
UM MODELO PARA A IMPLEMENTAÇÃO DE FEDERAÇÕES DE TRADERS

A MODEL FOR THE IMPLEMENTATION OF TRADER FEDERATIONS

Prof. Dr. Edmundo Roberto Mauro MADEIRA*
Luiz Augusto de Paula LIMA JÚNIOR**

ABSTRACT

This article describes a proposal of a **TRADER** which is the computational object that receives service offers from other objects (called "servers") storing them in a data base. When another object (a "client") needs a computational service, it can ask the trader about the service, and then the trader may return the "address" of a server which offers that service. The grouping of traders to form Federations is also discussed presenting the advantages and a model for building such organization. Algorithms and typical scenarios to negotiate the federation contract are also introduced. Finally, the details of the implementation of the prototype of the trader are discussed. This prototype is in process of development and it is a part of the *MULTIWARE Platform* which aims to provide an environment to support open distributed processing. The main algorithms and data structures used are also commented.

KEY WORDS: Trader, distributed system, Trader Federations, open distributed system.

RESUMO

Este artigo descreve uma proposta de um modelo para o **TRADER** que é o objeto computacional que recebe ofertas de serviços de outros objetos (chamados "servidores") colocando-as numa base de dados. Assim que um outro objeto qualquer (um "cliente") necessitar de um serviço computacional ele pode requisitar ao trader que então o informa a respeito da "localização" de algum servidor que oferece esse serviço. A união de traders em Federações também é discutida apresentando-se vantagens e modelos para a construção de tal organização, bem como algoritmos e cenários típicos para a negociação de contratos. Por fim, são apresentados com mais detalhes os esquemas de implementação do trader que está sendo implementado na UNICAMP como parte da *Plataforma MULTIWARE* que tem por objetivo criar um ambiente adequado para processamento distribuído aberto. Os principais algoritmos e estruturas de dados utilizados são comentados.

PALAVRAS-CHAVE: Trader, sistema distribuído, Federações de Traders, sistema distribuído aberto.

1. INTRODUÇÃO

O futuro da computação inclui o desenvolvimento de sistemas distribuídos e abertos. Sistemas distribuídos *homogêneos* já têm sido desenvolvidos e estudados extensivamente porque permitem que usuários compartilhem objetos que podem ser recursos periféricos e computacionais, serviços e informações. Esses sistemas consistem de várias estações de trabalho individuais e recursos periféricos (por exemplo, discos, impressoras) conectados por uma rede local (LAN). Estes

computadores têm hardware e software similares e estão sob o controle de um sistema operacional distribuído. Através da interconexão de vários sistemas distribuídos locais para formar um grande sistema distribuído homogêneo, os usuários de cada sistema local se tornam capazes de compartilhar recursos que não estão disponíveis nos seus sistemas locais.

A necessidade de utilizar uma variedade de objetos e o crescimento da diversibilidade de hardware e software

(*) Professor do DCC - IMECC - UNICAMP

(**) Mestrando - DCC - IMECC - UNICAMP

fizeram com que a *heterogeneidade* se tornasse algo corriqueiro. Se usuários individuais estão conectados a um ambiente heterogêneo o número de objetos que eles podem acessar é significativamente maior do que aqueles disponíveis em suas redes locais (por exemplo, banco de dados, sistemas de arquivos, serviços de diretório, supercomputadores e outros hardwares especializados). Assim, através da interconexão de sistemas distribuídos locais e de diferentes computadores, um grande sistema heterogêneo e distribuído de computadores (sistema aberto) pode ser construído. Este sistema aberto é gerenciado por um sistema operacional aberto.

Apesar das vantagens e do seu potencial, sistemas distribuídos abertos trazem consigo várias dificuldades:

1. Um usuário de um sistema distribuído perde a autonomia que possuía em ambientes "stand-alone", e a necessidade de mecanismos de segurança se torna um verdadeiro desafio;
2. A heterogeneidade acaba com o acesso direto a um objeto, e se usuários querem permitir com que outros usuários tenham acesso aos seus objetos, esses acessos devem ocorrer em condições determinadas em comum acordo;
3. Como o número de objetos que oferecem serviços normalmente é muito grande, deveria existir um

meio de localizar determinados servidores que satisfizessem uma lista de propriedades desejadas.

