

## AGENTES MÓVEIS EM OPERAÇÃO REMOTA: UMA APLICAÇÃO EM TELE-ROBÓTICA

L. M. Camarinha-Matos\*, Walter Vieira\*\*\* e Octavio Castolo\*

\* *Universidade Nova de Lisboa — Faculdade de Ciências e Tecnologia  
Quinta da Torre, 2825-114 Monte Caparica, Portugal*

*Tel.: +351-1-2954464 / ext. 3728 Fax: +351-1-2957786*

\*\* *Instituto Superior de Engenharia de Lisboa, DEEC, Portugal  
{cam, wv, lgc}@uninova.pt*

**Resumo:** Neste artigo é apresentado o uso de agentes móveis adaptáveis na operação remota, permitindo respostas em tempo real apesar das limitações dos canais de comunicação, tais como os atrasos de tempo, a sua pouca disponibilidade e falta de segurança. É proposta uma arquitectura envolvendo agentes estacionários e agentes móveis autónomos, com capacidade de monitoração de execução e recuperação de erros. As aplicações potenciais abrangem desde a tele-robótica tradicional até os laboratórios virtuais ou assistência à terceira idade, onde os agentes móveis prestam assistência aos utilizadores em comunidades virtuais. Também são apresentados os resultados práticos num cenário onde um robô tipo SCARA é comandado remotamente através da Internet.

Palavras chave: Agentes móveis, Tele-operação, Monitoração.

### 1. INTRODUÇÃO

A operação e supervisão remotas, em geral, bem como os laboratórios virtuais, registam um interesse crescente devido às suas potenciais aplicações, abrangendo desde a partilha de equipamentos caros, o comando remoto de máquinas em ambientes perigosos ou inacessíveis, até a operação de veículos espaciais com grande autonomia. Em particular, o número de aplicações usando a operação remota através da Internet tem crescido significativamente em vários domínios, tais como os robôs e telescópios operados remotamente, os sistemas de manufactura, a assistência remota de idosos, sistemas remotos de vigilância, entre muitos outros.

Neste artigo é considerado o caso específico da Internet como infraestrutura de suporte, já que os seus baixos custos e ampla disponibilidade a tornam atraente como uma base para a operação remota. A

aproximação proposta tenta responder a algumas dificuldades, tais como:

- i) Quando são considerados domínios práticos de aplicação com alguma complexidade podem esperar-se altos níveis de heterogeneidade na riqueza sensorial e de equipamento nos sítios remotos, os quais podem degradar a flexibilidade e a escalabilidade do sistema;
- ii) A Internet é ainda caracterizada por padecer de longos e variáveis atrasos de tempo e por sofrer, frequentemente, de baixos níveis de disponibilidade, levantando novos desafios em termos quer da segurança do sistema implementado, quer da sua dependência em relação às características da rede; e
- iii) Os ambientes de execução remotos são potencialmente não estruturados e incertos, o que significa que não é adequado recorrer a sistemas programados deterministicamente.

Deste modo, num sistema de operação remota com base na Internet deve ser dada especial atenção aos aspectos relacionados com os atrasos de tempo e a fraca disponibilidade das conexões da rede, bem assim como a heterogeneidade e a incerteza dos ambientes de execução.

Neste artigo é discutida uma solução para permitir a operação remota em canais de comunicação com grandes atrasos de tempo e/ou deficiente disponibilidade, e são apresentados os resultados da sua aplicação no controlo remoto de um robô em tarefas de montagem. A aproximação é baseada em agentes móveis adaptáveis (Fuggeta, *et al.*, 1998; Kotz e Gray, 1999; White, 1997) os quais transportam planos abstractos hierárquicos de alto nível que são depois adaptados às capacidades ou recursos encontrados em cada sítio remoto visitado (Camarinha-Matos e Vieira, 1999a), (Camarinha-Matos, *et al.*, 2000). Os agentes móveis são entidades de software autónomas dotadas de capacidade de, durante a sua execução numa máquina, decidirem deslocar-se para outra máquina, sendo o seu estado corrente “congelado”, transferido juntamente com o código do agente e recuperado na máquina destino. Os planos hierárquicos aqui considerados incluem informação que permite guiar os agentes na adaptação do plano, na monitoração de execução e nos processos de recuperação de erros.

A aproximação proposta de (multi-)agentes móveis constituiu também uma base para a implementação de sistemas de manipulação remota ou de laboratórios virtuais. As potenciais aplicações incluem o acesso remoto a experiências num cenário de pesquisa cooperativa, especialmente no caso de instalações caras somente disponíveis em alguns centros. Outra área de aplicação é o ensino à distância, especialmente para programas de treino ao longo da vida activa. O acesso remoto aos laboratórios virtuais de aprendizagem dá ao estudante a oportunidade de se “sentir próximo” à experiência em curso. Para além das mencionadas vantagens genéricas da aproximação proposta de agentes móveis, a aplicação aos laboratórios virtuais pode também beneficiar da autonomia dos agentes a fim de não ser necessária a sincronização dos participantes numa determinada experiência. Os participantes podem delegar nos agentes que os representam a realização efectiva de alguma tarefa, que será levada a cabo quando estejam satisfeitas as condições necessárias para o seu sucesso. Esta aproximação permite altos níveis de disjunção entre os participantes na experiência, em termos espaciais e temporais, pois eles não precisam estar fisicamente presentes no sítio onde decorre a experiência e também não necessitam participar sincronizadamente *on-line* na mesma.

