



## Projet PLACID - Intégration de -outils d'analyse fonctionnelle

Alain-Jérôme Fougères

### ► To cite this version:

Alain-Jérôme Fougères. Projet PLACID - Intégration de -outils d'analyse fonctionnelle. 2005. <hal-00576510>

**HAL Id: hal-00576510**

**<https://hal.archives-ouvertes.fr/hal-00576510>**

Submitted on 14 Mar 2011

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

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



**M3M**

**Rapport d'avancement de projet**

**RT/PLACID/1.02**

**Septembre 2005**

# **Projet PLACID**

## **Intégration de $\mu$ -outils d'analyse fonctionnelle**

**-**

**Alain-Jérôme FOUGERES**

**M3M - UTBM**

**Rapport d'avancement du projet**

**PLACID :**  
**IHM et plateforme**  
Alain-Jérôme FOUGERES  
M3M - UTBM

cf : RT/PLACID/1.01, Septembre 2003

**Sommaire :**

1. Introduction .....	4
2. Des $\mu$ -outils pour le travail collaboratif.....	5
2.1. Les concepts du travail collaboratif.....	5
2.2. Le concept de $\mu$ -outil.....	7
3. Des agents pour assister la coopération .....	9
3.1. Eléments de modélisation .....	9
3.2. Des agents communicants.....	10
3.3. Des agents coopérants.....	10
4. La plate-forme PLACID .....	11
4.1. Description de la plate-forme PLACID .....	11
4.2. Architecture agent de la plate-forme.....	12
4.3. L'intégration des $\mu$ -outils par agentification.....	13
5. Application : l'intégration de $\mu$ -outils d'analyse fonctionnelle.....	14
5.1. Des $\mu$ -outils pour l'analyse fonctionnelle.....	14
5.2. Description de l'activité et identification des $\mu$ -outils .....	15
5.3. Conception des $\mu$ -outils d'AF.....	15
5.3.1. Spécification des $\mu$ -outils .....	15
5.3.2. Conception des agents .....	17
5.3.3. Spécification et conception des interfaces des $\mu$ -outils .....	18
5.4. Intégration des $\mu$ -outils à PLACID.....	18
6. Conclusion.....	20
7. Bibliographie .....	20

## **Placid : une plate-forme agent pour assister la conception collaborative**

**Alain-Jérôme Fougères**

*Laboratoire M3M - Equipe Conception Innovante et Distribuée  
Université de Technologie de Belfort-Montbéliard  
Rue du Château Sévenans  
90010 BELFORT - France  
alain-jerome.fougeres@utbm.fr*

---

*RÉSUMÉ. Nous relatons dans cet article la conception d'une plate-forme multi-agents adaptée à la conception innovante et distribuée de systèmes mécaniques et supportant des applications d'assistance aux concepteurs dénommés  $\mu$ -outils. Cette plate-forme (PLACID : Plate-forme Logiciel d'Aide à la Conception Innovante et Distribuée) est développée dans le but d'apporter une assistance au travail de co-conception, guidé ou non par des processus complexes de type workflow pour leur capacité à gérer des flots de travaux coopératifs (contrôle et exécution de procédés coopératifs). L'utilisation du paradigme agent concerne aussi bien la modélisation et le développement des différentes couches de la plate-forme que celles des interfaces. A ces objectifs s'ajoutent des contraintes fortes de souplesse et d'adaptabilité, visant à faciliter l'intégration de nouveaux outils de travail collaboratif.*

*ABSTRACT. We report in this article the design of a multi-agent platform adapted to innovating and distributed design mechanical systems. This platform (innovating and distributed design platform, PLACID in French) must support applications of assistance to actors implies in a design process whom we call  $\mu$ -tools. It is developed with an aim of bringing an assistance to the work of Co-design. The use of the paradigm agent as well relates to the modeling and the development of the various layers of the platform that those of the interfaces. With these objectives constraints are added to facilitate the integration of new co-operative tools.*

*MOTS-CLÉS : systèmes multi-agents, agents coopérants, plate-forme collaborative, co-conception distribuée,  $\mu$ -outils, processus de développement.*

*KEYWORDS: multi-agent system, co-operative agents, multi-agent platform, co-design, distributed design, micro-tools, development process.*

---

## 1. Introduction

De façon générale, le travail coopératif et les activités collaboratives intégrées dans des bureaux virtuels nécessitent des outils :

- de communication interpersonnelle ou de groupe (de types synchrones et/ou asynchrones),
- d'organisation et de coordination des groupes et des activités,
- de distribution et de partage d'informations, d'applications, de ressources,
- de définition spatio-temporelle de la coopération : distance spatiale entre les membres d'une équipe (réalité géographique ou virtualité d'une salle), et distance temporelle dans l'interaction (séquentialité et/ou parallélisme de la réalisation des tâches).

Dans cette perspective, nous relatons la conception d'une plate-forme agent adaptée à la conception collaborative (ou co-conception), ainsi que l'intégration à la plate-forme des applications d'assistance aux concepteurs dénommés micro-outils ( $\mu$ -outils) (Van Handenhoven *et al.*, 1999). Le concept de  $\mu$ -outil correspond à des applications logicielles légères, faciles d'utilisation, insérables dans un environnement partagé, connectées entre elles à l'aide d'une base de données. La plate-forme (PLACID : Plate-forme Logiciel d'Aide à la Conception Innovante et Distribuée) a été développée dans le but d'apporter une assistance au travail de co-conception, guidé ou non par des processus complexes de type workflow pour leur capacité à gérer des flots de travaux coopératifs (contrôle et exécution de procédés coopératifs). A ces objectifs s'ajoutent des contraintes fortes de souplesse et d'adaptabilité, visant à faciliter l'intégration de nouveaux outils de travail collaboratif. Le système s'appuie sur une approche orientée agent de la distribution des composants.

Les principales caractéristiques des agents (autonomie, adaptabilité, coopération et communication) permettent, d'une part, de gérer efficacement des composants distribués, hétérogènes et autonomes, et, d'autre part, de faciliter les échanges d'informations et le partage de ressources entre les composants (interaction, communication et coopération). Les agents sont de type *application* ( $\mu$ -outils et autres outils d'aide à la co-conception), *coordinateur/médiateur*, *système* et *interface*. Le système d'agents assure, quant à lui, l'organisation et le contrôle de la communauté d'agents. L'utilisation effective du système (via une interface elle-même conçue selon une approche agent) se fait dans un contexte d'interactions fortes et multiples, de multi-utilisateurs.

Les  $\mu$ -outils supportés par la plate-forme ne sont pas nécessairement intégrés dans un processus prédéfini de conception. Leur utilisation peut être simplement ponctuelle, apportant un service bien ciblé dans une phase de conception. Il participe alors pleinement au processus d'émergence intrinsèque à la conception distribuée (Garro, 1996). Quoiqu'il en soit, chaque  $\mu$ -outil est connecté au système multi-agent par l'intermédiaire d'un agent hôte. Celui-ci sert d'interface

Placid : une plate-forme agent pour assister la conception collaborative

de communication (entrées/sorties) entre les  $\mu$ -outils et le système d'information coopératif.