Como podemos então resolver todos estes problemas? A ISO¹ propôs uma solução introduzindo uma nova classe de servidores de nomes e gerenciadores de objetos conhecidos como **TRADERS** (19) (20) dentro do Modelo de Referência para Processamento Distribuído Aberto (RM-ODP) (15) (16) (17) (18) (7) (8).

"Trading", em ODP, pode ser considerado como sendo uma forma melhorada de nomeação e "binding" (acoplamento) (5). No caso de trading, o nome de um serviço pode ser substituído por uma descrição das propriedades desejadas de um servidor.

Poderíamos comparar o trader a um serviço de "classificados" por telefone onde pessoas que estão oferecendo produtos (ou serviços) informam a uma central (o trader) sobre os dados pertinentes àquele tipo de serviço. Assim que a central receber um telefonema de alguém procurando por aquele produto (ou serviço) ela se encarrega de achar a melhor oferta (se é que há alguma) que mais se adequa aos requisitos do cliente que ligou. A central passa então as informações (inclusive o número de telefone) do fornecedor do produto (ou serviço).

Num exemplo mais realista, consideremos uma rede heterogênea de computadores como na figura 1.

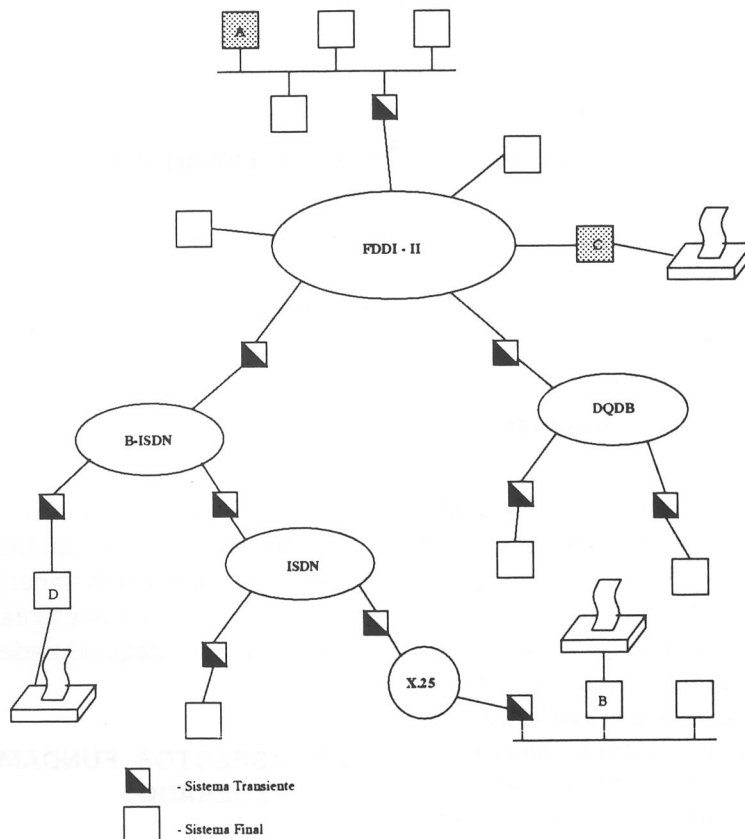


Figura 1 - Rede Heterogênea

(1) ISO: International Organization for Standardization

Se o usuário (objeto computacional ou ser humano) em A deseja utilizar uma impressora laser que esteja no "Centro de Computação" da empresa em que ele trabalha, cujo preço por página impressa seja mínimo, como descobrir o "endereço" de alguém que ofereça este serviço (se é que existe alguém e se é que A tem autorização para isso)? E ainda mais: como fazer isso de forma transparente, ou seja, de modo a esconder os detalhes da topologia da rede? No modelo ODP, é somente através da utilização do trader que objetos podem saber a respeito da existência de serviços os quais ele deseja requisitar. No exemplo, os servidores de impressão B, C e D podem ter informado previamente ao trader a respeito das características dos serviços de

impressão que eles oferecem ("PRINTER" na figura 2). Assim, quando A faz o pedido de serviço ao trader, ele se encarrega de achar o servidor (digamos, C) que melhor se adequa às exigências feitas por A, acoplando A a C. O serviço exportado por C também está apresentado na figura 2.

