

Collaboration / Coopération

Khalid Benali¹, Grégory Bourguin², Bertrand David³, Alain Derycke⁴, Christine Ferraris⁵

¹ LORIA, Campus Scientifique, Nancy
benali@loria.fr

² Laboratoire d'Informatique du Littoral
BP 719, 62228 Calais Cedex
bourguin@lil.univ-littoral.fr

³ Laboratoire ICTT, Ecole Centrale de Lyon, 69134 Ecully
Bertrand.David@ec-lyon.fr

⁴ Laboratoire TRIGONE, Université des Sciences et Technologies
de Lille, 59665 Villeneuve d'Ascq cedex,
alain.derycke@univ-lille1.fr

⁵ Equipe SysCom, Université de Savoie,
Campus Scientifique, 73376 Le Bourget du Lac cedex,
Christine.Ferraris@univ-savoie.fr

Résumé. Le travail coopératif assisté par ordinateur (TCAO) se développe de façon considérable ces derniers temps. Sous cet acronyme, sont rassemblées des activités très différentes dont nous proposons une classification. Nous montrons également des aspects importants tant technologiques qu'organisationnels et humains s'y attachant. Nous terminons par les orientations de recherche qui émergent actuellement.

1 INTRODUCTION

L'objectif du TCAO (Travail Coopératif Assisté par Ordinateur), surtout connu sous le sigle anglo-saxon de CSCW (Computer-Supported Collaborative Work), est de permettre à un collectif d'acteurs de travailler ensemble via une infrastructure informatique. Différentes formes et classifications du TCAO ont été dégagées. Parmi celles-ci, celle fondée sur la matrice espace-temps proposée par Ellis [15], garnie par Rodden et

complétée par Dix et Grudin [24] caractérise assez bien des situations typiques du travail coopératif. La collaboration est exprimée par rapport au temps (asynchrone ou synchrone) et à la distance (local ou distant). D'autres caractéristiques souvent utilisées concernent la granularité informationnelle de manipulation et la granularité temporelle de mise à jour d'actions utilisateurs, la conscience des actions des autres (awareness), le partage de vues, Globalement, ces caractéristiques permettent de qualifier assez précisément chaque contexte coopératif.

2 CARACTERISATION ET EVOLUTION

2.1 Besoins

Les besoins identifiés pour les différentes applications coopératives dépendent des paramètres listés ci-dessus. Il semble toutefois important d'introduire des nuances car distinguer seulement interaction asynchrone et interaction synchrone ne permet pas de tenir compte des besoins qui sont à la fois qualitatifs et quantitatifs. En effet, entre les interactions fortement couplées (de type « electronic meeting ») et les interactions lâchement couplées (de type édition collaborative) les délais souhaitables d'action et de perception ne sont pas les mêmes. Il en est de même pour la notion de vue, vue partagée (WYSIWIS strict ou relâché), le besoin d'observabilité ou au contraire de « privacy ». Sans vouloir exprimer toutes ces nuances, il semble souhaitable de rappeler le trèfle d'Ellis [16] qui caractérise l'activité coopérative par trois espaces permettant respectivement de communiquer, de co-produire, de se coordonner. Toutes les applications ne couvrent pas de la même façon ces trois espaces; certains outils peuvent se situer à l'intersection de deux voire de trois espaces. Dans tous les cas, il semble souhaitable tant en analyse qu'en mise en oeuvre de se positionner par rapport à ceux-ci [11].

2.2 Classification

Lorsqu'il est question de classification, la plupart des travaux existants procèdent à une distinction selon le type de tâche auquel les collecticiels considérés apportent un support. Il est ainsi habituel de distinguer entre les Group Decision Support Systems, les « recommender systems », les éditeurs collaboratifs et les supports à la co-conception. Une place spécifique est faite aux infrastructures ainsi qu'aux systèmes utilisés dans un contexte éducatif (e-learning au sens large du terme). Cette

classification semble naturelle et correspond de plus à des développements et travaux de recherche entrepris dans ces différents cadres depuis plusieurs années.

Il nous a cependant semblé opportun de considérer un autre critère de classification. Il s'agit du degré de liberté laissé aux usagers engagés dans une activité coopérative pour intervenir sur leur environnement, critère que nous croiserons avec le type de tâche considérée (spécifique ou pas) ainsi que l'explicitation ou pas du modèle de la tâche. Nous pourrions ainsi classer les collecticiels présentés selon un axe allant du plus spécifique et du plus restreint en termes de fonctionnalités au plus générique et au plus ouvert pour les usagers.