## 2. A INTERNET

Vários exemplos de conexão de robôs, câmaras e outros dispositivos à Internet têm sido desenvolvidos

nos últimos anos. Um dos primeiros exemplos foi o célebre *The Telegarden* (ver no sítio *web*: [http://ranier.oact.hq.nasa.gov/telerobotics\\_page/realrobots.html](http://ranier.oact.hq.nasa.gov/telerobotics_page/realrobots.html)) em que um robô munido duma câmara permite que utilizadores remotos observem e interajam com um pequeno jardim sob “guarda” desse robô. Os utilizadores podem observar o desenvolvimento das plantas e mesmo comandar o robô para semear novas plantas ou regar o jardim. Outras experiências similares têm sido realizadas em áreas como a montagem robotizada, operação de robôs móveis, visitas (virtuais) a museus, etc. Importantes desenvolvimentos têm também tido lugar na tele-medicina.

Contudo, como em muitas outras infra-estruturas de comunicação, a Internet, na sua versão actual, impõe restrições severas no que concerne à operação remota em tempo real, principalmente no que respeita a:

**Latência (atrasos de tempo).** Esta característica tem sido reportada por vários autores, e as soluções que têm sido propostas para domínios muito específicos caem em duas classes principais. Numa das classes a latência é estimada e apresentada de algum modo ao operador (por exemplo, um cursor pode ser colocado na imagem da cena que descreve o estado remoto actual apontando para a nova posição esperada para o braço do robô após o atraso de tempo calculado). Na segunda classe, a inteligência e a autonomia do sítio remoto são aumentadas a fim de permitir estratégias de supervisão tais que o operador apenas fornece comandos de alto nível ao sítio remoto quando este não pode prosseguir pelos seus próprios meios. O primeiro caso apenas é aplicável a domínios muito específicos, enquanto o segundo reduz a dependência das características da rede à custa da flexibilidade, isto porque qualquer mudança de funcionalidade ou a alteração dos ambientes remotos requer a actualização de software em todos os sítios remotos. Em alternativa, a solução aqui apresentada é baseada em agentes móveis que são enviados para o sítio remoto para implementar alguma funcionalidade desejada (Fig. 1). Como esses agentes se executam no sítio remoto, são alcançados altos níveis de independência nas características da rede, preservando contudo os altos níveis de flexibilidade (já que podem ser construídos novos agentes móveis sempre que sejam necessários para realizar as funcionalidades desejadas).

**Deficiente disponibilidade da rede.** Isto requer altos níveis de autonomia no sítio remoto, o qual deve possuir os recursos para permitir a supervisão da execução, mesmo em caso de perda temporária da ligação com a estação de comando. Essa autonomia deve permitir adoptar estratégias inteligentes de recuperação aquando da ocorrência de erros. Na nossa proposta, os agentes móveis transportam consigo planos abstractos hierárquicos (missões) que incluem anotações com informação para monitoração e recuperação de erros. As anotações de monitoração guiarão o agente nas acções de supervisão. É possível

embeber estratégias sofisticadas de execução e recuperação de erros nos planos abstractos porque estes podem ser gerados em *off-line* por planeadores de iniciativa mista. Também, a estrutura hierárquica dos planos permite um controlo mais fácil da complexidade do plano de reparação durante a recuperação do erro.

**Heterogeneidade dos ambientes de execução.** Este é um aspecto crucial quando se considera a operação remota em grande escala. Não é prático ter um agente diferente para cada sítio remoto equipado de forma diferente, mesmo naqueles casos onde a funcionalidade desejada a certo nível de abstracção seja a mesma. Neste trabalho, os planos abstractos carregados pelos agentes são adaptados (isto é, refinados) conforme as características dos sítios remotos “visitados” (Fig. 1). Por outras palavras, ao chegar a um determinado local remoto, o agente móvel refinará o seu plano, transformando-o num plano executável, de acordo com os recursos e funcionalidades aí encontrados.

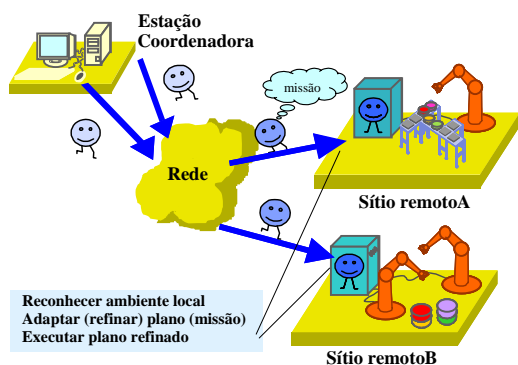


Fig. 1 - Agentes móveis na operação remota

Tal como no ponto precedente, a natureza hierárquica dos planos abstractos é a chave para controlar a complexidade da adaptação do plano. Esta capacidade de refinamento é, naturalmente, limitada ao domínio alvo para o qual o agente foi projectado.

