

Player Cooperativo de Documentos Hipermedia

António Silva, Joaquim Sousa Pinto, Joaquim Arnaldo Martins

Resumo - Este artigo pretende apresentar a implementação de uma arquitectura de um sistema distribuído, adequado à edição cooperativa de documentos. Deve permitir a vários utilizadores trabalhem em conjunto cooperando entre si partilhando um documento, o qual poderá ter texto, gráficos, imagens, etc.

Para tal, foram especificados um conjunto de serviços genéricos, organizado em *conferências*, que permitem a edição cooperativa de documentos independente do protocolo de rede utilizado, bem como vários algoritmos de controlo de cooperação.

A implementação prática foi suportada numa interface para redes TCP/IP[1] e para redes WfW[2] Um *Player* hipermedia baseado na tecnologia OLE 2.0[3] permite aderir aos vários modelos de cooperação. Todo este sistema foi desenvolvido em C/C++ para *Windows*.

Abstract - This article presents an system to edit cooperative documents. Should allow many users to work together cooperate between them, share an document that may have text, graphics, images, etc.

For that, had been specified some general services, organised in conferences, as well algorithms to control the cooperation. It had been supported by an interface to TCP/IP and WfW network. An hypermedia *Player* based on the technology OLE 2.0 allow align to the many models of cooperation.

I. INTRODUÇÃO

O trabalho cooperativo suportado por computador (*Computer Supported Collaborative Work* - CSCW) foi motivado pela revolução das redes de comunicação, a chamada auto-estrada da informação e devido à tecnologia actualmente disponível, possibilitar o desenvolvimento de sistemas que suportem a realização de tarefas que requeiram a colaboração simultânea de vários utilizadores, tais como, o ensino remoto, o estudo assistido, resolução de problemas e a preparação de documentos em modo colaborativo.

Define-se trabalho cooperativo suportado por computador ao trabalho desenvolvido por diversas pessoas em conjunto sobre um produto, um tema de investigação ou qualquer assunto com ajuda de computadores.

A ideia base do *Player Cooperativo de Documentos de Hipermedia* consiste em apresentar sobre a forma de conferências, um conjunto de serviços necessários ao

estabelecimento de uma conversação por parte de vários participantes. Para tal, foi projectado para suportar vários utilizadores, os quais supõe-se que queiram colaborar entre si (não competir), trabalhando em diferentes estações de trabalho ligados por uma rede local, com o fim de criar um ambiente de trabalho cooperativo. O objectivo dos utilizadores é suposto ser o de trabalharem para um objectivo comum, apoiar o trabalho dos restantes e assim promover o progresso em grupo.

A implementação usa o paradigma Cliente-Servidor onde o processo Servidor coordena o fluxo de informação e tem como obrigação manter o sincronismo entre todos os participantes numa conferência. Podem coexistir simultaneamente diversas conferências.

O sistema designado por *Player* baseia-se na metáfora do "retroprojector", pois é um equipamento usado com frequência na educação. Quando um professor apresenta uma lição usando o retroprojector, coloca transparências com diversas ilustrações que vai passando de uma para outras transparências consoante o seguimento e reacção da audiência. A estas ilustrações, no *Player*, chamamos objectos. Os objectos são construídos por aplicações *Microsoft Windows* e são integrados no documento usando as facilidades de comunicação entre aplicações fornecidas por este ambiente, nomeadamente o mecanismo de OLE.

Exemplos de documentos multimédia a serem produzidos desta forma são aqueles que envolvem pessoas que tenham dificuldade em deslocar-se para se encontrar (devido aos factores tempo, distância, dinheiro, etc.) ou grupos com tarefas diferentes no tempo. Concretamente, refiram-se relatórios científicos, propostas de projectos ou trabalhos conjuntos entre várias instituições.

II. ALGUNS CONCEITOS

Antes de se começar a descrição do trabalho desenvolvido é conveniente definir alguns termos que vão ser usados:

- *Conferência* - é o suporte do trabalho cooperativo entre os utilizadores. É constituída por um conjunto de atributos que a define, os quais são: o nome, o tipo, o modo, a duração e o número de participante. É iniciada por qualquer utilizador.

- *Proponente* - utilizador que inicia uma conferência. É este utilizador que dirige e controla a conferência, possuindo algumas ferramentas inacessíveis aos restantes *Cooperantes*. Está ligado aos outros *Proponentes* e aos utilizadores que participam na sua conferência e só ele pode terminar a conferência, quando o entender.

