

## JamEdit : un outil interactif et coopératif pour l'édition de documents

**Abdelmadjid Zidani\*** — **Mahmoud Boufaïda\*\*** —  
**Mahieddine Djoudi\*\*\***

\* *Institut d'Informatique, Université de Batna (05000), Algérie*  
*zidani@univ-batna.dz*

\*\*

*Institut d'Informatique, Université de Constantine (25000), Algérie*  
*boufaïda@ist.cerist.dz*

\*\*\*

*IRCOM-SIC (UMR 6615 CNRS)*  
*Université de Poitiers - UFR Sciences*  
*Bat. SP2MI - Boulevard Marie et Pierre Curie*  
*BP 30179, 86962 Futuroscope Chasseneuil Cedex*  
*djoudi@sic.sp2mi.univ-poitiers.fr*

---

*RÉSUMÉ. L'objectif de cet article est la présentation de JamEdit, un système d'édition coopérative de documents intégrant les concepts de base de l'édition et de l'interaction de groupe. Il repose sur une architecture en couches qui implique plusieurs objets, prenant en considération la gestion des ressources, l'interaction, la communication et la collaboration entre les utilisateurs. Pour assurer la transparence, nous exploitons la structure du document à l'aide d'un mécanisme de notification d'événements, représentant les actions des participants au sein de l'espace de travail partagé. Dans cet article, nous passons d'abord en revue les principaux travaux dans ce domaine. Nous présentons ensuite les concepts de base et les fonctionnalités du système.*

*ABSTRACT. This paper presents a shared editor system, called JamEdit, taking human factors considerations into account, and embedding concepts that are essential for collaborative editing and group interaction. It is based on oriented layer software architecture, implying many objects to support resource management, interaction, communication and collaboration between users. To achieve group awareness, we explore the document structure with an event notification mechanism representing each participant action into the shared workspace. We first review ongoing research, and then present the system basic concepts and functionalities.*

*MOTS-CLÉS : Collecticiel, Travail coopératif, Edition partagée, Interfaces multiutilisateurs*

*KEY WORDS : Groupware, Cooperative Work, Shared Editing, Multi-user Interfaces.*

---

## 1. Introduction

Les systèmes d'édition ont longtemps eu comme objectif de fournir l'assistance nécessaire à une seule personne pour rédiger des documents. Aujourd'hui, la tendance est plus que jamais orientée vers le principe stipulant que les groupes constituent une base pour toute organisation et permettent d'accomplir des tâches irréalisables par une seule personne. Suivant ce principe, les éditeurs coopératifs tentent de répondre à des besoins de plus en plus pressants des organisations qui souhaitent *réunir virtuellement* [BEL 97] plusieurs personnes avec différents domaines d'expertise autour d'un même document. Par conséquent, ils représentent une réponse technologique pour satisfaire aux besoins des activités collectives d'édition.

Le développement des éditeurs coopératifs a ainsi pu réaliser une avancée considérable ces dernières années. Aussi, nous assistons de plus en plus à l'émergence de plates-formes collectives qui facilitent énormément la tâche des concepteurs. Certaines d'entre elles proposent des outils prêts à l'emploi qui permettent de gérer les processus distribués et leurs communications. Ainsi que des outils de désignation et d'annotation destinés à l'usage des groupes et facilement intégrables. Citons par exemple, GroupKit [ROS 96].

Le processus d'écriture coopérative implique en général deux ou plusieurs personnes qui travaillent ensemble pour produire un document. Il comporte des phases d'écriture et de communication. Ainsi que des périodes d'activités synchrones, où le groupe travaille ensemble au même moment et des périodes d'activités asynchrones, où les participants travaillent à des moments différents. Les premiers travaux du domaine ont considéré ces aspects de manière séparée. Cependant, il semble que l'intégration de ces techniques au sein d'une même application faciliterait bien les choses.

Prenant en considération ces constatations, nous avons développé le prototype d'édition coopérative *JamEdit*<sup>1</sup> qui permet à plusieurs utilisateurs d'éditer simultanément et de façon coordonnée un même document. *JamEdit* est fondé sur deux principaux modes d'interaction à travers lesquels les utilisateurs peuvent travailler individuellement ou collectivement. Ceci est motivé par le souci de ne pas affecter l'aptitude d'écriture individuelle des auteurs comme l'a suggéré Grudin [GRU 94]. Pour développer le support de partage et faciliter le contrôle de la concurrence, le document partagé est décomposé en parties sur lesquelles les participants interviennent avec des rôles bien définis. L'attribution de ces rôles permet aux participants d'écrire, de consulter, d'annoter et de commenter des parties spécifiques. Ce choix découle du processus d'écriture qui est caractérisé par un enchaînement d'écritures, d'annotations, de commentaires et de révisions.

---

<sup>1</sup> Jam est une abréviation du mot arabe *Jamaa* qui signifie : "Groupe".

*JamEdit* repose sur une architecture en couches qui est constituée d'objets, prenant en compte la gestion des ressources, ainsi que les fonctionnalités de communication, de coordination et de partage des informations. Notre approche à base d'objets répliqués sur plusieurs sites distants est guidée tant par les principes structurels du génie logiciel (modularité, réutilisation, extensibilité, etc.), que par la modélisation des interactions entre les participants. Pour gérer les accès concurrents des différents participants sur les parties du document, nous adoptons une politique de *blocages souples*<sup>2</sup> qui ne permet qu'à un seul rédacteur de modifier une partie du document à un instant donné. L'intérêt principal de *JamEdit* concerne la collection et la distribution des événements générés par les interventions des différents participants sur le document partagé. Ceci est facilité par l'exploitation de la structure du document et l'utilisation du mécanisme de notification que nous avons développé. Enfin, l'expérimentation de *JamEdit* constitue un gage quant à sa perception par les utilisateurs potentiels comme un *médium* et non un mécanisme.

L'objectif de cet article est de présenter le système coopératif d'édition de documents *JamEdit*. La section 2 passe en revue les principaux systèmes d'édition partagée et présente les caractéristiques essentielles de *JamEdit*. La section 3 décrit les choix de conception retenus pour *JamEdit*. Nous discutons au passage la problématique liée à la nécessité de considérer l'aspect social impliqué par l'écriture coopérative. La section 4 détaille les différents niveaux que recouvre l'architecture logicielle et décrit les fonctionnalités principales supportées. Nous montrons que la conception de ce système est centrée sur les concepts de partage et d'échange de données pour favoriser la collaboration entre les participants. La section 5 résume l'expérimentation d'une première implémentation sur un réseau local. Finalement, nous concluons en donnant quelques perspectives du travail accompli.

## 2. Edition coopérative de documents

La majorité des documents produits au sein des organisations sont rédigés collectivement. Par conséquent, le développement de supports informatiques pour l'édition coopérative constitue l'un des objectifs les plus importants du collectifiel. L'écriture coopérative est définie selon Ellis [ELL 91] comme un processus au cours duquel les auteurs (i.e. rédacteurs, experts, lecteurs, etc.) interagissent avec différentes expertises durant la création et la révision d'un document commun.

Plusieurs travaux ont déjà porté sur le support de l'écriture coopérative, parmi lesquels nous distinguons : Alliance [DEC 95], Duplex [PAC 94], CES [GRE 92], GROVE [ELL 91], DocMan [BAE 96], SASSE [BAE 93], GroupDesign [KAR 94], IRIS [KOC 95], PREP [NEU 94]. Le point commun entre ces systèmes est qu'ils permettent à plusieurs participants d'éditer de manière synchrone ou asynchrone un même document (texte, graphiques, etc.).

---

<sup>2</sup> Les participants peuvent consulter une partie verrouillée en attendant de recevoir les mises à jour.

