

# Une Architecture en Couches pour un Editeur Partagé de Documents

A. M. Zidani<sup>1</sup> et M. Djoudi<sup>2</sup>

<sup>1</sup> Institut d'informatique de l'université de Batna

<sup>2</sup> Département d'informatique, Université de Poitiers, France

## Résumé

*Le collecticiel ( travail-coopératif-assisté-par-ordinateur ou Computer Supported Cooperative Work (CSCW) en anglais), est un domaine récent des applications de groupe qui est caractérisé par l'aspect fondamental du partage de la connaissance et du savoir faire. Les éditeurs partagés constituent l'une des applications les plus importantes du collecticiel. Ces systèmes permettent à plusieurs utilisateurs distants de travailler simultanément sur un même document, avec un support assurant la transparence et l'intégrité des données. L'objectif de cet article est la présentation de la conception d'un éditeur coopératif synchrone intégrant les concepts essentiels de l'édition coopérative et de l'interaction de groupe. Nous passons d'abord en revue les principaux travaux qui ont été réalisés dans ce domaine, nous présentons et discuterons ensuite les concepts de base pour la conception d'un éditeur coopératif qui repose sur une architecture en couches répliquée.*

## Mots Clés

Collecticiel, CSCW, Travail coopératif, Edition partagée, Interfaces graphiques multi-utilisateur

## Abstract

*The recent field of computer supported cooperative work (CSCW) is characterized by the fundamental aspect of knowledge and cleverness sharing. Collaborative editing is a major research topic within CSCW applications. While preserving some kind of consistency and transparency, it allows different users at different locations to edit a shared document at the same time. This paper presents a synchronous shared editor design, embedding concepts that are essential for collaborative editing and group interaction. We firstly passed ongoing research in review and then presented basic concepts for a cooperative editor design with an oriented-layers replicated architecture.*

## Keywords

Groupware, CSCW, Cooperative Work, Shared Editing, multi-user Graphical Interfaces

## 1. Introduction

Les systèmes informatiques ont longtemps eus comme objectif de fournir l'assistance à une seule personne pour réaliser ses tâches. Cependant, l'exploration de ces systèmes dans un contexte coopératif demeure très limitée. Aujourd'hui une nouvelle vision est apparue, elle s'appuie sur le principe que les groupes constituent une base pour toute organisation. Et permettent d'accomplir des tâches irréalisables par une seule personne. Cette démarche a entraîné d'importants changements au sein des organisations, qui sont passées des grandes directions verticales à de petites équipes créatives mobilisées ponctuellement sur des sujets précis. Les "logiciels de groupe" qui font l'objet de nombreux travaux de recherche, tentent d'accompagner cette nouvelle vision. Ils visent à aider des personnes à travailler ensemble, et représentent par conséquent une réponse technologique idéale pour satisfaire aux objectifs fixés par les activités collectives.

Nous visons dans le cadre de nos travaux à créer un espace de travail commun à un ensemble d'utilisateurs, rassemblant les outils nécessaires à l'édition coopérative de documents. Comme toute application coopérative, notre système tente de satisfaire les contraintes du travail de groupe. Par conséquent, nous avons conçu un support pour gérer les interactions entre

les utilisateurs et assurer la transparence sur leurs actions mutuelles.

L'objectif de cet article est la présentation d'un modèle pour un éditeur coopératif, prenant en considération la façon naturelle de collaboration d'un groupe de rédacteurs. Nous passons d'abord en revue les principaux travaux du domaine. Nous présentons ensuite notre approche ainsi que les concepts de base pour la conception d'un éditeur coopératif qui repose sur une architecture en couches répliquée. Finalement, nous décrirons les fonctionnalités essentielles de notre éditeur.