Cet article sera structuré comme suit : dans la section 2 nous présenterons les différents concepts impliqués dans le travail collaboratif assisté par ordinateur (TCAO), ainsi que la notion de C et leur processus de développement. La section suivante sera consacrée à la description de la plate-forme PLACID : son architecture agent, la distribution et la coopération assurée par la plate-forme et l'intégration de  $\mu$ -outils coopératifs. Dans la section 4 nous illustrerons l'utilisation coopérative de la plate-forme avec des  $\mu$ -outils d'analyse fonctionnelle. Finalement, en section 5, nous évoquons les prolongations de notre travail, d'une part sur la conception de systèmes de médiation adaptés à des systèmes coopératifs, et, d'autre part, sur l'extension du concept de  $\mu$ -outils à d'autres types d'activités coopératives.

## **2. Des $\mu$ -outils pour le travail collaboratif**

### **2.1. Les concepts du travail collaboratif**

Le développement des technologies informatiques, la démocratisation de l'Internet, l'utilisation des nouvelles ressources sur Internet ont donné naissance à de nouvelles méthodes de travail. Nous faisons bien entendu, allusion au travail coopératif assisté par ordinateur (CSCW) (Courbon *et al.*, 1999).

Par définition le collecticiel est un logiciel qui assiste un groupe d'utilisateurs à la réalisation d'un projet commun. Les membres du groupe collaborent à distance, soit au même moment (activité synchrone), soit à des moments différents (activité asynchrone). Les champs d'application sont très nombreux. Ainsi des activités aussi diverses que la conception de produits industriels, l'enseignement, les relations commerciales ou les jeux sont susceptibles d'être assistées par l'ordinateur. Les systèmes informatiques destinés au support du travail coopératif doivent permettre aux utilisateurs de réaliser une tâche collaborative à partir de postes de travail respectifs ou depuis des installations spécialisées (salles de vidéoconférence, salles virtuelles de co-conception, ...). Les collecticiels doivent permettre à plusieurs utilisateurs de collaborer dans des espaces partagés explicites. Les concepts à considérer sont ainsi :

— Le temps et l'espace : on peut dans un collecticiel vouloir réunir plusieurs personnes distantes géographiquement (bureau à proximité ou distant) ou ne travaillant pas en même temps (rythmes différents, incompatibilité des emplois du temps,...).

— Un ensemble de modes de coopération : la *coopération asynchrone* lorsque les différents acteurs interagissent dans un projet en échangeant des données et en travaillant quand ils peuvent (mode de travail autonome) ; la *coopération en session* lorsque les différents participants travaillent en même

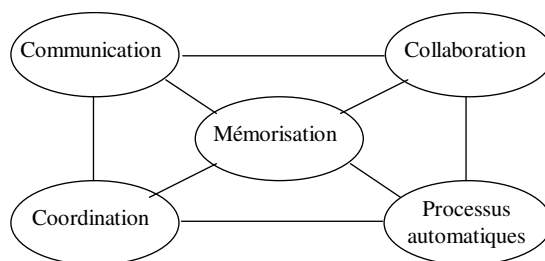
temps sur des données qui leur sont propres, tout en restant accessibles pour communiquer (l'objectif étant de faire diminuer les délais d'interaction entre les différents acteurs d'un projet) ; la *coopération en réunion* lorsque tous les utilisateurs, clairement identifiés, travaillent et communiquent en co-temporalité tout en partageant les objets de leurs travaux et discussions (des rôles leur sont attribués, et chacun participe à son tour) ; la *coopération étroite* lorsque les acteurs communiquent, interagissent en temps réel sur tous les objets partagés du projet (accroissement de la coproduction).

— La flexibilité opérant dans des domaines hétérogènes : dans l'interaction, dans la distribution de données, dans le partage, dans le contrôle d'accès, dans la représentation de l'information, dans la planification pour l'exécution des tâches.

Les activités liées au travail collaboratif sont essentiellement des échanges, du partage et de la coopération entre participants. Il est alors d'usage de présenter les systèmes d'information coopératifs comme une réponse aux enjeux de la coopération (Hoogstoel, 1995) - ci-dessous 6 enjeux, les 3 premiers pour des coopérations ponctuelles, les 3 autres pour la structuration d'un groupe :

- Faciliter le partage de ressources.
- Assister la coordination.
- Améliorer la communication de groupe.
- Favoriser l'implication individuelle.
- Entretenir la cohésion des équipes.
- Favoriser le développement de l'organisation

La figure suivante (Fig. 1) schématise les liens unissant les fonctionnalités de base d'un collecticiel : la collaboration (visioconférence, vidéoconférence, outils de localisation, d'annotations, de réunion électronique, de décision de groupe, édition conjointe), la mémoire de groupe (base d'informations partagées dans un groupe, bases de documents), la circulation de documents et le workflow. Elle reprend, puis complète le modèle 3C (communication, coopération, coordination) (Ellis, 1994), définissant les espaces nécessaires aux artefacts de collaboration.



**Figure 1.** *Fonctions basiques pour un collecticiel*

## Placid : une plate-forme agent pour assister la conception collaborative

La nécessité de standards qui permettraient l'interopérabilité d'applications de workflow a depuis un certain temps été reconnue. Dans le contexte de l'internet, des groupes de travail s'efforcent de proposer de tels standards. C'est autour de la WfMC (*Workflow Management Coalition*) que s'élaborent de telles spécifications empruntant aux approches objets (WfMC, 1999).

### 2.2. Le concept de $\mu$ -outil

Le concept de  $\mu$ -outil (Van Handenhoven *et al.*, 1999) s'oppose à la tendance actuelle des outils de conception, souvent lourds, prescriptifs, et parfois peu employés. Ces outils sont idéalement (Fig. 2) :

- faciles à apprendre (quelques minutes) et faciles à utiliser,
- peu complexes (même s'ils peuvent être développés sur une théorie élaborée),
- rapidement implémentables et modifiables (évolution), y compris par les concepteurs eux-mêmes,
- autonomes, mais aussi réactifs lorsqu'ils sont définis pour des processus coopératifs ; nous parlerons alors de *MOC* (Micro-Outils Coopératifs). Ces derniers sont distribués entre des acteurs qui interviennent selon leurs domaines de compétences, répondant ainsi aux besoins de l'ingénierie concurrente.