2.2.1 Les outils élémentaires

A la base, nous trouvons les outils qui peuvent répondre à des besoins primitifs de communication entre acteurs : chat, forum, tableau blanc partagé, mail. Ils peuvent également servir à la coordination et à l'organisation, à travers la communication, et dans une moindre mesure d'espace de production (cas du tableau blanc – mais il s'agit là d'un espace non structuré et bien souvent volatile). Ils sont en général considérés comme des briques élémentaires qui vont entrer dans la composition de collecticiels plus élaborés.

2.2.2 Les collecticiels centrés « tâches »

Au niveau intermédiaire, nous plaçons les collecticiels construits autour d'une tâche spécifique (édition collaborative, mutualisation documentaire, co-conception, ...). La tâche devient le sujet d'étude des concepteurs de collecticiels : ils cherchent à l'explicitier, en tout ou partie, étape après étape, dans le but de fournir un guide organisationnel ou un support logistique au groupe, dans un souci de coordination des actions des uns et des autres. Le focus retenu ici est celui du « process » et il est fait peu de cas des individus qui composent le groupe. Les fonctionnalités offertes dans les collecticiels concourent à fournir une aide dans ce sens.

Celles-ci varient grandement en fonction de l'explicitation ou non du modèle de la tâche. L'outil RtW (« Reading through Writing ») du projet européen Nimis [45] repose sur un modèle de tâche implicite dans lequel les ressources sont partagées entre les acteurs (des enfants). La tâche consiste à construire phonétiquement des mots correspondant à des images. Elle n'est réalisable que par la mise en commun des ressources (les phonèmes). La collaboration va alors émerger de l'impossibilité qu'a

chaque enfant de la mener à bien seul. L'outil leur fournit de fait un espace commun de construction du mot, dans lequel chacun peut déposer un phonème. Le projet « Memo-net » [32] s'appuie lui aussi sur un modèle de tâche implicite. Des acteurs sont invités à résoudre un problème d'analyse ou de synthèse en s'appuyant sur des concepts issus de ce modèle; ils disposent pour cela d'un forum structuré (étiqueté par ces concepts) qui leur permettra d'avoir une trace de la résolution du problème traité. Ces deux outils, parce qu'ils n'explicitent pas le modèle sur lequel ils sont construits, ne permettent aucune intervention des acteurs sur le processus et l'outil lui-même. Les acteurs se contentent d'utiliser les fonctionnalités et composants logiciels mis à leurs disposition pour résoudre la tâche.

L'explicitation du modèle, même partielle, fournit un premier moyen d'intervenir sur la tâche, et donc l'activité du groupe au delà. Ainsi, dans l'éditeur coopératif Byzance [8], il est supposé que les acteurs vont travailler sur des parties disjointes de documents, en écriture pour les modifier ou en lecture uniquement. La tâche de rédaction est alors vue comme l'agrégation des productions des uns et des autres, après fragmentation d'un texte initial. Les éléments issus de ce modèle de tâche sont des concepts tels que les rôles, les droits ou les fragments. Les acteurs disposent de fonctionnalités permettant de les manipuler. Ils acquièrent ainsi le moyen de commencer à influencer sur leur activité.

Lorsque nous parlons d'explicitation dans ce qui précède, cela relève davantage d'une « intuition » de ce que doit être la tâche que d'une véritable explicitation. Considérons des systèmes servant de support à des tâches complexes tels que BSCW [3] dans le cadre de la mutualisation de ressources documentaires ou [10] pour le développement de logiciels en équipe. Leurs concepteurs ont avant tout songé à fournir des fonctionnalités pratiques à destination des acteurs (par exemple annotation de documents, recommandations pour le premier; découpage en sous-projets, « todo list » pour le second) plutôt qu'une description rigoureuse de la tâche. Celle-ci serait en effet très difficile voire impossible à obtenir; tout au plus peut-on en donner les grandes lignes. Il semble ici préférable de fournir les moyens d'agir sur les systèmes (ex : création de groupes et de rôles) et de les adapter en fonction des besoins.

Seuls les systèmes intégrant la technologie du workflow conduisent à une explicitation complète du modèle de la tâche. Le workflow fournit en effet un formalisme rigoureux pour le décrire et des mécanismes pour l'automatiser. En ce sens, il s'agit d'une aide importante au travail de groupe pour la coordination des actions. A cet avantage, nous pouvons cependant opposer la difficulté d'explicitation du modèle, le manque de

support au sein des groupes à la négociation de ces modèles (celle-ci est bien souvent inexistante) ainsi que la rigidité qui en découle. L'explicitation du modèle devrait cependant permettre sa modification en situation et c'est actuellement l'objectif des travaux menés dans le domaine du workflow adaptatif (voir ci-après).

