

## Un modèle architectural pour l'édition coopérative de documents

Abdelmadjid. Zidani 1, Mahmoud. Boufaïda 2 et Mahieddine. Djoudi 3

**1** Institut d'Informatique, Université de Batna, (Algérie)

**2** Laboratoire LIRE, Institut d'Informatique, Université de Constantine, (25000)

Tél. & Fax. (213) (4) 92- 28 - 35. E-Mail. boufaïda@ist.cerist.dz

**3** Laboratoire d'Informatique de l'Université de Poitiers, UFR Sciences Fondamentales et Appliquées, 40, Avenue du Recteur Pineau, 86022 Poitiers Cedex - France

Tél. & Fax. (33) (5) 49- 01 – 96 - 64. E-Mail. djoudi@matpts.univ-poitiers.fr

---

### 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 email facilities and 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 layer replicated architecture.*

### Keywords.

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

---

## 1. Introduction

Le développement des collecticiels a réalisé une avancée considérable ces dernières années. En effet, puisqu'il y a quelques années, le collecticiel était un terme brut sans aucun attrait. Les travaux de recherches qui lui ont été consacrés par la suite vont rapidement lui permettre d'émerger en tant que domaine de recherches à part entière. Aujourd'hui, plusieurs systèmes performants commencent déjà à susciter l'intérêt de nombreuses organisations, comme le montre le degré d'utilisation des systèmes de messagerie électronique. Aussi, nous assistons de plus en plus à l'émergence de plateformes collecticielles qui facilitent énormément la tâche des concepteurs. Certaines d'entre elles proposent des outils de désignation et d'annotation destinés à l'usage des groupes et facilement intégrables. D'autres outils prêts à l'emploi permettent même d'assurer la gestion des processus distribués et leurs communications. Citons par exemple **GroupKit** [ROS 95].

Les éditeurs coopératifs dits synchrones permettent à plusieurs utilisateurs de manipuler simultanément les mêmes informations, constituant ainsi une base pour tous les projets basés sur la rédaction de documents. De ce fait, ils représentent une réponse technologique idéale pour les organisations. Cependant, il est nécessaire pour ces systèmes d'assurer une transparence complète sur l'évolution du processus de rédaction et de garantir la cohérence des informations. De plus, il est aussi important de prendre en considération le contexte social régissant les activités de groupes.

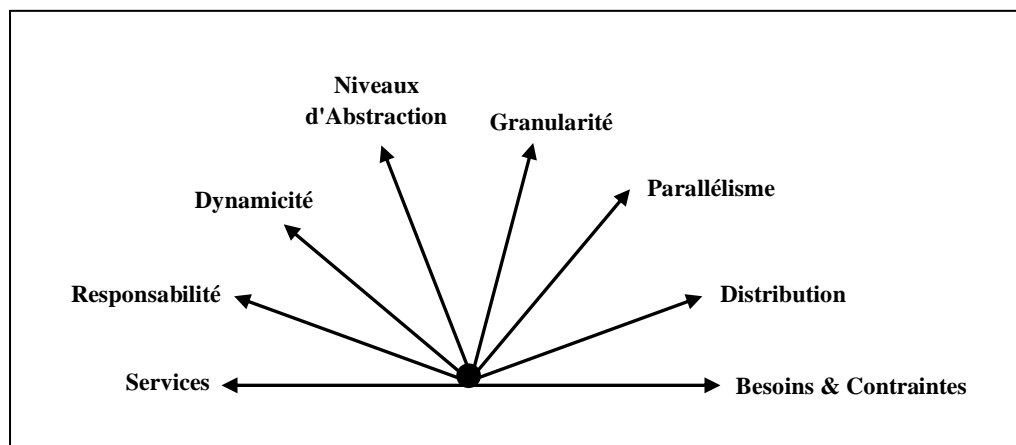
Ainsi, nous nous fixons comme objectif la création d'un espace de travail commun à un ensemble d'utilisateurs, rassemblant les outils nécessaires à l'édition coopérative de documents. Comme toute application coopérative, notre conception tente de satisfaire les contraintes du travail de groupe. Par conséquent, nous avons conçu un support pour gérer les interactions entre les utilisateurs et assurer la transparence sur leurs actions mutuelles.