- *Cooperante* - qualquer um dos utilizadores que participe na conferência. Qualquer utilizador pode intervir e participar na conferência, incluindo o utilizador que inicia a conferência.

- *Documento* - ficheiro hipermedia constituído por uma sequência de transparências que poderão conter todo o tipo de informação hipermedia: texto, gráficos, imagens, assim como ligações entre transparências, âncoras, que podem ser percorridas por um utilizador.

III. INFRA-ESTRUTURAS DE COMUNICAÇÃO

A. Introdução

A Fig. 1. representa a forma como se enquadram os vários sistemas computacionais ligados entre si por meio de uma rede (*Network*) de comunicações.

Em pormenor, estão representadas duas aplicações (*Cplayer* e *Ckernel*), constituída cada uma por várias camadas que estão ligadas e comunicam entre si, através de canais que fazem chegar a informação aos utilizadores.

Foram implementados dois canais de comunicação para efectuar troca de informação através da rede. Existe paralelamente a estes dois, um outro canal, que controla apenas a abertura e fecho daqueles dois. Um dos canais foi denominamos por *Canal Global* e está destinado à troca de informação entre os *Proponentes* ou, entre utilizadores que ainda não se encontram ligados a nenhuma conferência. Estes podem comunicar entre si por meio deste canal sem interferir com os utilizadores, *Cooperantes*, de uma determinada conferência.

O outro, denominado por *Canal Local*, é usado para transmitir informação dentro de uma conferência entre os vários utilizadores, *Cooperantes* e o *Proponente*, da qual fazem parte. Podem existir várias conferências paralelas a decorrer sem haver interferência na informação resultante da comunicação entre os seus participantes.

B. Protocolo TCP/IP

Por uma questão de universalidade nas comunicações, a comunicação entre utilizadores do sistema é feita através de uma interface com o protocolo TCP/IP. O protocolo TCP/IP é responsável pela ligação física à rede. A transmissão de dados na rede é feita através do envio de datagramas UDP (*User Datagram Protocol*).

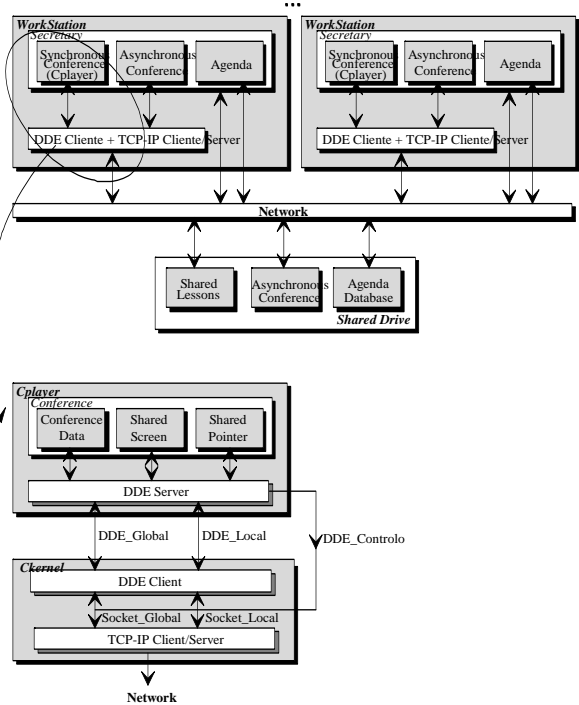


Fig. 1: Esquema geral da arquitectura do projecto.

Este procedimento permite enviar pacotes para muitos destinos e receber pacotes provenientes de vários destinos, ou seja, o estabelecimento de grupos.

C. Sockets

Para estabelecer uma ligação entre vários sistemas computacionais foi utilizada a especificação designada por *Windows Socket API*. A finalidade das *Sockets* é fornecer às aplicações uma abstracção do *software* de rede que está a ser utilizado a um nível mais baixo (Fig. 2), e que permita criar uma interface com a qual as aplicações possam trabalhar em conformidade, independente de qualquer protocolo de rede usado.