## 2. Travail coopératif

Le collecticiel dans sa définition la plus large, fait l'objet d'un champ d'étude pluridisciplinaire appelé CSCW ou "Computer supported cooperative work" (travail coopératif assisté par ordinateur). Cette discipline étudie les mécanismes individuels et collectifs du travail de groupe. Les recherches actuelles tentent de comprendre comment les technologies de l'information peuvent-elles faciliter la réalisation d'une tâche par un groupe de personnes. Plusieurs définitions du collecticiel ont été proposées [1, 2, 3]. Mais aucune ne s'est véritablement imposée entre toutes. Nous retenons parmi ces dernières la définition suivante de Palmer[4] "Un collecticiel est un système qui intègre le traitement de l'information et

les activités de communication dans le but d'aider les utilisateurs à travailler ensemble au sein d'un groupe". De cette définition, il en découle qu'un système collecticiel est caractérisé par deux aspects fondamentaux: la "technologie" et la "communication humaine". Et que le collecticiel doit pouvoir les intégrer tout en veillant à ne pas favoriser un aspect au détriment de l'autre.

### 3. Edition Coopérative de Documents

Les éditeurs coopératifs constituent l'une des applications les plus importantes du collecticiel. L'écriture coopérative est définie selon Ellis[1] comme un processus dans lequel, les auteurs (i.e. rédacteurs, experts, utilisateurs, lecteurs, etc.) interagissent avec différentes expertises durant la création et la révision d'un document commun. Ces systèmes permettent de produire un document par plusieurs acteurs qui peuvent se trouver éventuellement à des endroits distants et peut être à des moments différents.

Plusieurs travaux ont déjà été proposés pour le support de l'écriture coopérative. Un premier type de travaux appelés éditeurs "asynchrones", qui ne permettent qu'à un seul acteur à la fois d'agir sur le document, comme par exemple les éditeurs: CES[5], QUILI[6], SHARED-BOOKS[7], PREP[8], etc. Par contre les éditeurs "synchrones", permettent à plusieurs utilisateurs d'intervenir simultanément sur le même document comme par exemple les éditeurs de texte GROVE[1], SASSE[9] ou encore les éditeurs de graphiques structures GROUPEDESIGN[10], GROUPEDRAW et GROUPESKETCH [11], etc. Avec ce type d'éditeurs il est possible de modifier les données avec une granularité fine. Par exemple, deux utilisateurs peuvent modifier le même mot en même temps. Ainsi, les modifications portées par un utilisateur sont immédiatement visibles par les autres participants.

L'éditeur coopératif CES est l'un des premiers outils qui a fait la distinction entre l'information textuelle et l'information structurelle. Le texte est décomposé en sections qui sont propres à chaque rédacteur. Tout comme CES, QUILI, utilise les sections comme granularité d'édition.

Par ailleurs, des éditeurs tels que GROVE ou des boîtes à outils telles que DISTEDIT[12] offrent des informations non structurées. GROVE est caractérisé par une granularité fine et supporte des mises à jour en temps réel. Il utilise un algorithme pour la gestion des accès concurrents qui est fondé sur les priorités et la sémantique des opérations. Le tour de rôle adopté par DISTEDIT permet à un utilisateur d'intervenir seul sur le document partagé, tandis que tous les autres participants observent et suivent son travail en temps réel.

MMM[13] supporte la collaboration simultanée en temps réel avec une granularité de partage fine. Ceci inclut l'accès simultané à la même chaîne de texte ou à un objet graphique. MACE[14] supporte une granularité éditable variable, un utilisateur peut appliquer une paire de verrous à des fragments de texte. Les verrous "début" et "fin" marquent une zone de texte qui peut être modifiée sans interférences avec les autres, aussi longtemps que les

verrous sont détenus par un éditeur serveur centralisé. Les autres approches basées sur le verrouillage pour l'édition concurrente sont MULE[15] et MultimETH[16]. La granularité éditable de MULE est une ligne ou un bloc de texte, alors que MultimETH permet à un groupe de créer de façon coopérative, d'éditer et gérer des documents multimédia composés d'un nombre d'éléments structurés (i.e. Titres, entêtes, chapitres et sections). Les éléments structurés ont une relation hiérarchique et un contenu spécifique (i.e. texte, graphiques ou icônes). Il existe une variété d'autres approches telles que MMCONF[17], COAUTHOR[18], GROUPSYSTEMS[19], etc.

