moléculaires de Monte Carlo : amélioration de l'efficacité statistique de l'échantillonnage grâce aux algorithmes d'évolution artificielle. Thèse de doctorat de l'Université Paris XI, Mars 2002
  
- **Jurys de thèses**
  
- [Bourseau 93t] Bourseau P., « Modélisation des Connaissances de l'Ingénieur de Procédés. Application de l'Intelligence Artificielle ». Thèse de doctorat d'état ès sciences physiques, Université Pierre et Marie Curie, Janvier 93
  
- [Fauveau 93t] Fauveau A., Construction d'un système expert en économétrie, application à la demande d'énergie, Thèse d'Economie de l'Université Paris II, Mai 1993
  
- [Jourda 96t] Jourda L., Composants logiciels orientés objet pour la modélisation et la simulation des procédés chimiques, Thèse de l'Institut National Polytechnique de Toulouse, Spécialité Génie des Procédés, Décembre 1996
  
- [Moysse 00t] Moysse A., Odysseo, Plate-forme orientée objet pour la modélisation dynamique des procédés, Thèse de l'Institut National Polytechnique de Toulouse, Spécialité Génie des Procédés, Novembre 2000
  
- **Post-doctorants encadrés**
  
- [Junker 95p] Junker U., Braunschweig B. (1995) « History-based Interpretation of Finite Elements Simulations of Seismic Wave fields ». International Joint Conference on Artificial Intelligence, Montréal.
  
- [Surma 96p] Surma J. (1996) Supporting flowsheet design by cased-based retrieval, rapport interne IFP 43060
  
- [Lafargue 98p] Lafargue E., Naanaa W., Braunschweig B. (1998), Méthode pour interpréter des caractéristiques pétrolières de sédiments géologiques, Brevet français 98/15102
  
- [Batres 2002p] Batres R. Documents internes divers sur le projet européen CHEM, 2001-2002

## 8.2 Publications personnelles

- *Ouvrages, articles dans mes ouvrages, numéro spécial de revue*
- [Braunschweig 85L] Braunschweig B., *La simulation sur micro-ordinateur*, Eyrolles Editeur, collection Informatique et Entreprise, 1985, 195p.
- [Braunschweig 92L] Braunschweig B. (éditeur) *Revue de l'Institut Français du Pétrole, Special issue on Artificial Intelligence : Knowledge-based Systems, Neural Networks, Fuzzy Logic*, Mai-Juin 1992
- [Braunschweig 95La] Braunschweig B., Day Ron, eds. *Artificial intelligence in the petroleum industry. Symbolic and computational applications. Series on artificial intelligence in the petroleum industry, Vol. 1*. Editions Technip, Paris, 1995, 480 p
- [Braunschweig 95Lb] Braunschweig B., Day R. « Prolegomena : an Overview of AI Techniques and of their use in the Petroleum Industry » *Artificial Intelligence in the Petroleum Industry, Symbolic and Computational Applications Vol. 1* .Editions Technip, 1995.
- [Braunschweig 95Lc] Braunschweig B., Barreau Alain, Emami E "Genetic algorithms for the automatic adjustment of thermodynamic models" In : *Artificial intelligence in the petroleum industry. Symbolic and computational applications*, B. Braunschweig, R. Day eds., pp. 417-441.
- [Braunschweig 96La] Braunschweig B., Bremdal Bernt-A ,*Artificial intelligence in the petroleum industry. Symbolic and computational applications. Series on artificial intelligence in the petroleum industry, Vol. 2*. Editions Technip, Paris, 1996, 385 p.
- [Braunschweig 96Lb] Braunschweig B., Bremdal Bernt-A , *Successful applications of artificial intelligence in the petroleum industry*. In : *Artificial intelligence in the petroleum industry. Symbolic and computational applications, Vol. 2*, B. Braunschweig, B.A. Bremdal eds., pp. 1-20. Editions Technip, Paris, 1996, 385 p.
- [Braunschweig 02La] Braunschweig B., Gani Rafiqul, *Software Architectures and Tools for Computer-Aided Process Engineering*, Elsevier, Oct. 2002, 600p.
- [Braunschweig 02Lb] Braunschweig B., Gani Rafiqul, "Introduction", *Software Architectures and Tools for Computer-Aided Process Engineering*, Elsevier, Oct. 2002,
- [Braunschweig 02Lc] Belaud Jean-Pierre, Braunschweig B., White Michael: *The CAPE-OPEN Standard: Motivations, development process, technical architecture, examples*, in Braunschweig, GANI, *Software Architectures and Tools for Computer-Aided Process Engineering*, Elsevier, Oct. 2002
- [Braunschweig 02Ld] Batres Rafael, Braunschweig B. et al., *Software Agents*, in Braunschweig, GANI, *Software Architectures and Tools for Computer-Aided Process Engineering*, Elsevier, Oct. 2002



- Articles parus dans des revues <sup>27</sup>
- [Braunschweig 83r] Braunschweig B. "Maturité ?" , Annales des Composites, Vol 3, 1983
- [Braunschweig 85r] Braunschweig B. " Dissipative structures, systems dynamics, and oil exploration" Environment and Planning B : Planning and Design, Special issue on new approaches in dynamical systems modeling, Vol. 12 #1, 1985
- [Braunschweig 87r] Braunschweig B., Papaz B. (1987) « TOPSI User's Manual », Document Elf Aquitaine, Paris
- [Braunschweig 88r] Braunschweig B., Papaz B. « TOPSI : The Overall Project Simulator », Trait d'Union, Revue Interne Elf Aquitaine, Paris 1988
- [Braunschweig 90ra] Braunschweig B., Papaz B., Vial D. (1990) « Topsis, The Overall Project Simulator », 1990 Afitep & International Cost Engineering Congress, Paris, Mars 1990
- [Braunschweig 90rb] Braunschweig B., Artificial intelligence in the petroleum world, Revue de l'Institut Français du Pétrole, vol. 45, n° 5, sep.-oct. 1990, p. 683-698.
- [Braunschweig 91ra] Braunschweig B. "Pour une intelligence des systèmes dynamiques", Revue Internationale de Systémique, vol. 5, n° 2, 1991, p. 133-141
- [Braunschweig 91rb] Braunschweig B., Artificial intelligence in the petroleum world. Pétrole et techniques, n° 362, mars 1991, p. 22-31.
- [Braunschweig 92r] Braunschweig B. , Introduction. Artificial intelligence and the oil and gas industry : a variety of methods for a variety of applications. Revue de l'Institut Français du Pétrole, vol. 47, n° 3, mai-juin 1992, p. 299-304.
- [Cauvin 92r] Cauvin Sylvie, Braunschweig B., Galtier Pierre, GLAIZE Yves "Alexip : an expert system coupled with a dynamic simulator for the supervision of the alphanol process. Revue de l'Institut français du pétrole, vol. 47, n° 3, mai-juin 1992, p. 375-382.
- [Cauvin 93ra] Cauvin Sylvie, Braunschweig B. "Graphical knowledge representation in the Alexip system for petrochemical process supervision". In : Application of artificial intelligence in engineering, Vol. 2, G. Rzevski et al. eds., pp. 219-233. Editeur: PCMP, Southampton/Elsevier Applied Science, London, 1993.
- [Cauvin 93rb] Cauvin Sylvie, Braunschweig B., Galtier Pierre, Glaize Yves "Model-based diagnosis for continuous process supervision : the Alexip experience." Artificial Intelligence - Engineering Applications, Vol. 6, No. 4, 1993, pp. 333-343.