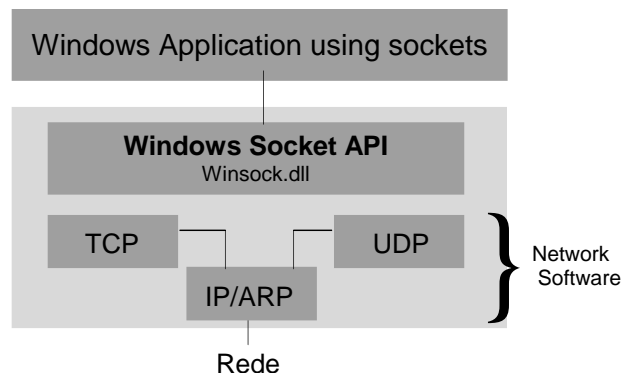


Fig. 2: Arquitectura *Windows Socket API*

Para o suporte da comunicação sobre o canal físico, foram desenvolvidas estruturas usando as *Sockets*, que permitiram definir ligações bidireccionais, através das quais se pode transmitir ou receber informação de diferentes sistemas computacionais.

A *Windows Socket API* contém as primitivas necessárias ao estabelecimento de uma ligação (i.e., canal de

comunicação). Para a implementação destes canais foram implementadas funções fazendo uso dessas primitivas, que permitiram criar os respectivos canais de comunicação, iniciar a estrutura de comunicação e ligar os utilizadores à rede.

Estes canais ao nível físico estão representados por *Socket_Global* e por *Socket_Local* (Fig. 1).

D. Mecanismo DDE

O mecanismo DDE[4] permite a diferentes aplicações numa máquinas comunicarem entre si (Fig. 1). Este mecanismo contém primitivas necessárias à iniciação, estabelecimento e troca de informação através de um canal de comunicação.

A este nível, foram implementados dois canais, denominados *DDE_Global* e *DDE_Local*, que estabelecem a ligação entre a camada inferior e superior (*DDE_SERVER* e *DDE_CLIENTE*) das duas aplicações (*Cplayer* e *Ckernel*). Estes canais dão continuidade à informação proveniente ou destinada à rede através da *Socket_Global* e/ou da *Socket_Local*.

IV. PROTOCOLO DE COOPERAÇÃO

Uma vez estabelecidos os vários canais de comunicação é indispensável, garantir que a cooperação se processe de uma maneira fiável, eficiente, fácil e organizada, ao qual se torna necessário impor um conjunto de regras, ou seja, elaborar um protocolo de cooperação.

O protocolo caracteriza-se por uma gestão e encaminhamento de mensagens, da responsabilidade do utilizador que inicia a conferência.

A. Estrutura do protocolo

O protocolo desenvolvido caracteriza-se pela permanente troca de mensagens do tipo *Request-&-Reply* entre um qualquer utilizador e o responsável por uma conferência.

Para tal, existe um conjunto de estruturas de controlo, organizadas por vários grupos de funções, que permitem o envio e recepção de mensagens, através das quais todos os participantes têm de passar, para receberem e/ou fornecerem a informação necessária, para que todos fiquem em sincronismo entre si. Essas funções (que representam as mensagens transmitidas entre *Cooperantes*) encontram-se enquadradas nos seguintes grupos:

⇒ “*Management of Conferences*” - Grupo de funções necessárias à gestão das conferências. Este grupo de mensagens flui apenas no *Canal Global*, ou seja, destinam-se a todos os utilizadores do sistema de conferências.

⇒ “*Management of Users in a Conference*” - Grupo de funções para controlar o acesso e a permanências dos utilizadores numa conferência. Tal como todos os grupos

de mensagens que se seguem fluem apenas no canal local, ou seja, são destinadas apenas aos utilizadores de uma conferência.

⇒ “*Global Cursor Control (Queues, Requests,...)*” e

⇒ “*Global Cursor Transmission*” - Grupo de funções para controlar o acesso, pedidos e transmissão do cursor comum.

⇒ “*Control of Public Window*” - Grupo de funções para manter o sincronismo da mesma área de trabalho entre os participantes.

⇒ “*File Handling*” - Grupo de funções que permitem identificar o acesso ao documento partilhado.

⇒ “*Conference Voting Mechanism*” - Grupo de funções que permitem a apresentação de uma proposta e a posterior votação.

⇒ “*Conference Message Transmission*” - Grupo de funções que permitem trocar comentários públicos ou privados entre um só ou vários utilizadores da conferência.

As entidades remotas de um sistema estabelecem uma comunicação virtual entre si por intermédio de várias camadas. Cada camada comunica apenas com as adjacentes através de um protocolo bem definido. Esta organização promove uma divisão clara de funções e da responsabilidade entre camadas (Fig. 3.).