### 4. Présentation du modèle

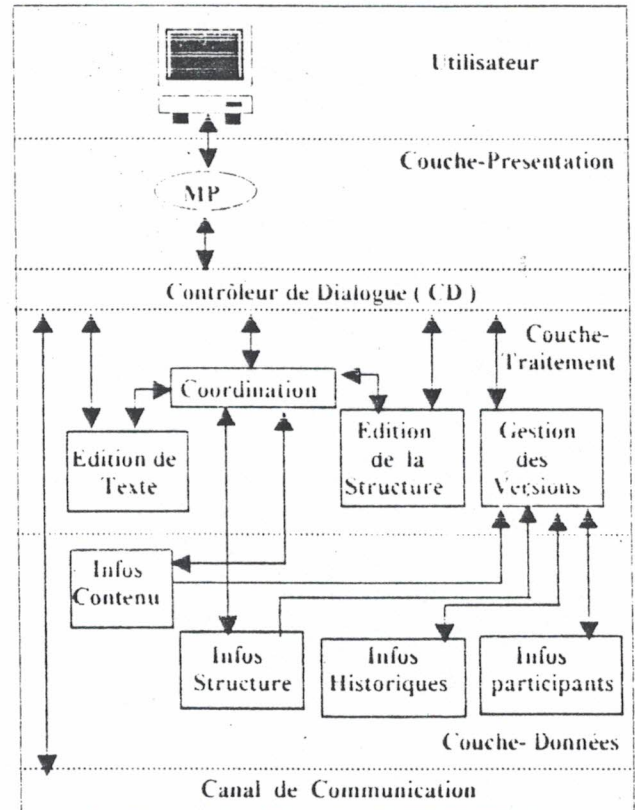


Figure 1 : Architecture en couches de l'éditeur

Dans notre modèle d'édition coopérative, nous nous appuyons sur la structure logique du document pour exprimer et gérer la coopération. Contrairement à un éditeur mono-utilisateur, l'existence des autres utilisateurs qui travaillent simultanément sur le même document crée la nécessité d'avoir une vue logique commune sur la structure globale du document. Cette structure permet la spécification d'unités et de positions logiques au sein du document. Elle constitue un moyen efficace pour répondre aux questions que peut se poser l'utilisateur telles que: " Qui a opéré telle ou telle opération sur le document? Quel rôle suis-je entraîné de jouer et sur quelles parties du document? Que font mes collègues? " etc. L'existence de cette structure logique suggère naturellement un ensemble d'opérations pour l'utiliser et la manipuler (mécanismes de notification et d'affichage des changements de la structure). Ceci permet alors de gérer les différentes versions du document au fur et à mesure de son évolution. Ainsi, il sera possible aux utilisateurs de

revoir les étapes précédentes du travail et de connaître l'identité de tous les acteurs impliqués.

La figure 1 montre le site d'un utilisateur participant au projet de rédaction coopérative. Nous distinguons une "couche Présentation" qui est chargée de gérer le dialogue entre l'utilisateur et l'éditeur. La "couche Traitement" est responsable des tâches d'édition, de coordination de notification et de gestion des versions. Enfin, la "couche Données", qui rassemble toutes les informations nécessaires à la conduite d'un projet coopératif (contenu du document, les informations associées à la structure, à l'historique et aux participants).

#### 4. 1 Couche Présentation

Le Module de présentation (MP) regroupe un ensemble organisé d'objets interactifs définissant la partie perceptible du système. Cette couche regroupe les traitements d'analyse et d'affichage qui sont chargés respectivement de traquer les gestes de l'utilisateur pour déterminer ses intentions ainsi que d'afficher la représentation appropriée à la situation courante. L'objectif de la couche présentation est de montrer à l'utilisateur une représentation simple, facile à comprendre et à manipuler pour réduire ses charges cognitives [20]. L'affichage d'une vue explicite qui ne comporte aucune ambiguïté, permettra alors de faciliter et d'encourager le dialogue entre les utilisateurs.

#### 4. 2 Couche Données

**Contenu du document:** Il s'agit du contenu correspondant à la version courante du document. Il est pratiquement similaire au document d'un éditeur mono-utilisateur.

**Informations de la structure:** Elle correspond à une structure explicite du document. Elle est visualisée et manipulée par les utilisateurs. Elle est également utilisée par l'éditeur pour fournir aux utilisateurs une vue compacte de leurs actions sur le document. L'information structurale et textuelle ont aussi une "structure implicite" qui n'est visible qu'à l'éditeur.

