

## Sistema Multimodular de Trabalho Cooperativo

Pedro Brêda dos Santos, Nelson Pacheco da Rocha

**Resumo** – O presente artigo descreve um Gestor de aplicações para sistemas de trabalho cooperativo. O Gestor, além de disponibilizar uma plataforma para o desenvolvimento de sistemas de trabalho cooperativo, pretende apostar na multimodularidade como forma de promover suporte ao desenvolvimento de sistemas específicos e/ou para utilizadores específicos. Isto é feito através da concentração das funcionalidades genéricas no Gestor, do reaproveitamento e reconfiguração de módulos e tirando partido da segmentação de trabalho no desenvolvimento de software. A multimodularidade tem, no entanto, o inconveniente de desagregar o sistema nos seus módulos constituintes, pelo que ao Gestor cabe também suportar a integração local dos módulos numa estação compacta.

**Abstract** – This paper presents an application Manager for cooperative work systems. Besides making available a basis for the development of computer cooperative work systems, the Manager also pretends to use multimodularity as a means to promote support to the development of specific systems and/or for specific users. This is done by the concentration of the generic functions on the Manager, by the reuse and reconfiguration of modules and by taking advantage of the segmentation of work within the development of the software. The multimodularity has, nevertheless, the disadvantage of disintegrating the system in its component modules, thus making it necessary for the Manager to support the local integration of the modules in a compact station.

### I. INTRODUÇÃO

O trabalho cooperativo suportado pelo computador (*Computer-Supported Cooperative Work - CSCW*) tem duas vertentes principais: a adição de capacidade cooperativa a tarefas que já são realizadas individualmente no computador, e a utilização do computador como suporte alternativo a trabalho cooperativo que já é efectuado de forma convencional (local e pessoal). A primeira destina-se a utilizadores familiarizados com o computador e promove a interacção entre o sistema cooperativo e o ambiente do computador. Por seu lado, a segunda destina-se a utilizadores que podem nunca ter tido um contacto prévio com o computador, devendo por isso isolar o utilizador do ambiente do computador e implementar todas as

funcionalidades do sistema de acordo com as especificidades dos utilizadores.

A generalidade do software para trabalho cooperativo não prevê ou não dá suporte a esta segunda vertente. Alguns, como o sistema CoNus [1], têm como uma das principais vertentes a gestão de dados existentes no computador e a sua delimitação entre dados privados e dados públicos, organização de dados, autorizações, negociações, etc. Isto porque o sistema foi encarado para utilizadores que trabalham com o computador, e o sistema é apenas um alargamento do seu trabalho para trabalho cooperativo. O espaço de trabalho é todo o ambiente do computador (sistema de ficheiros e memória), incluindo aplicações e informação, pelo que implementam apenas um conjunto de ferramentas e uma interface de acesso ao conteúdo do computador, ou pelo menos a parte dele, funcionando à base de colaboração transparente, genérica (partilha de aplicações) ou particular (partilha de informação, ou seja, espaços de trabalho). Neste caso o sistema de trabalho cooperativo funciona como uma extensão ao sistema operativo, sendo, por isso, genérico.

Na outra perspectiva o computador é usado apenas como suporte alternativo a actividades de colaboração entre utilizadores, sendo, por isso, apenas o meio. A base e o destino são os utilizadores, que possivelmente nunca tiveram antes contacto com computadores. A generalidade dos sistemas que se enquadram nesta segunda vertente é específica a cada actividade, podendo basear-se em colaboração transparente ou consciente, mas apenas a segunda consegue dar resposta à especificidade de todos os casos.

O presente trabalho pretende dar resposta à segunda vertente, isto é, à especificidade do sistema e do utilizador, sem deixar de o fazer relativamente à primeira relacionada com a extensão do sistema operativo. No entanto, a solução para a capacidade de resposta aos vários sistemas específicos tem de ser a generalidade, através de uma plataforma comum, mas de suporte e não de soluções finais. Desta forma consegue-se abarcar as duas vertentes, permitindo o suporte à construção de sistemas mais ou menos específicos. Não são por isso disponibilizadas interfaces de utilizador, mas interfaces a conjuntos de funcionalidades, preferencialmente opcionais. Cabe aos módulos implementar as especificidades do sistema com as respectivas interfaces finais. Trata-se de separar a cooperação em parte genérica para o *Gestor* e em parte específica para os módulos.

Como se disse, o maior entrave a esta vertente é a especificidade a que obriga, implicando elevados tempos de desenvolvimento e pouco reaproveitamento de software de uns sistemas para outros. Para ultrapassar esta situação, este trabalho propõe a multimodularidade. Esta, quando aplicada a qualquer sistema de software, permite obter as seguintes vantagens:

- Isolar uma componente genérica a todos os sistemas (um *Gestor*) que sirva de plataforma de suporte para o seu desenvolvimento, evitando o começar do zero.
- Facilitar a segmentação de trabalho por uma equipa de desenvolvimento, permitindo reduzir o tempo de desenvolvimento.
- Dividir um sistema específico em funcionalidades mais básicas, potencialmente com maior possibilidade de reaproveitamento de uns sistemas para outros.

A multimodularidade, além das vantagens apontadas, acarreta também alguns inconvenientes, nomeadamente ao nível da desintegração local da estação. Cabe então ao *Gestor* anular essas desvantagens, disponibilizando suporte à integração local dos módulos.

Ao combinar a cooperação e multimodularidade espera-se também combinar as vantagens da cooperação entre sistemas integrados com os sub níveis de cooperação entre módulos gémeos em terminais remotos.