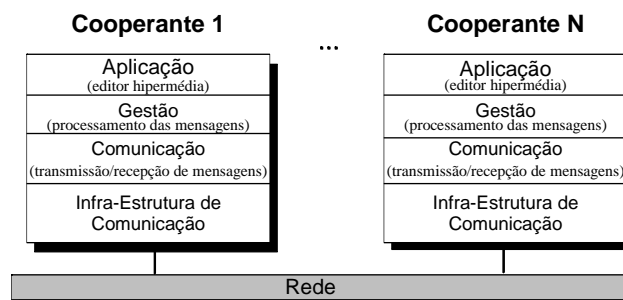


Fig. 3: Estrutura de comunicação.

•A **Infra-Estrutura de Comunicações** é responsável pela ligação física dos sistemas computacionais à rede. É aqui que se enquadra o protocolo TCP/IP e as aplicações desenvolvidas com Sockets e DDE's.

•A camada de **Comunicação** é responsável pela transferência fiável de informação vinda através da rede para ser posteriormente processada e, de enviar para a rede dados de camadas superiores, através da infra-estrutura de comunicação.

•A camada de **Gestão** é responsável pelo processamento da informação recebida e pelo correcto encaminhamento da informação a ser transmitida para cada participante através dos respectivos canais. Para isso, existem associadas a cada canal duas estruturas, uma, encarregue de processar a informação para ser transmitida no sentido **Canal_DDE** ⇒ **Canal_Socket** ⇒ **Rede**, a qual foi denominada por **GlobalRequest** e **LocalRequest**,

consoante o canal de que se trate. A outra, encarregue de receber a informação no sentido **Rede** ⇒ **Canal_Socket** ⇒ **Canal_DDE**, neste caso denominada por **GlobalPoke** e **LocalPoke**, consoante os respectivos canais.

•A camada da **Aplicação** comunica com a camada de Gestão para trocar mensagens específicas da aplicação (*Player* cooperativo) que informam os vários *Cooperantes* das alterações que vão sendo feitas sobre o estado, conteúdo ou apresentação do documento por um determinado *Cooperante*. Por outro lado, recebe essa informação através da camada de Gestão e processa essas alterações visualizando-as nos restantes *Cooperantes*.

B. Protocolo de gestão da conferência

O protocolo de gestão da conferência consiste na troca de mensagens que tratam da organização da cooperação, ou seja, são responsáveis por ligar os *Cooperantes* à conferência, em manter o sincronismo entre quem participa, em estabelecer uma conversação com quem coordena, organiza e controla a cooperação. Deve ainda difundir entre todos, quem tem a palavra, qual o documento presente no trabalho cooperativo, etc.

Quem inicia a cooperação fica com a coordenação e o controlo da conferência cooperativa. É também responsável pela gestão da cooperação pois atribui ou retira a palavra, quando achar necessário, a um *Cooperante*.

Os *Cooperantes* só “falam” quando o *Proponente* lhes atribui a palavra, isto é, só podem produzir acções sobre o documento nessa situação. Quando não têm a palavra limitam-se a receber mensagens vindas do *Proponente* que podem alterar o estado da cooperação, como por exemplo, modificação da área comum de trabalho, a atribuição do “*Cursor Distribuído*”, a existência de um novo documento aberto, etc. Nesta situação podem enviar um pedido de requisição da palavra, podem transmitir uma ideia ou processar um comentário público a um ou a vários *Cooperantes*. Podem de igual modo requerer um pedido de votação ao *Proponente* ou enviar uma mensagem no sentido de pretenderem abandonar a conferência.

Na atribuição da palavra os conflitos são evitados estabelecendo a regra de que apenas um utilizador *Cooperante* pode “ter a palavra” em cada instante, isto é, em cada momento só esse utilizador pode modificar o documento hipermedia. Ter a palavra reflecte-se no estado da cooperação pela posse do “*Cursor Distribuído*”. Enquanto o utilizador *Cooperante* altera o documento todos os outros *Cooperantes* reproduzem as suas acções, mantendo-se assim o sincronismo sobre o documento partilhado.

V. DEFINIÇÃO DO *PLAYER* COOPERATIVO

Os sistemas hipermedia são um mecanismo eficaz para aceder a grandes quantidades de informação com uma estrutura complexa. A sua característica não linear e

multimédia facilita a apreensão de conhecimentos e por esta razão o seu uso principalmente em sistemas de aprendizagem quer autónoma, quer assistida.

A. Conceitos básicos

Como já foi referido, o sistema designado por *Player* baseia-se na metáfora do “retroprojector”[5]. Assim, este sistema permite colocar sobre ele uma transparência com diversas ilustrações, passar para outras transparências consoante o seu seguimento e encadeamento. Pode-se inclusive sobrepor mais que uma transparência para combinar diversas informações. A seguir definiremos mais formalmente as entidades que constituem este sistema.