---

<sup>27</sup> Je n'ai que très peu de traces de mes publications dans des revues et dans des conférences avant 1989. Je n'ai pas gardé les archives de ce que j'ai fait dans le groupe Elf Aquitaine. Il y avait un certain nombre de publications dans le milieu de la dynamique des Systèmes, sur des modèles et des logiciels; mais impossible de restituer ces informations début 2002.

- [Ferraz 93r] Ferraz-Simha Claudia, Braunschweig B., Muratet G "SACRE : système d'aide au contrôle de résultats expérimentaux. Application à un pilote de reformage catalytique" In : Récents progrès en génie des procédés, vol. 7, n° 29, 1993, p. 167-172. Editeur: Technique et Documentation-Lavoisier, Paris, 1993.
- [Gondran 00r] Gondran Michel, Braunschweig B. et coll., Architecture des logiciels et reutilisation de composants, Recommandations, Observatoire Français des Technologies Avancées, Architectures Logicielles et Réutilisation de Composants, Arago 24, Oct. 2000.
- [Surma 96r] Surma J., Braunschweig B. « Case-Based Retrieval in Process Engineering : Supporting Design by Reusing Flowsheets ». Engineering Applications of Artificial Intelligence, Special Volume on Engineering Design, 1996
- [Braunschweig 01r] Braunschweig B., Triz, The Evolution of Software Objects and Soft Computing in Reservoir Modelling, Foreword to "Soft Computing for Reservoir Characterization and Modeling", Wong P., Aminzadeh F., Nikravezsh M. eds. , Physica-Verlag, 2001
  
- **Actes de conférences**
- [Arosio 96C] Arosio P., Benjelloun-Dabaghi Z., Braunschweig B., « Outils d'aide à la Simulation en Cinétique Chimique ». SIMO 96, Toulouse, 1996
- [Braunschweig 82C] Braunschweig B. "Structures dissipatives, dynamique des systèmes et exploration pétrolière", 7<sup>ème</sup> Conférence Internationale de Dynamique des Systèmes, Bruxelles, 1982.
- [Braunschweig 83C] Braunschweig B. "A technico-economical simulation model focusing on enhanced recovery in oil fields", 1983 International System Dynamics Conference, Boston
- [Braunschweig 84C] Braunschweig B., Huot J.C. "A simple model of project management", 1984 American Association of Cost Engineers Congress, Montréal
- [Braunschweig 90Ca] Braunschweig B. , Artificial intelligence in the E&P world.: In : CAIPEP 1990 - Conference on Artificial Intelligence in Petroleum Exploration and Production, College Station, May 16-18 1990, Proceedings, pp. 7-17.
- [Braunschweig 90Cb] Braunschweig B. "Pour une intelligence des systèmes dynamiques", Congrès Européen de Systémique, In : Systématique (La). Congrès européen, Lausanne, 3-6 oct. 1989, Actes, t. 2
- [Braunschweig 91Ca] Barreau Alain, Braunschweig B., Emami E., Behar Emmanuel , "A knowledge-based system for the automation of thermodynamic models adjustment". In : CAIPEP 1991 - Conference on Artificial Intelligence in Petroleum Exploration and Production, College Station, May 15-17 1991, Proceedings, pp. 55-62.
- [Braunschweig 91Cb] Braunschweig B., Lambert JM., Naim P. "Neural networks applications within IFP".European oil and gas conference (The) - A multidisciplinary

- approach in exploration and production R&D, Palermo, Oct. 9-12 1990, Proceedings, G. Imarisio et al. eds., pp. 464-475. Editeur: Graham & Trotman, London, 1991.
- [Braunschweig 91Cc] Braunschweig B., "Modèles Dynamiques de Simulation et Systèmes à Base de Connaissances" Séminaire CNRS, Grenoble, 1991
  - [Braunschweig 92C] Braunschweig B. Genetic algorithms with some application examples. In : CAIPEP 1992 - Conference on Artificial Intelligence in Petroleum Exploration and Production, Houston, July 22-24 1992, Proceedings, pp. 171-186.
  - [Braunschweig 95Ca] Braunschweig B. Pavone D., , « Classification d'expériences de forages par réseaux de neurones non supervisés ». ICANN 95, Paris, 1995
  - [Braunschweig 95Cb] Braunschweig B. Surma J. « Réutilisation d'Etudes de Procédés ». Interchimie 95, Paris, 1995
  - [Braunschweig 01C] Braunschweig B., Jaecker-Voirol Anne, Le Thiez Pierre, How advanced modelling improves environmental assessment and process design/operation, Bridging The Gap Conference, Stockholm, May 2001
  - [Cauvin 91Ca] Cauvin Sylvie, Braunschweig B., Galtier Pierre, Glaize Yves; "Alexip : système expert couplé à un simulateur dynamique pour la supervision du procédé Alphabutol. "In : GP - Mesures-Capteurs-Simulation-Commande. 3e congrès français de Génie des Procédés, Compiègne, 4-6 sep. 1991, G. Antonini, R. Ben Aim eds., vol. 5, n° 13, 1991, Lavoisier - Technique et Documentation, Paris, 1991.
  - [Cauvin 91Cb] Cauvin Sylvie, Braunschweig B., Galtier Pierre, Glaize Yves "Alexip : expert system coupled with a dynamic simulator for the supervision of the Alphabutol process". In : EuroCAIPEP - European Conference on Artificial Intelligence in Petroleum Exploration and Production, Rueil-Malmaison, Oct. 16-18 1991, Proceedings, 7 p.
  - [Cauvin 92C] CAUVIN Sylvie, Braunschweig B., Galtier Pierre, Glaize Yves, Model-based diagnosis for continuous process supervision : the Alexip experience., In : IFAC/IFIP/IMACS 1992 - Artificial intelligence in real-time control, International symposium, Delft, June 16-18 1992, Preprints, pp. 515-521.
  - [Cauvin 93C] Cauvin Sylvie, Braunschweig B. : "Graphical knowledge representation in the Alexip system for petrochemical process supervision". AIENG - Application of Artificial Intelligence in Engineering symposium, Toulouse, June 29-July 1 1993.
  - [Cauvin 96C] Cauvin Sylvie, Braunschweig B. "Représentation graphique de la connaissance dans le système de supervision de procédés pétrochimiques Alexip". In : Evolution de l'information, incidence dans l'entreprise. Congrès, Martigues, 22 mars 1996, comptes rendus, 7 p.
  - [Cauvin 99C] Cauvin Sylvie, Braunschweig B., Systèmes à base de connaissances pour la supervision de procédés de raffinage et de pétrochimie. In : ISAI'99 - Congrès 'Intelligence, Systèmes, Applications et Innovation', Palaiseau, juin 1999, actes, 14 p.