2.2.3 Les collecticiels centrés « modèle »

Nous rassemblons ici l'ensemble des collecticiels qui reposent sur un modèle non plus d'une tâche spécifique à réaliser mais de l'activité conjointe en général. Ces modèles sont inspirés de travaux en sciences humaines. Ils fournissent des concepts permettant de décrire ce qu'est une activité de groupe et par là même de la construire. Mis à la disposition des usagers, ces concepts vont leur permettre de structurer leur groupe, leur espace de travail et de décrire les conditions de leur engagement dans l'activité. Il s'agit ici d'explicitier un niveau « méta » et les fonctionnalités qui vont avec pour permettre non seulement une construction conjointe de l'activité du groupe mais aussi son évolution dans le temps, au cours de l'activité elle-même.

Parmi les systèmes relevant de cette catégorie, nous trouvons des « méta-collecticiels » tels que Worlds [20] [46] fondé sur la théorie des « locaux », Prospero [13] qui s'appuie sur l'ethnométhodologie ou encore DARE [5] qui lui a recours à la théorie de l'Activité. Dans la même veine, les travaux réalisés à l'Université de Savoie proposent un modèle original (le modèle de participation) qui s'inspire à la fois de la théorie des locaux, de l'ethnométhodologie, de la linguistique, de la pragmatique et de la sémantique différentielle. Ce modèle est à la base des réalisations effectuées dans le cadre du projet de « cartable électronique » [35] qui, si elles visent un cadre applicatif particulier (celui de l'Education), voire pour certaines une tâche spécifique (le dessin entre enfants) [18], n'en fournissent pas moins un niveau « méta » et des composants et fonctionnalités associés à ce niveau, réutilisables dans d'autres applications.

3 ASPECTS IMPORTANTS

3.1 Fonctionnalités et services

Les éléments technologiques nécessaires pour supporter le travail coopératif sous toutes ses formes sont nombreux. Nous pouvons lister les

suivants : distribution et synchronisation des objets partagés, accès concurrents à ces objets, gestion des droits d'accès, nommage, réplication, etc. Ne pouvant pas traiter tous ces aspects ici nous nous limiterons à la manipulation des objets par les acteurs coopérant et le contrôle de leur cohérence. En effet, les collecticiels reposent sur les mécanismes développés dans les systèmes distribués ou les gestionnaires de configurations: verrouillage des objets, passage de jeton (ou « prise de tour »), détection des dépendances (conflits résolus par les utilisateurs).

Afin de contrôler les mises à jour concurrentes effectuées par les différentes activités d'un système coopératif, il est possible d'utiliser un outil de gestion de configurations (RCS, ClearCase, Continuous, Adèle,...). Son rôle est d'assurer le stockage des données partagées, appelées ressources, tout en gardant une trace de leur évolution (généralement sous la forme de leurs versions successives) et en contrôlant les accès concurrents effectués par les activités. Par exemple, si deux activités modifient en parallèle une même ressource, elles vont chacune développer, à partir d'une version initiale de cette ressource, ce que l'on appelle une branche de versions. De cette façon, chacune des activités travaille sur sa propre copie de la ressource, sans être perturbée par les modifications effectuées par les autres. Lorsqu'elles ont terminé leur travail, une activité (éventuellement différente) est chargée de fusionner les branches afin de ne produire qu'une seule nouvelle version, sans que les modifications d'une des activités n'écrasent celles des autres. Le rôle du gestionnaire de configurations est alors de mémoriser le fait que cette nouvelle version est dérivée des précédentes.

La plupart des gestionnaires de configurations reposent sur une architecture client/serveur (référentiel centralisé, éventuellement répliqué et/ou partitionné sur plusieurs serveurs). Ils sont essentiellement concernés par les problèmes de concurrence d'accès à un référentiel : gestion des versions et des configurations de ressources partagées. Si l'on se place dans un cadre distribué dans lequel les activités réclament une certaine autonomie, on peut tirer profit d'une approche transactionnelle avec des modèles transactionnels étendus, des transactions de longues durées qui accèdent à une multitude de ressources distribuées. Le problème se « réduit » alors à ceux d'affaiblissement de l'atomicité et d'isolation, parmi les propriétés ACID (Atomicité, Cohérence, Isolation et Durabilité), pour les transactions longues, tout en assurant la cohérence et la fiabilité de l'exécution des tâches concurrentes et en offrant des mécanismes de reprise (ou de recouvrement) après une panne, ainsi que des mécanismes de compensation permettant d'annuler logiquement les effets d'une activité.

Plusieurs travaux ont choisi l'approche transactionnelle en se basant sur des échanges via un système de gestion de bases de données ou sur un moniteur transactionnel. Parmi ces travaux, nous pouvons mentionner: [7], le projet ECOO (Environments pour la COOperation) [2][22], le projet DOM (Distributed Object Management) [21], le projet WISE (Workflow based Internet Services) [1][41], le projet WIDE (Workflow on Intelligent Distributed database Environment)[23], le projet COW (Cooperative and Open Workflow) [47], le projet TRANSCOOP (Transaction Management Support for Cooperative Applications) [39], etc.