### 3. UMA ARQUITECTURA PARA AGENTES MÓVEIS ADAPTÁVEIS

A fim de suportar a aproximação referida acima, foi desenvolvida uma arquitectura para agentes móveis adaptáveis designada de IMAJ (*Intelligent Mobile Agents in JAVA*), a qual compreende três componentes principais:

1) A componente de mobilidade, que implementa o processo de migração do agente (ver (Camarinha-Matos e Vieira, 1998) para mais detalhes desta componente). IMAJ utiliza mobilidade fraca. Quando um agente se move de um servidor para outro, os atributos de todos os seus objectos estáticos são guardados e transferidos juntamente com o seu código (usando o mecanismo de serialização do JAVA) para o servidor destino. Cada servidor, quando recebe um agente, cria uma

linha de execução própria sob a qual o agente se executará (Vieira e Camarinha-Matos, 2000).

2) A componente de coordenação é responsável pela implementação da infraestrutura que permite aos agentes coordenar a sua actividade. A solução adoptada é baseada no modelo de coordenação Linda (Rowston e Wood, 1996), como forma de reduzir os acoplamentos espacial e temporal. Basicamente, cada agente tem acesso a um espaço de informação partilhado por todos os agentes que se encontram a executar no mesmo sítio, através do qual se envolve em interacções com outros agentes que se encontrem nesse sítio. Para além deste espaço de informação local, cada agente tem acesso ao espaço de informação do sítio onde foi originalmente lançado, através do qual se pode envolver em interacções com agentes que aí permaneçam e com outros agentes lançados do mesmo sítio (Cabri e Zambonelli, 1998). Este último espaço de informação é objecto de memorização local no agente e permite a sincronização com o sítio de origem sempre que se consegue uma ligação estável.

3) A componente de supervisão da execução, que é a mais elaborada, implementa os mecanismos que permitem a um agente adaptar o plano abstracto de alto nível que transporta ao ambiente particular que encontra em cada sítio visitado e executar o plano adaptado, incluindo a monitoração da execução e os processos de recuperação de erros, em caso de falhas (Vieira e Camarinha-Matos, 1999).

A supervisão da execução envolve a adaptação do plano, a monitoração da execução e a recuperação de erros. A adaptação do plano permite a um agente refinar o seu plano de alto nível para a sua execução em cada sítio, de acordo com as capacidades ou recursos aí encontrados; a monitoração da execução e a recuperação de erros são cruciais porque os agentes podem estar em operação autonomamente por longos períodos de tempo.

Em contraste com a maioria dos trabalhos precedentes em supervisão da execução, que supõem domínios de aplicação muito restritos, neste trabalho propõem-se soluções mais gerais visando um amplo leque de domínios de aplicação, tendo por consequência que a actual composição de cada ambiente de execução não pode ser conhecida com antecedência. Para além disso, ao tratar da operação remota, os agentes devem funcionar com um alto grau de autonomia em ambientes incertos. Para conseguir este objectivo foi adoptada uma aproximação baseada na monitoração geral e em métodos de recuperação, os quais junto com as anotações inseridas nos planos abstractos ajudarão na monitoração da execução e na recuperação de erros. A estrutura hierárquica dos planos adoptada, além das tradicionais vantagens do planeamento hierárquico, permite a especificação dos monitores a vários níveis de detalhe, o que se torna muito apropriado para domínios complexos.

Outro aspecto importante quando se considera a resposta em tempo real é que a execução de uma tarefa não deve causar degradações significativas na execução de outras tarefas, o que torna inapropriadas as soluções de execução sequencial encontradas em muitos outros sistemas (Vieira, 2001). Por este motivo a arquitectura da componente de supervisão da execução adoptada faz uso exaustivo de programação concorrente (Fig. 2). Os agentes móveis têm uma arquitectura de programação concorrente, a qual é uma característica fundamental para permitir a implementação do paralelismo necessário entre a execução do plano e a monitoração de eventos.

Nesta arquitectura, as acções executáveis (EA – *executable actions*) são realizadas de forma concorrente e interagem com o ambiente local através dum conjunto de actuadores. Admite-se que o ambiente local de execução está equipado com os adequados mecanismos de monitoração e recuperação de erros ao nível de execução, de tal maneira que se a execução de uma EA falhar, é obtida suficiente informação do ambiente para caracterizar a falha. Esta informação é usada para a recuperação de erros de alto nível.

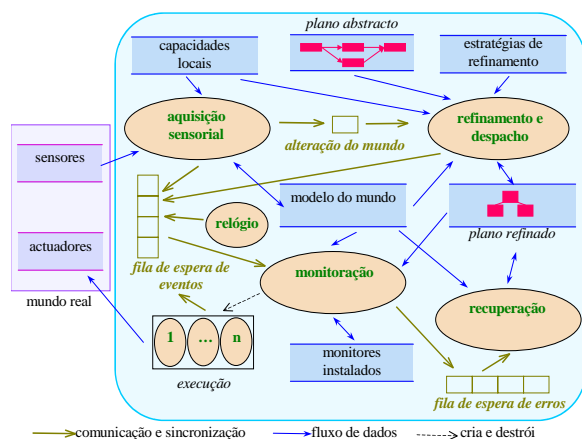


Fig. 2 - Arquitectura da componente de supervisão da execução de um agente móvel

A tarefa de **aquisição sensorial** é responsável por recolher informação que permita a manutenção de um modelo actualizado do estado do mundo. O ritmo de recolha deve ser suficiente para que, atendendo à dinâmica da aplicação, não se percam variações significativas. Esta tarefa gera um evento sempre que detecta uma alteração do estado do mundo, altura em que também sinaliza o semáforo binário **alteração do mundo**.