Objectos - Constituem as unidades elementares de informação multimédia e que podem ser comparadas às diferentes ilustrações de uma transparência.

Transparência - Unidade de apresentação e estruturação de informação multimédia manipulada pelo *Player* Cooperativo. A transparência é uma entidade constituída por vários objectos.

Lição - É uma forma de estruturação constituída por uma rede de transparências com ligações que as permitem percorrer de uma forma não linear. Uma lição é constituída por várias transparências relacionadas que permite uma navegação mais livre e não limitada a um número pré-definido de transparências.

VI. APRESENTAÇÃO DA INTERFACE GRÁFICA

⇒ O utilizador ao iniciar a aplicação encontrará a seguinte interface gráfica do *Player* Cooperativo.

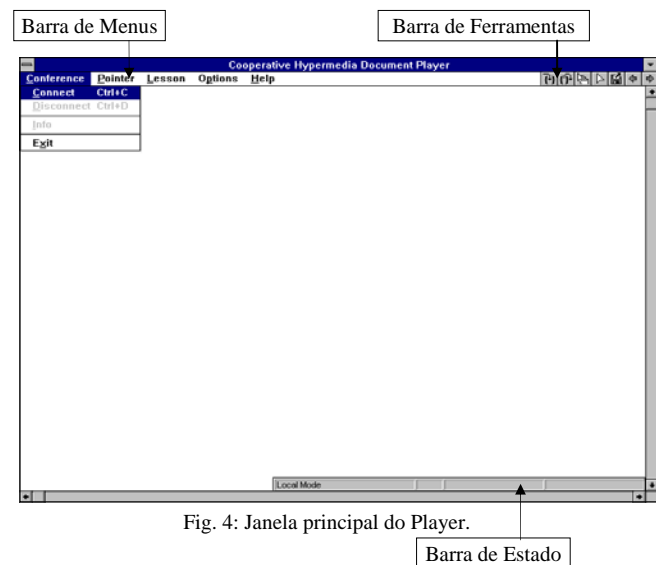


Fig. 4: Janela principal do Player.

Barra de Menus - possibilita o acesso às diversas opções disponíveis na aplicação.

Barra de Estado - fornece ao utilizador informação acerca do estado da conferência.

Barra de Ferramentas - permite um acesso directo às ferramentas disponíveis pelo *Player*.

⇒ Após o utilizador ter seleccionado a opção *Conference Connect* surge imediatamente uma janela de diálogo que permite que ele se identifique, escrevendo as iniciais do seu nome, *ShortName*, e o seu nome completo, *UserName*.

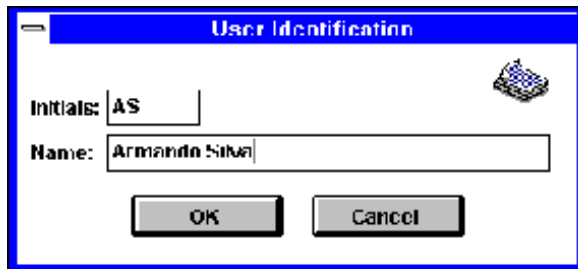


Fig. 5: Identificação do utilizador.

⇒ Caso não exista nenhuma conferência é dada imediatamente ao utilizador a possibilidade de criada uma nova, tornando-se este utilizador o *Proponente* da nova conferência. Este deve atribuir para tal um nome à conferência, seleccionar o seu tipo e modo, atribuir uma duração e o número de possíveis participantes.

A partir deste instante, a conferência está criada, a Barra de Estado é actualizada com o nome da conferência e caso o utilizador selecione a opção "*Pointer Request*" da Barra de Menus ou da Barra de Ferramentas, fica também com essa informação através do *ShortName* desse utilizador. Este utilizador *Proponente* está apto a aceitar a ligação de novos possíveis utilizadores.



Fig. 6: Atribuição dos parâmetros necessários para criar uma conferência.

⇒ Caso tenha conseguido a posse do "*Cursor Distribuído*", pode seleccionar a opção "*Lesson Open*" da Barra de Menus, para abrir um novo documento. Este que não estava anteriormente inactivo, ficou activada após a atribuição do "*Cursor Distribuído*".

⇒ Um exemplo de um documento hipermedia está representado na seguinte figura.