L'éditeur CES est l'un des premiers outils qui a fait la distinction entre l'information textuelle et l'information structurée. Le texte est décomposé en sections qui sont propres à chaque rédacteur. Tout comme CES, Alliance et IRIS définissent un support de coopération basé sur une structure composée de sections et d'unités textuelles. Par contre, l'éditeur GROVE traite des informations non structurées. GROVE est caractérisé par une granularité de partage fine et supporte des mises à jour en temps réel. Il utilise un algorithme pour la gestion des accès concurrents. DocMan utilise le schéma de blocages auquel il associe un mécanisme de notifications pour informer les utilisateurs sur l'état de chaque partie du document. Le contrôle répliqué utilisé par Alliance permet à un utilisateur de verrouiller un fragment de texte pour travailler sans interférence avec les autres. Avec l'éditeur Duplex, les auteurs ont la possibilité de contrôler la granularité pour partager une section ou une ligne de texte.

La conception de *JamEdit* s'inspire en grande partie des travaux décrits précédemment. Son objectif principal est de fournir un support d'interaction suffisamment flexible qui impose le moins de contraintes possible aux participants. Ainsi, les phases du processus d'écriture coopérative (planification, rédaction, commentaire, révision, discussion et annotation) prennent place naturellement sans qu'elles ne soient imposées, du fait que l'interaction repose sur un protocole associant les aspects social et technique. De plus, *JamEdit* intègre les styles de travail synchrone et asynchrone pour tenir compte de la caractéristique de l'écriture coopérative qui implique le travail collectif et le travail individuel, notamment en définissant plusieurs modes de coopération entre les différents participants. Ainsi, comme l'éditeur Alliance, *JamEdit* n'impose pas aux utilisateurs de travailler en même temps. Chaque rédacteur décide de valider ses modifications pour les rendre accessibles aux autres. Les participants peuvent décider de voir immédiatement ces nouvelles contributions ou de les ignorer.

*JamEdit* repose sur une architecture en couches dupliquée sur plusieurs sites. Par conséquent, la fonction de contrôle est assurée localement au niveau de chaque site et les sites communiquent entre eux pour préserver la cohérence des données partagées. Les expériences précédentes ont montré que les auteurs doivent avoir la possibilité d'ajuster la granularité et de choisir le moment pour diffuser leurs contributions. Ainsi comme l'éditeur Duplex, *JamEdit* est caractérisé par une granularité variable déterminée par les utilisateurs.

La particularité de *JamEdit* est que le processus d'interaction avec l'utilisateur est concentré sur la structure du document. La visualisation de la structure permet aux participants d'interagir avec les objets du document pour prendre connaissance de leurs actions mutuelles et de tout ce qui concerne le projet commun tandis qu'au niveau du contenu du document, nous ne diffusons aucune information événementielle, à l'exception de la coloration des fragments textuels (optionnelle) pour informer explicitement les utilisateurs de leurs auteurs respectifs. Ceci évite de surcharger le contenu du document et de risquer ainsi de perturber la concentration des auteurs. Un cas typique d'utilisation de *JamEdit* conduit un participant à

visualiser la structure pour s'informer de ce qui se passe. Il accède ensuite au contenu de sa partie pour travailler. Pendant ce temps, des notifications lui sont adressées (conformément à ses choix) à propos des événements qui se produisent sur le document partagé.

L'interaction avec la structure permet aux participants de réagir en fonction des événements qui surgissent. Ainsi, le support de la rétroaction de groupe fournie par *JamEdit* informe les participants sur les événements courants et les événements historiques. Ils pourront savoir quels changements ont été opérés en leur absence, à quel moment, sur quelles parties, et les auteurs impliqués. Nous adoptons l'approche utilisée par DocMan basée sur l'association des notifications au schéma de blocages. Ceci évite aux participants de voir des fragments rédigés apparaître instantanément et permet de masquer les hésitations éventuelles des auteurs. Enfin, *JamEdit* intègre un outil de messagerie pour faciliter la négociation et la collaboration entre les participants. Ceci leur permet de partager leurs idées à propos de leur tâche commune.

### **3. Aspect social de l'écriture coopérative**

Bien qu'en perpétuelle évolution, les systèmes d'édition coopérative butent toujours sur un obstacle de taille. Il s'agit de l'absence d'une étude sociale explicite qui soit capable de cerner effectivement les besoins d'un groupe engagé dans un processus de rédaction. Les membres de ces groupes sont sujets à un enchaînement complexe d'influences mutuelles dû notamment à leurs différences interpersonnelles (compétences, personnalités, etc.). La difficulté générée par la modélisation du processus d'écriture provient du fait que certaines de ses activités ne peuvent être ni observées ni décrites, comme par exemple la part importante de ce processus que les auteurs consacrent à l'effort mental. Chaque groupe est donc caractérisé par sa spécificité, notamment ses stratégies d'écriture qui peuvent évoluer sur la base des interactions entre les différents participants [POS 92].

Il est pratiquement impossible de développer un support informatique tenant compte de tous ces aspects. Néanmoins, les expériences précédentes [BEC 93] nous fournissent des informations suffisantes pour nous permettre de dégager quelques caractéristiques utiles au développement des outils de support les plus appropriés. L'expérimentation de ces outils par la suite, dévoilera les difficultés à surmonter ainsi que les perspectives à suivre.

La conception de *JamEdit* est centrée sur des besoins des groupes. Ceci est délicat à réaliser car les participants génèrent souvent des exigences diverses. Pour satisfaire aussi les exigences individuelles dans un environnement coopératif, nous avons décidé que *JamEdit* devrait être suffisamment flexible pour permettre aux utilisateurs de configurer leur environnement, notamment en fournissant des fonctionnalités paramétrées.

*JamEdit* est destiné à aider les utilisateurs à collaborer et à partager la responsabilité d'un document. Kraut [KRA 92] insiste sur la nécessité pour un support effectif de tenir compte de la complexité procédurale, intellectuelle et sociale du processus d'écriture coopérative. La diversité d'opinions au sein du groupe génère une activité intellectuelle énorme. Le support le plus approprié requiert des outils de communication. Par conséquent, *JamEdit* intègre un outil de messagerie afin de faciliter la négociation et aider les participants à se comprendre pour aboutir rapidement à un consensus lors de situations conflictuelles.

Posner [POS 92] a mené une étude basée sur des interviews et la collecte d'informations auprès de plusieurs groupes impliqués dans des projets de rédaction coopérative. Il en déduit que l'un des points communs à ces groupes est que les participants ont assumé les rôles de rédacteur, de lecteur ou de superviseur. Nous avons retenu ces rôles pour *JamEdit*, auxquels nous avons ajouté ceux de commentateur et d'annotateur, pour accroître la clarté et faciliter la lisibilité du contenu du document. Ainsi, nous avons défini pour les participants les rôles suivants :

*Chef de projet* — Il a pour tâches d'organiser le travail, d'attribuer les rôles selon les compétences et de les changer, de coordonner le travail, de vérifier que les délais sont bien respectés, etc.

*Responsables* — Ce sont des participants désignés par le chef de projet pour superviser l'avancement du travail sur des parties spécifiques du document. Leurs prérogatives sont restreintes aux parties auxquelles ils sont affectés.

*Rédacteur & Lecteur* — Un rédacteur est un participant qui peut modifier les parties auxquelles il est affecté, tandis que le rôle de lecteur ne permet que la consultation.

*Commentateur & Annotateur* — Ces rôles permettent respectivement aux participants associés de commenter et d'annoter des parties développées par les autres.