## II. REQUISITOS

### A. Colaboração

Pretende-se que as estações de trabalho cooperativo sejam independentes, permitindo ligações a outros clientes sem dependência de estações centrais, e autónomas, permitindo as mesmas actividades em modo local, como forma de melhor aproveitamento dos recursos disponíveis e também de treino. A arquitectura suportada ou implementada pelo *Gestor* é, por isso, relativamente à distribuição, do tipo distribuição total. De qualquer forma, a computação deveria ser distribuída para rentabilizar a banda de transmissão e para promover a especificidade de cada estação. Como, por outro lado, pelo menos parte da colaboração genérica não se pode separar da computação, e os módulos implementam, além da computação, a parte específica das componentes de colaboração e coordenação, não se justifica a existência de unidade central, e a dependência que acarreta, apenas por algumas funções genéricas de coordenação e colaboração. Mesmo que algumas funções genéricas de coordenação sejam atribuídas dinamicamente durante uma sessão a um dos terminais, todos possuem essa potencialidade, podendo usá-la a qualquer instante.

Quanto à génese, a arquitectura é do tipo colaboração consciente em que as próprias aplicações contêm as componentes de colaboração e coordenação. A colaboração transparente (as aplicações partilhadas não se apercebem da colaboração) tem limitações ao nível da definição de uma configuração individual e ao nível de

uma configuração comum. A implementação de sistemas específicos (para utilizadores específicos) pode exigir ambos, uma ligação entre actividades remotas (estado comum) e o respeito pelas diferenças individuais (estado individual), obrigando à utilização de colaboração consciente.

A colaboração consciente exige um desenvolvimento dedicado e muitas vezes trabalhoso, para a obtenção dos sistemas finais, mas é a única via possível para a especificidade pretendida; a colaboração transparente é um compromisso. Por outro lado, é precisamente a função do *Gestor* facilitar esse desenvolvimento, fornecendo o suporte possível de modo a evitar o desenvolvimento de raiz.

### B. Estado e Espaço de Trabalho

Pretende-se também que os módulos sejam autonomamente executáveis em modo local sem dependência do *Gestor*. O papel do *Gestor* deve ser apenas de adicionar as dimensões de cooperação e multimodularidade. Por isso o *Gestor* não deve implementar o estado e espaço de trabalho da estação (os objectos de cooperação) nem a sua interface de acesso, mas apenas a sua distribuição e o controlo de concorrência. O suporte relativamente aos objectos deve ser apenas relativo à distribuição remota (cooperação) e local (multimodularidade).

Por outro lado, sendo uma das funções do *Gestor* a de manter os objectos de cooperação durante a ausência dos módulos e inicializar novos módulos com objectos, o *Gestor* deve permanecer sempre activo. Paralelamente, o estado e espaço de trabalho, depois de criados, têm de continuar a existir independentemente da presença dos módulos.

Portanto, o estado e o espaço de trabalho devem ser independentes quer dos módulos quer do *Gestor*, mas acessíveis por ambos, pelo que os objectos não podem ser encapsulados por nenhum deles. Para isso, existem duas hipóteses, ou existem numa área global comum, ou são reproduzidos no(s) módulo(s) e no *Gestor*. O estado de um sistema CSCW é, devido à sua pequena dimensão, facilmente reproduzido nos vários módulos e *Gestor*. Aliás a sua cópia é não só fácil como conveniente, por forma a permitir uma maior eficiência no processamento individual dos módulos. Ao contrário, a reprodução do(s) espaço(s) de trabalho pode ser muito pouco eficiente, devido ao espaço adicional ocupado e, principalmente, à necessidade de distribuição local. Mesmo quando um espaço de trabalho é exclusivo de apenas um módulo local, deve ser implementado numa área comum, por forma a facilitar a sua distribuição remota, bastando ao módulo indicar ao *Gestor* as áreas alteradas. Os espaços privados de um módulo podem ser implementados internamente pelo mesmo, a menos que se pretenda também a sua manutenção entre execuções.

O suporte à distribuição de objectos pode ser indirecto através de funcionalidade genéricas de distribuição de

dados, ou directo através de um mecanismo genérico de distribuição automática de estado e espaços de trabalho com controlo de concorrência. É objectivo do *Gestor* concentrar as funcionalidades genéricas de cooperação, pelo que deve tomar a seu cargo a distribuição automática dos objectos de cooperação. Além disso, a gestão de objectos comuns a várias entidades locais e remotas simultaneamente (requerendo controlo de concorrência) pode ser bastante complexo, sendo de todo o interesse retirar essa carga aos módulos. O *Gestor* deve então disponibilizar um mecanismo automático genérico em que os módulos apenas indiquem ao *Gestor* directamente (no caso do estado) ou indirectamente (no caso do espaço de trabalho) as alterações e este distribui-as a todos os restantes módulos, locais ou remotos, encarregando-se do controlo de concorrência. Este mecanismo deve, como se disse atrás, ser capaz de manter os objectos independentemente dos módulos através de réplicas no caso do estado e referências no caso dos espaços de trabalho. O mecanismo também deve ter em conta a natureza dos objectos: privados, públicos ou comuns.

O controlo de concorrência entre os módulos locais é facilmente resolvido pelo *Gestor*, que devido à sua condição de instância única, assume o papel de árbitro. Já o controlo de concorrência entre módulos remotos obriga a um protocolo mais complexo entre os vários *Gestores* das diversas estações.