La structure implicite repose sur une séquence d'enregistrements qui peuvent être verrouillés par chaque membre du groupe. Par contre la structure explicite est composée de plusieurs niveaux (telle que la structure d'un document avec des chapitres, sections, sous-sections, paragraphes, appendices, en-têtes, notes de bas de page, figures et tables) pour fournir aux utilisateurs des vues logiques sur les différentes parties du document et établir le lien entre chaque contenu et l'élément de la structure associé. Pour être plus précis, la structure logique du document constitue le moyen à travers lequel nous pouvons décrire le document en cours d'édition tel qu'il est vu par l'utilisateur. Du point de vue technique, la structure explicite est essentiellement utilisée pour identifier des parties ou des positions dans le document. Son exploitation dans un environnement coopératif permet d'une part d'informer l'utilisateur sur les actions opérées sur le document. Et d'autre part de coordonner et synchroniser les actions des utilisateurs. Il faut préciser cependant, que la structure du document peut être utilisée à d'autres fins qui ne nécessitent pas de

coordination. Par exemple, le positionnement au début ou fin d'une partie du document ou la sélection d'une partie devant subir une opération de suppression, déplacement, etc.

#### 4. 3 Couche Traitement

**Tâches d'édition de texte:** Ce module regroupe toutes les fonctionnalités qui utilisent seulement les informations textuelles. Ces fonctionnalités sont similaires à celles des éditeurs mono-utilisateur existants.

**Tâches d'édition de la structure:** La structure du document évolue normalement durant l'existence du document. Ainsi, de façon analogue à l'édition des contenus, l'éditeur permet l'édition de la structure qui est une fonctionnalité est totalement indépendante de l'édition des contenus. Nous distinguons par exemple comme opérations appliquées à la structure: la définition d'une nouvelle partie, la suppression d'une partie existante, le déplacement d'une partie vers une autre position dans la structure, etc.

**Coordination:** Les deux modules d'édition des contenus et de la structure s'appuient sur le module de coordination pour gérer les interventions concurrentes des utilisateurs. Le module de coordination (fig. 1) interagit d'une part avec les deux modules d'édition des contenus et de la structure et permet la réalisation effective de leurs actions sur les données. Il accomplit évidemment au préalable un certain nombre de contrôles (droits d'accès, priorités, état des parties manipulées (libres ou bloquées), etc.). Et d'autre part, il est chargé d'avertir instantanément et de façon explicite l'utilisateur (via le contrôleur de dialogue) sur les actions qui ont été effectuées.

En dehors du moyen implicite de la coordination (cachée), les opérations de la coordination explicite doivent être fournies au niveau de l'interface à l'utilisateur. Un exemple sont les blocages explicites des parties du document par un utilisateur, une requête de relâchement du blocage, etc. Il faut préciser pour le cas des blocages, qu'ils peuvent être "simples" (i.e. l'unique possibilité offerte aux autres est la consultation) ou bien "strictés" (i.e. l'éditeur garantit l'accès exclusif aux parties bloquées).

Pour assurer la transparence et la coordination, nous avons adopté le concept de notification des événements [21]. De même que pour intégrer notre application dans un environnement coopératif, on s'interpose entre son noyau fonctionnel et son interface utilisateur (fig. 1), de telle sorte qu'une action réalisée par un utilisateur puisse être diffusée aux autres. Pour développer les mécanismes de partage, nous adoptons l'approche de manipulation des "événements de niveau-haut" [22] qui permet une interprétation simple des événements. Pour cette approche les différents sites échangent entre eux des messages récupérés au niveau de l'application (par exemple (Ouvrir (document), effacer (élément), ...)).

Nous attribuons les opérations de contrôle au contrôleur de dialogue (CD). Par conséquent, le module de gestion de la concurrence doit informer instantanément le CD sur les événements qui surgissent. Le CD informe alors

l'utilisateur et décide s'il est nécessaire d'initier une communication avec les autres sites

#### 4. 4 Contrôleur de dialogue

Le contrôleur de dialogue (CD) s'interpose entre la première et la deuxième couche. Le CD est responsable de la gestion des relations et des échanges entre la couche éditeur et la couche présentation ainsi que les communications avec les autres sites (fig. 1)