- [De Saint-Etienne 88C] de Saint-Etienne A., Braunschweig B. 1988, « Un système expert d'aide à l'ajustement de modèles de simulation », Groupe de travail Dynamique des Systèmes de l'AF CET, Janvier 1988.
- [Emami 93C] Emami E., Barreau Alain, Braunschweig B. , Genetic algorithms for the automatic adjustment of thermodynamic models. In : EuroCAIPEP 93 - European Conference on Artificial Intelligence in the Petroleum Exploration and Production, Aberdeen, Sep. 20-22 1993, Proceedings, 8 p.
- [Ferraz 93C] Ferraz-Simha Claudia, Braunschweig B., Muratet G , "SACRE : système d'aide au contrôle de résultats expérimentaux. Application à un pilote de reformage catalytique." GP - Conduite et commande des procédés. 4e congrès français de Génie des Procédés, Grenoble, 21-23 sep. 1993.
- [François 98C] Francois Olivier et Braunschweig B., Une étude de la pratique des algorithmes évolutionnaires en France, Journée Evolutionnaire Trimestrielle, Décembre 1998
- [Guérillot 89C] Guérillot D., Braunschweig B., Blanc G. "Application of AI concepts to build interfaces for reservoir simulators", 2<sup>nd</sup> Conference on Artificial Intelligence in Petroleum Exploration and Production, College Station, 1989
- [Heim 02C] Bruno Heim, Sylviane Gentil, Sylvie Cauvin, Louise Travé-Massuyes, Braunschweig B., Fault diagnosis of a continuous process using imprecise quantitative knowledge and causal models, Soumis à PAIS 2002, Prestigious Applications of Intelligent Systems, Lyon, Juillet 2002.
- [Junker 95Ca] Junker U., Braunschweig B. « History-based Interpretation of Finite Elements Simulations of Seismic Wave fields ». International Joint Conference on Artificial Intelligence, Montréal, 1995.
- [Junker 95Cb] JUNKER U., Braunschweig B., Bamberger Alain Nicoletis Laurence , "Sismonaute: history-based interpretation of wave propagation in synthetic seismic simulations. " In : AI Petro'95 - International conference on artificial intelligence in the petroleum industry, Lillehammer, Sep. 13-15 1995, Proceedings.
- [Leblanc 01Ca] Benoit Leblanc, Evelyne Lutton, Braunschweig B., Hervé Toulhoat, Improving molecular simulation : a meta optimisation of Monte Carlo parameters. International Conference on Evolutionary Computation, Corée, Mai 2001
- [Leblanc 01Cb] Benoit Leblanc, Evelyne Lutton, Braunschweig B., Hervée Toulhoat, Experimenting history and never ending life in Evolutionary Computation, Evolution Artificielle 2001, Le Creusot, Oct. 2001.
- [Leblanc 02Ca] Benoit Leblanc, Evelyne Lutton, Hervé Toulhoat, Bertrand Braunschweig (2002) "History and Immortality in Evolutionary Computation",Lecture Notes in Computer Science, 2002
- [Leblanc 02Cb] Benoît Leblanc, Evelyne Lutton, Braunschweig B., Hervé Toulhoat, "Mixing Monte Carlo moves more efficiently with an Evolutionary Algorithm"

Symposium on Enhanced Sampling Methods for Molecular Dynamics and Monte Carlo Simulations", 2002 Spring National ACS Meeting, Orlando, Avril 2002

- [Schoenauer 98C] Schoenauer Marc, Ehinger Andreas, Braunschweig B. , Non-parametric identification of geological models In : IEEE - Evolutionary computation. International conference of the Institute of Electrical and Electronic Engineers, Anchorage, May 4-9 1998, Proceedings, pp. 136-141.
- [Surma 96C] Surma J., Braunschweig B. « REPRO : Supporting Flowsheet Design by Case-Based Retrieval ». European Workshop on Case-Based Reasoning, Oct 1996
- [Wahl 95C] Wahl F., Braunschweig B., « Prince, A Knowledge-Based System which Binds Process Specification to Experimental Data ». Intelligent Systems in Process Engineering, Snowmass, USA , 1995
  
- **Mémoires**
- [Braunschweig 77m] Braunschweig B. "Application de la dynamique des systèmes au développement des champs pétroliers", mémoire d'ingénieur I.I.E., 1977
- [Braunschweig 98m] Braunschweig B. Aides à l'interprétation de simulations dynamiques. Application aux modèles de cinétique chimique. Thèse de docteur en méthodes scientifiques de gestion, université Paris IX, 1998. Editions Technip, Paris, 1998, 196 p.
  