Para uma distribuição automática eficiente de espaços de trabalho de média ou grande dimensão torna-se necessário a partição do espaço de trabalho em subsecções, permitindo ao *Gestor* distribuir apenas as secções alteradas. Quando um módulo escrever no espaço de trabalho deve notificar o *Gestor*, indicando as secções alteradas, por forma a serem distribuídas. Esta partição é também conveniente relativamente ao controlo de concorrência. Naturalmente, pela sua pequena dimensão, as variáveis de estado não levantam este problema.

Com vista à optimização de tráfego, nas operações sobre espaços de trabalho possíveis de serem reproduzidas remotamente, deve-se distribuir apenas a operação e não o resultado. Por outro lado, apenas se deve distribuir os objectos aos módulos neles interessados, sendo por isso conveniente a utilização de um mecanismo de subscrição.

Apesar de tudo o referido com vista a obter o máximo de generalidade, devido às especificidades de alguns casos, principalmente nos espaços de trabalho, nem sempre será possível ou conveniente a utilização do mecanismo distribuição automática. Por isso, como suporte alternativo, o *Gestor* deve disponibilizar também funcionalidades genéricas de comunicação. Neste caso, a componente de coordenação (controlo de concorrência) e o restante da componente de colaboração ficarão a cargo dos módulos, que poderão para a primeira usar variáveis de estado comum, como se verá adiante. Por outro lado, os canais de comunicação podem ser também usados para outros fins, como por exemplo, teleponteiros, áudio, vídeo, etc. As funcionalidades genéricas de comunicação

servem, portanto, como suporte geral à implementação da parte específica de cooperação por parte dos módulos.

Os mecanismos de distribuição, automáticos de objectos ou de comunicação simples deverão suportar vários formatos de informação, incluindo ficheiros por forma a disponibilizar suporte ao conteúdo do computador como espaço de trabalho.

### C. Coordenação de Participantes

A coordenação de um sistema CSCW é composta por duas funções, uma ao nível do utilizador e outra ao nível da informação: a gestão do universo de participantes (incluindo acesso ao sistema) e a resolução de conflitos provenientes da distribuição de informação. Da segunda falou-se nas secções anteriores. Vamos agora abordar a coordenação de participantes.

Vários artigos sobre CSCW discutem métodos de coordenação de participantes, mas esta função é, naturalmente, dependente do tipo de sistema e de participantes. Por exemplo, alguns sistemas, como sejam os casos de distribuição de ensino e assistência remota, requerem uma coordenação desequilibrada em termos de terminais.

Se a coordenação de dois participantes pode desde logo requerer as suas especificidades, a coordenação de grupos de participantes pode ser, além de específica, bastante complexa. De facto, um dos segmentos importantes do estudo de sistemas CSCW refere-se à coordenação de grupos de utilizadores, mas não é possível definir um mecanismo genérico quando se pretende dar resposta à especificidade do utilizador. Uma ligação simultânea entre mais do que dois utilizadores acarreta um acréscimo significativo de complexidade na coordenação dos vários utilizadores, ao nível do funcionamento e ao nível da interface, mesmo para os participantes que sejam apenas "coordenados". Esta complexidade pode ser indesejável para certo tipo de utilizadores, pelo que a coordenação deve ser adaptada a cada caso. As crianças são um exemplo típico de utilizadores difíceis de coordenar. Por um lado, se não forem alvo de atenção permanente podem perder interesse ou atenção pela conferência. Por outro, pode ser difícil disciplinar crianças (e não só ...) em termos de limitação de participação, de forma a não se sobreponem à participação dos outros elementos. No caso de áudio e vídeo, existe também a problemática da multiplexagem, que pode ser feita em modo comutado ou em modo misturado.

Esta função deve, pelo exposto, ser deixada a cargo dos módulos, incluindo-se na parte de cooperação específica a implementar pelos mesmos. Isto não invalida que se disponibilize um mecanismo opcional genérico de gestão de participantes que permita uma plataforma de partida para os sistemas específicos, e eventualmente uma definitiva para os que nesse campo não apresentem especificidades.

#### D. Conferência Multi-ponto

Para a implementação da capacidade multi-ponto, é necessária a existência de um mecanismo de difusão simultânea, directa ou indirecta, a todos os participantes. Com vista a negociações individuais e/ou de inicialização, pode também ser conveniente, embora não absolutamente necessário, a manutenção paralela da capacidade de transmissão individualizada.

Nas redes de comutação de pacotes, como a Ethernet ou Internet, a difusão é um mecanismo simples, sendo a única limitação a banda de transmissão, que além de não ser constante, é limitada superiormente pela capacidade de tráfego da rede.

As redes de comutação de circuitos, apesar de permitirem ligações de banda fixa, e em princípio dimensionáveis à necessidade em causa, são orientadas a ligações ponto-a-ponto. Por isso, para formar uma conferência multi-ponto numa rede de comutação de circuitos, é necessário recorrer a uma topologia em anel, malha ou estrela. Numa primeira análise, todas elas implicam um acréscimo das infra-estruturas de hardware de comunicação. A topologia em malha é a menos conveniente dada a multiplicação de ligações necessária em cada terminal. Ambas as topologias, anel ou estrela, usam o número total de ligações mínimo, igual ao número de terminais. A topologia em anel necessita de capacidade para duas ligações por terminal e apresenta outras desvantagens ao nível do tempo de transmissão, complexidade e robustez. A topologia de estrela é mais simples e rápida, e apenas utiliza uma ligação por terminal, concentrando todo o acréscimo de infra-estruturas num terminal ou servidor central. Tem, no entanto, por inconvenientes o custo elevado do servidor central e a dependência da sua presença em todas as conferências, o que retira autonomia aos terminais e pode reduzir a fiabilidade do sistema.