### 5. Fonctionnalités de l'éditeur coopératif

Nous proposons ici notre éditeur coopératif (voir [23] pour plus de détails). Il utilise une structure qui est assez flexible pour être adaptée à une large variété de contenus de documents. Dans ce qui suit, nous décrivons d'abord les fonctionnalités concernant l'édition coopérative, nous présenterons ensuite l'utilisation et le traitement de la structure explicite

#### 5. 1 Fonctionnalités de base

Contrairement aux autres approches, la notre repose sur une architecture répliquée et utilise un contrôle distribué basé sur le mécanisme de blocages (voir Table 1)

Editeurs Coopératifs	Données	Réplication	Contrôle de concurrence
CES	Distribuées	Partielle	Transaction
DistEdit	Distribuées	Totale	Tour de Role
GROVE	Distribuées	Totale	Priorités
MACE	Centralisées	Aucune	Blocages
MMM	Centralisées	Aucune	Blocages
MULE	Distribuées	Totale	Blocages
MultimETH	Centralisées	Aucune	Blocages
PREP	Centralisées	Aucune	Blocages
Quilt	Centralisées	Aucune </td <td>Blocages</td>	Blocages
SharedBooks	Centralisées	Aucune	Blocages
Notre système	Distribuées	Totale	Blocages

Table-1: Différentes approches d'édition coopérative

Ainsi, notre système assure les fonctionnalités suivantes :

- La manipulation de documents stockés sous format ASCII. Les éléments graphiques et multimédias ne sont pas supportés. Les documents sont structurés (§ 5.2). La description de la structure du document est enregistrée indépendamment du contenu. Tout comme CES et MultimETH, notre système permet les modifications structurelles sans endommager les informations textuelles. Le contenu du document peut être composé de plusieurs parties qui sont enregistrées dans des fichiers distincts. Les fichiers peuvent être distribués à travers plusieurs sites interconnectés. La structure du document permet la localisation des données et facilite donc leur récupération.
- Les parties du document qui sont utilisées durant une session de travail sont enregistrées localement. Donc un utilisateur retardataire se voit fournir automatiquement la partie entière du document qui l'intéresse. Manifestement, le degré de répllication d'une partie particulière du document peut varier dynamiquement. Elle est égale au nombre d'utilisateurs actifs éditant cette partie.

- Pour accroître les performances de défilement, les accès en lecture sont assurés à tout instant à toutes les parties du document. Cependant, les utilisateurs sont avertis si des parties du document sont bloquées ou modifiées. Ce mécanisme de notification associé aux parties bloquées constitue une aide appréciable pour gérer les accès concurrents.

- De même, que les accès en écriture sont synchronisés aussi bien par l'utilisation de blocage implicite qu'explicite. La structure du document suggère une unité de mise à jour avec une granularité éditable variable. Les utilisateurs sont informés par les changements aussitôt que la modification est terminée.