Même si les techniques de base ne sont pas nouvelles, la gestion des données manipulées dans un collecticiel reste primordiale et la façon dont cette gestion sera faite influera grandement sur la perception des utilisateurs quant à la collaboration en cours.

3.2 Awareness

Lorsque des personnes travaillent en « co-présence », c'est-à-dire en même temps et dans le même lieu, elles échangent un certain nombre d'informations, certaines implicites et d'autres explicites. Ces informations créent chez chaque participant une conscience de groupe, qui lui permet :

- de comprendre et mesurer l'activité et la dynamique du groupe ;
- de situer sa propre action au sein du groupe, et ainsi de coordonner ses propres activités avec celles des autres.

Cette notion de conscience de groupe (awareness) est originellement définie par Dourish et Belloti [14] comme étant « la compréhension des activités des autres, qui permet de donner un contexte à sa propre activité ». Dans une équipe virtuelle, la disparition de la co-présence entraîne la disparition de certaines informations, particulièrement les informations implicites, qui entraîne à son tour la disparition de la conscience de groupe. Le but, dans le contexte des applications de collaboration/coopération, est donc de reconstruire la conscience de groupe grâce à des modèles et par le biais de mécanismes adaptés. Modéliser cette conscience de groupe nécessite :

- l'identification des informations utilisées en co-présence pour construire la conscience de groupe ;
- une synthèse claire et utilisable des nombreuses informations que l'on peut obtenir des systèmes informatiques ;

- une représentation à l'utilisateur d'une information pertinente mais non obstructive.

On distingue généralement trois formes de conscience de groupe : la « group awareness » a pour but de rendre tangible le travail en groupe et d'être conscient de l'état et des activités des partenaires; la « Workspace awareness » (conscience de l'espace de travail commun) se situe à un plus grand niveau de détail et permet aux utilisateurs travaillant sur les mêmes documents d'être au fait des modifications apportées par les partenaires situés dans le « même » espace de travail; enfin, la « process awareness » (conscience du procédé commun) permet aux participants de situer leurs actions dans le contexte plus large d'un projet « global » et coordonné [6].

Parmi les différents modèles proposés pour résoudre le problème de la conscience de groupe dans les outils de collaboration/coopération nous pouvons citer les modèles basés sur la publication/souscription et les modèles spatiaux. Le modèle publication/souscription peut grossièrement être décrit par : les producteurs d'information publient des événements et des messages vers le système et les consommateurs d'information souscrivent des abonnements pour certaines catégories d'événements déterminées soit par leur sujet, soit par leur contenu. Le modèle spatial, quant à lui, est fondé sur une métaphore spatiale et définit la conscience de groupe en terme de positions, proximité et distance à l'intérieur d'un espace virtuel commun.

Parmi les infrastructures permettant l'implantation de la conscience de groupe dans les systèmes coopératifs, nous pouvons citer Elvin [17] (développé par DSTC), Gryphon [25] (IBM) et Nessie [38] (GMD - Fraunhofer). Gryphon et Elvin utilisent la publication / souscription sur des données différentes : sujets et contenus pour le premier, contenus pour le second. Elvin vise par ailleurs au passage à l'échelle. Il fonctionne avec de très grands nombres de messages et d'utilisateurs et a servi au développement de collecticiels comme TickerTape et Orbit. Nessie, quant à lui, permet la notification synchrone ou asynchrone d'événements qui sont présentés de différentes façons à l'utilisateur (popup, animation, sons,...).

3.3 Workflow

La coordination sociale s'effectuant sous la responsabilité exclusive des acteurs ayant montré ses limites, il s'agit d'utiliser l'ordinateur pour cette activité ou au moins partager cette activité entre l'ordinateur et les

acteurs. Le Workflow constitue une réponse technologique à la gestion des processus. Malheureusement, s'il est facilement utilisable dans les processus très rigides, la souplesse et la flexibilité ne sont pas les points forts des outils actuellement disponibles. Dans la coordination de nombreux processus coopératifs, la rigidité n'est pas acceptable. Il est en effet nécessaire de permettre à la fois des interventions opportunistes au sein d'un processus globalement structuré et de respecter une certaine rigueur dans les différents sous-processus. Les travaux sur les Workflows adaptables tentent de proposer des approches permettant au Workflow de pouvoir être utilisé dans la coordination au sein des collectifs synchrones [40].