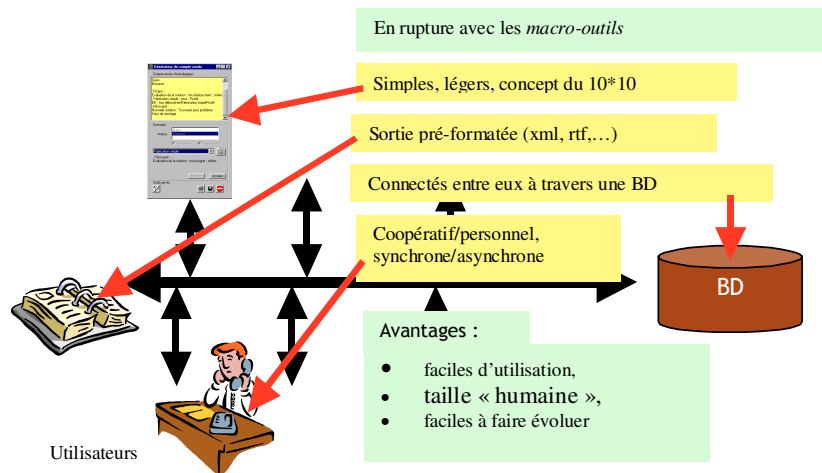


Figure 2. Le concept de  $\mu$ -outil

Nous venons de définir le concept de  $\mu$ -outil comme pouvant apporter une aide à la réalisation d'une tâche élémentaire et ponctuelle entrant dans une activité bien définie. Le produit de la tâche pouvant être un objet intermédiaire de



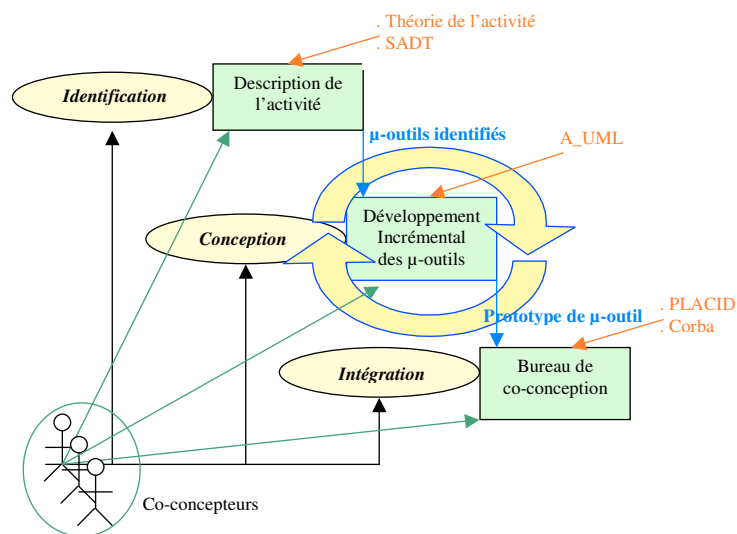
conception (de Terssac *et al.*, 1996). Le développement d'un  $\mu$ -outil s'inscrit donc spontanément dans une démarche orientée activité. Par ailleurs, les  $\mu$ -outils, comme tout type de système, peuvent être décrits selon le processus : *identification – classification – qualification – quantification*. Nous énonçons ci-dessous les différents principes conduisant à leur structuration informatique :

— l'usage du micro-outil est individuel ou collectif (lorsque l'activité est collaborative, notamment). Les tâches peuvent être structurées en plan d'action. Il est alors nécessaire de bien spécifier les conditions d'usage du micro-outil, en décrivant notamment le cycle de vie des objets traités et leurs conditions de partage ;

— l'interaction entre l'acteur et le micro-outil porte principalement sur l'acquisition de données (objets de l'activité), leur mise en relation, avec des moyens graphiques par exemple, puis leurs accès et leur gestion ;

— l'identification et la description du micro-outil étant le fruit d'un travail collectif et pluridisciplinaire, la réalisation de maquettes est recommandée, comme facilitateur d'échange d'idées ;

— le micro-outil est développé en suivant le respect des principes d'une démarche de qualité logicielle.



**Figure 3.** Le processus ICI de développement de  $\mu$ -outils de co-conception

Le processus de développement du  $\mu$ -outil (Fig. 3) débute par l'analyse de l'activité et aboutit sur les produits logiciels correspondants, ainsi que sur sept documents (livrables) constituant la mémoire de leurs conceptions. Trois grandes

Placid : une plate-forme agent pour assister la conception collaborative

phases structurent ce processus nommé *ICI* (Identification, Conception, Intégration) :

— la phase d'*identification* des  $\mu$ -outils, renvoyant aux niveaux supérieurs de l'ingénierie système et nécessitant une collaboration étendue de tous les acteurs, consiste à identifier, parmi les tâches qui composent une activité spécifique celles, qui pourront être instrumentées, puis à les spécifier ;

— la phase de *conception* vise, selon une approche incrémentale, propice à l'établissement d'un dialogue permanent entre tous les acteurs du processus de développement, à élaborer l'architecture du  $\mu$ -outil et de ces composants, à développer et à tester ces derniers (UML/Java), puis à valider l'interface utilisateur (Fougères, 2004) ;

— et enfin la phase d'*intégration* à la plate-forme agent PLACID reliée à un ORB chargé de la gestion des échanges et du partage des informations (*cf.* §4).

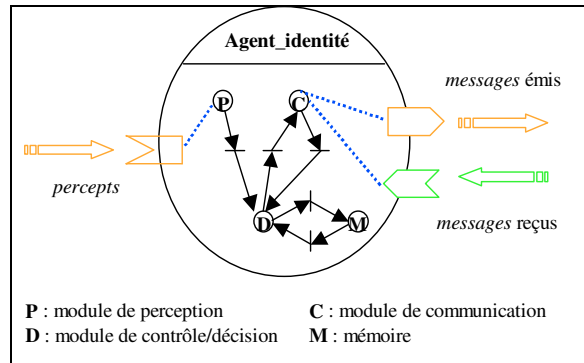
### 3. Des agents pour assister la coopération

Le principal intérêt des SMA est qu'ils permettent de distribuer des agents, entités communicantes, autonomes, réactives et dotées de compétences (Shoham, 1993 ; Ferber, 1995). Ces propriétés correspondent à celles attendues aussi bien pour une plate-forme d'aide à la conception collaborative, que pour les applications qu'elle est amenée à supporter ; d'autant plus quand il s'agit de MOC (*cf.* §2.2).

#### 3.1. *Éléments de modélisation*

La définition de nos agents est adaptée du modèle à 3 niveaux de l'opérateur de Rasmussen (comportement réflexe, comportement à base de règles, comportement à base de connaissances avec interprétation, décision et plan). Nous l'avons interprété comme modèle de processus de nos agents dont les comportements sont adaptés aux tâches qu'ils réalisent. Dans (Fougères, 2003) nous avons proposé l'architecture générale de tels agents, respectant les trois propriétés d'indépendance, de communication et d'intelligence. Celle-ci (Fig. 4), inspirée de la théorie de modularité de J. Fodor est composée de cinq modules gérant les connaissances, la perception, la communication, le contrôle et le raisonnement de l'agent, dont nous avons proposé un modèle réseau de Petri.