- **Rubrique "CAPE-OPEN"**
- [Banks 01o] Peter Banks, Braunschweig B., Rafiqul Gani, Kerry Irons, Knut Mathisen, Peter Mauer, Michel Pons, Pascal Roux , Global CAPE-OPEN (GCO) Project Results to Date, 2001 AIChE Annual Meeting, Reno, NV, Nov 4-9, Modelling and Computations for Process Design #2
- [Belaud 01oa] Belaud Jean-Pierre, Braunschweig B., Halloran Michael, Irons Kerry, PINOL Daniel, "New generation simulation environment: technical vision of the CAPE-OPEN standard" 2001 AIChE Annual Meeting, Reno, NV, Nov 4-9, Modelling and Computations for Process Design #2
- [Belaud 01ob] Jean-Pierre Belaud, Braunschweig B., Michel Pons, Open Software Architecture For Process Simulation : The Current Status of CAPE-OPEN Standard, accepté à ESCAPE-12, La Haye, Mai 2002
- [Braunschweig 96o] Braunschweig B., « Cape-Open et OS-CAPE : Vers un Standard pour la Simulation de Procédés ». Conférence invitée, SIMO 96, Toulouse, Oct. 1996
- [Braunschweig 98o] Braunschweig B., « How Online Fault Diagnosis and Supervision Systems can Benefit from the CAPE-OPEN standard interfaces in Process Simulation», IFAC Workshop on Online Fault Detection and Supervision in the Chemical Process Industries, June 1998, Solaize, France

- [Braunschweig 99oa] Braunschweig B. Britt H., Pantelides C., Sama S., Open Software Architectures for Process Modelling: Current Status and Future Perspectives, FOCAPD'99 Conference, Breckenridge, Colorado, July 1999
- [Braunschweig 99ob] Braunschweig B., Global CAPE-OPEN, Delivering the power of component software and open standard interfaces in computer-aided process engineering, IST Conference, Helsinki, Nov. 1999
- [Braunschweig 00oa] Braunschweig B. , Architectures ouvertes pour l'ingenierie de procedes: le standard CAPE-OPEN, Observatoire Français des Technologies Avancées, Architectures Logicielles et Réutilisation de Composants, Arago 24, Oct. 2000.
- [Braunschweig 00ob] Braunschweig B., Irons Kerry: "Global CAPE-OPEN: delivering the power of component software and open standard interfaces in computer-aided process engineering", 2000 AIChE Annual Meeting, March 2000
- [Braunschweig 00oc] Braunschweig B., Pantelides Constantinos-C., Britt Herbert-I., Sama Sergi, Process modeling: the promise of open software architectures. Chemical Engineering Progress, September 2000, pp. 65-76
- [Braunschweig 01oa] B.Braunschweig, K. Irons, M. Pons, "The CAPE-OPEN Laboratories Network, An IMS project gives birth to a non-profit neutral industry and academic association", IMS Forum, Locarno, Oct. 2001
- [Braunschweig 01ob] Braunschweig B., Paen Didier, Roux Pascal, Vacher Philippe An interface to make unit operations interoperable. Petroleum Technology Quarterly, Autumn 2001, pp. 79-85.
- [Braunschweig 01oc] Braunschweig B., Paen Didier, Roux Pascal, Vacher Philippe, The use of CAPE-OPEN interfaces for interoperability of Unit Operations and Thermodynamic Packages in Process Modelling,. ERTC - European Refining Technology Conference, Paris, 18-20 June 2001.
- [Braunschweig 02oa] Braunschweig B., Interopérabilité de composants de modélisation: opportunités et perspectives, Conférence SIMO 2002, Oct. 2002, Toulouse
- [Braunschweig 02ob] Braunschweig B., Irons Kerry, TRIZ and the evolution of CAPE tools, From FLOWTRAN® to CAPE-OPEN-® and beyond, ESCAPE-12, La Haye, Mai 2002 (*prix de la meilleure présentation orale*).
- [Braunschweig 02oc] B. Braunschweig, E.S. Fraga, Z. Guessoum, D. Paen, D. Piñol, A. Yang, (2002) Cogents: Cognitive Middleware Agents to Support e-CAPE. e-Business e-Work conference, Prague, Octobre 2002

- [Freund 00o] Freund E., *Global CAPE-OPEN: a new collaboration framework between users and suppliers of CAPE software*, Hyprotech 2000 Conference, Amsterdam, Nov. 2000<sup>28</sup>
- [Jarke 99oa] Jarke M., Becks A., Tresp C., Zlatintsis S., Braunschweig B., *Designing Standards for Open Simulation Environments in the Chemical Industries: A Computer-Supported Use-Case Approach*, INCOSE'99, Brighton, UK
- [Jarke 99ob] Jarke M., Köller J., Marquardt W., von Wedel L., Braunschweig B., *CAPE-OPEN: Experiences from a Standardization Effort in Chemical Industries* 1st IEEE Conference on Standardisation and Innovation in Information Technology, Aachen, Germany, September 15-17, 1999
- [Köller 01o] J. Köller, B. Braunschweig, K. Irons, M. Jarke, M. Pons, *The CAPE-OPEN Laboratories Network: Standards for Interoperable Process Engineering Software Components*, e-Business e-Work 2001 Conference, Venice, Oct. 2001
- [Pons 01oa] M. Pons, B. Braunschweig, K. Irons, *CO-LaN: missions and means of the CAPE-OPEN laboratories network*, 8<sup>e</sup> Congrès Français de Génie des Procédés, Nancy, Octobre 2001
- [Pons 01ob] M. PONS, Braunschweig B., K. Irons, J. Köller, A. Kuckelberg, and P. ROUX, *CO-LaN: Maintaining the CAPE-OPEN standard through a virtual organization*, World Congress of Chemical Engineering, Melbourne, Sept. 2001
- [Pons 01oc] M. PONS, Braunschweig B., K. Irons, R. Gani, P. Mauer, P. Banks, P. Roux, K. Mathisen, *Global CAPE-OPEN (GCO) Project Results to Date*, World Congress of Chemical Engineering, Melbourne, Sept. 2001
- [Pons 01od] Michel Pons, Braunschweig B., Kerry Irons, Alexander Kuckelberg, Jörg Koeller and Pascal Roux, *CAPE-OPEN (CO) standards: implementation and maintenance*, European Congress of Chemical Engineering 3, Nuremberg, June 2001
- [Pons 01oe] Michel Pons, Braunschweig B., Kerry Irons, Jörg Köller and Alexander Kuckelberg, *CAPE-OPEN (CO) standards: implementation and maintenance*, 2<sup>nd</sup> IEEE Conference on Standardisation and Innovation in Information Technology, Boulder, Oct. 2001
- [Pons 01of] Michel Pons, Braunschweig B., Kerry Irons, Jörg Köller, Alexander Kuckelberg, and Pascal Roux, *A virtual organisation for managing CAPE-OPEN (CO) standards*, fifth Italian Conference on Chemical and Process Engineering, Florence, Mai 2001
- [Pons 01og] Michel Pons; Braunschweig B., Kerry Irons, *Global CAPE-OPEN (GCO) Project Results To Date: June 2001*, European Congress of Chemical Engineering 3, Nuremberg, June 2001

---

- <sup>28</sup> (mon nom n'apparaît pas mais j' ai écrit cet article pour le DGA de l'IFP...)