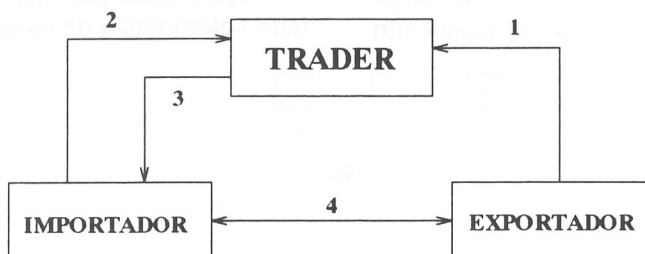
Em suma, a interação do trader com outros objetos (cliente e fornecedor) está representada na figura 3.

2. FEDERAÇÃO DE TRADERS - VISÃO GERAL

Uma Federação de Traders é uma coleção de traders cooperativos mas autônomos.

Tipo de Serviço (PRINTER)	Serviço Exportado por C
<i>Printer type</i> : DotMatrix, Laser	<i>Printer type</i> = Laser
<i>Identifier</i> : String	<i>Identifier</i> = "HP"
<i>Localization</i> : String	<i>Localization</i> = "Centro Comp. - Empresa de A"
<i>Paper size</i> : A4, A3 (default A4)	<i>Paper size</i> = A4
<i>Cost per page</i> : Float	<i>Cost per page</i> = 0.5

Figura 2 - Tipo de serviço e serviço exportado por C.



1. Exportador informa trader sobre serviços oferecidos.
2. Importador solicita serviços.
3. Trader informa importador a respeito de exportador do serviço solicitado.
4. Interação entre importador e exportador.

Figura 3 - Interações do trader com seus usuários

Através da federação, uma oferta de um trader componente poderá ser conhecida por uma audiência mais ampla e um trader componente da federação terá um mercado maior para suprir as suas necessidades.

Uma federação de traders também pode ser formada quando o número de objetos administrados por um trader se torna tão grande que não é possível que a administração sirva eficientemente aos seus usuários. Através do particionamento das responsabilidades administrativas do banco de dados do trader em traders cooperativos, uma federação de traders pode ser formada.

Numa federação, devem existir mecanismos para permitir cooperação entre traders que são autônomos,

heterogêneos e distribuídos. Além do mais, os objetivos do ODP exigem a transparência de distribuição de modo que um usuário de um trader federado será associado a somente um trader e acessará transparentemente outros traders através daquele trader.

2.1. ASPECTOS FUNDAMENTAIS DE TRADERS FEDERADOS

Existem cinco princípios básicos que devem caracterizar o relacionamento entre os traders componentes de uma federação:

1. um componente não pode ser forçado a executar uma atividade para outro componente;
2. um componente deve ser livre para entrar e sair de uma federação;
3. um componente deve ser capaz de determinar quais os dados que ele deseja compartilhar com os outros;
4. um componente determina como ele vê e combina os dados existentes;
5. usuários locais, atividades e dados devem sofrer alterações mínimas (se é que há alguma) quando se incluem os mecanismos de federação.

Esses princípios devem reger toda formulação de modelos e algoritmos para a federação de traders.

2.2. UMA PROPOSTA DE MODELO PARA FEDERAÇÃO DE TRADERS

O modelo proposto para a federação de traders é o modelo descentralizado que tem a característica de garantir a **autonomia** de cada trader (que é um dos princípios básicos que regem a formação de federações) (3).

Para ser parte de uma Federação, um trader deve importar serviços de pelo menos um outro trader ou deve exportar serviços para pelo menos um outro trader na federação.