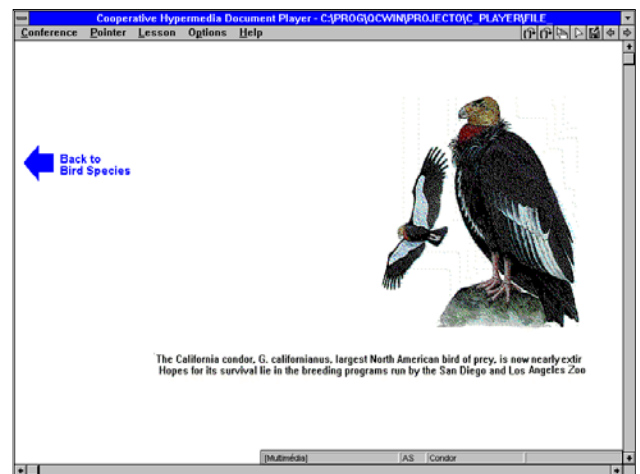


Fig. 7: Exemplo de um documento hipermedia.

⇒ Quando surge um novo utilizador, a sua aplicação encarrega-se de enviar um pedido de informação acerca das possíveis conferências que possam estar a decorrer, dirigido a todos os *Proponentes*. Caso estes respondam, o novo utilizador depara-se com uma janela de diálogo contendo as conferências existentes, o qual pode escolher a conferência à qual se pretende ligar ou, caso pretenda, ser ele mesmo a criar uma nova conferência. A partir deste momento, estabelece-se uma fase inicial de conversação com o *Proponente* dessa conferência através de um canal local previamente aberto para a ligação a essa conferência.

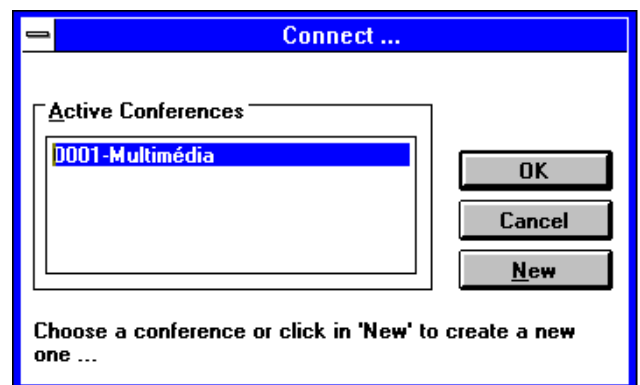


Fig. 8: Ligação de um utilizador a uma conferência seleccionada.

⇒ O *Proponente* pode aceitar, recusar ou propor a uma votação a entrada na conferência do novo utilizador. Caso seja aceite, este utilizador juntar-se-á à conferência, recebendo informação relativa à uniformização da área pública de trabalho, da lição que está a ser utilizada, do possuidor do "*Cursor Distribuído*", da existência de outros utilizadores, da lista de espera para ter a posse do "*Cursor Distribuído*".

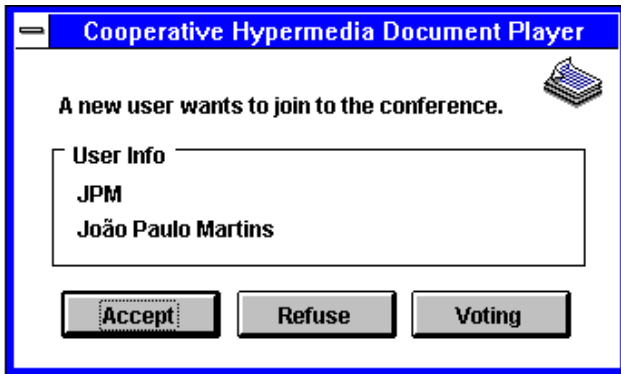


Fig. 9: O *Proponente* é informado da intenção de um utilizador pretender entrar na conferência.

⇒ Caso o novo utilizador seja aceite na conferência, o seu *Player* poderá apresentar por exemplo o seguinte documento.

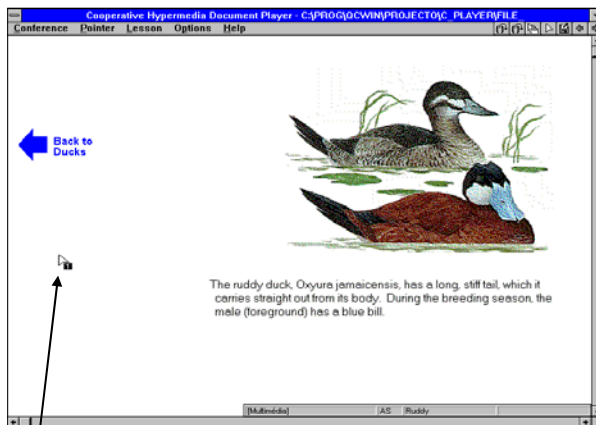


Fig. 10: Documento presente na conferência.