Cet article est consacré à la présentation d'un modèle architectural pour l'édition synchrone de documents partagés. La conception de ce modèle, est centrée sur les concepts de partage et d'échange de données pour favoriser la collaboration entre les utilisateurs. La section 2 décrit de façon générale le travail coopératif assisté par ordinateur et présente un cadre taxonomique utile aux concepteurs. La section 3, passe en revue les principaux systèmes d'édition partagée et présente leurs caractéristiques essentielles. La section 4 présente notre approche ainsi que les concepts de base pour la conception d'un éditeur coopératif reposant sur un modèle architectural à plusieurs niveaux. Enfin, la dernière section décrit les fonctionnalités essentielles du modèle proposé.

## 2. Notion de Collecticiel

Le collecticiel est un domaine d'étude pluridisciplinaire appelé **TCAO**<sup>1</sup> (Travail Coopératif Assisté par Ordinateur). Cette discipline a pour but de développer des systèmes informatiques supportant la communication, la coordination et la collaboration entre les utilisateurs. Les recherches actuelles tentent d'explorer les technologies de l'information pour faciliter la réalisation d'une tâche par un groupe de personnes. Plusieurs définitions du collecticiel ont été proposées [BAN 89], [ELL 91], [AFC 94], [PAL 94], etc. Parmi ces dernières nous citons la définition de Palmer [PAL 94]: " 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". Il doit pouvoir les intégrer tout en veillant à ne pas favoriser un aspect au détriment de l'autre.

Par ses caractéristiques très diversifiées, le collecticiel reste difficile à cerner [KAR 94]. Sa décomposition en sous domaines s'est avérée efficace pour l'aborder. 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] et la matrice 3x3 [GRU 94]. Salbert et Coutaz [SAL 94] proposent un espace de référence (figure 1) permettant de raisonner sur le comportement des systèmes coopératifs et montrent clairement qu'il peut servir à classer les systèmes actuels et futurs. Cet espace de référence permet de dégager les caractéristiques et donne une vue globale du système à développer et constitue de ce fait une aide appréciable aux concepteurs.



**Figure 1.** Espace du problème pour le développement de systèmes coopératifs [SAL 94]

Comme le montre la figure 1, l'espace de référence 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 coopératif. La seconde exprime la responsabilité d'assurer un service particulier dans un

<sup>1</sup> En Anglais, CSCW(Computer Supported Cooperative Work)

environnement coopératif. La dynamicité couvre la capacité du système à supporter les changements durant une session de travail. La quatrième dimension permet au concepteur d'envisager le développement de son 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. Avec une granularité fine les utilisateurs peuvent modifier une donnée en même temps. Le parallélisme concerne tous les systèmes concurrents et 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 dynamicité des interactions des utilisateurs. Cet aspect sera discuté dans le premier paragraphe de la section 5. Enfin, la huitième dimension constitue l'étape de départ du processus de conception, et couvre la description des outils logiciels et matériels mis en jeu pour le développement d'un système coopératif.

Cet espace constitue donc 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 de façon précise nos choix, 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 "responsabilité" et "dynamicité", nous pouvons exprimer que la responsabilité sur la modification est échangée dynamiquement entre les utilisateurs. Enfin, comme l'a suggéré Grudin [GRU 90], nous avons mené une étude sur les systèmes existants pour tirer profit de leurs expériences.

### **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 [ELI 91] 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, parmi lesquels nous distinguons les éditeurs asynchrones **SHARED-BOOKS** [LEW 88] **PREP** [NEU 90], **QUILT** [LEL 88], **CES** [GRE 86], etc. Ces systèmes ne permettent qu'à un seul utilisateur d'intervenir sur le document. Un deuxième type de travaux sont les éditeurs synchrones, ils permettent à plusieurs utilisateurs d'agir simultanément sur le même document. Parmi ces systèmes nous pouvons citer les éditeurs de texte **GROVE** [ELI 91], **SASSE** [BAE 92] ou encore les éditeurs de graphiques structurés **GROUPDESIGN** [BEA 92], **GROUPDRAW** et **GROUPSKETCH** [GRE 92], etc. Ces éditeurs sont caractérisés par une granularité fine. Par exemple, deux utilisateurs peuvent modifier le même mot en même temps.