A tarefa de **refinamento e despacho** funciona em duas fases: i) adaptação do plano abstracto ao ambiente de execução, e ii) selecção, após obtido o plano refinado, das acções executáveis a colocar em execução, em cada momento, na **fila de espera de eventos**. O semáforo binário **alteração do mundo** serve para sinalizar esta tarefa de que ocorreram alterações no mundo e proceder ao teste de existência de novas acções para execução.

A tarefa de **monitoração** é responsável por controlar a entrada em execução e finalização das acções executáveis, pela instalação e remoção dos monitores especificados e pelo teste das suas condições. Em caso de detecção de erro por parte de um monitor, esta tarefa coloca uma descrição adequada desse erro na **fila de espera de erros** para que a tarefa de recuperação os processe.

A tarefa de **recuperação** é responsável pela execução das estratégias de recuperação associadas aos vários monitores e por proceder às alterações do plano refinado que delas resultem.

A tarefa de **relógio** gera eventos ao ritmo de um relógio com determinado frequência, cujo objectivo é marcar tempos no sistema para efeitos de detecção de falhas por excesso de tempo.

Para mais pormenores sobre a arquitectura de execução, consultar em (Viera e Camarinha-Matos, 1999) e em (Vieira, 2001).

A especificação dos planos abstractos é feita através da linguagem MAAPL (*Mobile Agents Abstract Plan Language*) (Camarinha-Matos e Vieira, 1999b). Esta linguagem permite a definição de planos abstractos hierárquicos incluindo a anotação de diversos monitores (Reece e Tate, 1994) e indicação de diversas estratégias de recuperação quando o monitor é activado.

Na implementação actual, os seguintes tipos de monitores podem ser especificados: **monitores de pré-condições**, que são activados se uma condição definida nas pré-condições da acção falha; **monitores de efeitos**, os quais são activados se uma condição definida nos efeitos da acção falha; **monitores de manutenção**, os quais são usados para a observação contínua de alguma condição; **monitores de falha**, que verificam se uma acção ao nível hierárquico imediatamente inferior termina com sucesso; e **monitores de excesso de tempo**, que se activam se a acção não termina dentro do limite de tempo especificado. Cada monitor tem associado uma das estratégias de recuperação seguintes: **repair** é a estratégia por omissão dos monitores de pré-condições e tenta encontrar um plano para reparar as condições que falharam; **retry** implica a repetição da acção com o seu actual refinamento; **redo** indica que se deve tentar um novo refinamento para a acção, caso seja possível; **ignore** é a estratégia por omissão dos monitores de efeitos e ignora o erro; **fail** é usada para propagar a falha para os níveis superiores na hierarquia do plano; e **user** permite especificar uma estratégia de recuperação definida pelo utilizador, através de uma função indicada.

O refinamento hierárquico na linguagem MAAPL pode seguir um dos seguintes padrões: **seq**, que descreve uma sequência de sub-acções; **choice**, que especifica um conjunto de acções, onde uma delas deve ser escolhida; **par**, que especifica um conjunto

de acções cujos refinamentos serão executados concorrentemente; **execute**, que permite ao utilizador (programador) especificar a função que implementa a funcionalidade associada à acção; e **adapt** e **goal**, que especificam que um plano deve ser obtido localmente tomando em conta as capacidades do ambiente. A diferença entre **adapt** e **goal** é que **adapt** toma as pré-condições da acção como o estado inicial do planeador enquanto **goal** considera o estado real no momento de começar a sua execução. O primeiro é usado para obter planos que são executados como protocolos e, também, requerem que a execução do plano refinado tenha origem nas pré-condições definidas na acção de nível superior; o segundo é usado para situações menos restritivas onde só se pretende alcançar com sucesso os efeitos definidos na acção.

Como ambiente de desenvolvimento, a plataforma IMAJ é baseada em Java e nos sistemas JESS (sistema de regras “forward”) e JINNI (um interpretador de Prolog).

#### 4. UM CENÁRIO DE AVALIAÇÃO

Para a avaliação da solução proposta foi considerado um cenário onde é executada uma versão simplificada do teste de Cranfield (montagem robotizada dum pêndulo especialmente desenhado para avaliar o desempenho de robôs de montagem). As experiências consistem em montar três peças (ver no esquemático da Fig. 4) usando uma estratégia de controlo a nível de tarefa.

A estação remota é composta por um robô tipo Scara (SONY SRX-4CH), um conjunto de três garras, uma paleta para montagem, e dois dispensadores de peças: um para peças do tipo A e outro para peças do tipo B.