### **3.1. Modes de coopération**

La définition d'un modèle d'interaction coopératif efficace repose sur le choix de modes d'interaction naturels et sans contraintes. Le processus d'écriture coopérative est caractérisé par un enchaînement de phases au cours desquelles les participants travaillent individuellement ou collectivement. De même, pour favoriser l'échange du savoir faire, il est nécessaire de permettre à un participant de s'inspirer du travail de personnes plus expérimentées. Par conséquent, nous avons défini pour *JamEdit* les modes de coopération suivants :

*Responsabilité individuelle* — Le document est décomposé en parties distinctes, chacune d'elles est manipulée par au moins un participant. Plusieurs participants peuvent obtenir le rôle de rédacteur potentiel sur une même partie, mais un seul d'entre eux aura la permission de modifier la partie à un instant donné.

*Echange dynamique* — Pendant une session de travail, les participants peuvent échanger dynamiquement des parties. Un participant désirant intervenir sur une partie qui est verrouillée par quelqu'un d'autre peut lui envoyer un message afin qu'il la libère.

*Version alternative* — Les différents acteurs peuvent développer des versions alternatives d'une même partie, qui pourront alors être fusionnées ultérieurement. Une même partie qui est développée en parallèle est explicitement signalée au niveau de l'interface (figure 3).

*Responsabilité collective* — Deux ou plusieurs personnes peuvent travailler sur la même partie au même moment en mode *WYSIWIS relâché* [GRE 96]. Ce mode constitue un moyen de partage de la vue plus naturel et plus flexible que le mode *WYSIWIS strict*<sup>3</sup>. Les opérations exécutées par l'un des rédacteurs sont immédiatement rendues visibles aux autres participants partageant la même vue. Leur consentement mutuel déclenchera l'opération d'enregistrement des données.

Les trois premiers modes sont des modes de coopération typiquement asynchrones. Le troisième s'inspire de la démarche réelle suivante : *réfléchissons séparément à la question puis comparons nos résultats*. Le quatrième est un mode synchrone. L'objectif visé à travers ce mode n'est pas d'offrir le mode synchrone en lui-même, qui reste très délicat à réaliser, mais de permettre à un nombre relativement réduit de participants (tels que les responsables hiérarchiques), lorsque le projet arrive à terme, de mettre au point la version finale du document.

Ainsi, à travers ces modes, les participants accèdent aux parties du document et la coopération prend place à travers des annotations et des commentaires liés aux parties développées par les autres. La spécification du mode à appliquer sur une partie du document est décidée de façon collective. Cependant, l'exécution de cette opération est réservée au chef de projet et au responsable de cette partie.

#### 4. Présentation de JamEdit

Le collecticiel est caractérisé par sa pluridisciplinarité qui implique un large éventail de compétences. Plusieurs classifications visant à faciliter le développement des collecticiels ont vu le jour ces dernières années. Nous distinguons parmi ces dernières la matrice *Espace-Temps* [ELL 91], la *matrice 3x3* de Grudin et l'*Espace-Problème* [SAL 94]. La caractéristique principale des deux premières classifications est la séparation entre collecticiels synchrones et asynchrones. Cependant, l'évolution rapide des recherches a engendré des applications supportant des interactions à la fois synchrones et asynchrones. De plus, l'intégration d'autres outils comme la messagerie au sein d'une même application rend sa classification difficile. Par exemple, deux utilisateurs peuvent interagir en mode synchrone suite à un échange asynchrone de messages. Par conséquent, nous avons utilisé la

<sup>3</sup> Contrairement au WYSIWIS strict, le WYSIWIS relâché permet aux participants de ne pas avoir exactement la même vue sur l'espace de travail partagé.

classification de Salber et Coutaz (figure 1), qui donne une vue globale du système à développer et permet de dégager ses caractéristiques.

L'espace-problème comprend huit dimensions permettant d'identifier l'ensemble de concepts inhérents à un domaine d'intérêt particulier. La première dimension permet de recenser les *services* devant être assurés par le système. La seconde exprime la *responsabilité* d'assurer un service particulier dans un environnement coopératif. La *dynamacité* couvre la capacité du système à supporter les changements durant une session de travail. La quatrième dimension permet d'envisager la conception du système en termes de *niveaux d'abstraction*, facilitant ainsi l'identification des niveaux pour fournir un service particulier. La *granularité* concerne de façon particulière la taille minimale de l'information qui peut être partagée ou mise à jour en temps réel. Le *parallélisme* permet d'exprimer les relations temporelles entre les différentes activités réalisées. La *distribution* des données et des traitements pour les collecticiels est fondée sur le degré de dynamacité des interactions des utilisateurs. Enfin, la huitième dimension couvre la description des outils logiciels et matériels mis en jeu.

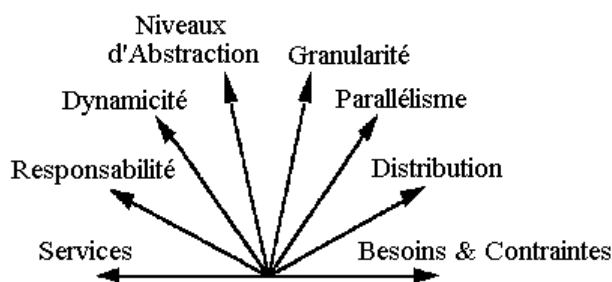


Figure 1. Espace-Problème des collecticiels [SAL 94]

Cet espace constitue un cadre qui nous a permis de mener une analyse objective des systèmes existants pour mieux structurer notre espace de conception. Il a aussi guidé nos choix de façon précise, notamment en raisonnant non seulement sur les différentes dimensions de façon singulière, mais encore en les couplant pour approfondir le raisonnement. Par exemple, en couplant les dimensions *dynamacité* et *responsabilité*, nous pouvons exprimer que la responsabilité sur la modification est échangée dynamiquement entre les utilisateurs.

Dans la suite de cette section, nous présentons l'architecture logicielle de *JamEdit*. Nous décrivons ensuite les solutions techniques adoptées ainsi que les fonctionnalités de base supportées.

#### 4.1. Architecture logicielle

*JamEdit* repose sur une architecture en couches (figure 2). Cette architecture s'appuie sur le principe de séparation entre le composant de présentation et le noyau fonctionnel. Nous distinguons la couche-*Interface* qui est chargée de gérer le



dialogue entre l'utilisateur et l'éditeur. La couche-*Traitements* est responsable des tâches d'édition, de coordination et de gestion des versions. La couche-*Données* rassemble toutes les informations nécessaires à la conduite d'un projet coopératif (contenu du document, informations associées à la structure, à l'historique et aux participants). Le *contrôleur de dialogue* permet de réaliser le couplage entre l'interaction de l'utilisateur et l'application. Il intercepte d'une part les actions réalisées localement par le participant au niveau de l'interface afin de les diffuser aux autres sites, et récupère d'autre part les modifications distantes via le réseau pour les délivrer à ce dernier.