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, QUILT utilise les sections comme granularité d'édition. Par contre, l'éditeur GROVE ou la boîte à outils **DISTEDIT** [KNI 90], 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. Cet algorithme est fondé sur les priorités et la sémantique des opérations. Le tour de rôle<sup>2</sup> 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** [BIE 91] 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** [NEW 91] supporte une granularité d'édition variable, il utilise un contrôle centralisé permettant à un utilisateur de verrouiller un fragment de texte pour travailler sans interférences avec les autres. Il existe d'autres approches comme **MULE** [PEN 90] et **MultimETH** [LUB 90] qui s'appuient sur le mécanisme de blocage pour gérer la concurrence. Pour l'éditeur MULE, le partage peut concerner une ligne de texte. MultimETH permet à un groupe d'utilisateurs de créer et d'éditer des documents multimédias structurés (i.e. titres, en-têtes, chapitres et sections). Ces éléments structurés ont chacun un contenu spécifique (i.e. texte, graphique ou icône) et sont organisés suivant une relation hiérarchique. La liste des travaux discutés dans cet article n'est pas exhaustive. Il existe une variété d'autres approches qui sont toutes aussi intéressantes telles que : **MMCONF** [KRO 90], **COAUTHOR** [HAH 91], **GROUPSYSTEMS** [NUN 91], etc.

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

**Table-1** : Différentes approches d'édition coopérative

Contrairement aux autres approches (cf. Table 1), notre modèle s'appuie sur une architecture répliquée et se limite à la manipulation de documents textuels. Il sépare les données textuelles et structurelles et leur définit des traitements totalement indépendants. La granularité d'édition est variable, elle dépend de la décomposition du document. La fonction de contrôle est assurée localement au niveau de chaque site et les sites communiquent entre eux pour préserver la cohérence des données. Enfin, pour gérer la concurrence nous exploitons un mécanisme de blocages basé sur la négociation entre les utilisateurs.

<sup>2</sup> En Anglais, Floor-Control

## 4. Présentation du Modèle Architectural

Notre objectif n'est pas de construire une nouvelle méthode satisfaisant aux besoins des applications d'édition partagée, mais d'utiliser aux mieux les méthodes existantes. Le modèle que nous proposons, repose sur une architecture en couches répliquée (figure 2), il s'appuie sur la structure logique du document pour exprimer et gérer la coopération entre les utilisateurs. 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 de spécifier explicitement des parties et des positions au sein du document. Elle constitue un moyen efficace pour répondre aux questions que peut se poser l'utilisateur sur les activités courantes de ses collègues et lui facilite leur localisation. L'existence de cette structure suggère naturellement un ensemble d'opérations pour l'exploiter 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.