- [Pons 01oh] Michel Pons; Braunschweig B., Kerry Irons, Global CAPE-OPEN (GCO) Project Results To Date: June 2001, fifth Italian Conference on Chemical and Process Engineering, Florence, Mai 2001
  
- **Bulletin AFIA**
  
- [Braunschweig 90d] Braunschweig B. : L'intelligence artificielle à l'IFP. Revue de l'Association française d'intelligence artificielle, n° 3, juil. 1990, p. 6-7.
- [Braunschweig 92d] Braunschweig B.: L'intelligence artificielle à l'Institut Français du Pétrole. Bulletin de l'Association française d'intelligence artificielle, n° 10, juil. 1992, p. 13-18.
- [Braunschweig 94d] Braunschweig B., Dossier pétrole-chimie: Bulletin de l'AFIA, n° 19, oct. 1994, pp. 21-52.
- Plusieurs autres contributions au bulletin (éditoriaux, rapports moraux, comptes rendus, etc.) dont je ne donne pas le détail.
  
- **Documents de projets coopératifs**
  
- [Braunschweig 96d] CAPE-OPEN, Proposition de projet BRITE-EURAM, 1996; 15 partenaires, 30 mois. Acceptée et financée par la Commission Européenne.
- [Braunschweig 98d] Global CAPE-OPEN, Proposition de projet BRITE-EURAM, 1998; 25 partenaires Européens et internationaux, 30 mois. Financée par la CE.
- [Braunschweig 99d] BF-CHARISMA, Proposition de projet HUMAN, 1999; Bourses Marie Curie Industrial Host Institution. Acceptée et financée par la CE.
- [Braunschweig 00d] GCO-Support, Proposition de mesure d'accompagnement GROWTH; 4 partenaires Européens, 1 US, 1 japonais, 24 mois. Financée par la CE.
- [Braunschweig 01da] Cogents, Proposition de projet IST, 2001; 5 partenaires, 24 mois. Acceptée et financée par la Commission Européenne.
  
- **Autres**
  
- [Braunschweig 01db] Braunschweig B. Perspectives pour les Logiciels Scientifiques et Techniques, note technique interne IFP, Mars 2001
- [Lafargue et coll. 98d] Lafargue E., Naanaa W., Braunschweig B. (1998), Méthode pour interpréter des caractéristiques pétrolières de sédiments géologiques, Brevet français 98/15102
- Rapports IFP produits par le groupe IA, 54 références internes IFP, non incluses.



### 8.3 Références citées dans ce mémoire

- [Aamodt 94] Aamodt A. and Plaza E. (1994) *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*. *AI Communication* 7, (1), 39-59
- [Abbassi 92] Abbassi A. (1992), « Réseaux Neuronaux pour l'Interprétation Quantitative de Diagraphies », Rapport IFP 40107, Novembre 1992
- [Abdel-Hamid 89] Abdel-Hamid T., Madnick S., (1989), "Software productivity : potential, actual, and percieved", *System Dynamics Review*, Volume 5 Number 2, Summer 1989
- [Altshuller 54] Altshuller G. (1954) *40 Principles : Triz Keys to Technical Innovation*, Uri Fedoseev, Steven Rodman (Translator), Lev Shulyak (Translator)
- [Altshuller 96] H. Altov. (1996) *And Suddenly the Inventor Appeared : TRIZ, the Theory of Inventive Problem Solving* (Translated and adapted from the Russian by Lev Shulyak.) Worcester, MA: Technical Innovation Center, 1996. (2nd Edition)<sup>29</sup>
- [Altshuller 00] Altshuller G. (2000) *The innovation algorithm : TRIZ, systematic innovation and technical creativity*; Technical innovation center, Worcester (Mass.)
- [Arosio 95a] Arosio P., (1995) « SPIKE , Software Platform for Interactive Chemical Kinetics Experiments », Rapport IFP, 1995
- [Arosio 95b] Arosio P., (1995) « SPIKE , Software Platform for Interactive Chemical Kinetics Experiments, Référence Manual », Rapport IFP, 1995
- [Arosio 97] Arosio P., Premoli G., (1997) « Spike. A System for Representing and Studying Chemical Mechanisms » Référence Manual. Document interne IFP, Janvier 1997
- [Batres 99] Batres, R., M. L. Lu and Y. Naka (1999). *A Multidimensional Design Framework and its Implementation in an Engineering Design Environment*, *Concurrent Engineering: Research and Applications*, Vol. 7, No. 1, pp. 43-54
- [Bogusch 01] Bogush R. (2001), *Eine Software-umgebung für die rechnergestützte Modellierung verfahrenstechnischer Prozesse*, Diploma-Thesis, Berichte aus dem Lehrstuhl für Porzesstechnik, Aachen
- [Boston 74] Boston, J. F., and S. L.Sullivan, Jr., (1974) "A New Class of Solution Methods for Multi-component, Multistage Separation Processes," *The Canadian Journal Of Chemical Engineering*, 52, 52-63(1974).
- [Bourseau 95a] Bourseau P. (1995) " Approches Objets en Genie Chimique : Applications et Enjeux". *Interchimie* 95