Fig. 3 - O cenário de avaliação

Este ambiente foi enriquecido com um conjunto de sensores infravermelhos para a detecção de peças em diversas posições, tal como é mostrado no desenho esquemático da Fig. 4. Também foi incluída uma pequena câmara de vídeo de forma de fornecer imagens do processo de execução das tarefas para o lado do cliente ou estação coordenadora.

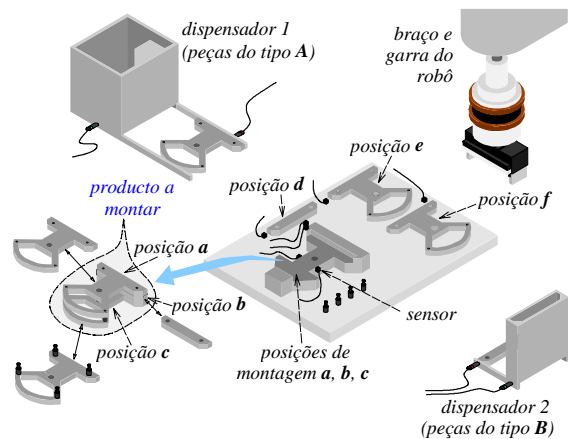


Fig. 4 - Esquema do ambiente remoto

Foi integrado neste ambiente um servidor IMAJ (o servidor remoto). Um segundo servidor IMAJ (o servidor cliente ou estação coordenadora) foi instalado noutra máquina a qual comunica com o primeiro servidor através da Internet. Neste servidor foi iniciado um ambiente para receber a informação sensorial do servidor remoto e para permitir ao utilizador enviar agentes móveis a esse mesmo servidor (Fig. 5). Dois agentes são determinantes neste cenário: o agente **Feedback Collector**, o qual é lançado no servidor cliente e imediatamente migra para o servidor remoto onde continuamente perscruta o ambiente e põe a informação recolhida no espaço de informação partilhado do lado do cliente; e o agente **Visualization and Simulation**, que corre no servidor cliente e mostra numa janela a informação enviada pelo **Feedback Collector**, como é ilustrado na Fig. 5.

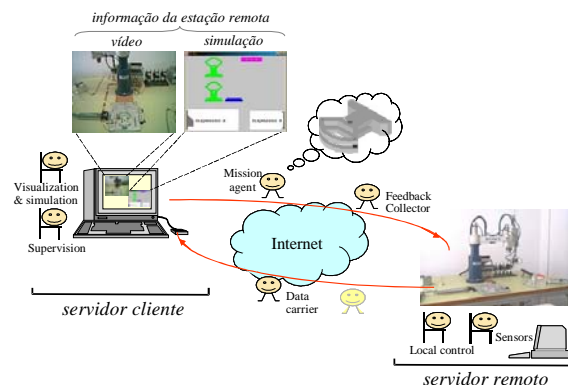


Fig. 5 - Os agentes **Feedback Collector** e **Visualization**

Neste cenário foram conduzidos dois tipos de experiências:

#### Implicações dos atrasos de tempo

Como todas as experiências foram conduzidas usando uma estratégia de controlo a nível de tarefa, a principal implicação dos atrasos de tempo é o tempo que leva completar uma tarefa de montagem. Na primeira experiência a montagem foi feita usando o mecanismo tradicional de chamadas a procedimentos remotos (*remote procedure call*, RPC) sob um atraso

de tempo (simulado) de envio e recebimento de dados de 10 segundos e usando comandos de baixo nível como **move**, **getgrip**, **setgrip**, **getpos**, **putgrip**, etc. A tarefa demorou 18 minutos para ser levada a cabo. Uma segunda experiência foi feita usando um agente móvel que foi enviado para o servidor remoto onde realizou a mesma tarefa de montagem demorando 3 minutos. Estes resultados confirmam que a operação remota usando comandos de baixo nível é impraticável quando os canais de comunicação têm grandes latências.

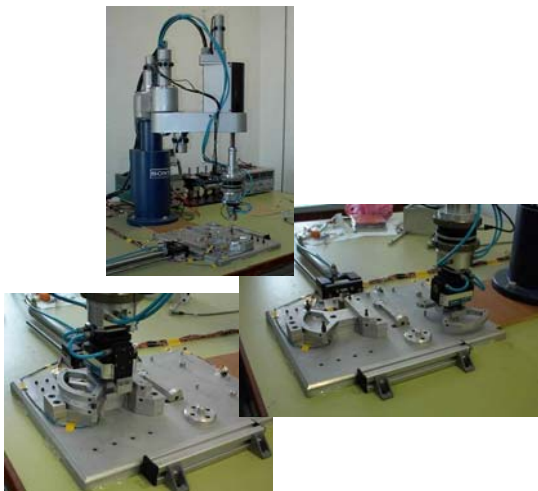


Fig. 6 - Diversas fases da montagem

Uma alternativa seria aumentar o nível dos comandos implementados no servidor remoto mas, como referimos acima, esta solução sacrificaria a flexibilidade do sistema. O uso de agentes móveis é uma boa solução, mas são requeridos altos níveis de autonomia.

### A autonomia dos agentes móveis

O conjunto seguinte de experiências foi realizado a fim de avaliar o nível de autonomia dos agentes móveis nomeadamente através das suas capacidades de monitoração de execução e recuperação de erros.