Um trader que importa serviços de um trader remoto tem um *contrato de importação* com aquele trader remoto. O contrato de importação define:

- os tipos de interfaces disponíveis no trader remoto;
- as regras para mapear requisições e respostas entre o trader local e o remoto de forma que sejam inteligíveis para ambos;

Um trader que exporta serviços para um trader tem um *contrato de exportação* com o trader remoto. O contrato de exportação define:

- a extensão do acesso ao banco de dados do trader local ao trader remoto;
- os tipos de interface disponíveis no trader remoto;
- as regras para mapear requisições e respostas entre o trader local e remoto de forma que sejam inteligíveis para ambos;

Para cada contrato de exportação existe um contrato de importação correspondente no trader remoto.

Um trader que exporta para um trader remoto oferece uma interface de "*Estabelecimento-Federação*" e uma interface de "*Operações-Federadas*" para o seu trader remoto (19).

O procedimento para fazer uma requisição de serviço é o seguinte: um cliente de um trader procura por um serviço no banco de dados do trader e, se necessário, o trader local busca através dos seus contratos de importação. Se o tipo de serviço solicitado está disponível num trader remoto, então o trader local mapeia a requisição em uma requisição remota e envia para o trader remoto através da interface "*Operações-Federadas*" desse trader remoto. O trader remoto inicia uma busca no seu banco de dados, de acordo com o contrato de exportação correspondente. Quando a busca for completada, os resultados (transformados, se necessário) são retornados ao trader que está importando os serviços e por fim ao cliente que requisitou os serviços. A figura 4 mostra as interfaces entre traders via contratos.

3. ASPECTOS DE MODELAGEM E IMPLEMENTAÇÃO

3.1. MODELO

Em nosso trabalho identificaram-se basicamente dois grupos de atividades desempenhadas por um trader: o gerenciamento de bases relacionadas às informações (estáticas e dinâmicas) mantidas pelo trader e a execução propriamente dita das operações utilizando-se das informações armazenadas (figura 5).

O modelo que propomos consiste de um refinamento desses dois módulos considerando separadamente funções locais e aquelas relativas ao estabelecimento e "manutenção" de federações bem como separando em módulos diferentes o gerenciamento/obtenção das informações estáticas e dinâmicas. Assim chegamos a um modelo (figura 6) que supre toda a funcionalidade requerida de um trader (inclusive para federação).

O administrador local e o de federação (objetos ou humanos), que podem ser um só, são responsáveis pela definição e cumprimento de certas políticas de negociação locais e no nível de federação, respectivamente. O *administrador local* acrescenta novos tipos de serviços a um dado contexto entre outras atividades, por isso deve estar acoplado diretamente ao repositório de tipos. Já o *administrador de federação* é responsável por preparar o catálogo, requisitar o catálogo, decidir quan-

do e com quem federar-se, e aceitar, recusar ou formular propostas de contratos definindo assim políticas para o estabelecimento da federação.

Dois módulos são responsáveis pelo armazenamento e/ou obtenção de informações no trader. São eles:

MÓDULO DE INFORMAÇÕES ESTÁTICAS

Concentra as operações que manipulam as informações estáticas que correspondem a tipos de serviços (no Repositório de Tipos) e ofertas de serviços (no Diretório).