---

<sup>29</sup> H. Altov est un pseudonyme de G. Altshuller

- [Bourseau 95b] Bourseau P., Emond F., Thomas G., Vacher P., (1995) "Approche orientee objet appliquee a la simulation statique des procedes" Rapport IFP N° 41 993- Mars 1995
- [Bourseau 96] Bourseau P. (1996) Modélisation des connaissances de l'ingénieur de procédé : application des techniques de l'intelligence artificielle, Thèse d'état ès-Sciences de l'Université Pierre et Marie Curie (Paris 6) ;
- [Brice 95] Brice A.A. and Johns W.R. (1995) Open Process Simulation.Parts 1,2,3. Technical Report for the Object-Oriented CAPE Consortium, Intera Information Technologies.
- [Briot Demazeau 01] Briot J.-P., Demazeau Y., (2001) *Principes et Architecture des systèmes multi-agents*, Hermès Science Publications, collection IC2, novembre 2001.
- [Cavalucci 99] Cavalucci, D. (1999) *Contribution à la conception de nouveaux systèmes mécaniques par intégration méthodologique*, thèse de l'ENSAIS/Université Louis Pasteur, décembre 1999
- [Chandrasekaran 93] B. Chandrasekaran and T.R. Johnson, (1995) "Generic Tasks And Task Structures: History, Critique and New Directions," *Second Generation Expert Systems*, eds., J.M. David, J.P. Krivine, and R. Simmons, Springer-Verlag, 1993, pp. 239-280.
- [Chauveau 91] Chauveau J.-M., Lardera S. (1991) *Regards sur la Gestion de Projet*, Cahiers de la Recherche, Ecole Supérieure de Commerce de Paris, Série Management de la Technologie
- [Chauvet 00] Chauvet J.-M., (2000) *Composants logiciels : un état (éphémère) de l'art*. Rapport Arago 28, Observatoire Français des Technologies Avancées, Oct. 2000
- [CHEM 02] Projet CHEM (2002) [www.chem-dss.org](http://www.chem-dss.org)
- [CO-LaN 02] CO-LaN, (2002) CAPE-OPEN Laboratories Network web portal, [www.colan.org](http://www.colan.org)
- [Collet 00] Pierre Collet, Evelyne Lutton, Marc Schoenauer, Jean Louchet, (2000), "Take it EASEA," *Parallel Problem Solving from Nature VI*, vol 1917, Springer pp 891-901, Paris, September 2000
- [Cooper 80] Cooper K.G. (1980), "Naval Ship Production : A Claim Settled and a Framework Built", *Interfaces* 10, December 1980
- [Currie 91] Ken Currie and Austin Tate (1991), *O-Plan: the Open Planning Architecture - Artificial Intelligence Vol. 52*, pp 49-86, 1991
- [Davallo 89] Davalo E., Naim P. (1989) *Des réseaux de Neurones*, Eyrolles, Paris
- [Dague 97] L.Trave Massuyes , P.Dague , F.Guerrin (1997), *Le raisonnement qualitatif pour les sciences de l'ingénieur*, Rapport LAAS No97234, Ed. Hermes "Diagnostic et Maintenance", N°ISBN 2-86601-611-4, 1997, 505p.

- [de Garis 02] Hugo de Garis, Michael Korkin, (2002) "THE CAM-BRAIN MACHINE (CBM) An FPGA Based Hardware Tool which Evolves a 1000 Neuron Net Circuit Module in Seconds and Updates a 75 Million Neuron Artificial Brain for Real Time Robot Control", *Neurocomputing journal*, Elsevier, Vol. 42, Issue 1-4, February, 2002. Special issue on Evolutionary Neural Systems,
- [de Kleer 84] Kleer J.de, Brown J.S. (1984) "A Qualitative Physics Based on Confluences" Special Volume on Qualitative Reasoning about Physical Systems, *Artificial Intelligence*, 24(1-3), 7-84, 1984
- [De Saint-Etienne 87] de Saint-Etienne A., (1987) "Un système expert d'aide à l'ajustement de modèles de simulation", *Mémoire d'ingénieur IIE*, Juin 1987
- [Donnadiou 02] Donnadiou, G. and M. Karsky, (2002) *La Systémique, penser et agir dans la complexité. Entreprises et Carrières*. Editions Liaisons, Paris.
- [Draelants 92] Draelants G. , (1992) *Vergelijkende Studie van de Schatting van Well-Logs met Behulp van Lineaire Regressie and Neurale Netwerken*, Verhandeling voorgedragen tot het bekomen van de graad van Burgerlijk Mijningenieur, Katholieke Universiteit Te Leuven (Sept. 1992, en néerlandais).
- [Drogoul 02] Drogoul A. (2002) *Cours de DEA/IARFA*, en ligne sur le site [www-poleia.lip6.fr/~drogoul/cours/iarfa/](http://www-poleia.lip6.fr/~drogoul/cours/iarfa/)
- [Eisenberg, 1991] M. Eisenberg. (1991) *The kineticist's work-bench: Combining symbolic and numerical methods in the simulation of chemical reaction mechanisms*. Technical Report 1306, MIT Artificial Intelligence Laboratory, 1991.
- [Farreny 87] Farreny H., Ghallab M. (1987) *Eléments d'Intelligence Artificielle, Hermès, Traité des Nouvelles Technologies*, Paris
- [Feigenbaum 88] Feigenbaum E., McCorduck P., Nii H.P., (1988) *The Rise of the Expert Company*, Times Books, New York
- [Fikes 93] Richard Fikes (1993) : STRIPS, A Retrospective. *Artificial Intelligence* 59(1-2): 227-232 (1993)
- [FIPA 02] FIPA (2002) *Foundation for Intelligent Physical Agents*, [www.fipa.org](http://www.fipa.org)
- [Forbus 90] Forbus K.D., Falkenhainer B. (1990) "Self-Explanatory Simulations: An Integration of Qualitative and Quantitative Knowledge" *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, AAAI Press/MIT Press, Menlo Park, CA, pp.380-387, 1990
- [Forbus 91] Forbus K.D. (1990) "Qualitative Spatial Reasoning." *Artificial Intelligence*, 51(1-3), 1991
- [Forrester 61] Forrester J. (1961) *Industrial Dynamics*, MIT Press, Cambridge
- [Forrester 68] Forrester J. (1968) *Principles of Systems* , Wright-Allen Press, Cambridge