4 VERS DES LOGICIELS DE TCAO EVOLUTIFS

4.1 Malléabilité et composants

Depuis déjà plusieurs années, le domaine de recherche sur le TCAO a identifié la malléabilité [37] des systèmes comme un besoin important pour toute application ou plate-forme [30]. Une première approche architecturale de la malléabilité assez répandue est celle liée aux composants et aux architectures cadres (frameworks) [3][26][27]. Le composant est l'unité de base réutilisable, évolutive et compréhensible supportant la malléabilité alors que l'architecture cadre fournit l'ossature de l'application. Ainsi, cette démarche offre un moyen simple d'évolution de l'application par la manipulation de sa structure au niveau d'abstraction des composants.

Néanmoins, si cette approche apparaît comme une solution intéressante, elle est loin de résoudre tous les problèmes liés à la mise en œuvre de la malléabilité. Il reste encore à déterminer les mécanismes d'évolution de ces derniers. Ces différents mécanismes ont été étudiés par Anders Morch [36] qui les définit en termes de paramétrage, d'intégration et d'extension. Le paramétrage offre une malléabilité limitée par un nombre de choix prédéterminés, comme par exemple choisir un composant parmi une liste figée de N. L'intégration consiste à greffer un nouveau composant dans l'architecture de l'application. L'extension revient à modifier le code du composant lui-même. Le paramétrage, l'intégration et l'extension offrent de plus en plus de malléabilité mais demandent à son utilisateur de plus en plus grandes compétences en informatique. Ce point explique en partie pourquoi la plupart des

systèmes de TCAO actuels orientent généralement leur malléabilité vers des utilisateurs développeurs ou experts du système plutôt que vers les utilisateurs finaux qui, paradoxalement et d'après les recherches en sciences humaines, sont ceux qui en ont le plus besoin [31].

4.2 Autres approches de la malléabilité

L'ensemble de ces problèmes et solutions ont été approfondis et pris en compte dans la création de l'architecture du système DARE développé par G. Bourguin [5]. Ce système est fondé sur la Théorie de l'Activité (TA) et veut offrir une malléabilité accessible aux utilisateurs finaux dans la démarche nommée Co-évolution [4] (voir ci-après). DARE est défini comme un collecticiel réflexif (au sens de Maes [33]) et repose sur une approche de type Implémentation Ouverte (IO) de Kickzales [28].

C'est ce même souci de donner la main aux utilisateurs finaux qui a conduit C. Martel à proposer un espace dédié à l'activité de configuration, reconfiguration, définition et modification du système [34]. En manipulant les concepts du modèle qui structure cet espace (acteurs, rôles, scénarios notamment), les utilisateurs ont la possibilité de décrire leur mode de participation à l'activité ainsi que les conditions du travail en groupe. Une architecture fondée sur ce modèle a été proposée et une implantation dans deux applications a permis de valider en partie le modèle [19][35].

Une autre approche de malléabilité, appelée flexibilité [44] a été mise en place dans la proposition de l'architecture AMF [42] pour les systèmes interactifs et dans sa version AMF-C [43] pour les systèmes coopératifs. Le modèle AMF-C se caractérise par l'approche multi-facettes qui conduit à identifier et mettre sous forme de pattern tout comportement stéréotypé. La possibilité de définir de nouvelles facettes chaque fois qu'une nouvelle fonctionnalité se stabilise, constitue un atout fondamental d'AMF-C. Les liens entre facettes qui expriment le « contrôle » sont présentés graphiquement à l'utilisateur, qui peut par leur manipulation choisir dynamiquement par exemple une présentation alternative, un autre comportement applicatif ou rendre la propagation des modifications plus ou moins globale (à aucun, à un certain nombre ou à la totalité des acteurs de la coopération).

4.3 La co-évolution

Notre vision est que le système coopératif produit doit pouvoir s'adapter aux besoins émergents, à la transformation de besoins ayant servis à l'analyse initiale, ainsi qu'aux évolutions des contextes d'usages et des technologies utilisées. Ces systèmes peuvent être vus à double titre comme une intermédiation (un passage plus qu'une séparation) entre les concepteurs et les usagers:

- intermédiation entre les besoins, compétences, désirs... des usagers et les contraintes, technologiques ou autres, les compétences et les désirs des concepteurs ;
- intermédiation entre les deux parties pour en assurer la maintenance et l'évolution dans la durée. Cela implique, nous le justifierons après, que le système coopératif doit contenir sa propre représentation (un méta-modèle) et être capable de réflexivité (introspection et intercession).

Nous pensons, utilisant pour cela l'expérience acquise au sein de la communauté du TCAO, que cette transformation continue a une dimension collective et historique. C'est au sein de la communauté de pratique, au sens de la sociologie des usages, que les artefacts informatiques proposés doivent être adaptés afin de refléter la cristallisation des compétences et des attitudes de ses membres.