Nous ajouterons à cette description que les agents sont des entités hétérogènes aux modes d'interactions variés et aux comportements complexes. Une modélisation de SMA doit aussi définir le type d'organisation des agents et sa capacité d'évolution.



**Figure 4.** Architecture modulaire d'un agent générique

### 3.2. Des agents communicants

La communication est le principal mécanisme d'interactions d'un agent avec la communauté des agents – en conformité avec la relation triangulaire (*Communication, Coopération, Compétition*). Nous insistons sur le fait que pour qualifier un agent d'"intelligent" il est essentiel de prouver ses capacités à communiquer dans un but individuel ou collectif.

Pour communiquer entre eux (information ou dialogue pour la coopération), les agents expriment leurs intentions selon le langage KQML, dérivé de la théorie des actes de langage. La forme générale d'un acte de langage est donnée par J. Searle sous l'expression  $F(p)$ , avec  $F = \{Affirmer, Demander, Promettre, Exprimer, Déclarer\}$  et  $p$  une proposition. Le format que nous avons retenu est défini par le quintuplet  $\langle intention, émetteur, récepteur, langage, message \rangle$ . Il permet de représenter le contexte, l'intention et le message de la communication.

### 3.3. Des agents coopérants

Les systèmes de travail collaboratif sont constitués de composants distribués, hétérogènes et autonomes. Les systèmes développés en intelligence artificielle distribuée (IAD), et notamment les SMA sont donc bien adaptés. L'apport potentiel des agents concerne :

- la prise en charge d'actions répétitives et la délégation de tâches sans intérêt pour l'utilisateur,
- la prise de décision par compréhension du contexte d'usage (pertinence),
- la personnalisation de l'information (préférences, buts et capacités des utilisateurs),
- l'interactivité plus naturelle (modalités, forme et présentation),

Placid : une plate-forme agent pour assister la conception collaborative

— et l'adéquation aux systèmes en réseau, notamment coopératifs.

Les comportements individuels et coopératifs des agents sont variés : initialisations, planification des actions, émission et réception de documents et de messages, recherche de documents ou d'information, supervision de procédures. Chacun de ces services correspond à la mise en œuvre de compétences.

#### **4. La plate-forme PLACID**

##### **4.1. Description de la plate-forme PLACID**

Le projet PLACID consiste en la définition et le développement d'une plate-forme logicielle apportant un certain nombre de services pour l'utilisation d'un environnement de co-conception virtuelle (partage d'objets, services de gestion de tâches, services de communications et peut-être outils d'aide à la décision). L'architecture de PLACID (Fig. 5) est composée de 4 couches, offrant par sa modularité des possibilités d'utilisation multi-plate-formes et d'évolutivité :

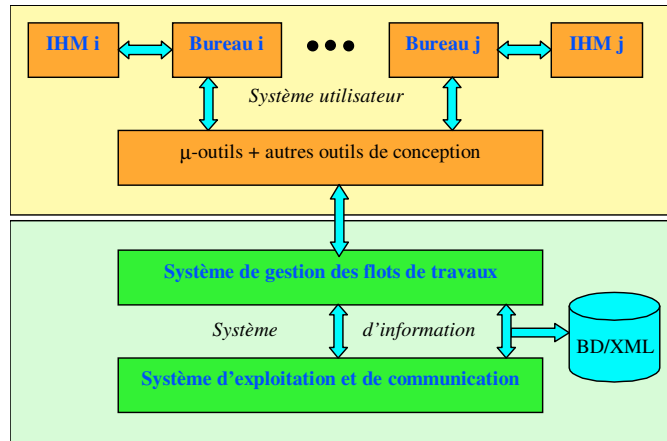
— Couche 1 : c'est la couche de présentation du système de conception dans un contexte d'interactions multiples, multi-utilisateurs et multi-modal.

— Couche 2 : couche de gestion des outils d'aide à la co-conception ( $\mu$ -outils et autres outils).

— Couche 3 : couche de gestion des travaux collaboratifs (de type workflow), permettant de contrôler et exécuter les procédés coopératifs.

— Couche 4 : couche du système d'exploitation et de gestion des communications de bas niveau.

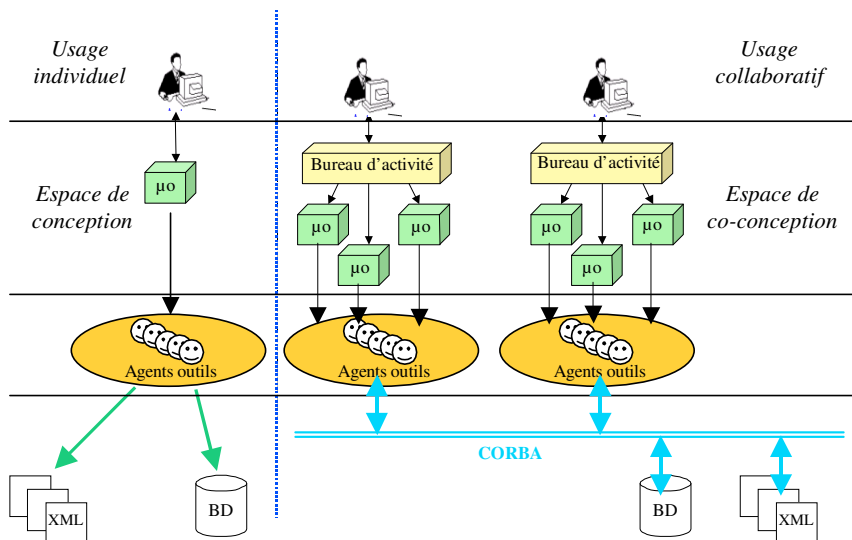
Le développement de PLACID s'intègre dans les 2 axes (points de vue sur le produit et coopération en conception) de notre équipe afin de faciliter l'utilisation de  $\mu$ -outils de conception par une équipe de concepteurs proches ou distants, dans le cadre de la conception distribuée structurée en modules (fonctionnel, structurel, fabrication, maintenance). Les différentes déclinaisons de PLACID permettent aussi bien l'usage ponctuel d'un outil de conception que la constitution de véritables "bureaux de co-conception".



**Figure 5.** Architecture de principe de la plate-forme PLACID

#### 4.2. Architecture agent de la plate-forme

La figure 6 présente l'architecture agent de la première couche de la plate-forme reliée à un ORB chargé de la gestion des échanges et du partage des informations. Cette couche d'agents se décompose en deux niveaux : des agents *acteurs*, proches des utilisateurs et des  $\mu$ -outils, et des agents *outils* (exécutants) dotés des compétences indispensables à la coopération.