- [Forrester 71] Forrester J. (1971) *World Dynamics*, Wright-Allen Press, Cambridge
- [Fraga 94] Fraga E.S. (1994) *The Implementation of a Portable Object-Oriented Distributed Process Engineering Environment*. Technical Report 1994-17. Department of Chemical Engineering, Edinburgh University (1994).
- [Fraga 00] E S Fraga, M A Steffens, I D L Bogle & A K Hind (2000), *An object oriented framework for process synthesis and optimization*, in *iFoundations of Computer-Aided Process Design*, MF Malone, JA Trainham & B Carnahan (Editors), AIChE Symposium Series 323.
- [Fraga 02] Fraga E. & A. Zilinskas (2002), *Experience with hybrid evolutionary/local optimization for process design*, in I C Parmee (editor), *Adaptive Computing in Design and Manufacture V*, Springer-Verlag
- [Gama 90] Groupe GAMA (1990) *La Gamme Automatique en usinage*, Hermès, 1990, ISBN 2-86601-255-0.
- [Gama 95] Gama, E., Helm, R., Johnson, R., and Vlissides, J., (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995
- [Gensym 02] Gensym (2002) *NeurOnLine Product*, [www.gensym.com](http://www.gensym.com)
- [Glansdorff 71] Glansdorff P., Prigogine I. (1971) *Structure, Stabilité et Fluctuations*, Masson et Cie, Paris
- [Goldberg 89] Goldberg, David E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Pub. Co. 1989.
- [Gondran 93] Gondran M. (1993) "Synthèse du Numérique et du Symbolique", note 93NJ00046 Coll. de notes internes de la Direction des Etudes et Recherches Mathématiques, informatique, télécommunications EDF, Mars 1991 (in french)
- [Guessoum 96] Z. Guessoum (1996) *Environnement de développement et de conception de systèmes multi-agents*. Thèse, Université Paris 6, 1996.
- [Guessoum 02] Guessoum Z., Briot J.P., (2002) *From Active Objects to Autonomous Agents*, Submitted to IEEE Concurrency - Special Series on Actors & Agents
- [Gutknecht 00] O. Gutknecht, J.Ferber & F. Michel, (2000) *MadKit: une plate-forme multi-agent générique*, Rapport de Recherche LIRMM 00061, Mai 2000
- [Hayes 85] P .J .Hayes. (1985) *Naive physics i: Ontology for liquids*. In J. Hobbs and B. Moore, editors, *Formal Theories of the Commonsense World*, pages 71-89. Ablex Publishing Corporation, 1985.
- [Heim 00] Heim B., Cauvin S., Gentil S., (2000) *Causal and fuzzy model-based reasoning methodology for cascaded loops diagnosis*. IAR-ICD 2000. Intelligent Control and Diagnosis workshop of the Institut franco-allemand pour les Applications de la Recherche, Nancy, Nov. 16 2000, Proceedings, 6 p.

- [Hérault 94] Hérault J., Jutten C. (1994) Réseaux Neuronaux et Traitement du Signal, Hermès, Traité des Nouvelles Technologies, Paris
- [IBM 01] IBM Corporation (2001), Autonomic Computing Manifesto, September 2001
- [IK-CAPE 95] IK-CAPE (1995) Thermodynamik-Schnittstelle für CAPE-Anwendungen, User's Guide, Projekt der IK-CAPE
- [IST 02] Commission Européenne (2002) Information Society Technologies, 2002 Work Programme, 2002
- [Jamin 91] Jamin F., (1991) Approche Connexionniste et Statistique pour l'Estimation du Cliquetis dans un Moteur à Allumage Commandé, Rapport IFP 39306, Janvier 1991
- [Jarke 95] Jarke, M. W. Marquardt, (1995). Design and evaluation of computer-aided process modelling tools. In : Intelligent Systems in Process Engineering, ISPE'95, Snowmass.
- [Jourda 93] Jourda L., (1993) " ATOM, Une bibliothèque d'objets pour l'estimation de propriétés thermodynamiques", ENSIGC -LEAP , Octobre 1993
- [Jourda 96] Jourda L., Joulia X. and Koehret B. (1996). Introducing ATOM, the Applied Thermodynamics Object-Oriented Model, 6th European Symposium on Computer Aided Process Engineering, May 26-29 1996, Rhodes, Greece
- [Junker 94] Junker U. (1994) « Sismonaute, A System for Detecting and Interpreting Wave Fronts in Seismic Simulations », dans Braunschweig B. & Day R. , Artificial Intelligence in the Petroleum Industries, Symbolic and Computational Applications, Vol. 1, Editions Technip, 1995
- [Keeler 95] Keeler S. (1995), Prediction and Control of Chaotic Chemical Reactions Via Neural Networks Models, dans [Braunschweig 95La]
- [Keijzer 01] M. Keijzer, J. J. Merelo, G. Romero, and M.Schoenauer (2001), Evolving Objects: a general purpose evolutionary computation library, Evolution Artificielle 2001, Le Creusot
- [Kohonen 89] Kohonen T. (1989) : Self Organization and Associative Memories; 3eme Edition, Springer-Verlag, 1989
- [Kohonen 93] Kohonen T. (1993): Things you Haven't Heard about the Self-Organizing Map; IEEE International Conference on Neural Networks; IEEE Neural Networks Council, pp. 1147-1156, 1993
- [Kohonen 97] Teuvo Kohonen (1997), Self-Organizing Maps, 2nd Edition, Series in Information Sciences , Springer-Verlag New York, May 1997
- [Kolodner 93] Kolodner , J. (1993) Case- Based Reasoning. Morgan Kaufmann., San Mateo (1993).
- [Kosko 88] Kosko B. (1988) Bidirectional Associative Memories, IEEE Transactions on Systems, Man and Cybernetics, vol. 18, no. 1, pp. 49-60, January-February 1988.