Cursor distribuído

⇒ Qualquer utilizador pode aceder à informação geral acerca da conferência, seleccionando a opção "*Conference Info*". Se o utilizador em causa for o *Proponente*, tem opções que permitem-lhe remover da conferência um determinado participante ou retirar qualquer pedido da lista de espera do "*Cursor Distribuído*", seleccionando o utilizador pretendido e premindo de seguida "*Revoke User from Conference*" ou "*Revoke User from Queue*", respectivamente. Estas opções estão disponíveis na janela de diálogo "*Conference Info*".

Na figura seguinte, por exemplo, o *Proponente* requereu um "*Conference Info*", na qual se pode verificar a existência de dois participantes e que o utilizador com o *UserID* 0001 - AS tem a posse do "*Cursor Distribuído*", para além de outra informação adicional.

Se tivesse sido qualquer outro utilizador participante, esta janela de diálogo não apresentaria os botões que permitem remover os utilizadores ou os seus pedidos da lista de espera, opções essas, só acessíveis ao *Proponente*.

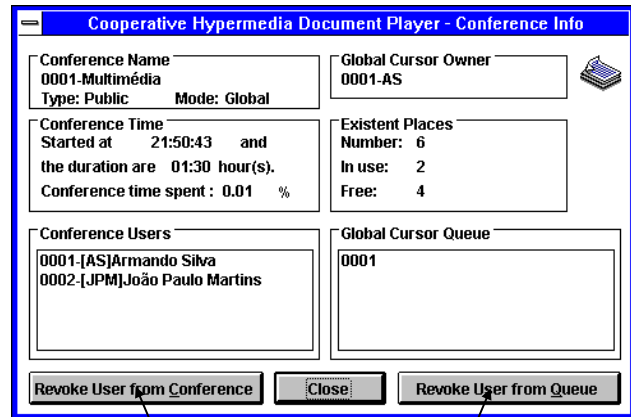


Fig. 11: Janela de diálogo da "*Conference Info*".

Botões só existentes para o *Proponente*

⇒ Qualquer participante pode utilizar vários serviços disponíveis, durante o período em que decorre a conferência. Uma dessas possibilidades consiste em exercer um mecanismo de votação.

Um utilizador participante deve pedir antes, autorização ao *Proponente* para poder levar a sua ideia a um processo de votação. Um exemplo deste processo está apresentado nas seguintes janelas de diálogo.

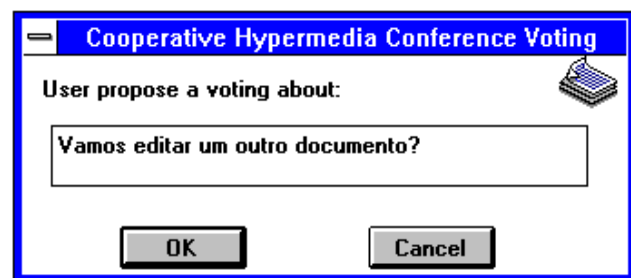


Fig. 12: Janela de diálogo usada para propor uma ideia a votação.

Todos os restantes participantes recebem esta mensagem e podem exercer a sua votação durante alguns instantes perante a seguinte janela de diálogo.

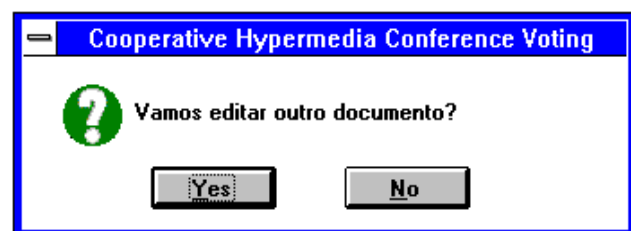


Fig. 13: Permite exercer um voto perante um assunto em votação.

O resultado desta votação é comunicado a todos os *Cooperantes* ao fim de alguns momentos, após se ter iniciado este mecanismo.

⇒ Um outro mecanismo disponível consiste que qualquer participante, poder exprimir uma ideia ou um comentário dirigido a um utilizador em particular ou a um conjunto de utilizadores previamente seleccionados. Para isso basta seleccionar a opção "*Send Message*", da Barra de Menus.