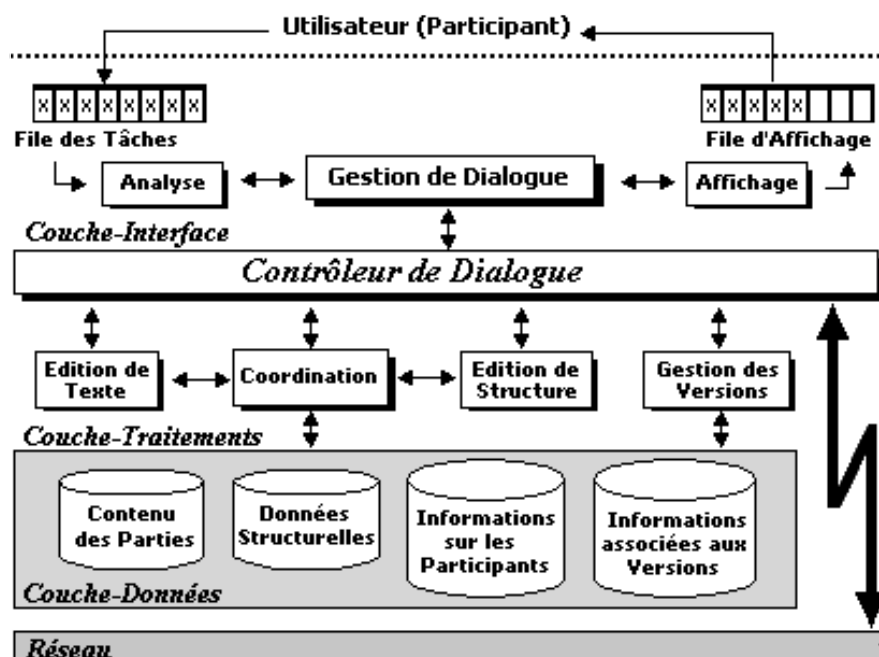


Figure 2. Architecture simplifiée de JamEdit

#### 4.1.1. Coordination

L'efficacité d'un groupe repose essentiellement sur la capacité des participants à coordonner leurs actions mutuelles [ABH 96]. Pour les collecticiels, la coordination est une partie intégrante de l'assistance fournie au groupe. Les objets de coordination sont chargés d'assurer cette tâche et leurs résultats sont répercutés au niveau de l'interface à travers des visualisations appropriées.

Le contrôle des accès concurrents sur la structure logique et les parties du document est basé sur le concept de blocages souples. Par conséquent, l'accès en lecture à une partie verrouillée est autorisé et les participants sont notifiés qu'un autre rédacteur est en train de travailler sur cette dernière. En attendant d'intervenir,

ils se contenteront de consulter la version courante en recevant les modifications distantes à chaque fois que le rédacteur associé les enregistre. Ceci permet une certaine rétroaction au cours du processus de collaboration. Les utilisateurs peuvent échanger des messages pour coordonner leur travail, entamer des négociations ou différer leurs interventions.

La structure logique du document évolue de la même manière que le contenu des parties. Les deux composants d'édition du contenu et de représentation de la structure s'appuient sur le composant de coordination pour gérer les interventions concurrentes sur les données structurelles et textuelles partagées. Le composant de coordination (figure 2) est informé régulièrement par le contrôleur de dialogue des événements qui se produisent au niveau des autres sites. Par conséquent, avant d'autoriser la réalisation d'une opération sur les données, il doit accomplir un certain nombre de contrôles (droits d'accès, priorités, état des parties manipulées : libres ou bloquées, etc.) et avertir les participants concernés. Par exemple, la modification de la structure nécessite une notification préalable de tous les participants travaillant sur les parties à déplacer ou détruire, et éventuellement, de ceux qui sont désignés pour travailler sur les nouvelles parties créées. Les effets des opérations de coordination apparaissent explicitement à l'utilisateur au niveau de l'interface. Le blocage de parties du document et leur libération en sont un bon exemple.

#### 4.1.2. *Gestion des versions*

La gestion des versions d'un document multi-auteur est une tâche délicate et coûteuse, puisque le nombre de copies peut croître rapidement au cours du processus de collaboration. Le participant risque alors d'être débordé et désorienté. Pour éviter ce problème, *JamEdit* se contente de gérer les différentes versions du document pour chaque session de travail, de manière distincte. Ainsi, au démarrage d'une session de travail, il ne sera gardé que la dernière version de la session précédente comme version de départ pour la session courante. Cependant, les participants peuvent enregistrer localement les versions qui les intéressent au cours d'une session de travail. Un numéro de session est alors associé à chacune des versions stockée localement. Ils peuvent également effacer les versions inutiles afin de limiter leur nombre.

*JamEdit* associe à la version de chaque partie spécifique un numéro unique et une donnée temporelle qui correspond à l'instant où les mises à jour associées ont été enregistrées par leur rédacteur. Par conséquent, une modification apportée à une partie du document n'écrase jamais son contenu précédent, mais une nouvelle version est créée à l'instant de son enregistrement. Le plus grand numéro correspondant à la version précédente de cette partie est incrémenté d'une unité et associé à la nouvelle version. Ainsi, aucune donnée ne risque d'être perdue accidentellement à travers les interventions concurrentes des participants.

Au cours d'une session de travail, les participants peuvent consulter la structure du document pour s'informer sur les parties qui les intéressent. L'utilisation des

possibilités d'affichage de *JamEdit* leur permet de distinguer les différentes versions des parties sélectionnées. Ils peuvent alors choisir l'une d'elles afin de consulter son contenu.

#### 4.1.3. *Contrôleur de dialogue*

Le contrôleur de dialogue se situe entre les couches *Traitements* et *Interface*. Il est responsable de la gestion des relations et des échanges entre ces deux couches. Il est aussi chargé d'intercepter les événements générés par l'utilisateur pour les communiquer aux autres sites. Les communications accomplies entre les différents sites sont assimilées à des événements. Pour transmettre un événement, le contrôleur émetteur construit le message associé, spécifie les contrôleurs récepteurs et le leur délivre. Nous distinguons deux principaux types d'événements, les événements internes du système qui sont générés par les traitements et les événements générés par le participant au niveau de l'interface.

Nous distinguons aussi deux types d'opérations exécutées par les participants. Les opérations locales qui ne nécessitent aucune coordination, par exemple l'accès à une partie bloquée affiche à l'utilisateur une fenêtre de travail qu'il exploitera pour rester actif, en attendant d'intervenir sur la partie verrouillée. L'exécution de ses tâches d'édition dans ce cas, est considérée comme une opération locale. Les opérations partagées par contre nécessitent de coordonner le travail avec les autres sites. Le contrôleur de dialogue associé communique avec les autres sites et attend la confirmation avant d'autoriser l'exécution d'une opération partagée. Par exemple, un rédacteur qui sollicite le droit de blocage sur une partie libre, ne pourra intervenir que si tous les autres sites confirment qu'ils ont modifié l'état de cette dernière. Il continuera alors à rédiger son texte localement sur sa fenêtre de travail. Ceci préserve la cohérence des données distribuées et permet le travail en mode déconnecté en cas de défaillances matérielles ou de pannes de transmissions.

Le contrôleur de dialogue récupère les traces des actions réalisées sur les parties du document sous forme d'événements. Chaque participant peut exécuter localement ses tâches de façon totalement autonome, jusqu'à ce qu'il décide d'invoquer une opération partagée. Le contrôleur de dialogue se charge d'établir dans ce cas des communications avec les autres sites pour coordonner le travail.

Au niveau local, nous attribuons au contrôleur de dialogue toutes les opérations de contrôle (déclenchement et interruption des traitements, etc.). L'algorithme suivant traduit une réaction simplifiée du contrôleur de dialogue à un événement généré par l'utilisateur :

```

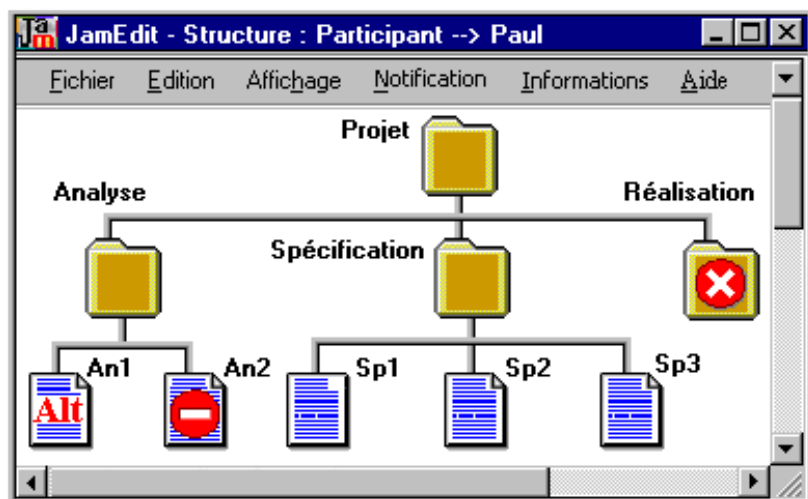
A l'arrivée d'un événement { * Dans la file des tâches * }
  - Consulter les résultats de l'Analyseur
Si la requête de l'utilisateur nécessite une réponse immédiate
Alors - Invoquer le composant d'Affichage          Fin-SI
Si l'opération à exécuter est locale

```

Alors - Lancer l'exécution de l'opération localement  
 Sinon { \* Opération nécessitant la communication avec les autres sites \* }  
 - Déterminer les sites à avertir et initier les communications  
 - Lancer l'exécution de l'opération localement Fin-SI

#### 4.1.4. Structure logique du document

Contrairement à un éditeur mono-utilisateur, l'existence d'autres utilisateurs qui travaillent simultanément sur le même document nécessite d'avoir une vue logique commune sur la structure globale du document. La décomposition d'un document en parties auxquelles seront affectés des participants avec des rôles bien déterminés doit apparaître explicitement au niveau de l'interface.