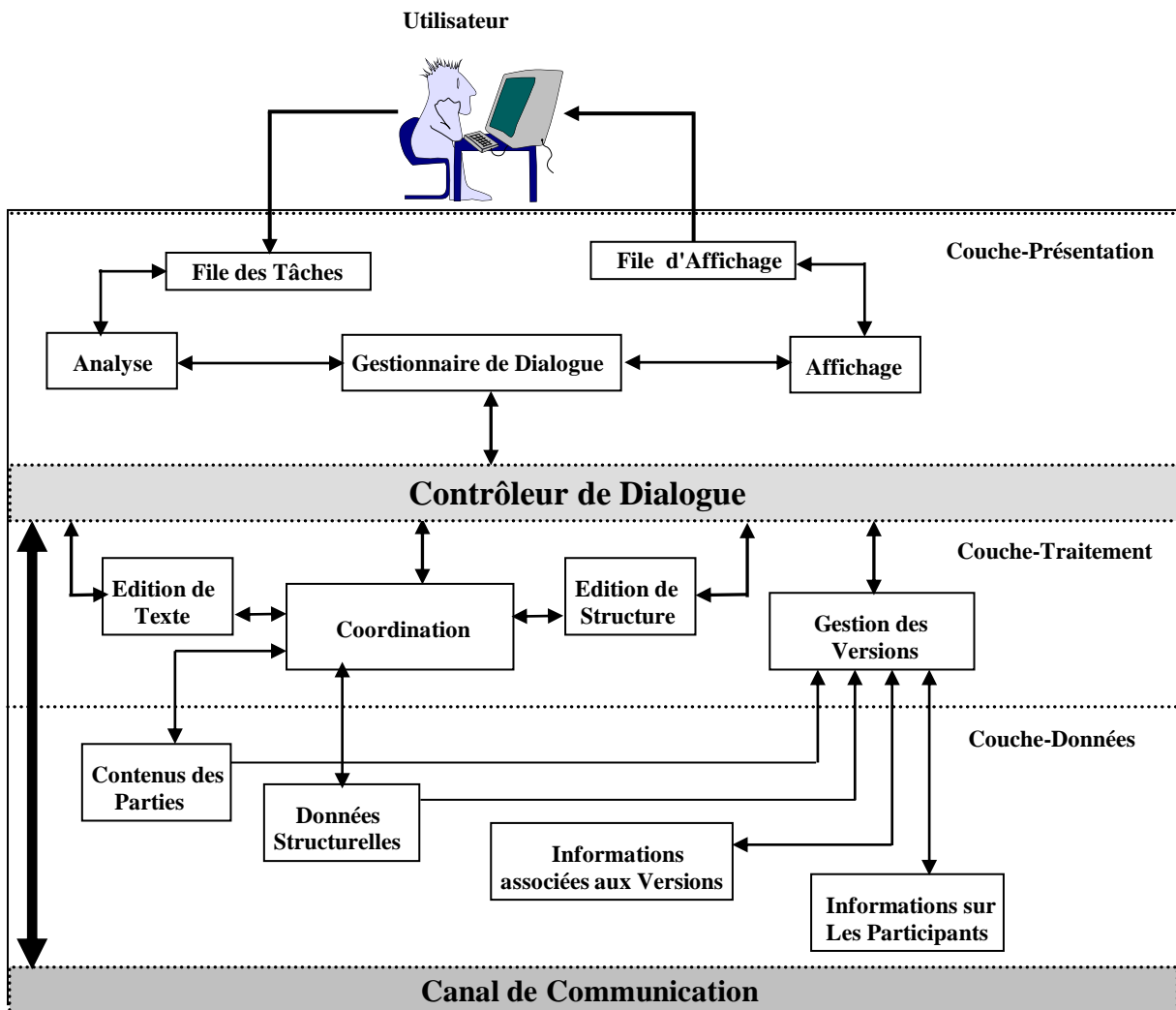


Figure 2 : Architecture simplifiée de l'éditeur coopératif

La figure 2 illustre l'architecture en couches de notre modèle. Cette architecture s'appuie sur le principe de séparation entre le composant de présentation et le noyau fonctionnel [COU 94]. Nous distinguons la 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 et de gestion des versions. La couche "**Données**" 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). Le "**contrôleur de dialogue**" permet de réaliser le couplage entre l'interaction de l'utilisateur et l'application. Enfin, le "**canal de communication**" sert de support aux échanges entre les différents sites. Ainsi, comme pour tout système interactif, nous devons gérer les interactions entre l'utilisateur, l'interface (couche Présentation) et l'application (couches Traitements et Données).

#### **4.1. La Couche Présentation**

La couche Présentation définit la partie perceptible du système et montre sa vision externe. Sa principale tâche est la gestion des interactions de l'utilisateur avec le système. Les objectifs essentiels de cette couche sont :

- Répondre rapidement aux différentes actions de l'utilisateur et accroître le taux d'interactions entre l'utilisateur et le système [MAR 93].
- Montrer à l'utilisateur une vue simple, facile à manipuler et à mémoriser pour réduire ses charges cognitives [COU 90].

Notre modèle est organisé comme une collection de composants totalement indépendants. Par conséquent, nous pouvons exploiter le parallélisme pour satisfaire la première exigence. Quant à la deuxième, nous avons choisi les signes visuels de sorte qu'ils véhiculent au mieux les concepts qu'ils représentent.