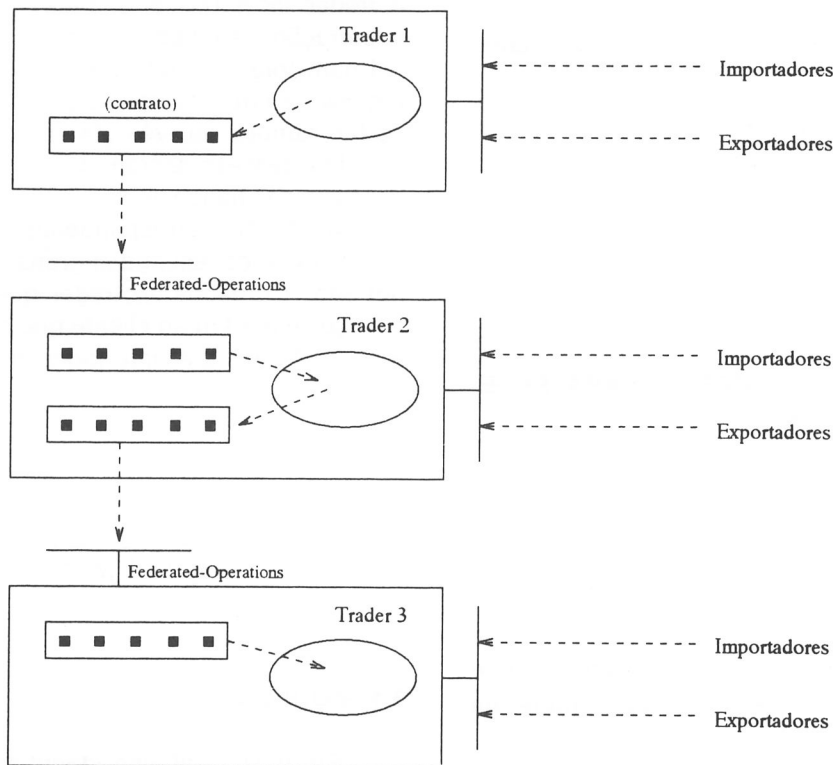


Figura 4 - Federação de traders



Figura 5 - Atividades básicas de um trader

Inicialmente foi construído um protótipo de um trader diretamente em cima do sistema operacional UNIX. São utilizadas as chamadas de sistemas padrão e chamadas a procedimentos remotos (RPC's). Além disso, utiliza-se na implementação "sockets" e uma simulação de "processos leves" ("threads"). Com isso, procura-se tornar o protótipo o mais portátil possível para outros ambientes dos quais a corrente implementação possa vir a fazer parte no futuro. O próximo passo será transportar o modelo implementado para utilizar as facilidades de uma das plataformas citadas acima.

Estão implementadas somente as funções básicas de um trader que permitem com que uma federação seja formada. A implementação dos algoritmos para a federação ainda se encontra em fase de desenvolvimento.

3.2.1. COMUNICAÇÃO - O "BROKER"

Esquemáticamente o protótipo construído está representado na figura 7.

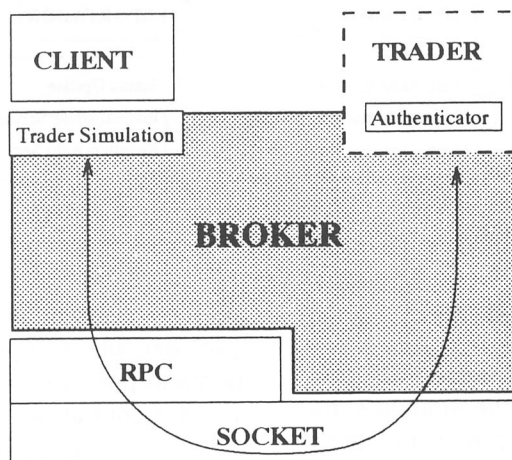


Figura 7 - Ambiente construído juntamente com o trader

- identificador da interface;
- nome de quem está requisitando a função (cliente);
- lista de parâmetros adicionais (pode ser vazia).

que são fixos para todas as operações. Caso os parâmetros a serem transmitidos para o trader sejam muito complexos, então o "Broker" se encarrega de transmiti-los via rede através de "stream sockets". Se não houver permissão para dado cliente efetuar a dita operação, os parâmetros complexos nem são enviados, o que melhora o desempenho nestes casos.

Está também embutido no "Broker" uma implementação básica do mecanismo de interfaces. Assim, ao chamar uma determinada função, é imprescindível que se especifique em qual interface aquela função

O módulo de *simulação do trader* faz com que o cliente do trader (um importador ou exportador de serviços) tenha a impressão de que todas as suas operações estão sendo feitas localmente, tornando assim totalmente transparente todos os detalhes da comunicação. Chamamos a parte da implementação que lida com a transparência para a comunicação via rede de "Broker", um termo já consagrado pela CORBA⁵ (4). O "Broker" tem a função de transformar uma chamada a uma função local em uma chamada de procedimento remoto.

Um cliente é construído utilizando uma biblioteca que esconde os detalhes de toda a comunicação com o trader remoto. Cada invocação de uma operação do trader é feita localmente ao "Broker" do lado do cliente, que, por sua vez, comunica-se com o "Broker" do lado do trader via RPC passando os seguintes parâmetros:

está localizada. O "Broker" do lado do trader, antes de tudo, sempre verifica a existência de tal operação na interface especificada, e então chama localmente a função do trader correspondente.