**Figure 3.** Visualisation de la structure logique du document

De manière similaire au contenu des parties du document, sa structure évolue. Les interventions concurrentes sont gérées par le mécanisme de blocage. Cependant, l'exécution des opérations associées à la structure est restreinte au chef de projet et aux responsables. Ainsi, pour déplacer une partie, la détruire ou en créer une nouvelle, le participant autorisé doit solliciter le droit de blocage avant d'intervenir. Des notifications alertant les participants sont alors envoyées et la nouvelle version est automatiquement délivrée à tous les participants actifs.

Les informations structurelles correspondent à la dernière version de la structure logique du document. Celle-ci apparaît au niveau de l'interface comme une hiérarchie à plusieurs niveaux (telle que la structure d'un document avec des chapitres, sections, paragraphes, appendices, notes, figures, tables, etc.). Cette structure est visualisée et manipulée par les participants.

La structure interne est organisée comme un arbre d'objets, objets composés ou objets atomiques (i.e. le contenu des parties du document). Nous associons à chaque objet une paire de numéros (numéro d'ordre, niveau). Le point de départ de tout processus de recherche est la racine de l'arbre. Au niveau de l'interface, chaque objet apparaît avec le nom logique que lui a attribué son créateur, comme par exemple : *Analyse ou Spécification* (figure 3). La figure 3 montre la structure globale du document *Projet* avec ses différentes parties. Pour réduire les charges cognitives des utilisateurs, nous associons aux parties visualisées des icônes<sup>4</sup>. Ceci permet aux participants de déduire intuitivement leurs états respectifs. Par exemple, l'icône sur la partie *Réalisation* indique que son développement n'intervient pas pour le moment. La partie *An1* est développée en versions alternatives, tandis que *An2* est bloquée par un rédacteur. Enfin, les composants de la partie *Spécification* sont à l'état libre, *Sp1* est fermé alors que *Sp2* et *Sp3* sont ouverts en lecture.

## 4.2. Fonctionnalités de JamEdit

### 4.2.1. Fonctionnalités associées à la structure

La structure logique du document décrit le document en cours d'édition tel qu'il est vu par les participants. Du point de vue technique, la présentation de la structure est essentiellement utilisée pour identifier des parties ou des positions dans le document. Son exploitation dans un environnement coopératif permet de coordonner et de synchroniser les actions des participants.

L'existence de cette structure suggère naturellement un ensemble d'opérations pour l'exploiter et la manipuler. Les opérations de coordination exploitent la sélection d'objets pour la spécification des parties à bloquer ou pour référencer des parties dans des messages destinés aux autres participants. Elles utilisent des marques pour désigner les parties bloquées. Les traitements associés à la gestion des versions utilisent également le marquage des objets du document pour distinguer les différentes versions. Ceci permet aux participants de sélectionner une version spécifique pour consulter son contenu. La visualisation de la structure logique permet aussi aux participants de sélectionner les objets à supprimer, à déplacer ou à renommer.

D'autres opérations pour le support de la collaboration sont également définies comme par exemple, la possibilité illustrée par la figure 4. Le participant *Paul* a sélectionné le composant *Analyse* pour s'informer sur tous les acteurs qui y sont affectés. Il peut ainsi consulter pour chacun d'eux son identité, son rôle, son expérience, son adresse, etc. Pour localiser l'un de ses collègues, il peut vérifier s'il est présent en session, l'instant de son entrée, ainsi que ses activités courantes.

---

<sup>4</sup> Donner à l'environnement la capacité de représenter et de communiquer les connaissances est fortement requis dans le contexte des applications de groupes.



**Figure 4.** Informations sur les participants affectés à une partie spécifique

#### 4.2.2. Fonctionnalités de base

Notre approche s'appuie sur la communication entre les participants pour faciliter la collaboration. *JamEdit* intègre un mécanisme pour le support des échanges de messages entre les participants. Ainsi, ceux-ci peuvent combiner la communication et la collaboration au sein d'une même application. Nous pensons que cette démarche constitue un moyen plus efficace pour l'introduction naturelle des règles sociales, que d'imposer des contraintes au niveau du logiciel. Ainsi, *JamEdit* assure les fonctionnalités suivantes :

- Au départ du projet, le chef de projet décompose le document en parties et spécifie pour chacune d'elles les participants et leurs rôles respectifs, ainsi que le mode de coopération associé. Les différents participants sont alors notifiés et peuvent échanger des messages pour discuter de leurs objectifs communs, leurs rôles, etc. Ainsi, le travail peut démarrer et chaque rédacteur sollicitera le droit d'écriture sur sa partie, qui lui sera accordé si elle est libre. Autrement, il établit un engagement afin de recevoir automatiquement les modifications (section 4.2.3). L'entrée d'un membre du groupe en session de travail et son départ sont explicitement signalés aux autres. Par conséquent, les participants connaissent à tout instant la liste des personnes présentes en session, qu'ils peuvent solliciter dans le contexte de leur activité courante.

- Au cours d'une session de travail, les participants peuvent échanger dynamiquement des messages pour discuter de l'état d'avancement du travail ou négocier le blocage des parties du document. L'émetteur rédige son message,

spécifie le nom du destinataire et le contrôleur de dialogue se charge de le transmettre au site concerné.

- Les informations structurelles et textuelles sont complètement indépendantes et leurs traitements sont séparés. Ceci nous permet de déplacer des parties du document sans endommager leur contenu.

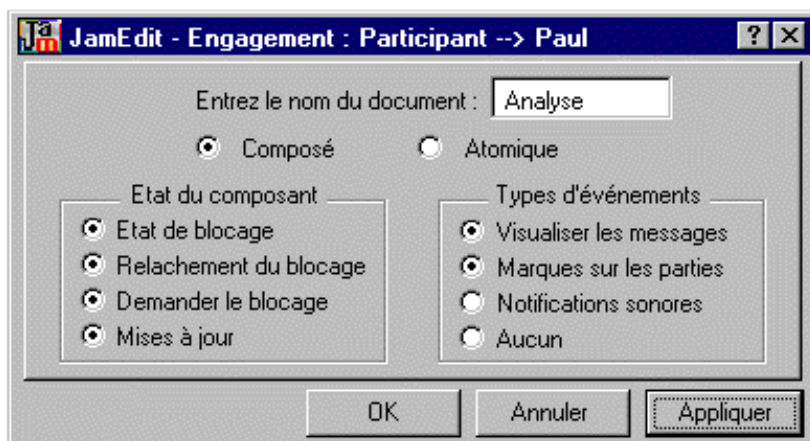
- Au cours d'une session de travail, plusieurs parties sont utilisées par les participants. Le contenu de ces parties est alors enregistré localement. Par conséquent, une même partie qui est utilisée par plusieurs participants est répliquée sur chacun des sites associés. A travers les trois premiers modes discutés précédemment (section 3.1), chaque participant peut verrouiller la partie qui l'intéresse pour travailler. Il enregistre à chaque fois ses mises à jour pour forcer leur apparition sur les écrans des autres. En mode synchrone, les modifications apparaissent instantanément chez tous les participants qui partagent la même vue sur l'espace de travail.