A escolha entre as topologias em anel e estrela é dependente de cada caso. A velocidade de transmissão é, naturalmente, um aspecto importante e muitas vezes crucial, mas quando se pretende uma forte difusão do sistema cooperativo, é necessário reduzir ao mínimo os seus custos, que mesmo para um terminal com hardware de suporte para apenas um canal pode ser bastante elevado. A topologia em anel pode apresentar menos custos em RDIS se se tirar partido dos dois canais B de um acesso básico para a implementação das duas ligações. É claro que isto elimina a possibilidade de usar os dois canais para a mesma ligação, seja para dados, áudio ou vídeo. A topologia em estrela, por seu lado, também pode ser mais económica se houver muitos terminais, casos em que geralmente apenas um subconjunto está em ligação simultaneamente. Nesse caso, a capacidade de ligações necessária no servidor central seria bastante menor, reduzindo portanto o conjunto de infra-estruturas do mesmo. A topologia estrela pode ainda ser mais conveniente nos casos em que se pretenda uma conferência desequilibrada controlada pela unidade

central, ou em que se necessita de um servidor central para as funções da computação.

A topologia estrela pode ganhar uma vantagem decisiva com o apoio dos novos serviços de teleconferência da rede pública, que são baseados precisamente em unidades centrais que têm funções de controlo e multiplexagem de dados, áudio e vídeo. A entidade responsável por definir as normas europeias de Telecomunicações, o European Telecommunication Standard Institute (ETSI), define uma série de serviços de conferência para a RDIS. Alguns (CONF [2], MMC [3] e 3PTY [4]) são apenas aplicáveis sobre serviços básicos de voz. Para conferências que incluam transmissão de outros dados para além de voz, existem os Serviços de Teleconferência (STC) [5], que são dois: o serviço de Conferência Audiográfica [6, 7] e o serviço de Videoconferência [8]. Ambos suportam áudio e dados, podendo incluir informação visual estática. O segundo diferencia-se do primeiro pelo suporte adicional ao vídeo animado. Os STC, quando a conferência inclui três ou mais participantes, fazem uso de uma unidade central com capacidade de ligação múltipla a vários terminais e de processamento/multiplexagem das várias ligações. Os STC apresentam versatilidade sobre vários aspectos, nomeadamente ligações originadas pelo participante ou pela rede, coordenação equilibrada ou desequilibrada e banda de transmissão por canal atribuída dinamicamente.

Os STC ainda não se encontram suficientemente difundidos, mas são, conjuntamente com a Internet, os suportes mais difundidos e com maiores perspectivas de futuro no que se refere a conferências remotas.

O *Gestor* deve separar a componente genérica de cooperação da especificidade da rede, canais de comunicação e respectivos serviços associados, permitindo, entre outras vantagens, facilitar futuras evoluções e adaptações. A topologia de rede deve, assim, ser isolada pelo nível de rede (nível 3) ou de transporte (nível 4) do modelo OSI [9].

#### E. Áudio e Vídeo

As comunicações áudio e vídeo podem ser importantes à cooperação remota, principalmente para pessoas pouco familiarizadas com o computador. No entanto, áudio e vídeo em tempo real acarretam ainda hoje um volume de processamento muito elevado para um sistema de software, pelo que são geralmente implementados por hardware, ficando o software apenas com o seu controlo. Por outro lado, os sistemas são rapidamente desactualizáveis, principalmente em vídeo, não permitindo estabelecer plataformas firmes.

Os novos serviços de teleconferência da rede pública (RDIS) incluem as funcionalidades de áudio e vídeo, e podem consistir numa alternativa de qualidade relativamente à Internet, em termos de velocidade e fiabilidade. Estes serviços permitem fazer a multiplexagem de áudio e vídeo por comutação através de

testemunho, ou por mistura. No caso do vídeo a mistura é feita através da composição de imagem.

O *Gestor* não inclui, nesta fase, suporte a áudio ou vídeo, que ficam tal como as restantes especificidades do sistema, a cargo dos módulos. Isto deve-se, além de alguns dos factores mencionados, ao facto de já haver um sistema de videotelefonía implementado na versão anterior do sistema em que o *Gestor* foi aplicado.

### III. APRESENTAÇÃO DO SISTEMA

O *Gestor*, corporiza, de uma forma geral, as funções e soluções até agora apontadas relativamente ao suporte à cooperação e multimodularidade para sistemas e utilizadores específicos. Neste contexto, segue a perspectiva da generalidade como melhor forma de suportar a especificidade.

A arquitectura implementada ou suportada pelo *Gestor* é de distribuição total e de colaboração consciente, cabendo aos módulos a implementação das funções específicas de cooperação (colaboração e coordenação). O *Gestor* segue, então, uma concepção de sistema cooperativo global tendencialmente equilibrado, formado por estações independentes, capazes de moldar o sistema global e de implementar as suas funcionalidades de acordo com as suas próprias especificidades. Cada estação é baseada num computador operado e controlado por vários módulos (e *Gestor*) em sintonia, do ponto de vista de execução, interface gráfica e informação. Cada módulo pode ser autónomo, executável independentemente dos restantes, mas, quando em execução simultânea, os módulos apoiam-se no *Gestor* e unem-se, formando um sistema completo e compacto. A estação assim formada assume-se com uma identidade própria, com capacidade de comunicação remota com outras estações suportadas pelo *Gestor*.