##### **4.1.1. Le composant d'analyse**

Ce composant est chargé de suivre les moindres gestes de l'utilisateur pour déterminer de façon exacte ses intentions. L'écran de l'utilisateur est partagé en zones de contrôle et de travail pour faciliter la distinction entre les activités d'édition et les traitements de synchronisation et de communication. Chaque zone contient un ensemble d'objets interactifs réagissant aux actions de l'utilisateur. Ainsi, lorsque le point d'intérêt de l'utilisateur (souris) est localisé en zone de contrôle, dès que l'utilisateur actionne le bouton de la souris, l'analyseur va agir en partant des coordonnées d'appui pour déterminer exactement l'objet interactif concerné. Il procède ensuite à l'insertion de la tâche de l'utilisateur dans la file des tâches (Figure 2). Le contrôleur de dialogue est chargé d'établir la liaison entre l'objet interactif actionné et l'objet du domaine d'application associé et décide de la suite des traitements à déclencher.

#### **4.1.2. Le composant d'affichage**

Le composant d'affichage est chargé de repeindre régulièrement l'écran de l'utilisateur avec les représentations appropriées à la situation courante. L'affichage nécessite l'utilisation d'une bibliothèque d'objets graphiques de présentation qui doivent être placés soigneusement dans des zones spécifiques de l'écran. Un exemple typique des traitements d'affichage sont les traitements d'animation. Lorsque par exemple, l'utilisateur déplace ou redimensionne un objet, les traitements d'affichage réagissent alors en lui montrant les représentations successives associées aux déplacements de l'objet manipulé.

### **4.2. La Couche Données**

La couche donnée regroupe toutes les informations nécessaires à la conduite d'un projet de rédaction coopérative. Nous distinguons (cf. Figure 2) les informations associées aux participants (identité, adresse, rôle, compétences, etc.). Ainsi que les informations permettant de gérer l'évolution dynamique du contenu et celles liées à la structure logique du document. Lesquelles sont organisées de façon à faciliter leur récupération, modification et enregistrement.

#### **4.2.1. Le 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. Les contenus des différentes parties du document sont enregistrés dans des fichiers distincts qui sont partageables par les utilisateurs.

#### **4.2.2. Les informations de la structure**

La décomposition d'un document en parties sur lesquelles seront affectés des utilisateurs avec des rôles bien déterminés, doit apparaître explicitement à l'utilisateur. Les informations structurelles correspondent à la dernière version de la structure logique du document. Cette structure est visualisée et manipulée par l'utilisateur, elle lui permet alors de s'informer sur l'état global du projet. L'information structurelle n'est visible qu'à l'éditeur, elle repose sur une séquence d'enregistrements qui peuvent être verrouillés par chaque membre du groupe. Elle 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.). La figure 3 montre la structure d'un document avec ses différentes parties. 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 présentation de la structure 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 réalisées sur le document. Elle permet 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 à la fin d'une partie du document ou encore la sélection d'une partie devant subir une opération, etc.

### **4.3. La Couche Traitement**

La couche traitement recouvre plusieurs types de traitements, parmi lesquels nous distinguons essentiellement ceux qui sont liés à la gestion de la structure logique et le contenu du document, le contrôle des droits d'accès, la gestion des mises à jour, la synchronisation des actions des utilisateurs, la gestion des versions des contenus et de la structure, etc.

#### **4.3.1. Les tâches d'édition de texte**

Ce composant regroupe toutes les fonctionnalités qui utilisent seulement les informations textuelles. Ces fonctionnalités sont similaires à celles des éditeurs mono-utilisateur existants.

#### **4.3.2. Les tâches d'édition de la structure**

La structure du document évolue normalement durant l'existence du document. L'édition d'une telle structure est une fonctionnalité totalement indépendante de celle 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. Toutes ces opérations nécessitent une sélection préalable de la partie concernée.