- Au cours d'une session de travail, les participants peuvent revoir les étapes précédentes du travail, notamment en visualisant les modifications successives qui ont été réalisées. Chaque participant peut ainsi enregistrer localement les versions qui l'intéressent. Ceci présente un double avantage : permettre aux participants d'avoir des regards distincts sur le passé et assurer une intégration naturelle des nouveaux participants ou des retardataires.

#### 4.2.3. *Notification des événements*

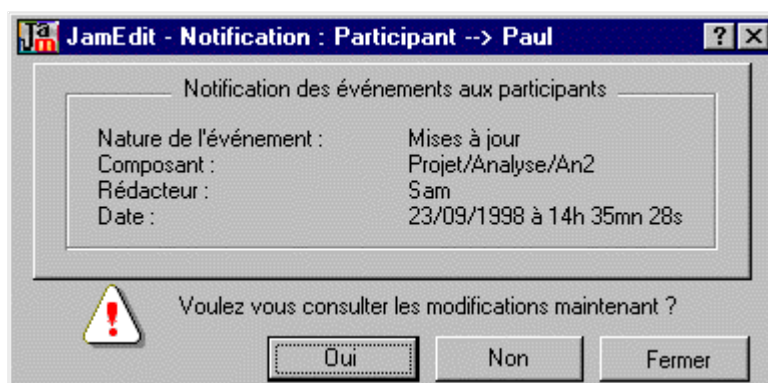
L'exploitation du concept de notification dans le contexte du travail coopératif est une approche prometteuse [WII 91]. Par conséquent, nous avons développé un mécanisme de notification d'événements pour le support de la conscience collective et de la coordination. Le processus de gestion des événements est à la charge du contrôleur de dialogue (figure 2). Comme le contrôleur encapsule les données par les traitements qu'il supervise, la notification d'événements liés aux différentes parties du document se trouve ainsi facilitée. Chaque participant peut demander à être informé à chaque fois qu'un événement se produit au sein de l'espace partagé. Par exemple, il peut établir un engagement afin de recevoir les modifications apportées à une partie spécifique du document, qui lui seront alors automatiquement délivrées. Les engagements sont établis manuellement par les participants (figure 5).

La figure 5 illustre les possibilités offertes au participant *Paul* pour configurer les événements dont il souhaite être notifié. Par exemple, en ce qui concerne la partie *Analyse*, il spécifie au système qu'il souhaite être notifié à chaque fois qu'elle est bloquée ou libérée. Il sollicite également le droit de blocage sur cette partie ainsi que les modifications réalisées par les autres auteurs. *Paul* peut également spécifier la manière selon laquelle il désire être informé par des marques sur les objets, des messages à l'écran ou des messages sonores. Enfin, il peut être concentré et décider, pour ne pas être perturbé, de fermer son espace de travail. De cette manière, nous permettons aux participants de contrôler eux-mêmes le degré d'ouverture de leur espace de travail.



**Figure 5.** Engagements des participants

A chaque fois qu'un rédacteur enregistre ses modifications, il force leur apparition sur les écrans des autres. Par exemple, sur la figure 6, le participant *Paul* est notifié de l'arrivée de modifications associées à la partie *An2* conformément à son engagement de la figure 5.



**Figure 6.** Notification des événements

#### 4.2.4. Rétroaction de groupe

D'un point de vue strictement coopératif, nous accordons une importance extrême au concept de *conscience collective* [DOU 92] [GUT 96]. Pour développer le support informatique approprié, nous avons dégagé les principaux facteurs concernés par la transparence au sein de l'espace de travail. La table 1 résume les éléments considérés ainsi que les questions que peuvent se poser les participants.



| <b>Facteurs</b>       | <b>Interrogations des participants</b>      |
|-----------------------|---|
| <i>Identification</i> | <i>Qui participe à l'activité ?</i>         |
| <i>Lieu</i>           | <i>Où est-il ?</i>                          |
| <i>Présence</i>       | <i>Est-il présent en session ?</i>          |
| <i>Actions</i>        | <i>Que fait-il ?</i>                        |
| <i>Rôle</i>           | <i>Avec quel rôle intervient-il ?</i>       |
| <i>Compétences</i>    | <i>Que peut-il faire ?</i>                  |
| <i>Modifications</i>  | <i>Qu'est ce qu'il a modifié ?</i>          |
| <i>Parties</i>        | <i>Sur quelles parties travaille t-il ?</i> |
| <i>Collaboration</i>  | <i>Que pouvons nous faire ensemble ?</i>    |

**Table 1.** Facteurs concernés par la transparence au sein de l'espace de travail

*JamEdit* exploite un mécanisme de notification qui lui permet de diffuser aux participants les différents événements qui se produisent au sein de l'espace de travail partagé. Au cours du processus de collaboration, les participants prennent alors connaissance de “*Qui fait quoi, Où et Quand ?*”. C’est ainsi que *JamEdit* peut fournir à chaque participant des informations concernant ses collègues (rôle, identité, compétences, etc.), les interactions au sein de l'espace de travail (activités courantes, parties manipulées, modifications réalisées, etc.), ainsi que les états de chaque partie du document (bloquée, libre, rédacteurs manipulant cette partie, instant de blocage, etc.). Les échanges de messages permettent aux participants de discuter de leurs tâches communes, par exemple pour comprendre ce qu'ils peuvent faire ensemble, la stratégie convenable, etc.

De plus, nous rendons *perceptible*<sup>5</sup> à chaque participant l'existence des autres au sein de l'espace de travail partagé [BUX 92]. Cependant, la perception mutuelle des participants dépend du mode de coopération appliqué. Par exemple, les actions de chaque participant sont immédiatement propagées vers les autres dans le cas des interactions synchrones.

La figure 7 illustre un exemple de gestion de la rétroaction de groupe. Nous distinguons les fenêtres appartenant aux participants *Paul* (7.a) et *Sam* (7.b) qui interagissent à travers le mode synchrone. Sur la figure 7.a, *Paul*, qui détient le droit de blocage sur la partie *An1*, construit sa figure (“+” en haut à droite de la fenêtre). Il constate la présence d'un autre participant qui consulte le texte (curseur noir en bas à droite). Il utilise une option de la fonction d’affichage pour connaître l'identité du propriétaire du curseur noir. Il est explicitement informé que c'est le participant *Sam*. Ainsi, la communication est supportée de façon efficace et en ne mobilisant que de petits espaces sur l'écran [GUT 98]. De cette manière, nous permettrons aux participants de se *rencontrer par hasard*<sup>6</sup> au sein de l'espace de travail. Un dialogue

<sup>5</sup> Concept de Téléprésence qui consiste à transmettre les informations dynamiques qui surviennent entre les participants au sein de l'espace de travail partagé.

<sup>6</sup> Rencontrer un partenaire par hasard génère une opportunité naturelle pour l'interaction et la collaboration.

peut alors prendre place et la collaboration entre en jeu naturellement. *Paul* décide, par exemple, d'envoyer un message à *Sam* pour lui demander d'intervenir, et ce message apparaît à droite de sa fenêtre (figure 7.b).