#### A. Funcionalidades

O *Gestor* implementa três conjuntos principais de funcionalidades. Os dois primeiros são associados, respectivamente, às vertentes local e remota do sistema, ou, o mesmo é dizer, às dimensões de multimodularidade e cooperação. O terceiro apoia-se nos dois primeiros para disponibilizar um serviço integrado de distribuição local e remota, fazendo a fusão, nesse serviço, das duas vertentes ou dimensões. Os três conjuntos de funcionalidades são compostos da seguinte forma:

- Suporte à integração (sincronização) local dos módulos num sistema compacto: controlo de execução dos módulos (código); suporte a ligações activas com parâmetros entre módulos (código e dados); integração das interfaces de utilizador (entrada e saída).
- Suporte à comunicação remota, ou seja, suporte base de cooperação: implementação e gestão de um canal comum de comunicação remota com as restantes estações para o qual o *Gestor* aplica o seu protocolo

interno de comunicação com *Gestores* gémeos remotos, isto é, entre estações; disponibilização aos módulos de transmissão livre de mensagens, blocos globais e ficheiros entre módulos remotos. Esta função permite aos módulos implementarem a parte específica de cooperação do sistema.

- Suporte integrado à distribuição local e remota de estado espaços de trabalho, ou seja suporte directo e completo à multimodularidade e cooperação. Este suporte integrado é feito através de mecanismos genéricos de distribuição automática de estado e espaços de trabalho do sistema a todas as estações e módulos que se designam de bases de distribuição ou simplesmente bases. Estes mecanismos asseguram, além da distribuição dos objectos, a sua manutenção na ausência dos módulos e o controlo de concorrência nos modos ordenado ou controlado.

O *Gestor* implementado disponibiliza ainda um conjunto de funcionalidades adicionais:

- Comunicações remotas privadas em modo transparente, ou seja, com protocolo deixado a cargo dos módulos. Este suporte não visa um fim específico mas pretende disponibilizar um serviço multi-uso. Exemplos podem ser o acesso a uma base de dados remota ou o controlo de dispositivos remotos.
- Serviços gráficos por omissão para gestão das interfaces com os utilizadores e gestão de ligações.
- Serviço de registo de eventos, com finalidade de *debug*, estatística, etc.

Por forma a obter um *Gestor* o mais genérico possível, todas as funcionalidades são opcionais, permitindo sistemas mono-módulo, mono-interface, sem comunicações, etc. Para tal, o *Gestor* é configurável por um ficheiro de texto, que tem como parâmetros principais uma lista de módulos e uma lista de interfaces com o mapa de módulos associados a cada uma delas. Entre outros parâmetros existe a configuração das interfaces gráficas por omissão e gestão de ligações. Estas permitem pequenos ajustes e podem ser simplesmente desactivadas e as interfaces implementadas pelas aplicações.

A relação do *Gestor* com o restante sistema em termos de interface gráfica é ilustrada na Figura 1. O *Gestor* implementa um painel, onde os vários módulos "afixam" a sua interface. A dimensão do painel é configurável, podendo ou não isolar completamente o sistema operativo. É também sobre este painel que operam as interfaces gráficas por omissão já referidas. A gestão da ocupação do espaço do ecrã é deixada a cargo dos módulos que se afixam no painel implementado pelo *Gestor*.

#### B. Bases de Distribuição

As bases de distribuição são, como se disse, mecanismos genéricos de distribuição automática, local e remota, de variáveis de estado e espaços de trabalho, integrando, por isso, o suporte à multimodularidade e à cooperação. A distribuição consiste na notificação automática aos vários

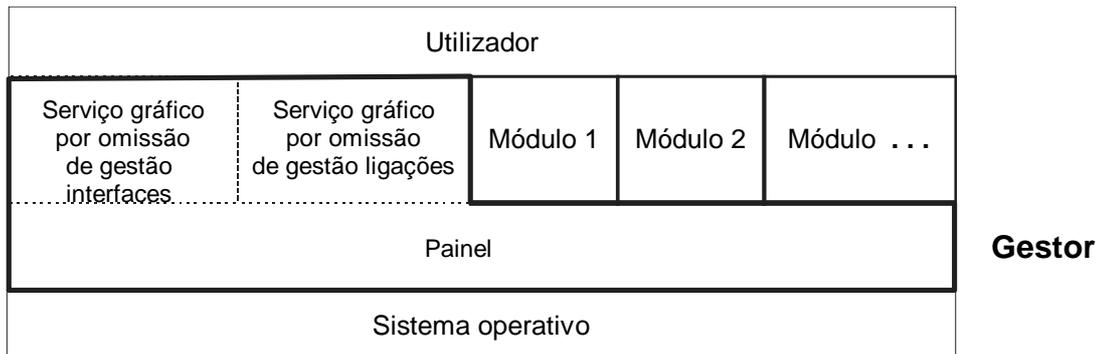


Fig 1 - Estrutura de um sistema de software baseado no *Gestor* relativamente à interface de utilizador.

módulos de todas as alterações efectuadas sobre um objecto. Esta distribuição garante controlo de concorrência em modo ordenado ou controlado, conforme indicado pelos módulos em cada operação de escrita. A distribuição consiste ainda na inicialização dos módulos com os objectos correntes, sendo as bases responsáveis pela manutenção dos objectos na ausência dos módulos. As naturezas distintas dos objectos a distribuir vai originar a existência de várias bases. Elas dividem-se em bases de estado e bases de trabalho por um lado, e em bases privadas, bases públicas e bases comuns por outro, totalizando seis bases.