Le terme co-évolution a été choisi pour traduire le fait que les systèmes coopératifs doivent être continûment évolutifs, sans l'être toutefois de façon autonome ou auto-adaptative, car ils doivent rendre compte, de manière consciente, des évolutions des besoins, des attitudes et des compétences des usagers, individuellement ou collectivement. Mais il n'y a pas que le système qui doive évoluer, l'utilisateur doit également transformer, adapter, ses pratiques et ses méthodes de travail, c'est-à-dire son rôle, pour satisfaire les besoins évolutifs de l'organisation [48]. C'est une vue plus large que la boucle « tâche/artefact » proposée par Carroll et Rosson [9] pour essayer de traduire la relation dialectique qui existait entre la spécification des besoins au travers de la tâche et l'artefact lui-même résultant du processus de conception. La co-évolution traduit tant les ajustements continus, négociés et socialement situés, des pratiques de la part des individus, que les ajustements apportés au comportement du système interactif considéré. Ceci peut-être synthétisé par le schéma donné dans la figure ci-après, où il apparaît que l'environnement global fixe le contexte de la tâche [29].

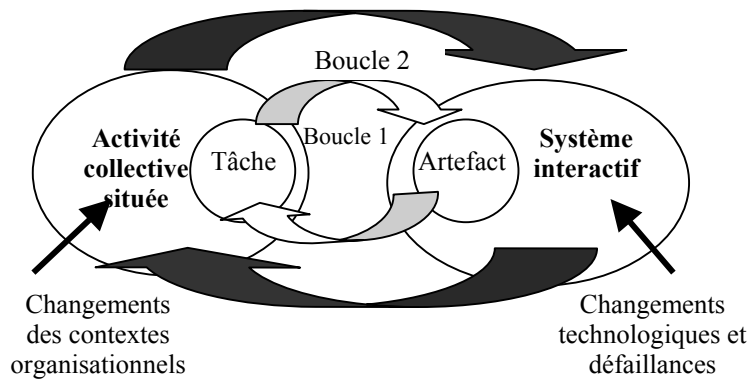


Illustration de la co-évolution

5 TENDANCES ACTUELLES

Après les développements d'applications spécifiques, puis de systèmes collaboratifs, une démarche plus industrielle se met en place en proposant des plates-formes intégrant des services et composants dédiés au travail de groupe. De nouvelles applications peuvent ainsi être plus rapidement développées en utilisant ces briques de base.

Les collecticiels actuellement développés tendent vers la co-évolution. Il s'agit de réaliser des plates-formes de travail de groupe, dans lesquelles les acteurs définissent eux-mêmes les modalités de l'organisation du groupe, son fonctionnement, les caractéristiques de l'espace partagé ainsi que les outils qu'ils souhaitent utiliser. L'espace de travail et ses lois sont ainsi configurés par les usagers eux-mêmes et ils évoluent de manière dynamique, au cours de l'activité elle-même. Cet idéal est encore loin cependant d'être atteint, la reconfiguration dynamique nécessitant bien souvent des compétences en informatique ou une connaissance des modèles théoriques sous-jacents que nous ne sommes pas en droit d'attendre des usagers. Elle bute de plus sur des verrous technologiques.

Par ailleurs, l'évolution actuelle du TCAO vers une plus grande disponibilité des acteurs passe entre autres par la prise en compte de leur mobilité et de l'utilisation de dispositifs légers. Cette approche est qualifiée de TCAO capillaire par [12]. Le TCAO capillaire répond à un

fort besoin émis par des utilisateurs qui ne travaillent plus seulement dans des bureaux devant des postes fixes. En plus de l'accès aux informations, de l'interaction et de la collaboration par la virtualisation de l'espace et du temps, il ajoute la mobilité des personnes, leur plus grande disponibilité et la contextualisation de leurs actions et informations grâce à la prise en compte du contexte précis, de leur localisation et de la location des objets sur lesquels portent le travail.

6 CONCLUSION

Dans cet article nous avons explicité l'évolution du travail coopératif. Nous en avons affiné la définition et décrit les principales caractéristiques. Nous avons pu constater le besoin d'allier flexibilité, robustesse et généricité pour répondre de façon satisfaisante aux nouvelles exigences liées entre autres à la prise en compte du contexte logique et/ou géographique obtenu par les capacités nouvelles des dispositifs nomades. Nous avons mis en évidence le besoin d'une nouvelle approche du cycle de vie des systèmes coopératifs qui prennent en compte la gestion, par les usagers en partie, des changements à opérer sur le système lui-même pour satisfaire les évolutions des tâches des usagers.