O "Broker" é também responsável por retornar os resultados das operações. Aqui novamente, se os resultados forem complexos (no caso da operação "search", por exemplo), eles são transmitidos para o cliente através de "stream sockets".

3.2.2 O TRADER

O protótipo está ainda em fase de desenvolvimento para incorporar operações para o gerencia-

(5) Um "broker" é uma pessoa que faz um determinado serviço no lugar de outra pessoa. (CORBA: Common Object Request Broker Architecture)

mento de federação de trader, porém, no estágio atual, estão oferecidas as operações de "export", "withdraw", "search" e "list_offer_details", que são essenciais.

Antes que qualquer operação seja executada, um módulo chamado "autenticador" verifica a autorização do cliente em executar a determinada operação. Se não há permissão, um código de erro

apropriado é retornado, caso contrário, a operação é executada.

3.3. FEDERAÇÃO DE TRADERS

A comunicação entre dois traders para se construir uma federação ocorre da mesma forma descrita para a comunicação entre um trader e o seu cliente⁶.

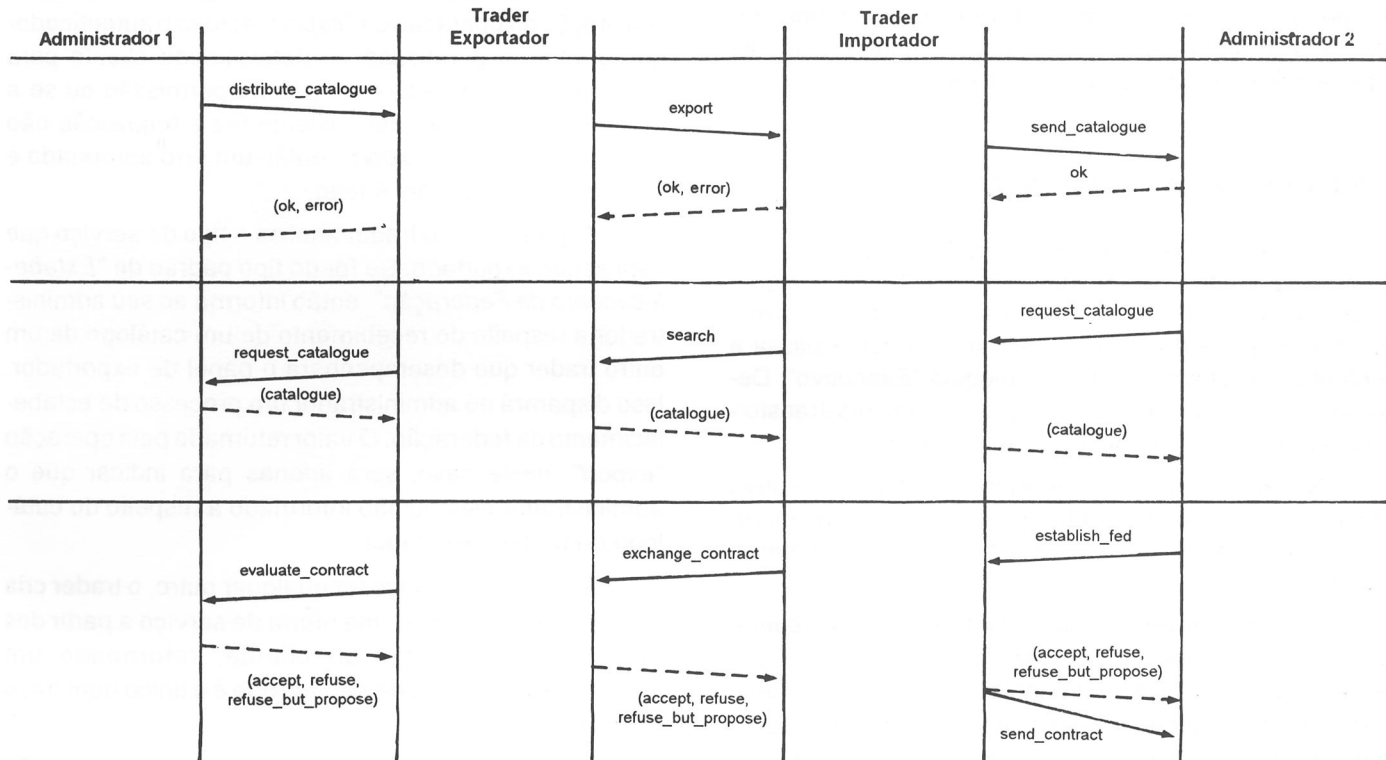


Figura 8 - Cenário para a negociação e o estabelecimento do contrato

Para o estabelecimento do contrato, é necessária a intervenção do *Administrador de Federação* (veja figura 6), pois é dele que partem as decisões de formar a federação e de avaliar e propor contratos.

O esquema da figura 8 apresenta o cenário para a negociação e o estabelecimento de um contrato de federação entre dois traders, onde um faz o papel de "exportador" de serviços e o outro de "importador" de serviços.

Antes de mais nada, para se estabelecer um contrato de federação entre dois traders é preciso que o trader que fará o papel de importador obtenha o *catálogo* do trader exportador. E isto pode ocorrer de duas maneiras.

No primeiro caso, o *administrador 1* (que é o administrador da federação do trader exportador na

figura 8) decide enviar o seu catálogo para um potencial trader importador. O trader exportador ao receber a requisição para distribuir o catálogo, juntamente com o próprio catálogo, age como um cliente qualquer do trader importador usando a operação "export" tendo como propriedades do serviço o catálogo, e como tipo de serviço o tipo padronizado "ESTABELECIMENTO DE FEDERAÇÃO". O que é retornado ao *trader importador*, ao *trader exportador* e ao *administrador 1* é apenas um indicador de que o administrador do trader importador está ciente ou não do catálogo enviado.