*Experiência 1:* Neste caso foi enviado um agente ao servidor remoto carregando consigo o plano abstracto ilustrado na Fig. 7. O plano refinado de execução da tarefa é mostrado na Fig. 8.

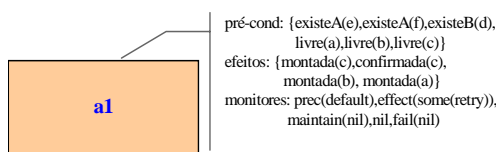


Fig. 7 - O plano abstracto para a experiência 1

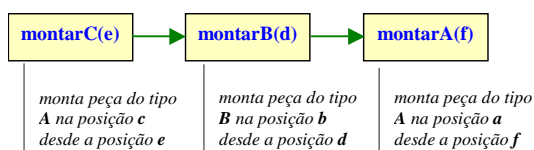


Fig. 8 - Refinamento do plano da Fig. 7

Foram provocados alguns erros aquando da execução deste plano simples. O agente recuperou os erros em todas as situações onde um plano de recuperação era teoricamente possível. Por exemplo, depois da acção **montarA(f)** estar concluída, foi provocado um erro dizendo ao agente que a primeira peça não estava montada o que causou que o agente activasse o monitor de efeitos da acção **a1**, o qual indica uma estratégia de recuperação **retry**. Como consequência disto, o agente tenta voltar a executar o mesmo plano executável, mas agora as pré-condições de **a1** não são mais satisfeitas, pelo que é activado o seu monitor de pré-condições. Já que **repair** é a estratégia por omissão dos monitores de pré-condições, o agente gera e executa um plano de reparação que desmonta todas as peças antes de voltar à execução do plano inicial. De forma análoga foram ainda provocados outros erros e o agente pôde recuperar quando os erros se encontravam dentro das suas capacidades de recuperação autónoma.

*Experiência 2:* A fim de mostrar as vantagens do uso de planos hierárquicos, foi enviado ao servidor remoto um agente com o plano abstracto descrito na Fig. 10. Neste plano, a acção **a1** é refinada numa sequência de acções **a2**, **a3** e **a4**. A acção **a2** é então refinada numa eleição (*choice*) entre as acções **a5** e **a6**; e **a4** é refinada numa eleição entre **a7** e **a8**.

O plano executável (refinado) é o mesmo que o da experiência anterior (ver na Fig. 8). As condições iniciais para esta experiência são mostradas na Fig. 9.

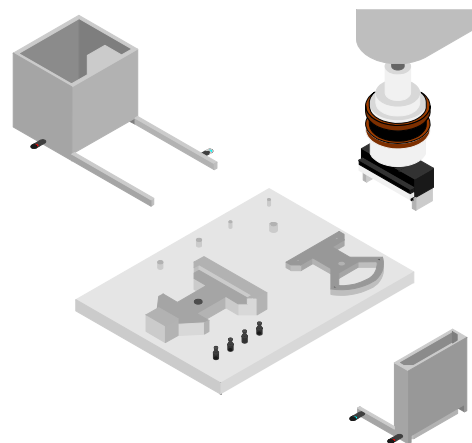


Fig. 9 - Condições iniciais da experiência 2

Quando o agente iniciou a sua execução aconteceu um erro nas pré-condições da acção **a5** pois a pré-condição **existeA(e)** falhou e activou o seu monitor de pré-condições, isto causou para que a falha fosse propagada ao nível imediato superior, activando pela sua vez o monitor de falhas da acção **a2**, cuja estratégia de recuperação é **redo**. Como consequência, o segundo ramo de eleição de **a2** foi seleccionado e o novo plano composto da sequência "**levarA(e)**, **montarC(e)**", correspondente à acção **a6**, substituiu o subconjunto composto só pela acção **montarC(e)** do plano executável original (Fig. 11).