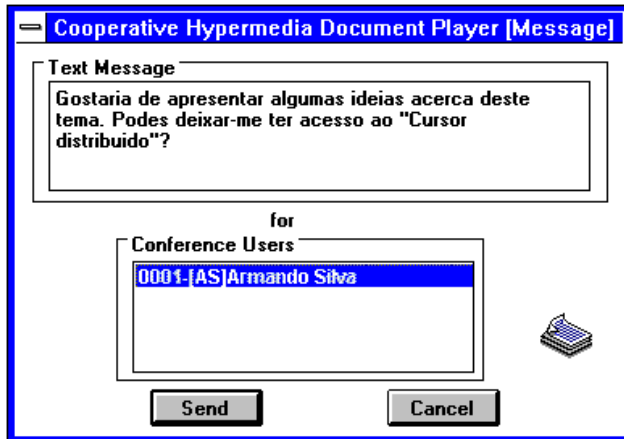


Fig14: Mensagem ou comentário dirigida a um ou a vários *Cooperantes* seleccionados.

Neste exemplo, o participante 0001-AS foi o seleccionado para receber a mensagem, o qual respondeu usando a mesma janela de diálogo ao utilizador que lhe enviou a mensagem.

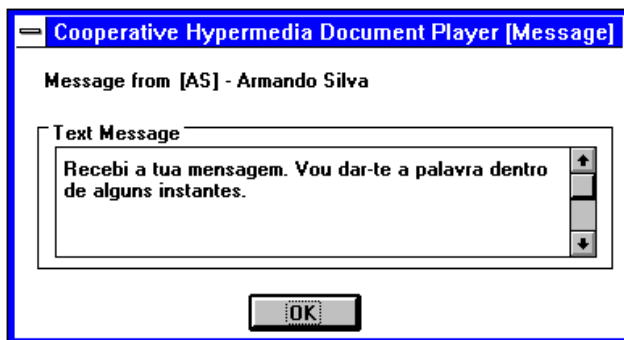


Fig. 15: O *Cooperante* seleccionado recebe um comentário na sua estação de trabalho.

⇒ Em determinada altura dar-se-á por concluída a conferência. Alguns instantes antes, o *Proponente* recebe essa informação e caso pretenda poderá atribuiu-lhe uma nova duração ou questionar os participantes se valerá apenas ou não ser prolongada a sua duração. A conferência termina deste modo ou por iniciativa do *Proponente* antes da sua duração expirar.

Existem muitas outras mensagens que poderão surgir no decorrer da conferência, que permitem manter um diálogo ou manter informados os participantes de algum procedimento ocorrido, como por exemplo um procedimento incorrecto da parte desse utilizador.

VI. CONCLUSÃO

Este projecto permitiu adquirir o conhecimento dos mecanismos utilizados na comunicação entre computadores em ambientes de trabalho cooperativo, utilizando principalmente o protocolo de comunicação TCP/IP.

Como foi referido, este protocolo caracterizou-se pela constante troca de mensagens do tipo *Request-&Reply* entre os participantes. Durante a sua elaboração veio a verificar-se que em determinadas situações este processo não era totalmente eficiente, resultando daí a perda de algumas mensagens, com a conseqüente perda de sincronismo dos sistemas computacionais.

De referir ainda o grande estudo que teve que ser feito para a aquisição de conhecimentos no ambiente funcional onde se enquadra este projecto (contacto com a filosofia CSCW) e devido ao facto de este projecto já estar numa fase inicial de elaboração (por parte do eng. Joaquim Sousa Pinto). Ainda de referir, a necessidade que houve da aprendizagem de uma nova linguagem e filosofia de programação (programação C/C++ para ambiente Windows) que foi a base de elaboração deste projecto.

REFERÊNCIAS

- [1] Douglas E. Keiser, "Transmission Control Protocol/Internet Protocol - Internet Working with TCP/IP", Practice Hall.
- [2] Charles Petzold's, "Windows For Workgroups - Programming Windows 3.1", Microsoft Press, 1992.
- [3] Microsoft Press, "Object Linking and Embedding - Programmer's Reference", 1992.
- [4] Gordona S. Smith, "Dynamic Data Exchange - Simplify Client-Server Applications Using DDE Management Library", MSJ 1992 (Dec.);
- [5] Joaquim Sousa Pinto, Hubert W.J. Borst Pauwels, Joaquim Arnaldo Martins, Beatriz Sousa Santos, "HyDE: a hypermedia document editor based on OLE technology", Actas do IEEE ICMAS'94, Boston, EUA, Maio de 1994.