#### 5. 2 Structure explicite

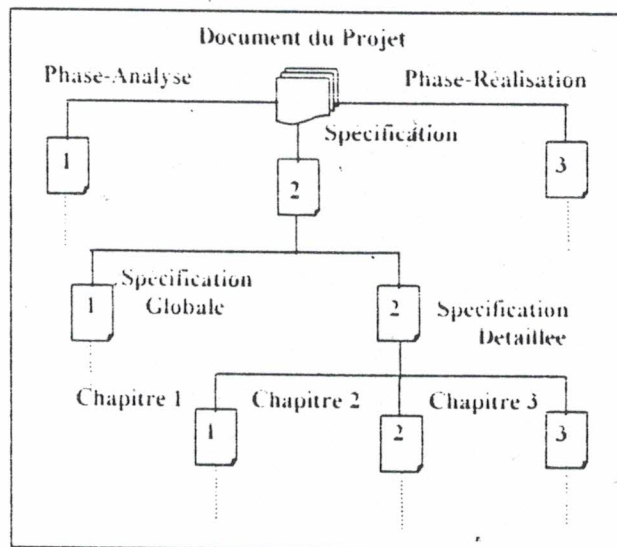


Figure 2: Structure explicite d'un document

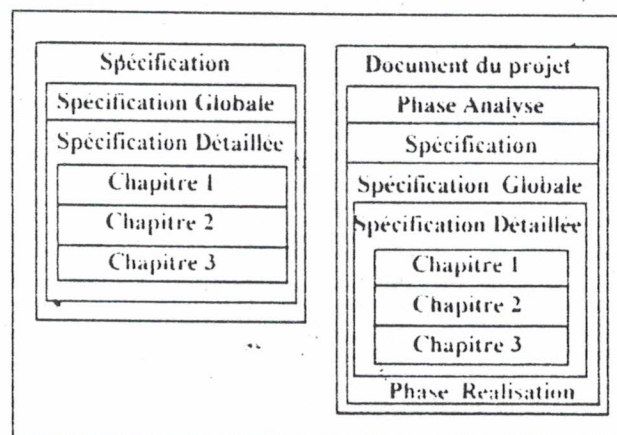


Figure 3: Vues de structure

La structure logique dans notre modèle est inspirée de celle de l'architecture ODA[24]. Cette structure est un arbre à objets, qui sont soit des objets composés, soit des objets contenus (i.e. des contenus des parties du document). Nous associons à chaque objet un numéro qui sert à son identification. Le point de départ de tout processus de recherche est la racine de l'arbre. De même que chaque objet est désigné explicitement par un nom logique attribué par l'utilisateur comme par exemple "Spécification Détaillée" ou "Chapitre 1" sur la figure 2.

Ainsi, l'unité de modification n'est autre que l'objet contenu quelle que soit sa taille. Les objets contenus forment ensemble le contenu global du document, qui est stocké distinctement de la structure.

La figure 3 représente deux structures de vues possibles des nœuds "Document de Projet" et "Specification" de la structure (fig 2). Les modules de l'éditeur décrits (cf. Parag 4) exploitent les vues de structure pour permettre à l'utilisateur de sélectionner des objets à supprimer, des positions pour créer de nouveaux objets ou déplacer des objets existants, et de sélectionner des noms ou des nombres à modifier.

La coordination explicite utilise la sélection d'objets pour la spécification des parties à bloquer ou pour référencer des parties dans des messages destinés aux autres utilisateurs. Elle utilise le marquage des objets pour montrer les parties bloquées. Le module de gestion des versions utilise également le marquage des objets pour montrer l'endroit où les activités précédentes se sont déroulées dans le document.

### 6. Conclusion

Dans cet article nous avons présenté les concepts de base de la conception d'un éditeur coopératif. Nous avons largement montré que notre système permet à plusieurs utilisateurs de collaborer au sein d'un espace de coopération (i.e. l'ensemble des documents et outils d'édition). De même qu'il permet aussi bien aux actions individuelles que collectives de s'articuler sur une tâche commune. L'exploitation de la structure du document a été largement discutée, car elle permet d'assurer la transparence en informant effectivement les utilisateurs sur leurs actions mutuelles.

Nous pouvons aussi montrer que notre approche est facilement applicable à tous les projets de conception qui sont basés sur la création et la manipulation des documents. Pour mieux appréhender le fonctionnement de notre système, nous avons implémenté une première version en C++ avec Windows. L'expérimentation de cette version nous a permis de tester certaines hypothèses que nous avons proposées.

Actuellement, nous consacrons la majorité de nos recherches à l'affinement du travail accompli. Enfin nous envisageons à l'avenir de combiner cet éditeur avec un modèle qui permet de répartir les tâches sur un groupe d'utilisateurs pour l'adapter d'avantage aux besoins des projets de groupes.

### Références

[1]: ELLIS, C.A., GIBBS, S.G., REIN, G.L. "Groupware: Some Issues and Experiences", *Com. ACM*, vol 34, n° 1, (Jan 1991), p. 38-58.

[2]: BANNON, L., SCHMIDT, K., "CSCW - Four Characters Search of a Context", in *Proc of the 1st Euro conf on CSCW (CSCW '89)* (Sep 1989), p. 13-15.

[3]: AFCET, "Enquête sur la pratique de la collective (groupware) en France", *Rapport d'étude* (Sep 1991), 83 pages.

[4]: PALMER J, et FIELDS N. "Guest editors' introduction: computer-supported cooperative work", *Com. H.E.I. Computer*, vol 27, n° 5, (1994), pp. 15-18.