#### **4.3.3. Le composant de coordination**

Les deux composants d'édition des contenus et de la structure s'appuient sur le composant de coordination pour gérer les interventions concurrentes sur les données partagées. Le composant de coordination est informé régulièrement par le contrôleur de dialogue sur les événements qui se produisent au niveau des autres sites. Par conséquent, avant d'autoriser la réalisation d'une action sur les données, il doit accomplir un certain nombre de contrôles (droits d'accès, priorités, état des parties manipulées : libres/bloquées, etc.). Les effets des opérations de la coordination apparaissent explicitement à l'utilisateur au niveau de l'interface. Les blocages explicites des parties du document par un utilisateur ou un relâchement du blocage en sont un bon exemple. Ils peuvent être "souples" (i.e. l'unique possibilité offerte aux autres est la consultation) ou bien "strictes" (i.e. l'éditeur garantit l'accès exclusif aux parties bloquées).

### **4.4. Contrôleur de dialogue**

Pour assurer la transparence et la coordination, nous avons adopté le concept de notification des événements [GRO 94]. De même que pour intégrer notre application dans un environnement coopératif, un contrôleur de dialogue s'interpose entre son noyau fonctionnel et son interface utilisateur (cf. figure 2), de telle sorte qu'une action réalisée par

un utilisateur puisse être diffusée aux autres participants. Pour développer les mécanismes de partage, nous adoptons l'approche de manipulation des “ événements de niveau-haut ” [ATA 94] qui permet une interprétation simple des événements. Pour cette approche les différents sites échangent entre eux des messages récupérés au niveau de l'application (ex. Ouvrir (document), effacer (élément), bloquer (élément), etc.). Le contrôleur de dialogue se situe entre les couches traitement et présentation. Il est responsable de la gestion des relations et des échanges entre elles. Il est aussi chargé d'intercepter les événements générés par l'utilisateur pour les communiquer aux autres sites. Nous lui attribuons localement toutes les opérations de contrôle (déclenchement et interruption des traitements, etc.). L'algorithme suivant traduit une réaction très 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
              Invoquer le composant d'Affichage
    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
```

Le contrôleur de dialogue permet grâce à sa disposition de récupérer les traces des actions réalisées sur les parties du document sous forme d'événements, ceci le prédispose à prendre les décisions appropriées. Comme le montre l'algorithme, le contrôleur de dialogue permet à l'utilisateur d'exécuter ses tâches, jusqu'à ce qu'il invoque une opération nécessitant la coordination avec les autres participants. Le contrôleur de dialogue se charge d'établir dans ce cas des communications avec les autres sites pour coordonner le travail.

## 5. Fonctionnalités de l'Editeur Coopératif

L'éditeur coopératif proposé utilise une structure 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 logique.

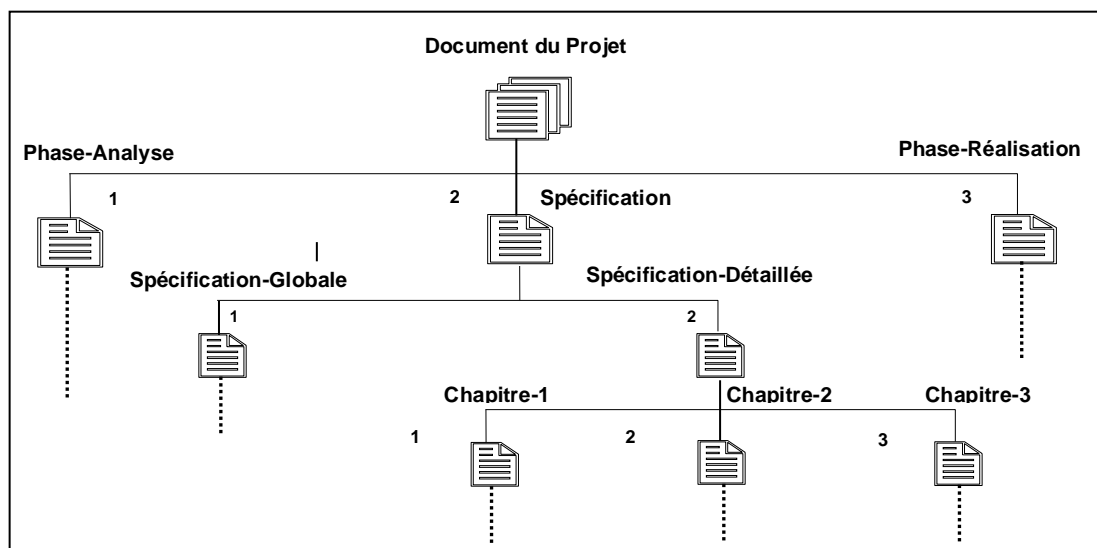
### 5.1. Fonctionnalités de base

Notre modèle s'appuie sur la communication entre les utilisateurs pour faciliter la collaboration. Il intègre aussi la messagerie électronique pour permettre aux utilisateurs de négocier l'intervention sur les parties du document. Nous pensons que ceci constitue un moyen efficace pour l'introduction naturelle des règles sociales plutôt que d'imposer des contraintes au niveau du logiciel. Ainsi, notre système assure les fonctionnalités suivantes :