**Figure 6.** Architecture agent du système d'information coopératif de PLACID

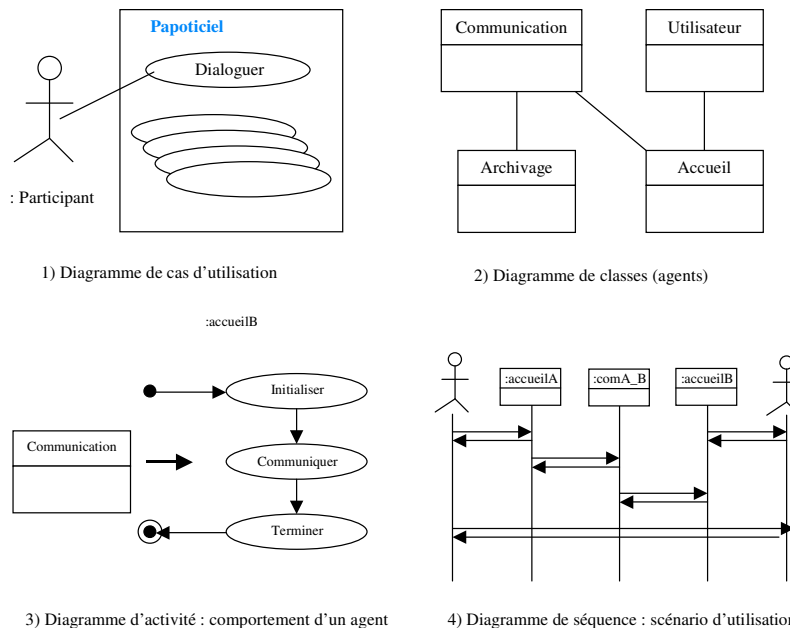
Placid : une plate-forme agent pour assister la conception collaborative

### 4.3. L'intégration des $\mu$ -outils par agentification

Les agents sont des entités possédant un certain nombre de compétences qui leur permettent de jouer un ou plusieurs rôles dans une organisation. Ils sont regroupés au sein d'un SMA organisé selon une structure hiérarchique (agents *spécialistes*, agents *médiateurs* et agents *superviseurs*).

Pour la conception du SMA nous reprenons les propositions faites dans la définition du langage A\_UML (Odell *et al*, 2000), ainsi que nos propres propositions méthodologiques (Fougères, 2003) (schématisées en Fig. 7) :

- 1) réaliser le diagramme de cas d'utilisation (services rendus par le système), et pour chacun des usages identifiés réalisés les 3 phases suivantes ;
- 2) réaliser le diagramme de classes mettant en relation les agents concernés par l'usage (on peut aussi faire usage du diagramme de collaboration) ;
- 3) définir le comportement de chaque agent au moyen d'un diagramme d'états ou d'activités ;
- 4) sur la base de scénarios d'usage, réaliser les diagrammes de séquence qui précisent les échanges de messages (et leur ordonnancement) entre les agents concernés par les scénarios.



**Figure 7.** Méthodologie de conception de SMA, illustrée pour une application typique de réunion électronique : «Papoticiel».

## 5. Application : l'intégration de $\mu$ -outils d'analyse fonctionnelle

Pour tester le premier prototype de PLACID nous avons modélisé puis conçu une application de réunion électronique, que nous avons nommé « Papoticiel » (Fougères, 2004). En effet cette application, outre son aspect communicationnel, met en relation des applications de coopération aussi variées que la gestion d'un groupe d'utilisateurs, la maintenance d'un agenda, la gestion d'une mémoire de groupe au travers de l'archivage des réunions et de ces éléments, ainsi que l'édition partagée de compte-rendu de réunion. Ce *Papoticiel* nécessite le déploiement de 8 types d'agents : les *agents micro-outil*, des *agents utilisateur*, des *agents d'accueil (ou bureau)*, un *agent groupe*, un *agent communication*, un *agent d'archivage* et un *agent BD*. Depuis nous avons lancé trois chantiers de développement de micro-outils pour la co-conception : un premier pour l'aide au déploiement de la méthodologie TRIZ (*Teoria Reschenia Izobretateliskih Zadaci*, théorie de résolution des problèmes innovants) (Weité *et al.*, 2005), un second pour l'évaluation des performances en ingénierie de systèmes manufacturiers (Deniaud *et al.*, 2005) et un troisième que nous détaillons dans la section suivante pour illustrer notre démarche d'intégration de micro-outils dans PLACID, et qui concerne l'Analyse Fonctionnelle.

### 5.1. Des $\mu$ -outils pour l'analyse fonctionnelle

L'analyse fonctionnelle (AF) est une méthode systématique et structurée de conception. Elle permet de décrire un produit sous forme fonctionnelle afin de prendre en compte les besoins de l'utilisateur. L'origine du projet de développement de  $\mu$ -outils d'AF provient :

— de constats : une méthode reconnue pour rationaliser la conception, mais aussi une méthode d'appropriation délicate, insuffisamment employée

— et d'un besoin d'outil bien identifié pour supporter la démarche, pour guider l'utilisateur, pour former et aider à l'appropriation

La démarche proposée de l'AF s'appuie sur une approche méthodique basée sur l'utilisation de 6  $\mu$ -outils :

- le  $\mu$ -outil *Millex* pour définir les milieux extérieurs (limites du système),
- le  $\mu$ -outil *Devo* pour définir l'évolution du système (cycle de vie),
- le  $\mu$ -outil *Beso* pour définir les besoins,
- le  $\mu$ -outil *Isys* pour établir la liste des fonctions (inventaire systématique),
- le  $\mu$ -outil *Caraf* pour caractériser les fonctions,
- le  $\mu$ -outil *Hiera* pour hiérarchiser les fonctions.

L'activité cible d'AF étant décrite, nous allons détailler le développement des 6  $\mu$ -outils précédents, conformément au processus ICI (Fig. 3).

Placid : une plate-forme agent pour assister la conception collaborative

## 5.2. Description de l'activité et identification des $\mu$ -outils

Cette phase doit permettre d'identifier les  $\mu$ -outils susceptibles d'apporter une assistance à l'analyse fonctionnelle. Ce processus d'identification conduit à l'élaboration du diagramme SADT, point de départ de la conception des  $\mu$ -outils.

- analyse de l'activité, puis construction de graphes d'activité de référence (*Milex*→*Beso*→*Devo* ou *Isys*→*Carf*→*Hiera*)
- identification des MO à l'intérieur des graphes d'activité,
- déduction de l'actigramme SADT, à partir des graphes d'activité.