As bases de estado são formadas por um conjunto de variáveis de estado, fazendo uso da reprodução da variável em cada módulo e no *Gestor*. Sempre que uma variável sofre uma operação de escrita, o seu novo valor é distribuído automaticamente. O controlo de concorrência é efectuado sobre cada variável individualmente, permitindo um aproveitamento muito mais eficiente do tempo de processamento do sistema. As bases de estado suportam os formatos numéricos e sequência de caracteres. Esta última pode funcionar como identificador, indicando desse modo um estado.

As bases de trabalho, por seu lado, não fazem uso da reprodução da informação, havendo apenas uma instância do espaço por estação em área comum. Estas bases são, por isso, formadas por um conjunto de referências para objectos partilhados. A escrita é feita directamente no objecto, e a distribuição utiliza uma referência para o objecto. As bases de trabalho operam não sobre espaços de trabalho, mas sobre segmentos de espaços, permitindo uma maior eficiência na transmissão e no controlo de concorrência que é efectuado relativamente aos vários segmentos individualmente. Os formatos suportados para espaços de trabalho são blocos globais de memória e ficheiros.

As bases privadas são formadas pelos objectos (variáveis de estado e espaços de trabalho) privados de uma estação, mas comuns a mais do que um módulo local. Os objectos exclusivos de um módulo local são naturalmente implementados internamente pelo próprio módulo. Dada a sua natureza privada, os objectos são totalmente de âmbito local, relativamente à distribuição e à escrita. São bases exclusivamente de multimodularidade. No caso da

utilização do *Gestor* em sistemas não cooperativos, apenas estas bases serão usadas.

As bases públicas gerem objectos públicos, isto é, individuais a uma estação em termos de escrita, mas com acesso de leitura pelas estações remotas. Nesta base cada objecto é formado, não por uma instância, mas por uma matriz unidimensional de instâncias, em que a primeira é a da estação local e as seguintes das estações remotas. A primeira apenas pode ser escrita localmente, e as restantes apenas pelas estações remotas correspondentes. Todas as instâncias são distribuídas localmente, enquanto que apenas a instância local é distribuída remotamente. Esta perspectiva assume uma correspondência entre os objectos públicos das várias estações. Esse é o caso mais natural, dada a usual similaridade de estações remotas e portanto dos módulos remotos. No entanto, se tal não acontecer, o mecanismo funciona igualmente, sendo a base formada pela reunião de todas os objectos públicos de todas as estações e ficando vazios os valores correspondentes às estações que não possuam esse objecto.

As bases comuns gerem objectos comuns, isto é, objectos partilhados por escrita, sendo, por isso, idênticas em todas as estações. Os objectos comuns podem ser escritos por qualquer entidade, local ou remota, e são distribuídos local e remotamente.

As bases comuns e públicas existem numa estação mesmo quando esta não está em ligação. Nesse caso um objecto dessas bases terá o mesmo efeito de um objecto privado. No entanto, quando a estação entra em ligação, as bases comuns e públicas são distribuídas automaticamente e cada tipo de objecto assume o seu papel específico. A distribuição das bases comuns e públicas é feita por fusão. No caso das bases públicas é feita a reunião dos objectos de todas as estações. No caso das bases comuns, a determinação da estação ou grupo de estações que prevalece é determinado por negociação no estabelecimento da ligação.

As bases de trabalho públicas podem, no caso de haver muitas estações em sessão, gerar muito tráfego na inicialização de estações ou módulos. Por isso, apenas devem ser usados espaços de trabalho públicos quando necessário, usando a base comum quando possível.

De um modo genérico cada módulo não utiliza todos os objectos, mas apenas um subconjunto dos mesmos. Para evitar tráfego desnecessário, faz-se uso de um mecanismo de subscrição, em que cada módulo regista no *Gestor* os objectos de cujas alterações pretende ser notificado. A escrita, por seu lado, pode acontecer em qualquer altura sobre qualquer objecto desde que não seja público remoto. Quando duas estações se ligam é feita a distribuição automática das bases entre *Gestores*, mas quando um módulo se inicializa a distribuição dos objectos não é automática, tendo o módulo de subscrever as variáveis que necessitar. Para evitar tráfego exagerado de subscrição em sistemas que usem muitos objectos e em que muitos são subscreitos, geralmente variáveis, a subscrição suporta o uso de "templates", isto é, referências a grupos de objectos.

Um objecto fica registado no *Gestor* quando este recebe a primeira referência, seja de escrita ou subscrição. Deste modo, o critério de criação, gestão e partilha dos objectos de estado é da responsabilidade das entidades utilizadoras, sejam módulos ou o próprio *Gestor* que também utiliza as bases de estado para comunicação interna, com módulos locais e com as estações remotas.

As bases apresentam, então, um conjunto significativo de vantagens relativamente aos métodos convencionais de difusão remota e partilha local de objectos, sendo as principais: notificação automática de alterações, manutenção dos objectos na ausência dos módulos, o controlo de concorrência e o sistema de subscrição.

As bases garantem transmissão em modo ordenado ou controlado, definível em cada operação de escrita. No modo ordenado apenas é evitado o conflito de ordem, enquanto que no modo controlado é evitado o conflito de decisão garantindo que apenas uma acção é aceite em cada instante. As bases privadas e públicas apenas suportam a escrita local, situação que garante o modo ordenado. Ao contrário, as bases comuns têm de lidar com escritas de entidades remotas pelo que, para garantir uma consistência dos objectos entre entidades remotas, é necessária a utilização de mecanismos de controlo de concorrência. Para a obtenção de modo controlado em qualquer das bases é também necessário o controlo de concorrência. Para o efeito é usado um mecanismo de selo ou um mecanismo de testemunho.