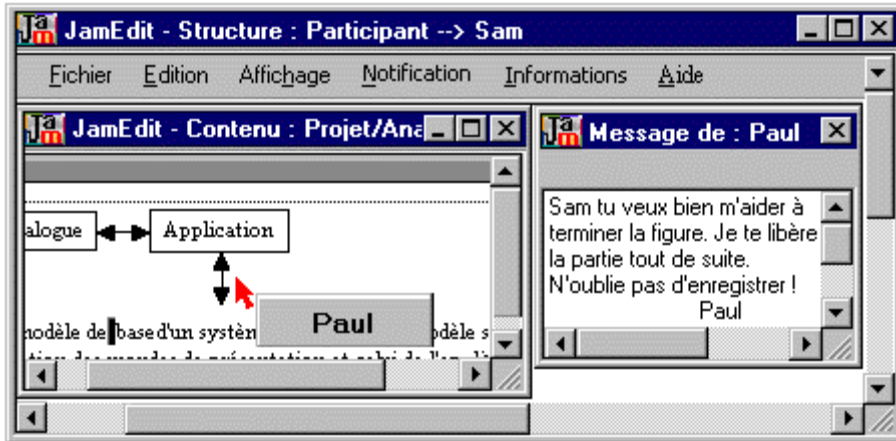
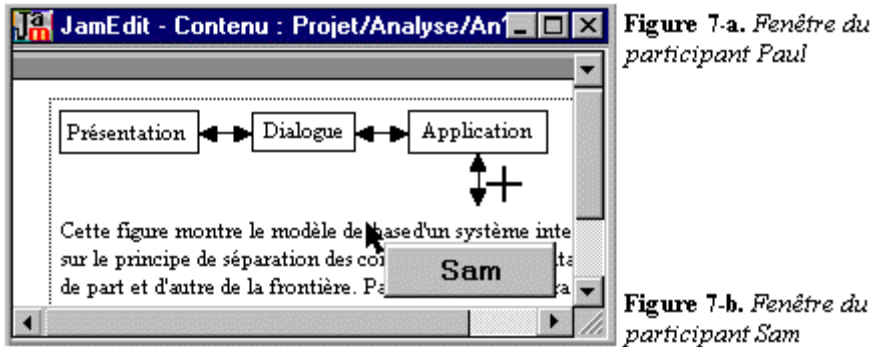


Figure 7. Perception mutuelle des participants au sein de l'espace partagé

Ainsi, en exploitant cette forme simple de communication gestuelle, nous permettons non seulement aux participants de se rencontrer fortuitement, mais encore d'être en position idéale pour anticiper leurs intentions mutuelles. Une autre option d'affichage est également utilisée, elle exploite la *coloration* des fragments pour faciliter l'identification de leurs auteurs.

## 5. Expérimentation

A travers cette expérimentation, nous visons deux objectifs essentiels. Recueillir d'abord des informations sur *JamEdit* pour comprendre comment il est perçu par les

utilisateurs. Ensuite, mesurer son aptitude à supporter la collaboration entre les participants. Par conséquent, nous avons considéré les facteurs suivants :

*L'interface de JamEdit* — Nous affichons à la fin de chaque session un questionnaire aux participants pour collecter leurs avis. Ces informations concernent les facteurs d'affichage, la réaction du système, les facteurs d'apprentissage, etc.

*L'interaction de groupe* — Nous avons développé un mécanisme (transparent) qui comptabilise les diverses interactions entre les participants au cours d'une session de travail : l'échange dynamique des parties du document et de messages, etc. Nous enregistrons également les aspects liés à la rétroaction de groupe tels que le souhait des participants de s'informer les uns sur les autres, de se localiser au sein de l'espace de travail partagé, les types d'événements dont ils souhaitent être notifiés, etc.

*Fonctionnalités* — Nous avons mené une étude basée sur des interviews avec les utilisateurs pour vérifier si les fonctionnalités de *JamEdit* répondent à leurs besoins, s'il faut en développer certaines ou en rajouter de nouvelles.

*Organisation du groupe* — Nous avons fait appel à des spécialistes en sciences humaines pour discuter avec les utilisateurs des aspects associés à l'organisation du groupe. Notre objectif est de comprendre si *JamEdit* favorise effectivement l'installation des règles sociales et permet une collaboration naturelle.

Cette démarche nous a apporté des renseignements appréciables. Ainsi, nous avons fait utiliser *JamEdit* par un groupe d'enseignants pour la rédaction de documents de projets et de supports de cours à l'institut. Il en découle les conclusions suivantes :

Les participants se sont rapidement familiarisés avec le système. Ceci démontre qu'au niveau interaction *JamEdit* s'est avéré efficace. En effet, les marques associées aux objets sont claires et répondent bien aux règles de l'ergonomie cognitive (composant bloqué, libre, versions alternatives, etc.).

Au cours des différentes sessions de travail, les participants ont tous pratiquement configuré leur espace de façon à ce qu'ils soient notifiés de la majorité des événements qui se produisent. Ceci démontre leur désir intense de s'informer sur leurs activités mutuelles. Ils ont également échangé un nombre considérable de messages<sup>7</sup>. Par conséquent, nous pouvons affirmer que le mécanisme de messagerie intégré a joué le rôle d'un outil de négociation et que sa disponibilité au sein de *JamEdit* a facilité la tâche des participants. Les modes de coopération supportés par *JamEdit* ont favorisé l'interaction et la collaboration entre les participants. Cependant, cette modeste expérience a montré que leur efficacité repose essentiellement sur une bonne organisation des tâches à réaliser, ainsi que sur la distribution des rôles aux différents participants.

---

<sup>7</sup> Au cours des différentes sessions de travail, les participants ont consacré une moyenne de 40% de leur temps à la communication entre eux.

Les fonctionnalités de *JamEdit* se sont avérées utiles aux participants. En particulier, celles qui leur permettent de contrôler les mises à jour, d'établir les engagements de notifications et d'archiver les versions choisies. Cependant, cette flexibilité qui nous paraissait l'un des points forts de *JamEdit* leur a généré quelques difficultés au début de l'expérience pour configurer leur contexte de travail courant. Quant aux fonctionnalités spécifiques à l'édition, les participants les ont trouvées moins développées que celles de leur outil de travail habituel *Word*. Ils ont par exemple exprimé le besoin d'utiliser un correcteur d'orthographe.

Les participants ont réussi rapidement à établir des objectifs communs et résolu les problèmes d'autorités et de rôles sur les parties du document. Ce constat qui semble a priori encourageant, ne nous permet de tirer aucune conclusion puisque les participants travaillent tous ensemble à l'institut et se connaissent parfaitement.

Enfin, l'aspect le plus positif de cette expérience fut sans doute l'introduction d'un collecticiel comme *JamEdit* au sein d'un groupe de participants d'origine ethnique arabe. Car bien que ces communautés vivent socialement de façon très regroupée, la collaboration et le travail de groupe sont des pratiques plutôt rares. Non seulement *JamEdit* a influencé leurs comportements, mais encore a constitué un terrain de partage des connaissances et du savoir-faire. En effet, à part une légère retenue au départ, les participants ont éprouvé une vive sensation de collaboration et de participation au cours des différentes sessions de travail.

## 6. Conclusion et perspectives

Dans cet article, nous avons présenté les concepts de base de la conception de l'éditeur coopératif *JamEdit*. Nous avons largement montré qu'il permet à plusieurs participants 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 d'édition.

L'exploitation de la structure du document a été largement discutée, car elle permet d'informer de manière effective les utilisateurs de leurs actions mutuelles. Au niveau visuel, l'association des marques aux objets du document facilite la tâche des participants et leur permet de déduire intuitivement leurs états respectifs ainsi que le mode de coopération appliqué.

Au cours d'une session de travail, les participants peuvent changer de rôle, échanger dynamiquement des parties et interagir à travers les différents modes de coopération définis. Cette flexibilité est dictée par la nécessité de centrer *JamEdit* sur la dynamique impliquée par le processus d'écriture coopérative. Par ailleurs, l'intégration du mécanisme de messagerie au sein de l'éditeur permet aux participants de combiner la communication et la réalisation de leurs tâches. Pour mieux appréhender le fonctionnement de *JamEdit*, nous avons développé une première version sous MS Windows en Borland C++. L'expérimentation de cette

version sur un réseau local donne une bonne idée des possibilités de collaboration et de coordination. Cependant, ces tests demeurent préliminaires.