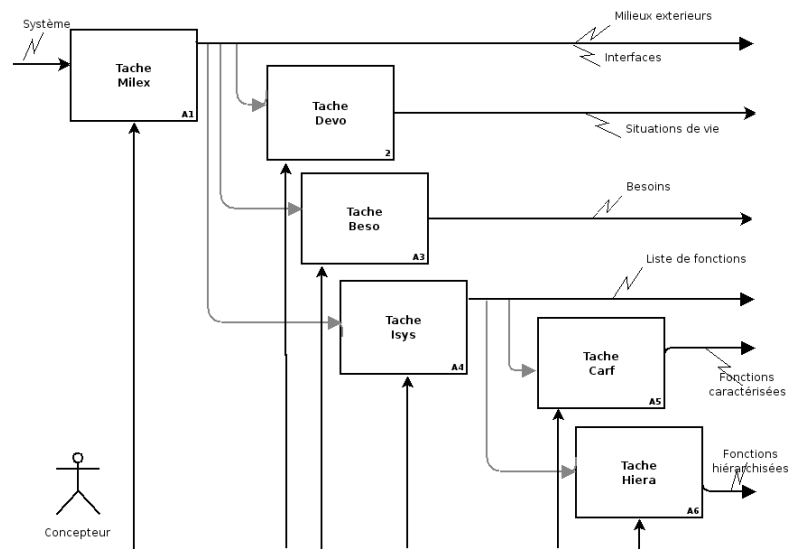


Figure 8. Diagramme SADT correspondant à une activité d'AF de référence.

## 5.3. Conception des $\mu$ -outils d'AF

Cette deuxième phase de la méthodologie ICI consiste à spécifier, concevoir (diagrammes UML), puis à développer en java les  $\mu$ -outils et leurs interfaces, en incluant le processus d'agentification présenté plus haut (cf. §4.3)

### 5.3.1. Spécification des $\mu$ -outils

Le diagramme de cas d'utilisation ci-dessous (Fig. 9) présente le contexte d'utilisation des  $\mu$ -outils d'AF. Une activité d'analyse fonctionnelle peut être déclenchée à l'initiative d'un membre du groupe, qualifié d'*initiateur*, ou par un agent logiciel *agenda*, si une séance a été préalablement programmée. Ce



diagramme est facilement déduit de l'actigramme SADT (Fig. 8) construit pour modéliser l'activité de référence d'analyse fonctionnelle.

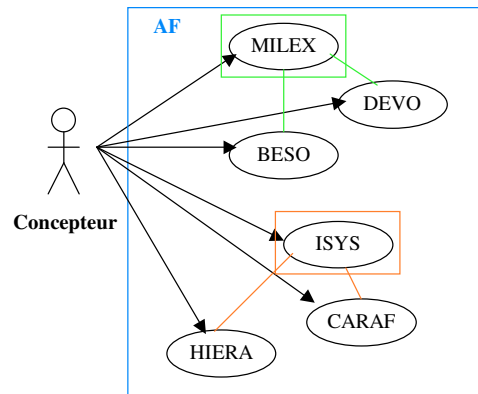


Figure 9. Diagramme de cas d'utilisation des 6  $\mu$ -outils d'AF

La construction du diagramme de classe de chaque  $\mu$ -outil est standardisée : une classe pour le  $\mu$ -outil lui-même, une classe pour son IHM et un ensemble de classes pour gérer les objets de conceptions traités par le  $\mu$ -outil (*ME* et *Interface* pour le  $\mu$ -outils *Milex*, par exemple). Un septième  $\mu$ -outil permet de créer des processus tels que ceux représentés sur la figure 9 (*Milex*→*Beso*→*Devo* ou *Isys*→*Caraf*→*Hiera*). L'utilisation des  $\mu$ -outils pourrait alors être prescrite par une méthodologie notamment, ce que représente le diagramme de classe complet de la figure suivante (Fig. 10).

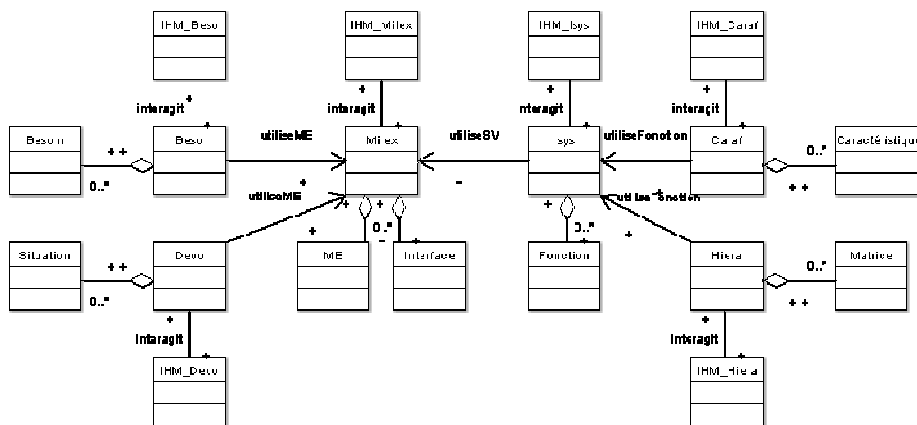
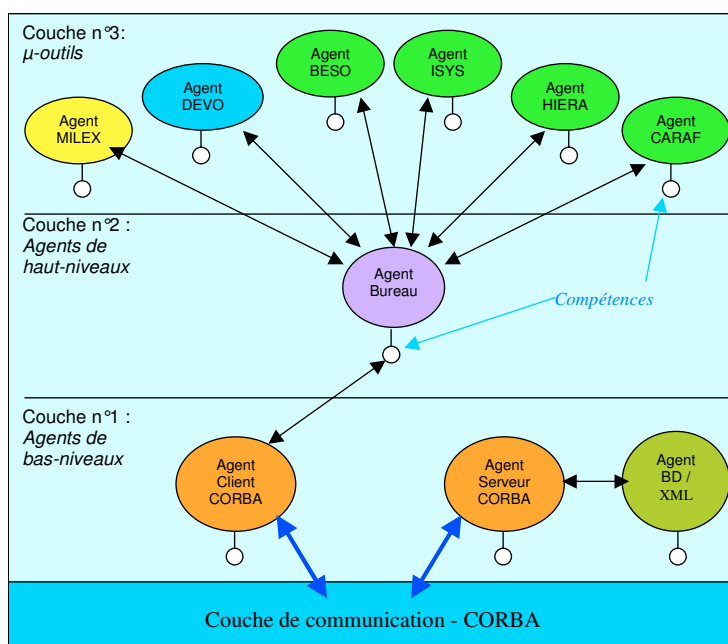


Figure 10. Diagramme de classe pour les  $\mu$ -outils d'AF

Placid : une plate-forme agent pour assister la conception collaborative

### 5.3.2. Conception des agents

L'atelier AF, en plus des agents déjà disponibles sur la plate-forme PLACID, nécessite le déploiement de 3 autres types d'agents (Fig. 11) : les agents micro-outil (*Milex, Devo, Beso, Isys, Hiera, Caraf*), un agent *bureau* pour assister la session de travail du concepteur, un agent *clientCorba* et un agent *serveurCorba* pour gérer la coopération ainsi que le contexte multi-utilisateur. Le tableau 1 recense les différentes compétences mises en œuvre par chaque agent.