Os objectos públicos podem também necessitar de um modo controlado remoto para evitar realimentação positiva em casos de dependência mútua. Como porém a dependência pode não ser da variável gémea, o controlo destes casos é deixado às entidades utilizadoras, que podem usar uma variável comum para desempenhar o papel de testemunho. Note-se que os participantes podem usar variáveis de estado comuns em modo controlado para a implementação de testemunhos para o controlo de outras acções, por exemplo comunicação sobre variáveis públicas ou comunicação convencional.

A interface das bases de distribuição é formada por três primitivas: uma para escrita do objecto, outra para subscrição do objecto e uma terceira usada pelo *Gestor*

para distribuição dos objectos. Os parâmetros destas primitivas diferem das bases de estado para as bases de trabalho, mas o significado de cada primitiva é idêntico em ambas as bases.

Ao contrário da escrita sobre variáveis de estado, a escrita sobre espaços de trabalho é mais complexa e pode levar a duas situações:

- A operação apenas pode ser realizada na estação origem, pelo que o resultado é escrito directamente no espaço e transmitido a partir do mesmo.
- A operação é possível de ser realizada em qualquer estação, pelo que é distribuída a operação e não o resultado. A operação é executada por todas as estações quando receberem a notificação, incluindo a origem.

As bases de trabalho suportam, então, três tipos de operações:

- A escrita inicial do resultado no espaço para a distribuição do resultado a partir do espaço local. Esta operação requer um testemunho, por forma a garantir que apenas uma entidade em cada tempo escreve sobre um segmento de espaço.
- A distribuição de operações definidas e implementadas pelos módulos. As operações são executadas pelos módulos quando recebem a notificação.
- Um conjunto de cinco operações básicas de manuseamento definidas e implementadas pelo *Gestor*. Estas operações são executadas pelas bases antes de notificarem os módulos do segmento alterado.

As cinco operações básicas disponibilizadas pelas bases são:

- Fill - Preenche um segmento indicado com um determinado número.
- Insert - Insere um segmento num espaço, inicializando-o com um determinado número.
- Cut - Elimina um segmento indicado, resultando num espaço final menor.
- Copy - Copia o conteúdo de um segmento para cima de outro segmento.
- Move - Elimina um segmento de um local e insere-o noutra posição.

Para a distribuição de outras operações, além destas cinco, possíveis de serem executadas remotamente, são distribuídos códigos de operação definidos pelos módulos que serão reconhecidos pelos módulos das várias estações. Nesse caso são os módulos a implementar a operação. Em qualquer dos casos as operações têm, de uma forma geral, parâmetros que podem ser indicação de segmentos, posições no espaço ou outros necessários à realização da operação pelas entidades remotas.

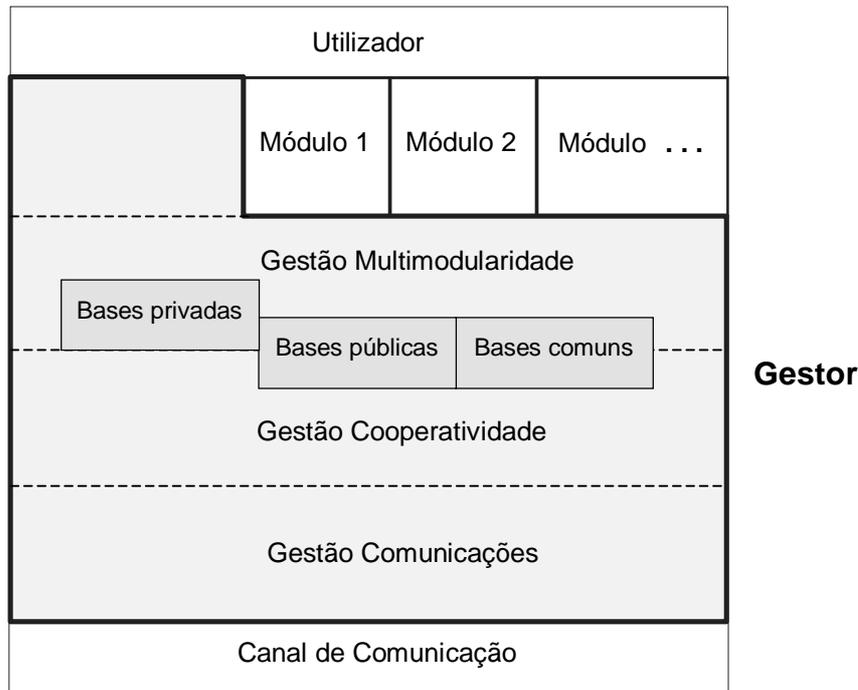


Figura 2 - Estrutura de um sistema baseado no *Gestor* sob o ponto de vista de comunicação.

### C. Dualidade Objecto de Cooperação - Canal de Comunicação

O mecanismo de base de distribuição é, apesar de simples, bastante versátil. Permite criar um canal de comunicação activo (com notificação) entre um conjunto de módulos através de um objecto. Sempre que um objecto é alterado, todos os subscritores são notificados, recebendo o seu valor ou referência. Assim, é possível a qualquer conjunto de módulos, locais ou remotos, implementar entre si um canal de comunicação multi-ponto através da criação de um objecto para o efeito. A entidade emissora não necessita de especificar os destinatários, pois estes ao subscreverem o objecto definem o canal. Temos então um mecanismo de dupla utilização: distribuição de objectos e criação de canais de comunicação. Estabelece-se, assim, uma dualidade objecto - canal de comunicação.