Uma outra maneira de fazer com que o catálogo seja conhecido pelo administrador do trader importador, é quando o próprio administrador faz um pedido de catálogo a um potencial trader exportador. Essa opera-

(6) No processo de negociação do contrato de federação, somente se torna necessária a inclusão de uma nova operação ("Exchange Contracts"). Toda a troca de informações que precede a "assinatura" do contrato é feita utilizando-se as operações normais do trader (alterando-se os algoritmos).

ção, no trader exportador, é transformada num "search" com o tipo de serviço padronizado "ESTABELECIMENTO DE FEDERAÇÃO". O catálogo é então passado do *administrador 1* até o *administrador 2* (veja figura 8)

Uma vez que o administrador do trader importador possui o catálogo, ele pode agora iniciar a negociação do contrato de federação fazendo uma proposta ao administrador do trader exportador. Este pode aceitá-la plenamente, recusá-la ou recusá-la e ainda fazer uma contra-proposta com um contrato reduzido. Deste modo um contrato de federação pode ser estabelecido e então as operações federadas podem ser executadas.

3.3.2. OPERAÇÕES FEDERADAS

O módulo "Executivo de Federação" (veja figura 6) do trader exportador, ao receber a requisição de qualquer operação, interpreta o seu contrato de exportação correspondente aplicando as restrições ali contidas e passa a execução propriamente dita ao módulo "Executivo". Depois disto, ele se encarrega de retornar os valores (transformados, se necessário) ao trader importador.

No lado do trader importador, o módulo "Executivo de Federação" é acionado pelo módulo "Executivo" quando este identificar a necessidade de alguma operação federada.

De um modo geral, o papel do "Executivo de Federação" do trader importador é transformar uma requisição local em uma operação federada que é passada ao trader exportador. As operações executadas via federação, passam pelo módulo "Executivo de Federação" que se encarrega de interpretar o contrato de exportação aplicando as restrições e operações combinadas.

Os algoritmos das operações que formam o conjunto mínimo a ser fornecido aos clientes de um trader (importador, exportador ou outro trader) estão comentados a seguir.