[5]: GREIF, I., SELIGER, R., WEHL, W., "Atomic Data Abstractions in a Distributed Collaborative Editing System", in *Proc ACM symp on principles of programming languages (POPL)*, (Jan 1986), p. 160-172.

[6]: LELAND, M., FISHER, S., KRAUT R., "Collaborative document production using Quilt", *Com. ACM*, (Sep 1988), pp. 200-215.

[7]: LEWIS, B., MANTEI, M., "Shared Books: Collaborative publication Management for an office information system", in *proc of the conf on office information systems*, Palo Alto, CA, (Mar 1988), p. 197-204.

[8]: NEUWIRTH, C., KAUFER, D., CHANDHOK, R., "Issues in the design of computer support for co-authoring and commenting", in *proc of the 3rd int conf on CSCW*, Los Angeles CA, ACM, (Oct 1990), p. 183-195.

[9]: BAECKER, R., NASTOS, D., POSNER L., MAWBY K., "The user-centred iterative design of collaborative writing software", in *proc of the (CSCW'92) conf*, Toronto, Ontario, 31 (Oct 1992).

[10]: BEAUDOUIN-LAFON, M., KARSENTY, A., "Transparency and Awareness in Real-Time Groupware Systems", in *proc of the ACM Symp on User Interface Software and Technology (UIST 92)*, November 1992.

[11]: GREENBERG, S., ROSEMAN, K., WEBSTER, D., "Human and Technical Factors of Distributed Group Drawing Tools: Interacting with Computers", Vol 4, n° 3, (Dec 92), pp. 304-302.

[12]: KNISTER, M., PRAKASH, A., "A Distributed Toolkit for Supporting Multiple Group Editors", in *proc of the 3rd int conf on CSCW*, Los Angeles, CA, Oct 1990, ACM, p. 343-355.

[13]: BIER, E., FREEMAN, S., "MMM: A User Interface Architecture for Shared Editors on a Single Screen", in *proc of the 4th ACM Symp on User Interface Software and Technology*, Hilton Head, SC, Nov 1991, ACM, p. 79-86.

[14]: NEWMAN-WOLFE, R., PELIMHANDIRAM, H., "MACH: A Time-trained Concurrent Editor", in *proc of ACM SIGOIS conf on Organizational Computing Systems*, Atlanta, GA, 1991, ACM, p. 240-254.

[15]: PENDERGAST, M., VOGEL, D., "Design and Implementation of a PC LAN-based Multi-User Text Editor", in *proc of Multi-User Interfaces and Applications conf*, (Sep 1990), p. 195-206.

[16]: LUBICH, H., PLATTNER, B., "A Proposed Model: Functionality Definition for a Collaborative Editing and Computing System", in *proc of Multi-User Interfaces and Applications conf* (Sep 1990), p. 215-232.

[17]: KROWLEY, T., MILAZZO, P., BAKER, O., FORSDICK, H., "MMCconf: An Infrastructure for Building Shared Multimedia Applications", in *proc of the 3rd int conf on CSCW*, Los Angeles, CA, Oct 1990, ACM, p. 329-342.

[18]: HAHN, U., JARKE, M., EHERER, S., KREPLIN, K., "COAUTHOR: A Hypermedia Group Authoring Environment", *Studies in CSCW*, 1991, p. 79-100.

[19]: NUNAMAKER, J., DENIS, A., VALACICH, J., VOGEL, D., "Electronic Meeting Systems to Support Group Work", *Com. ACM*, vol 34, n° 7, (Jul 1991), p. 40-61.

[20]: MARCUS, A., "Human Communications Issues in Advanced Us", *Com ACM*, vol 36, n° 4, (Apr 1993), p. 101-109.

[21]: GRONBAEK, K., HEM, J., MADSEN, O., SLOTH, L., "Hypermedia Systems - A Dexter-based architecture", *Com ACM*, vol 37, n° 2, (Feb 1994), pp. 68-74.

[22]: ATALLAH, S., KANAWATI, R., "Architecture de systèmes pour les collectifs synchrones", *Actes d'HHM 94*.

[23]: ZIDANI, A., "Conception d'un collectif pour l'édition partagée de documents", *Thèse de Magister*, Université de Batna, Juillet 1996.

[24]: ISO, 8613, *Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format*, ISO IS Parties 1,2,4,8, 1988.