Pode-se estabelecer um conjunto de canais cujos participantes serão apenas as entidades que saibam o nome desse canal (objecto) e o subscrevam. Se um sistema particular necessitar de um canal aberto genérico, pode implementá-lo através de um objecto, por exemplo "Geral", que todos os módulos subscreverão.

As bases de estado são bastante limitadas na dimensão do bloco de informação, sendo adequadas para a implementação de canais de controlo. Ao contrário, as bases de trabalho permitem blocos de informação de dimensão variável e utilização de segmentos diferentes, estando assim vocacionadas para a implementação de canais de dados.

Tal como a distribuição de objectos, também a acção de comunicação tem disponível o controlo de concorrência nos modos ordenado ou controlado. Tem também a possibilidade de usar objectos privados, públicos ou comuns, conforme o tipo de canal que pretender estabelecer e o controlo de concorrência que pretender obter.

Para que um objecto possa implementar um canal de comunicação, todos os acessos de escrita são notificados, mesmo que seja escrito o conteúdo já presente, nomeadamente nas variáveis de estado. Apenas desta forma é possível manter a sequência original.

As diferenças principais entre a utilização das bases de distribuição como canais de comunicação e os mecanismos convencionais de transmissão de dados são a definição de canais particulares através da subscrição e o facto de a última mensagem ficar armazenada.

### D. Estrutura Geral do Gestor

O *Gestor* organiza-se numa estrutura de quatro camadas de gestão conforme representado na Figura 2. Estas quatro camadas podem-se subdividir em dois grupos, relativos às duas vertentes do *Gestor* - cooperação e multimodularidade. Na base estão as camadas associadas ao trabalho cooperativo: a camada de comunicações e a camada de cooperação. Sobre estas assentam as camadas associadas à multimodularidade: a camada básica de multimodularidade e a secção suplementar de gestão das interfaces com os utilizadores. Esta última funciona paralelamente aos vários módulos, podendo ser considerada como um módulo implementado dentro do *Gestor*.

De acordo com esta arquitectura, a cooperação não é dividida em colaboração e coordenação, mas em cooperação genérica, composta pelas camadas de gestão de cooperação e gestão de comunicações do *Gestor*, e em cooperação específica, efectuada pelos módulos. A computação é formada pelos módulos e pelas camadas de multimodularidade e organização das interfaces do *Gestor*. A camada de comunicações, tal como foi sugerido anteriormente, deve encapsular as especificidades relativas a redes e canais de comunicação.

As bases de distribuição públicas e comuns reúnem ambas as vertentes de multimodularidade local e cooperação remota, pelo que atravessam as camadas de multimodularidade e cooperação, fazendo a fusão de ambas e encarando assim a multimodularidade como "cooperação local". As bases privadas, que apenas fazem uma distribuição local, vão naturalmente inserir-se apenas na camada de multimodularidade.

#### IV. CONCLUSÕES

Apresentou-se de uma forma funcional e genérica um *Gestor* para sistemas multimodulares de trabalho cooperativo designado de *Gestor*. A implementação actual do *Gestor* [10] foi suportada em Windows e utiliza a infra-estrutura de comunicação RDIS.

Deve-se salientar que a implementação do *Gestor* foi orientada de modo a permitir aos módulos serem aplicações autónomas em modo local, com vista a facilitar reutilizações dos serviços por eles implementados. Por outro lado, o modo como foi implementada a comunicação módulos-*Gestor* permite também a cada módulo comunicar com outra instância sua a correr no mesmo computador sem a presença do *Gestor*, como se estivessem em terminais remotos a comunicar através do *Gestor*.

O grau de modularidade e flexibilidade do *Gestor* foram testadas com sucesso no âmbito de um projecto europeu, o CANS (programa TELEMATICS) que teve como objectivo o estudo do impacto de serviços remotos de apoio à comunidade, o que serviu para demonstrar a utilidade de um *Gestor* genérico como plataforma para a implementação de sistemas específicos.

#### V. REFERÊNCIAS

- [1] Walter Reinhard, Jean Schweitzer, Gerd Volksen, Michael Weber, "CSCW Tools: Concepts and Architectures", IEEE Computer, Maio 1994.
- [2] ETS 300 183, "Integrated Services Digital Network (ISDN); Conference Call, Add-on (CONF) Supplementary Service", Outubro 1992.
- [3] ETS 300 164, "Integrated Services Digital Network (ISDN); Meet-Me Conference (MMC) Supplementary Service", Outubro 1992.
- [4] ETS 300 186, "Integrated Services Digital Network (ISDN); Three-Party (3PTY) Supplementary Service", Julho 1993.
- [5] CCITT Recommendation F.710 (1988) - "Teleconference Equipment".
- [6] ETS 300 101, "Integrated Services Digital Network (ISDN); International Digital Audiographic Teleconference", Fevereiro 1993.
- [7] ETS 300 675, "Integrated Services Digital Network (ISDN); Audiographic Conference Teleservice", Setembro 1995.
- [8] ETS 300 678, "Integrated Services Digital Network (ISDN); Videoconference Teleservice", Setembro 1995.
- [9] Andrew S. Tanenbaum, "Computer Networks", Prentice-Hall International, Inc., 1989.
- [10] Pedro Brêda "Sistema Multimodular de Trabalho Cooperativo", Dissertação de Mestrado Apresentada à Universidade de Aveiro, 1998.