**Figure 11.** Structuration des agents d'AF sous la plate-forme PLACID

Agents	Compétences
<i>Milex, Beso, Devo, Isys, Hiera, Caraf</i>	<ul style="list-style-type: none"> <li>Gestion de <math>\mu</math>-outils, à partir des interactions entre utilisateurs et IHM associées.</li> </ul>
<i>Bureau</i>	<ul style="list-style-type: none"> <li>Communication avec l'agent <i>ClientCorba</i>.</li> <li>Coordination des agents des <math>\mu</math>-outils.</li> <li>Transmission des données de projets.</li> <li>Communication à l'utilisateur des informations de coopération.</li> </ul>
	<ul style="list-style-type: none"> <li>Gestion des communications (messages et événements) côté client.</li> </ul>

<i>Client CORBA</i>	<ul style="list-style-type: none"> <li>• Communication avec l'agent <i>ServeurCorba</i> (requête de l'utilisateur).</li> <li>• Transmission des informations coopératives au bureau de l'utilisateur.</li> </ul>
<i>Serveur CORBA</i>	<ul style="list-style-type: none"> <li>• Gestion des communications (messages et événements) côté serveur.</li> <li>• Communication avec l'agent <i>ClientCorba</i> (réponse à l'utilisateur).</li> <li>• Gestion des accès concurrents aux données de projets.</li> <li>• Gestion des sessions des utilisateurs.</li> <li>• Envoi de messages d'informations coopératives.</li> </ul>
<i>BD et/ou XML</i>	<ul style="list-style-type: none"> <li>• Gestion de fichiers XML (création, ouverture, enregistrement, fermeture, destruction).</li> <li>• Gestion des échanges (requêtes/réponses) avec une BD.</li> <li>• Transfert de données.</li> </ul>

**Tableau 1.** *Le tableau des compétences des agents d'AF*

### 5.3.3. Spécification et conception des interfaces des $\mu$ -outils

Pour le cycle de développement des interfaces des  $\mu$ -outils nous avons retenu la démarche incrémentale : développement successif avec amélioration de prototypes. Cette démarche permet d'évaluer lors de chaque phase terminale d'un incrément la pertinence de l'interface et le respect des objectifs qui ont conduit l'incrément. Pour une discussion sur une démarche alternative de structuration de l'IHM, nous conseillons au lecteur de se référer à (David, 2001). L'analyse des tâches utilisateurs rentrant dans la spécification finale de chaque  $\mu$ -outils nous adoptons la méthode SADT/Petri (Abed *et al.*, 2001). En effet, cette association a été conçue pour réaliser l'analyse, la spécification et l'évaluation des systèmes interactifs comme les IHM. L'intérêt majeur de cette association est d'effectuer la structuration fonctionnelle du système en termes de tâches par SADT, et de décrire son comportement dynamique par les réseaux de Petri. Une fois l'IHM spécifiée il s'agit de concevoir son architecture logicielle. Nous adoptons pour celle-ci le modèle PAC (Coutaz *et al.*, 2001), particulièrement adapté à la modélisation des architectures logicielles des systèmes interactifs et/ou multimodaux et des collecticiels. En guise d'illustration, la figure suivante (Fig. 12) présente une vue d'écran de l'atelier AF.

### 5.4. Intégration des $\mu$ -outils à PLACID

L'intégration à la plate-forme PLACID du set de  $\mu$ -outils d'AF est réalisée en deux phases : la première correspond à l'agentification des 6  $\mu$ -outils conçus pour l'AF, la seconde consiste à définir puis inclure un agent *ClientCorba* à PLACID qui sera référencé par chaque bureau d'activité d'utilisateurs (Fig. 13).

Placid : une plate-forme agent pour assister la conception collaborative

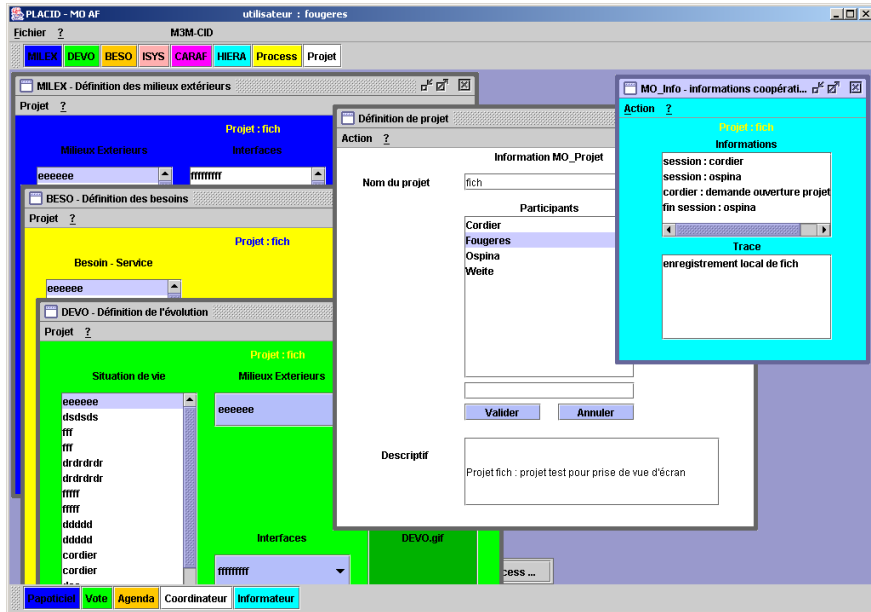


Figure 12. Vue d'écran du bureau d'Analyse Fonctionnelle

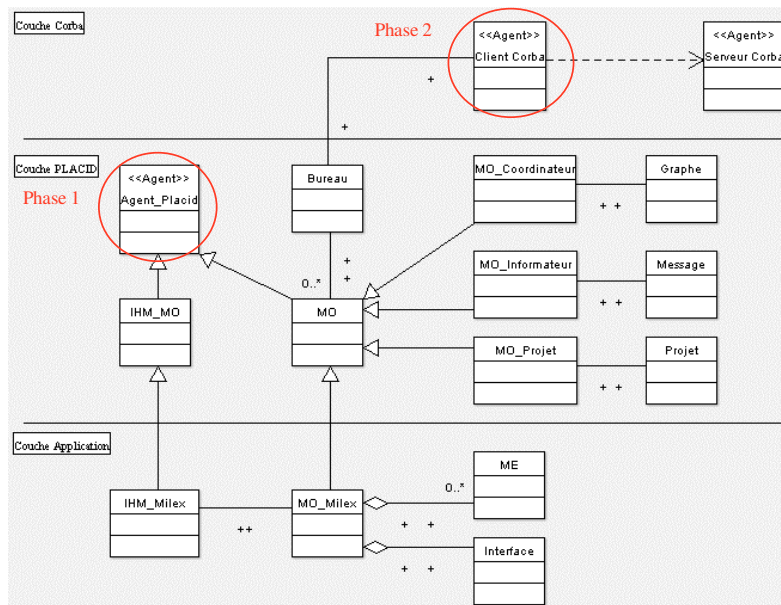


Figure 13. Intégration du  $\mu$ -outil «Milex» à la plate-forme PLACID

## 6. Conclusion

Du bilan d'une étude menée sur les perspectives des techniques de travail collaboratif pour la co-conception, utilisant le paradigme agents, est ressorti un ensemble de concepts prometteurs qui ont servi à la définition des objectifs de PLACID :

— définition d'une plate-forme agent de co-conception distribué offrant des services de partage de ressources, coordination de groupe, réunion électronique, décision de groupe, compte-rendus et mémoire de groupe (ou de conception).