Actuellement, nous sommes en train d'affiner les fonctionnalités existantes. Notre modeste expérience a révélé que les participants ont souvent eu recours à des contacts directs pour discuter du travail. Par conséquent, nous envisageons d'intégrer la messagerie vocale pour assurer une communication naturelle entre les participants. Ceci permettra sans doute aux règles sociales de s'installer naturellement et complétera les mécanismes de coordination que nous avons développés. A plus long terme, nos recherches visent à adapter *JamEdit* pour l'édition coopérative sur Internet, notamment en traduisant ses fonctionnalités en Java. Ceci nous permettra de monter des expériences à une échelle beaucoup plus large, afin de l'évaluer avec plus de précision.

#### Remerciements

Cet article est le fruit d'une révision minutieuse de lecteurs anonymes que nous tenons particulièrement à remercier. Nous adressons également nos remerciements aux enseignants de l'institut d'informatique de Batna pour leur patience et leur coopération durant cette étude, ainsi que les membres de l'équipe de recherche Ibn-Khaldoun des sciences humaines pour les échanges d'idées fructueuses.

#### 7. Bibliographie

- [ABH 96] ABHIJIT C., PI-SHENG D., « A Computational Model of Coordination », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, n° 1, May 1996, p. 132-151.
- [BAE 94] BAECKER R., GLASS G., MITCHELL A., POSNER I., « SASSE : The Collaborative Editor », *Proceedings of the ACM CHI'94 Conference on Human Factors in Computing Systems*, Vol. 2, 1994, p. 459-460.
- [BAE 96] BAEKER A., BUSBACH U., « DocMan : A Document Management System for Cooperative Support », *Proceedings of Hawaii International Conference on System Sciences*, IEEE Computer Society Press, January 1996, p. 82-91.
- [BEC 93] BECK E., BELLOTTI V., « Informed Opportunism as Strategy : Supporting Coordination in Distributed Collaborative Writing », In MICHELIS G., SIMONE C., SCHMIDT K. *Proceedings of the ECCSCW'93*. Milan 1993, p. 233-248.
- [BEL 97] BELATTAR B., ZIDANI A., DJOUDI M., « Vers un Environnement de Simulation Coopératif », *Conférence Francophone de Modélisation et de Simulation*, Rouen, Juin 1997, p. 483-500.
- [BUX 92] BUXTON W., « Telepresence : Integrating Shared Task and Person Spaces », *Proceedings of the Graphics Interfaces Conference*, p. 123-129, 1992.
- [DEC 95] DECOUCHANT D., QUINT V., ROMERO M., « Structured Cooperative Editing and Group Awareness », *Proceedings of the Sixth International Conference on Human-Computer Interaction*, July 9-14, Tokyo 1995, Vol. 20A, p. 403-408.

- [DOU 92] DOURISH P., BLY. S., « Portholes : Supporting Awareness in A Distributed Work Group », *Proceedings of CHI'92 Conference on Human Factors in Computing Systems*, Monterey, May 1992, ACM Press, p. 541-547.
- [ELL 91] ELLIS C., GIBBS S., REIN G., « Groupware, Some Issues and Experiences », *Communications of the ACM*, vol. 34, n° 1, January 1991, p. 38-58.
- [GRE 96] GREENBERG S., GUTWIN C., COCKBURN A., « Awareness Trough Fisheye Views in Relaxed-WYSIWIS Groupware », *Proceedings of the Graphics Interface*, Toronto, Canada, May 21-24, 1996, p. 28-38.
- [GRE 92] GREIF I., SELIGER R., WEIHL W., « A Case Study of CES : A Distributed Collaborative Editing System Implemented in Argus », *IEEE Transactions on Software Engineering*, vol. 8, n° 9, September 1992, p. 827-839.
- [GRU 94] GRUDIN J., « Groupware and social dynamics : Eight challenges for developers », *Communications of the ACM*, vol. 37, n° 1, January 1994, p. 92-105.
- [GUT 96] GUTWIN C., ROSEMAN M., GREENBERG S., « A Usability Study of Awareness Widgets in a Shared Workspace Groupware System », *Proceedings of CSCW'96*, Boston 1996, p. 258-267.
- [GUT 98] GUTWIN C., GREENBERG S., « Effects of Awareness Support on Groupware Usability », *Proceedings of the CHI'98 Conference on Human Factors in Computing Systems*, Los Angeles 1998, ACM Press, p. 511-518.
- [KAR 94] KARSENTY A., « GroupDesign : un collecticiel synchrone pour l'édition partagée de documents », Thèse de Doctorat, Université de Paris Sud, Février 1994.
- [KOC 95] KOCH M., « Design Issues for a Distributed Multi-User Editor », *Computer Supported Cooperative Work, An International Journal*, Vol. 3, n° 4, 1995, p. 359-378.
- [KRA 92] KRAUT R., GALEGHER J., CHALFONTE B., FISH R., « Task Requirements and Media Choice in Collaborative Writing », *Human Computer Interaction*, Vol. 7, 1992, p. 375-407.
- [NEU 94] NEUWIRTH C., KAUFER D., CHANDHOK R., MORRIS J., « Computer Support for Distributed Collaborative Writing : Defining Parameters of Interaction », *Proceedings of CSCW'94*, Chapel Hill, ACM Press, October 22-26, 1994.
- [PAC 94] PACULL F., SANDOZ A., SCHIPER A., « Duplex : A Distributed Collaborative Editing Environment in Large Scale », *Proceedings of the CSCW'94 Conference*, ACM Press, October 1994.
- [POS 92] POSNER I., BAECKER R., « How People Write Together ? », *Proceedings of the 25<sup>th</sup> Annual Hawaiian International Conference on System Sciences*, 1992, p. 127-138.
- [ROS 96] ROSEMAN M., GREENBERG S., « Building Real-Time Groupware With GroupKit, a Groupware Toolkit », *Proceedings of the ACM Transactions on Computer-Human Interaction*, Vol. 3, n° 1, March 1996, p. 66-106.
- [SAL 94] SALBER D., NIGAY L., COUTAZ J., « Extending the Scope of PAC-Amodeus to Cooperative Systems », *Proceedings of the CSCW'94 Workshop on Software Architectures for Cooperative Systems*, October 22-26, Chapel Hill, USA, 1994.

[WII 91] WIL U. K., « Using Events as Support for Data Sharing in Collaborative Work », *Proceedings of the International Workshop on CSCW*, Berlin 1991.

Article reçu le 3 mars 1999

Version révisée le 30 septembre 1999

Rédacteur responsable : Bernard PHILIPPE

*Abdelmadjid Zidani* prépare une thèse de doctorat à l'université de Batna (Algérie). Titulaire d'un Magister en informatique dans le domaine de l'édition coopérative, son sujet de doctorat concerne la conception d'environnements virtuels dédiés à l'enseignement coopératif sur Internet.

*Mahmoud Boufaïda* est maître de conférences à l'université de Constantine (Algérie) où il dirige une équipe de recherche au laboratoire LIRE. Ses domaines d'intérêt portent sur les systèmes d'information, et plus particulièrement les systèmes coopératifs et multimédia, et sur les interfaces homme-machine.

*Mahieddine Djoudi* est maître de conférences en informatique à l'université de Poitiers et chercheur au laboratoire IRCOM-SIC. Ses domaines d'intérêt portent sur les systèmes coopératifs, l'enseignement à distance, et sur l'usage des nouvelles technologies de l'information et de la communication.