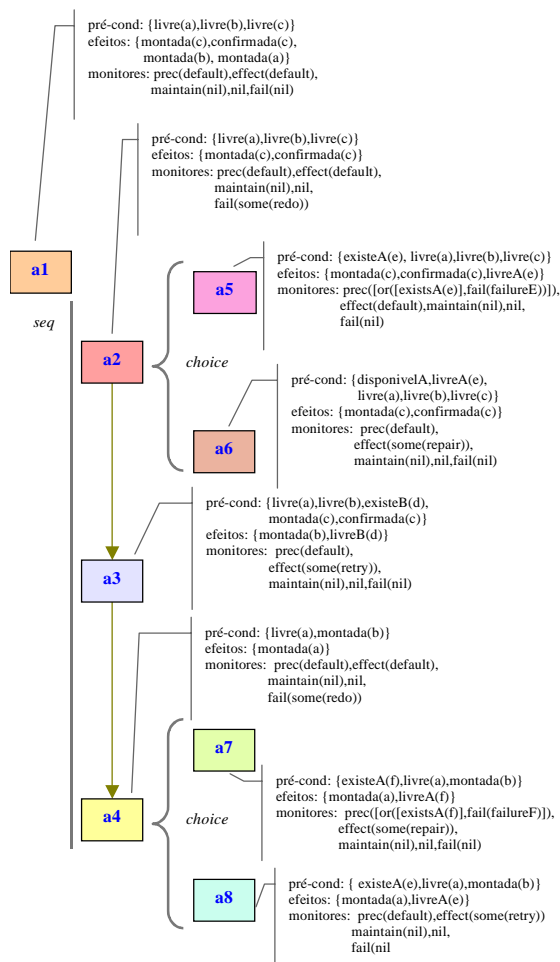


Fig. 10 - O plano abstracto para a experiência 2

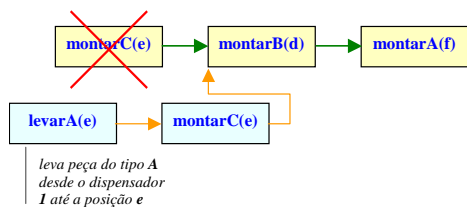


Fig. 11 - O novo plano executável resultante da falha da acção a5 no plano da Fig. 10

No entanto, como não havia nenhuma peça do tipo **A** disponível na saída do dispensador, a pré-condição **disponivelA** da acção a6 falhou e foi activado o seu monitor correspondente, cuja estratégia reparou o erro com uma simples acção **disporA** que accionou o dispensador para levar uma nova peça à sua saída (ver na Fig. 14). Estes dois passos são mostrados nos esquemas de Fig. 12 e Fig. 13.

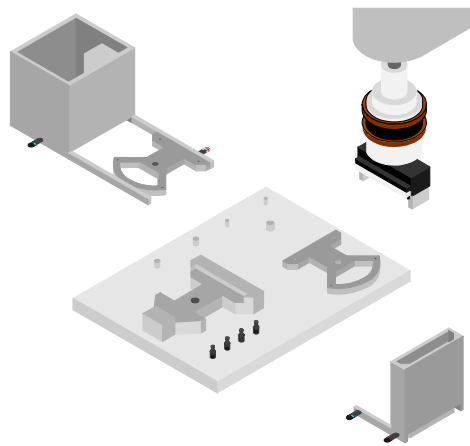


Fig. 12 - Reparação das pré-condições da acção a6

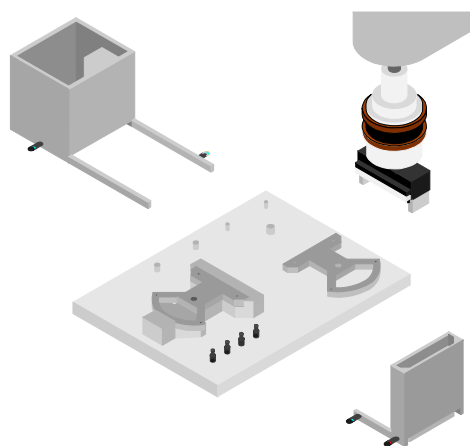


Fig. 13 - A tarefa da acção a6

Mais tarde, no decurso da mesma experiência, ocorreu mais um erro porque a peça do tipo **B** não estava disponível na posição **d**, o que causou que o monitor de pré-condições de a3 fosse activado e que fosse gerado um plano de reparação com a sequência “**disporB, levarB(d)**” mostrada na Fig. 14. O agente demorou por volta de 5 minutos para completar a montagem contra os 3 minutos que levaria se não houvesse ocorrido nenhum erro. De novo, o agente foi capaz de recuperar todos os erros onde era teoricamente possível obter um plano de recuperação. Pela sua vez, os planos de recuperação tenderam a ser mais pequenos e mais concisos do que quando não foi usada nenhuma decomposição hierárquica.

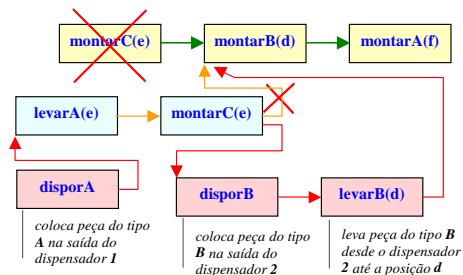


Fig. 14 - Reparando a pré-condição existeB(d) de a3

## 5. CONCLUSÕES

Neste artigo foi apresentada uma solução para a operação remota baseada em agentes móveis adaptáveis. Esta solução permite a conciliação entre a flexibilidade, alcançada na operação remota baseada em comandos de baixo nível implementados no sítio remoto, e a independência das características dos canais de comunicação, alcançada pela operação remota baseada em comandos de alto nível implementados no sítio remoto. Os agentes transportam planos hierárquicos não completamente especificados a fim de lidar com os altos níveis de heterogeneidade esperada nos sítios remotos. Os agentes obtêm um plano executável refinando estes planos abstractos de acordo com as capacidades que eles encontram em cada lugar a que chegam. As experiências conduzidas com um robô real mostraram que a solução proposta é independente das características dos canais de comunicação, tais como a latência e a sua disponibilidade. No que respeita à autonomia, estas experiências evidenciaram que os agentes podem recuperar de erros onde isso seja teoricamente possível. Embora o domínio considerado seja muito simples, o número de possíveis situações é notável e isso dificulta adoptar soluções determinísticas, especialmente quando se considerar que o mesmo agente pode visitar diversos sítios com diferentes ambientes de execução. De facto, a recuperação das muitas situações de erro inseridas nas experiências realizadas requereriam a intervenção humana se fosse adoptada uma aproximação tradicional. Consequentemente, é a opinião dos autores que a solução proposta é realmente eficaz no suporte à operação remota através da Internet e em cenários a grande distância, como é o caso do domínio da exploração espacial.