- Au cours d'une session de travail, les utilisateurs peuvent échanger dynamiquement des messages pour discuter sur l'état d'avancement du travail ou négocier le blocage des parties du document. L'émetteur rédige son message, spécifie l'adresse du destinataire et le contrôleur de dialogue se chargera de l'acheminer vers le site concerné.
- Le traitement des informations structurelles et textuelles qui sont enregistrées séparément. Ceci nous permet comme pour l'éditeur coopératif CES de modifier la structure du document sans endommager les contenus des parties déplacées. Le contenu du document est composé de plusieurs parties sur lesquelles les différents participants peuvent intervenir individuellement ou collectivement. La structure du document permet la localisation des différentes parties et facilite donc la récupération de leurs contenus.
- Au cours d'une session de travail, plusieurs parties sont utilisées par les participants. Les contenus de ces parties sont alors enregistrés localement. Par conséquent, une même partie qui est utilisée par plusieurs utilisateurs sera répliquée sur chacun des sites associés. Chaque utilisateur peut verrouiller la partie qui l'intéresse pour travailler et enregistrer à chaque fois ses mises à jour pour forcer leur apparition sur les écrans des autres.
- L'ouverture d'une partie du document nécessite la spécification du type d'accès (lecture ou écriture ). Toutes les parties sont accessibles en lecture. Cependant, le système informe les utilisateurs si une partie est bloquée ou modifiée. Ce mécanisme de notification des événements constitue un moyen efficace pour gérer les accès concurrents.
- D'un point de vue strictement coopératif, nous accordons une importance extrême au concept de transparence [DOU 92]. Ainsi, le système peut fournir à l'utilisateur des informations concernant aussi bien les autres participants (identité, adresse, activité en cours, etc.) que les états de chaque partie du document (bloquée ou libre, identités des acteurs manipulant cette partie, leurs adresses, instant de blocage, etc.).

## 5.2. Structure logique

La structure interne est organisée comme 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 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 l'utilisateur comme par exemple "Spécification Détaillée" ou "Chapitre 1" (cf. figure 3). 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 la structure globale du nœud "Document du Projet" avec ses descendants. Les composants de l'éditeur décrit (cf. Paragraphe 4) exploitent la structure logique pour permettre à l'utilisateur de sélectionner des objets pour les supprimer, en créer de nouveaux ou déplacer des objets existants et modifier leurs noms. L'utilisateur peut aussi sélectionner un objet pour s'informer de son état ou des acteurs qui le manipulent, etc.



**Figure 3** : Structure logique du document

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 utilisateurs. Elles utilisent des signes visuels pour marquer les parties bloquées. Le composant 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 qu'un système reposant sur notre modèle 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 peut être 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 commencé à implémenter une première version en C++ sous l'environnement Windows. L'étude de cette version donne une bonne idée sur les possibilités de collaboration et de coordination. L'idée d'intégration de la messagerie électronique au sein de l'éditeur, permettra sans doute aux utilisateurs de combiner la communication et la réalisation de leurs tâches. Enfin nous envisageons à l'avenir de permettre une interaction multimodale pour garantir une communication naturelle entre les utilisateurs et répondre davantage aux besoins des projets de groupes.