7 REFERENCES

- [1] Alonso G., Hagen C., Lazcano A. Process in Electronic Commerce, ICDS workshop on Electronic Commerce and Web-Based Applications, Austin, Texas, USA, June 1999.
- [2] Baïna K., Benali K., Godart C. A process service model for dynamic enterprise process interconnection, Proceedings of 6th International Conference on Cooperative Information Systems (CoopIS'2001), in cooperation with VLDB'2001, Trento, Italy, 2001, LNCS 2172.
- [3] Bentley, R., Appelt W., Busbach U., Hinrichs E., Kerr D., Sikkell K., Trevor J., Woetzel G. Basic support for cooperative work on the World Wide Web. International Journal of Human-Computer Studies Vol. 46, pp. 827-846, 1997.
- [4] Bourguin G., Derycke A., Tarby J.C. Beyond the Interface: Co-evolution Inside Interactive Systems – A proposal Founded on Activity Theory, Springer Verlag, People and Computer vol. 15 – Interaction without Frontiers, proceedings of Human Computer Interaction 2001 (HCI'2001), Blandford, Vanderdonckt, Gray (eds), pp. 297-310

- [5] Bourguin, G. Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : le projet DARE, Ph. D. Thesis, Informatique, n° 2753, Université des Sciences et Technologies de Lille, France, 210 p., 2000
- [6] Bouthier C., Canals G. Le contexte comme base de la conscience de groupe, CITE, Troyes, France, Novembre 2001.
- [7] Breitbart, A. Deacon, H-J. Schek, A. Sheth, G. Weikum, Merging Application-centric and Data-centric Approaches to Support Transaction-oriented Multi-system Workflows, SIGMOD Record, pp 23-30, Vol 22, N° 3, 1993
- [8] Byzance <http://www.inrialpes.fr/opera.html>
- [9] Carroll J., Rosson, M. Getting around the Task-Artifact Framework: How to Make Claims and Design by scenario, ACM Transaction On office Information Systems, vol. 10 (2), 181-212. 1992.
- [10] Charoy F., Godart C., Molli P., Oster G., Patten M., Valdes M. Services For Virtual Teams Hosting. ToxicFarm Introduction. Second International Workshop on Cooperative Internet Computing (CIC), August 18-19, 2002, Hong Kong, Kluwer Academics.
- [11] David B. IHM pour les collecticiels. In Réseaux et Systèmes Répartis. Hermès, Paris, vol. 13, novembre 2001, pp. 169-206. ISBN 2-7462-0303-0.
- [12] David B., Vaisman G., Saikali K. Evolution du Travail Coopératif Assisté par Ordinateur : Vers le TCAO « capillaire », CITE'2001, Coopération, Innovation et Technologie. 29 et 30 Novembre 2001. Université de Technologie de Troyes. Novembre 2001.
- [13] Dourish P. Using Metalevel Techniques in Flexible Toolkit for CSCW Applications, ACM Transaction on Computer-Human Interaction, vol. 5, n°2, pp. 109-155. 1998.
- [14] Dourish P., Bellotti V. Awareness and Coordination in Shared Workspaces, Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW-92), Toronto, Ontario, 1992, ACM Press, p. 107-114.
- [15] Ellis C.A., Gibbs S.J., Rein G.L. Groupware: some issues and experiences, Communication of ACM, 34(1), January 1991, pp. 39-58.
- [16] Ellis C., Wainer J., A conceptual model of Groupware, In Proc. CSCW'94, ACM Press, p. 79-88., 1994.
- [17] Elvin – <http://elvin.dstc.edu.au>.
- [18] Ferraris C., Martel C. Regulation in groupware: the example of a collaborative drawing tool for young children. Proceedings of CRIWG'2000, 6th international workshop on groupware, Madeira, Portugal, pp. 119-127. 2000.
- [19] Ferraris C., Brunier P, Martel C. Constructing collaborative pedagogical situations in classrooms: a scenario and role based approach. Computer Support for Collaborative Learning 2002 (CSCL 2002), Boulder, Colorado, USA, 7-11 January 2002.