- [Kuipers 94] Kuipers B. (1994) *Qualitative Reasoning, Modeling and Simulation with Incomplete Knowledge*, MIT Press, Cambridge
- [Lambert 91] Lambert J.M., (1991) *Neural Network Controller Project. Rapport final du séjour au sein du département 'Electrical and Computer Engineering' de l'Université de Californie, San Diego, Rapport IFP N° 39320, Janvier 1991*
- [Lutton 01] Evelyne Lutton, Pierre Collet, Jean Louchet, (2001) *EASEA Comparisons on Test Functions : GALib Versus EO, EA2001, Le Creusot, France.*
- [Marquardt 92] Marquardt W. (1992) -"An Object-Oriented Representation of Structured Process Models" -*Comp. Chem. Eng. 16S, Proceedings of ESCAPE-I, Elsinore (Denmark), p. S329-S336*
- [Marquardt 96] Marquardt, W. (1996) *Trends in computer-aided process modelling. Computers chem. Engng, 20, 591-609.*
- [Marquardt 02] Marquardt W. (2002) , *Adaptivity in process systems modelling, ESCAPE-12 Conference, La Haye, Mai 2002*
- [Microsoft 99] Microsoft Corp. (1999) *Site web de COM, [www.microsoft.com/com](http://www.microsoft.com/com)*
- [Mousset 91] E. Mousset, (1991) "Spikes Filtering in Seismic Data with Neural Networks", *Conf. on AJ. in Petr. Expl. & Prod., p. 115-123, 1991.*
- [Nabec 89], Nabec R. (1989), *Contribution au développement d'un système expert de sélection et d'ajustement de modèles thermodynamiques appliqués aux fluides de gisements, Rapport IFP N° 37933.*
- [NITRD 02] National Science and Technology Council, Office of the President of United States of America, (2002) *Network and Information Technology Research and Development, 2002*
- [OMG 97] Object Management Group (1997) *The Common Object Request Broker : Architecture and Specification. Document OMG disponible sur le site [www.omg.org](http://www.omg.org)*
- [Orfali 96] Orfali R., Harkey D., Edwards J. (1996), *The Essential Distributed Objects Survival Guide, John Wiley & Sons, New York*
- [Pantelides 94] Pantelides, C.C., and H. I. Britt (1995). *Multipurpose Process Modeling Environments. In L.T. Biegler. and M.F. Doherty (Eds.), Proc. Conf. on Foundations of Computer-Aided Process Design '94. CACHE Publications, Austin, Texas. pp. 128-141.*
- [Perrot 95] Nicolas Revault, Houari A. Sahraoui, Gilles Blain, Jean-François Perrot, (1995) *A Metamodeling technique : The MÉTAGEN system, Tools Europe 95*
- [Piela 91] Piela P.C., et al. (1991) *ASCEND: An Object-Oriented Computer Environment For Modeling and Analysis: The Modeling Language. Computers & Chemical Engineering 15, (1), 53-72 (1991).*
- [Quinlan 93] J.R Quinlan. (1993) *C4.5 Programs for machine learning. Morgan Kaufmann, San Mateo, Californie, 1993.*

- [Rea 01a] Rea K. (2001) TRIZ and Software - 40 Principle Analogies, Part 1, Triz Journal, [www.triz-journal.com/2001/09/e/index.htm](http://www.triz-journal.com/2001/09/e/index.htm)
- [Rea 01b] Rea K. (2001) TRIZ and Software - 40 Principle Analogies, Part 2, Triz Journal, [www.triz-journal.com/2001/09/e/index.htm](http://www.triz-journal.com/2001/09/e/index.htm)
- [Rea 02] Rea K. (2002) Applying TRIZ to Software Problems, Creatively Bridging Academia and Practice in Computing, Triz-Journal, Octobre 2002
- [Rosen 98] Rosen M. , Curtis D. (1998), Integrating CORBA and COM Applications, Wiley Computer Publishing, New York
- [Salamatov 99] Salamatov Y. (1999) *TRIZ : The Right Solution at the Right Time. A Guide to Innovative Problem Solving.*, Insytec B.V.
- [Sarda 90] Sarda S. (1990) Induction de règles de production à partir de bases de données expérimentales ; rapport IFP 38148.
- [Schieve 82] Schieve W., Allen P. (1982) Self-Organization and Dissipative Structures, University of Texas Press, Austin
- [Schlueter 01] Schlueter M. (2001) TRIZ for PERL programming, Triz Journal, [www.triz-journal.com/2001/09/e/index.htm](http://www.triz-journal.com/2001/09/e/index.htm)
- [Schoenauer 95] Marc Schoenauer, (1995) Habilitation à Diriger des Recherches, Equipe Evolution Artificielle et Apprentissage, C.M.A.P - Ecole Polytechnique
- [Schreiber 93] Schreiber G., Wielinga B., Breuker J. (1993) KADS, A principled Approach to Knowledge-Based System Development, Academic Press, London
- [Shrobe 88] Shrobe E. (1988) Exploring Artificial Intelligence, Survey Talks from the National Conferences on Artificial Intelligence, Morgan Kaufmann, San Mateo CA.
- [Siegel 96] Siegel J. (1996) CORBA Fundamentals and Programming, OMG Publications, Wiley & Sons, New York
- [Sportisse 97] Sportisse Myriam (1997) Modélisation des propriétés thermodynamiques des gaz à condensat par représentation de la fraction lourde à l'aide de fonctions de distribution , Thèse de l'Université Aix-Marseille II
- [Stephanopoulos 87] Stephanopoulos, G., J. Johnston, T. Kriticos, R. Lakshmanan, M. Mavrovouniotis and M. Siletti (1987). Design-Kit : An object-oriented environment for process engineering. *Comput. chem. Engng.*, 11, 655-674.
- [Stephanopoulos 90] Stephanopoulos, G., G. Henning and H. Leone (1990). Model-LA. A language for process engineering. Part I and II. *Comput. chem. Engng.*, 14,813-869.
- [Surma 96a] Surma J. (1996) Supporting flowsheet design by cased-based retrieval, rapport interne IFP 43060
- [Surma 96b] Surma J. (1996) REPRO version 1.3 User manual and implementation, rapport interne IFP 43059

- [Sycara et coll. 99] Sycara K., Klusch M., Widoff S., Lu J., (1999) Dynamic Service Matchmaking Among Agents in Open Information Environments, SIGMOD Record (ACM Special Interests Group on Management of Data), Vol. 28, No. 1, March, 1999, pp. 47-53.
- [Travé-Massuyès 02] Groupe de Travail Imalaia, Intégration de Méthodes alliant Automatique et IA, (2002) [www.afia.polytechnique.fr/node.php?lang=fr&node=80](http://www.afia.polytechnique.fr/node.php?lang=fr&node=80)
- [TRIZ 02] TRIZ Journal (2002) <http://www.triz-journal.com/>
- [von Wedel 00] Von Wedel L., (2000) An Object Model for Chemical Process Models, WP 5: Advancing Open Process Engineering Modeling Concepts and Exchange Language, internal Global CAPE-OPEN document, December 2000
- [Weld 90] D. S. Weld and J. de Kleer, editors (1990). Readings in Qualitative Reasoning About Physical Systems. Morgan Kaufmann Publishers, San Mateo, CA, 1990.
- [Westerberg 94] Westerberg, A.W., K. Abbot, B. Allan (1994). Plans for ASCEND IV: Our nest equational-based modelling environment. Presented at AspenWorld'94, Boston, Massachusetts, November, 1994.
- [Willamowski 94] Jutta Willamowski (1994), Modélisation de tâches pour la résolution de problèmes en coopération système-utilisateur, Thèse d' informatique, Université Joseph Fourier, Grenoble (FR), (6 avril) 1994
- [Zitney 02] Zitney S., Syamlal M. (2002) Integrated process simulation and CFD for improved process engineering, Conférence ESCAPE-12, La Haye