## 7. Bibliographie

- [AFC 94]: AFCET., "Enquête sur la pratique de la collectique (groupware) en France", Rapport d'étude, (Sep, 1994), 83 pages.
- [ATA 94]: ATALLAH, S., KANAWATI, R., "Architecture de systèmes pour les collecticiels synchrones", Actes d'IHM'94.
- [BAE 92]: 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).
- [BAN 89]: BANNON, L., SCHMIDT, K., "CSCW : Four Characters Search of a Context", in Proceedings of the First European conference on CSCW, (ECSW' 89), (Sep 1989), P 13-15.
- [BEA 92]: 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), (Nov 92).
- [BIE 91]: 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.
- [COU 90]: COUTAZ J., "Interfaces Homme-Machine : Un regard critique," Rapport. Actes d'IHM'90.
- [COU 94]: COUTAZ J., "Software Architecture Modelling for User Interfaces," Encyclopedia on Software Engineering. 1994.
- [DOU 92]: DOURISH P., et BLY, S., "Potholes : Supporting Awareness in A Distributed Work Group," proceedings of CHI'92 conference on Human Factors In Computing Systems, Monterey, (May 1992).
- [ELI 91]: ELLIS, C.A., GIBBS, S.G., REIN, G.L. "Groupware, Some Issues and Experiences," Communications of the ACM, vol 34, n° 1, (Jan 1991), P 38-58.
- [GRE 92]: 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 364-392.
- [GRE 86]: GREIF, I., SELIGER, R., WEIHL, 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.
- [GRO 94]: GRØNBÆK, K., HEM, J., MADSEN, O., SLOTH, L., "Hypermedia Systems : A Dexter-Based Architecture", Com. ACM, vol 37, n° 2, (Feb 1994), pp 65-74.
- [GRU 90]: GRUDIN J., "Groupware and Cooperative Work : Problems and Prospects", Communication. ACM, vol 37, n° 1, (January 1994), pp 93-105.
- [GRU 94]: GRUDIN J., "Groupware and social dynamics : Eight challenges for developers," Communication. ACM, vol 37, n° 1, (January 1994), pp 93-105.
- [HAH 91]: HAHN, U., JARKE, M., EHERER, S., KREPLIN, K., "CoAUTHOR: A Hypermedia Group Authoring Environment", Studies in CSCW, 1991, p 79-100.

- [KAR 94]: KARSENTY A., “ Le collectif : de l’interaction homme-machine à la communication homme-machine-homme ,” *Technique et Science Informatiques*, vol 36, n° 1, ( 1994 ), pp 105-127.
- [KNI 90]: 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.
- [KRO 90]: KROWLEY, T., MILAZZO, P., BAKER, O., FORSDICK, H., MMConf: 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.
- [LEL 88]: LELAND, M., FISHER, S., KRAUT R., “ Collaborative document production using Quilt ”, *Com*, ACM, (Sep 1988), pp 206-215.
- [LEW 88]: 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.
- [LUB 90]: 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.
- [MAR 93]: MARCUS, A., “Humain Communications Issues in Advanced Uis”, *Com ACM*, vol 36, n° 4, (Apr 1993), p 101-109.
- [NEU 90]: 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.
- [NEW 91]: NEWMAN-WOLFE, R., PELIMUHANDIRAM, H., “ MACE: A Fine-Grained Concurrent Editor”, in proc of ACM SIGOIS conf on Organizational Computing Systems, Atlanta, GA, 1991, ACM, p 240-254.
- [NUN 91]: 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.
- [PAL 94]: PALMER J, et FIELDS N. “ Guest editors' introduction : computer-supported cooperative work ”, *Com*, IEEE Computer, vol 27, n° 5, (1994), pp 15-18.
- [PEN 90]: PENDERGAST, M., VOGEL, D., “ Desing 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.
- [ROS 95]: ROSEMAN M., GREENBERG S., “Building Real Time Groupware with GroupKit, A Groupware Toolkit”, *Communication. ACM*, (April 1995).
- [SAL 94]: SALBER D., NIGAY L., et COUTAZ J., “ Extending the Scope of PAC-Amodeus to Cooperative Systems ,” proceedings of ACM conference on CSCW, North Carolina, (October 1994).