Uma área particularmente interessante onde a aproximação e arquitectura descritas se encontram sendo aplicadas é a da assistência remota à terceira idade, no âmbito do projecto europeu TeleCARE. Nesta aplicação visa-se o desenvolvimento duma infra-estrutura baseada em agentes móveis para o estabelecimento de comunidades virtuais de assistência aos idosos (Fig. 15).

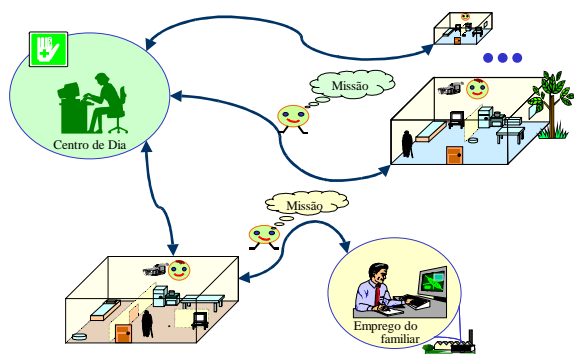


Fig. 15 - Agentes móveis na assistência à 3ª idade

Conforme se ilustra na Fig. 15, um Centro de Dia, por exemplo, poderá enviar agentes móveis com missões específicas de assistência às residências de

vários idosos. Esses agentes terão de adaptar (refinar) a sua missão de acordo com os recursos encontrados em cada residência. Exemplos de missões podem ser recolha de informação sensorial, apoio às rotinas de medicação, ou mesmo realização de tarefas na cozinha (assumindo que os electrodomésticos passarão a estar progressivamente integrados em redes de computadores). De forma semelhante, também um familiar poderá, a partir do seu emprego e através da Internet, enviar agentes com missões específicas de apoio ao idoso que se encontra em casa.

## Agradecimentos

Este trabalho foi financiado em parte pelo Ministério da Ciência e da Tecnologia através do projecto PRAXIS-XXI Telecare e pela Comissão Europeia através do projecto IST TeleCARE.

## 6. REFERÊNCIAS

- Cabri, G. Leonardi e F. L. Zambonelli (1998). How to coordinate internet applications based on mobile agents. In: *Proc. of WETICE'98 — workshop on coordination architectures for distributed web applications*.
- Camarinha-Matos, L. M. e W. Vieira (1998). Adaptive mobile agents for telerobotics and telesupervision. In: *Proc. of INES'98*. pp. 79-84.
- Camarinha-Matos, L. M. e W. Vieira (1999a). Intelligent mobile agents in elderly care. *Robotics and Autonomous Systems* 27(1-2), 59-75.
- Camarinha-Matos, L. M. e W. Vieira (1999b). MAAPL: A language for adaptive mobile agents with execution monitoring. In: *Proc. of INES'99*. pp. 171-176.
- Camarinha-Matos, L. M.; W. Vieira, e O. Castolo, (2000). A mobile agents approach to virtual laboratories: Enabling remote operation over the internet, *Proceedings of IFAC Symposium on Artificial Intelligence in Real-Time Control*, Budapest, Hungary, 2-4 Out, pp. 301-306.
- Fuggeta, A., G. P. Pico e G. Vigna (1998). Understanding code mobility, *IEEE Transactions on Software Engineering* 24(5), 346-361.
- Kotz, D. e R. S. Gray (1999). Mobile agents and the future of the internet. *ACM Operating Systems Review* 33(3), 7-13.
- Rowston, A. e A. Wood (1996). Solving the Linda multiple rd problem. In: *Coordination Languages and Models, LCNS* (P. Ciancarini e C. Hankin, Eds.). Vol. 1061, pp. 357-367, Springer-Verlag.
- Reece, G. e A. Tate (1994). Synthesizing protection monitors from causal structure. In: *Proc. of the 2nd Int. Conf. on Planning Systems*. AAAI Press.
- Vieira, W. (2001). *Agentes Móveis Adaptáveis para Operação Remota*. Tese de doutoramento, Universidade Nova de Lisboa.
- Vieira, W. e L. M. Camarinha-Matos (1999). Execution monitoring in adaptive mobile agents. In: *Cooperative Information Agents III, LNAI* (M. Klusch, O. Shehory e G. Weiss, Eds.). Vol. 1652. pp. 220-231. Springer Verlag. Berlin et al.
- Vieira, W. e L. M. Camarinha-Matos (2000). Agentes móveis na operação remota. In: *Jornadas da Engenharia Electrónica e de Telecomunicações do ISEL-DEEC*.
- White, J. E. (1997). Mobile agents. In: *Software Agents* (J. M. Bradshaw, Ed.). pp. 437-472. MIT Press.
- Wooldridge, M. (1999). Intelligent agents. In: *Multiagent systems* (G. Weiss, Ed.). The MIT Press.