- [20] Fitzpatrick G., Tolone W. J., and Kaplan S. M. Work, Locales and Distributed Social Worlds. Proceedings of the ECSCW'95, Stockholm, September, pp. 1-16. 1995.
- [21] Georgakopoulos D., M. F. Hornick, F. Manola, M. L. Brodie, S. Heiler, F. Nayeri, B. Hurwitz, " An Extended Transaction Environment for Workflows in Distributed Object Computing", IEEE Data Engineering Bulletin, Vol.16, N° 2, June 1993
- [22] Godart C., Perrin O., Skaf H., COO : a workflow operator to improve cooperative modelling in virtual processes", 9th International Workshop on Research Issues on Data Engineering : Information Technology For Virtual Enterprises (RIDE-VE'99), March 1999, Sydney, Australia
- [23] Grefen P. Advanced Architectures for Transactional Workflows or Advanced Transactions in Workflow Architectures, International Process Technology Workshop (IPTW'99), 1999.
- [24] Grudin J. CSCW: History and Focus. In Computer, 29(6), IEEE Computer Society, June 1992, pp. 27-35.
- [25] Gryphon – <http://www.research.ibm.com/gryphon>
- [26] Hoogstoel F. Une approche organisationnelle du Travail Coopératif Assisté par Ordinateur. Application au projet Co-Learn, Th. de Doctorat en Informatique, Université des Sciences et Technologies de Lille, 1995, n° 1487
- [27] Hummes J., Merialdo B. Design of extensible component-based groupware, soumis pour être publié par Kluwer dans le Journal of CSCW, 1999, <http://www.eurocom.fr/~hummes/docs/JCSCW/JCSCW.html>
- [28] Kiczales G., Bobrow D.G., Des Rivieres J. The Art of the Metaobject Protocol, MIT Press, August 1991, 335 p
- [29] Kirsh, D. Distributed Cognition, Coordination and Environment Design. In Proceedings Of the European Cognitives Sciences Society (<http://icl-server.ucsd.edu/~kirsh/articles/italy/published.html>) 1999.
- [30] Koch M. Teege G., Support for tailoring CSCW systems: adaptation by composition. Seventh Euromicro Workshop on Parallel an Distributed Processing PPD'99, 1999, pp 146-153.
- [31] Kuutti K. The concept of activity as a basic unit of analysis for CSCW research, Proceeding of the second ECSCW'91 conference, Kluwers Academics Publishers, 1991, pp 249-264.
- [32] Lewkowicz M., Zacklad M. A structured groupware for a collective decision-making aide. European Journal of Operational Research 136, 2002, pp.333-339.
- [33] Maes P. Computational Reflection, Ph.D. Thesis, V.U.B, Brussels, 1987.
- [34] Martel C., Ferraris C., De la régulation dans les collecticiels, Actes de IHM99, Montpellier 1999.

- [35] Martel C., Vignollet L. Educational Web Portal based on personalized and collaborative services. Proceedings of ICALT (International Conference on Advanced Learning Technologies), Madison, USA, August 6-8. 2001, <http://www.cartable-electronique.fr>,
- [36] Morch A. Three levels of end-user tailoring: customization, integration, and extension, dans [37], 1995, pp 41-51
- [37] Morch A. Method and Tools for Tailoring of Object-oriented Applications: An Evolving Artifacts Approach, part 1, Dr. Scient. Thesis Research Report 241, University of OSLO, Department of Informatics, 1997
- [38] Nessie - <http://orgwis.gmd.de/projects/nessie>
- [39] Puustjärvi J. Transactional Workflows, Department of Computer Science, University of Helsinki, Finland, 1999
- [40] Saikali K., David B. Vers l'usage du Workflow pour la coordination dans les collecticiels. In Actes de IHM-HCI 2001, Lille, France, septembre 2001. ISBN 1-85233-515-7
- [41] Schuldt H., Schek H.J., Alonso G. Transactional Coordination Agents for Composite Systems, Proceedings of the International Database Engineering and Applications Symposium, (IDEAS'99), Montreal, Canada, August 1999
- [42] Tarpin-Bernard F., David B. AMF : un modèle d'architecture multi-agents multi-facettes. Techniques et Sciences Informatiques. Hermès. Paris. Vol. 18. No. 5. p. 555-586. Mai 1999.
- [43] Tarpin-Bernard F., David B.T., Primet P. Frameworks and patterns for synchronous groupware: AMF-C approach, In Proceedings of EHCI'98, IFIP Working Conference on Engineering for HCI, Greece, Sept. 1998, 13 p.
- [44] Tarpin-Bernard F. La flexibilité dans les collecticiels, Ecole thématique Documents & évolution du GDR I3, Tome 2, Cépaduès, 2000.
- [45] Tewissen F., Lingnau A., Hoppe U., Mannhaupt G., Nischk D. Collaborative Writing in a Computer-integrated Classroom for Early Learning. Proceedings of Euro-Csel 2001, Maastricht, Netherlands, Mars 22-24, 2001
- [46] Tolone W. J. Introspect: a Meta-level Specification Framework for Dynamic, Evolvable Collaboration Support. Ph.D. Thesis. University of Illinois at Urbana-Champaign, 1996.
- [47] Vantrois V., Peter Y. A WMF-Based workflow for e-learning, European Research Seminar on Advances in Distributed Systems, Advanced School and Workshop (ERSADS'01), Bologna, Italy, May 2001
- [48] Vicente, K. J. HCI in the Global Knowledge-Based Economy: Designing to Support Worker Adaptation, ACM Transactions on Computer-Human Interaction, vol. 7, n°2, 263-280, 2000.