— distribution d'activités coopératives et de composants d'aide à la conception ; aide à la décision dans le processus de conception distribuée et prise en charge d'actions (tâches répétitives ou implicites dans un cadre de coopération), conception des IHM.

A la suite de cette étude et des premiers éléments de modélisation de PLACID, nous avons réalisé un prototype de la plate-forme, validé par l'utilisation et la coordination d'un *Papoticiel* (réunion électronique artificielle) et d'un outil d'assistance à la « gestion de groupes ». Depuis, 4 chantiers de développement de  $\mu$ -outils ont été lancés, dont celui d'*Analyse Fonctionnelle* présenté dans la section 5, respectant le processus ICI que nous avons défini.

Nos travaux actuels portent sur l'identification et la définition de MOC (micro-outils coopératifs) génériques et sur la généralisation de l'aide aux activités de co-conception à tout type d'activités coopératives, notamment dans le domaine du génie logiciel (Fougères *et al*, 2004).

## 7. Bibliographie

- Abed M., Ezzedine H, Kolski C., «Modélisation des tâches dans la conception et l'évaluation des systèmes interactifs : la méthode SADT/Petri», in *Analyse et conception de l'IHM*, sous la direction de C. Kolski, Hermès, 2001.
- Courbon J.-C., Tajan S., *Groupware et intranet. Vers le partage des connaissances*, 2<sup>ème</sup> édition, DUNOD, 1999.
- Coutaz J., Nigay L., «Architecture logicielle conceptuelle des systèmes interactifs», in *Analyse et conception de l'IHM*, sous la direction de C. Kolski, Hermès, 2001.
- David B., « IHM pour les collecticiels », *Réseaux et systèmes répartis*, Hermès, nov. 2001.
- Deniaud I., Micaëlli J.-P. et Fougères A.-J., «Déployer et définir de façon collaborative une performance en ingénierie système manufacturière : l'exemple de la réactivité», *Actes du 6<sup>ème</sup> Congrès international de Génie Industriel*, Besançon, 7-10 juin 2005.
- Ellis C.A., Gibbs S.J., and Rein G.L., «Groupware : some issues and experiences», *Communications of ACM* 34(1), p. 38-58, 1991.
- Ellis C.A., Wainer J., «A conceptual model of Groupware », In *Proc. CSCW'94*, ACM Press, pp. 79-88, 1994.

Placid : une plate-forme agent pour assister la conception collaborative

- Ferber J., *Les systèmes multi-agents. Vers une intelligence collective*, InterEditions, 1995.
- Fougères A.-J., « Des agents communicants pour simuler et détecter des épidémies », *Revue Ingénierie des Systèmes d'Information*, (8)1, Hermès, 2003.
- Fougères A.-J., « Agents to cooperate in distributed design », *IEEE International Conference on Systems, Man and Cybernetic (SMC'04)*, The Hague, Netherlands, October 10-13 2004.
- Fougères A.-J., « Agent-based micro-tools development for a co-operative design platform », *ITI 3rd International Conference on Information and Communication Technology*, (ICICT'05), pp. 715-730, Cairo, Egypt, December 5-6 2005.
- Fougères A.-J., et Ospina V., « Un système de médiation pour la gestion et le suivi de projets d'étudiants », *Colloque International TICE'04*, Compiègne, octobre 2004.
- Garro O., « Conception distribuée – de l'industrie... à l'industrie », *Actes des 4<sup>e</sup> Journées Francophones d'Intelligence Artificielle et Systèmes Multi-Agents (JFIADMSA'96)*, Hermès, Port Camargue, 1-3 avril 1996.
- Grislin-Le Strugeon E., Adam E., Kolski C., « Agents intelligents en interaction homme-machine dans les systèmes d'information », in *Environnements évolués et évaluation de l'IHM*, sous la direction de Christophe Kolski, Hermès, 2001.
- Hérin D., Espinasse B., Andonoff E. et Hanachi C., « Des systèmes d'information coopératifs aux agents informationnels », in *Ingénierie des systèmes d'information*, sous la direction de Corine Cauvet et Camille Rosenthal-Sabroux, Hermès, 2001.
- Hoogstoel F., Une approche organisationnelle du travail coopératif assisté par ordinateur. Application au projet CO-LEARN, Thèse de Doctorat de l'Université des Sciences et Techniques de Lille, 1995.
- Odell J., Parunak H.V.D., Bauer B., « Extending UML for agents, Proceedings of the Agent-Oriented Information Systems », *Workshop at the 17<sup>th</sup> National conference on Artificial Intelligence*, Austin, Texas, july, 30, 2000.
- Ospina V., Fougères A.-J., Zacklad M., « Modélisation de connaissances pour un système de médiation », Actes des cinquièmes journées Extraction et Gestion des Connaissances, Paris, France, 18-21 janvier 2005, 2 Volumes. *Revue des Nouvelles Technologies de l'Information (RNTI-E-3)*, Cépaduès-Éditions, p. 575-586, 2005.
- Routier J.-C. et Mathieu P., « Une contribution du multi-agent aux applications de travail coopératif », *Technique et science informatiques*, 14(4), 473-500, 1995.
- Schmidt K. and Bannon L., « Taking CSCW seriously », *Computer Supported Cooperative Work Journal*, 1(1), 1992.
- Shoham Y., « Agent-Oriented Programming », *Artificial Intelligence*, vol. 60, p. 51-92, 1993.
- Terssac G. de, Friedberg E., *Coopération et conception*, sous la direction de Gilbert de Terssac et Erhard Friedberg, Collection Travail, Editions Octares, Toulouse, 1996.
- Van Handenhoven E., Trassaert P., « Design knowledge and design skills », *International Conference on Engineering Design (ICED 99)*, Munich, Allemagne, 24-26 août, 1999.

RT/PLACID/1.02

Wagner G., Towards Agent-Oriented Information Systems, Technical Report, University of Leipzig, Germany, March 1999.

Weité P.-A., Fougères A.-J., Gazo C., « Les micro-outils, vecteur d'appropriation des nouvelles méthodologies de conception et d'innovation », *Actes du 6<sup>ème</sup> Congrès international de Génie Industriel*, Besançon, 7-10 juin 2005.

WfMC, Workflow Management Coalition, Terminology & Glossary, WFMC-TC-1011, Issue 3.0., 1999.