OPERAÇÕES DISPONÍVEIS NA INTERFACE "EXPORTER OPERATIONS"

As operações disponíveis nesta interface são:

EXPORT: corresponde à criação de uma oferta de serviço no diretório do trader;

WITHDRAW: remove uma oferta de serviço do diretório;

MODIFY: corresponde à uma operação de "withdraw" seguida de uma operação de "export" feitas ativamente

As operações "withdraw" e "modify" não sofrem modificações para incorporar os mecanismos de federação propostos neste trabalho, por isso não discutiremos aqui o seu algoritmo.

A OPERAÇÃO "EXPORT"

Antes de mais nada, o trader que recebeu uma solicitação da operação de "export" aciona o autenticador que verifica a permissão do determinado cliente para executar essa operação. Se não há permissão ou se a interface através da qual o cliente fez a requisição não possui a operação "export", então um erro apropriado é retornado e nada mais é feito.

Depois disso, o trader analisa o tipo de serviço que está sendo exportado. Se for do tipo padrão de "Estabelecimento de Federação", então informa ao seu administrador a respeito do recebimento de um catálogo de um outro trader que desempenhará o papel de exportador. Isso disparará no administrador um processo de estabelecimento da federação. O valor retornado pela operação "export", neste caso, será apenas para indicar que o administrador está ou não informado a respeito do catálogo do trader exportador.

Se o tipo de serviço for qualquer outro, o trader cria em sua base de dados uma oferta de serviço a partir dos parâmetros recebidos do cliente, retornando um identificador da oferta de serviço que é o único num dado contexto⁷.

Um código de erro será retornado se a operação não puder ser executada por algum motivo.

OPERAÇÕES DISPONÍVEIS NA INTERFACE "IMPORTER OPERATIONS"

As operações disponíveis nesta interface são:

LIST_OFFER_DETAILS: retorna os detalhes de uma determinada oferta de serviço;

SEARCH: retorna todas as ofertas de serviços que satisfazem os requisitos do importador;

SELECT: seleciona a melhor oferta de serviço (a partir de critérios determinados) dentre as que satisfazem às propriedades requeridas pelo importador;

Aqui novamente a operação "search" é a única alterada para permitir o estabelecimento de federação e as operações federadas.

(7) Um contexto pode ser considerado como sendo um "subdiretório".

A OPERAÇÃO "SEARCH"

Após verificada a permissão do cliente para efetuar a operação, o tipo de serviço pedido é analisado. Se for do tipo padrão de "Estabelecimento de Federação", o trader pede ao seu administrador que crie um catálogo com as possíveis operações via federação e restrições de acesso. Então retorna o catálogo ao cliente (que é um outro trader, neste caso) juntamente com o identificador para a sua interface de estabelecimento de federação, se a operação for bem sucedida.

Se o tipo de serviço pedido não for de estabelecimento de federação, o trader faz a busca dentro de um escopo adequado para obter as ofertas de serviço que satisfazem aos critérios pedidos pelo cliente. É nesse momento que são obtidas tanto as informações estáticas quanto as dinâmicas das ofertas de serviços.

Se não acha ofertas de serviço adequadas localmente, verifica os seus contratos para fazer uma requisição de busca ao "Executivo de Federação" que se encarregará de executar o mesmo "search" no trader remoto.

É retomada ao cliente uma lista de ofertas de serviços com as propriedades pedidas juntamente com os identificadores das interfaces onde os serviços estão disponíveis.

A OPERAÇÃO "EXCHANGE-CONTRACT"

Esta operação está disponível na "interface de estabelecimento de federação". Através desta operação, um trader importador poderá fazer uma proposta de um contrato para a federação. O trader exportador ao receber esta proposta, pede ao seu administrador para avaliá-la após verificar permissões e interfaces. Se a proposta de contrato for aceita pelo administrador, o trader cria uma nova interface para interações federadas e exporta para a sua própria base de dados o tipo de serviço padronizado "INTERAÇÃO DE NEGOCIAÇÕES FEDERADAS" tendo o contrato de exportação como sendo a propriedade do serviço. O identificador da interface é então retomado ao trader importador se não houver problemas. Caso contrário, um código de erro é retomado.

O protótipo implementado a partir destes algoritmos segue o modelo proposto na figura 6 e está incluído no projeto da plataforma Multiware(1).

4. CONCLUSÃO

O Trader desempenha um papel muito importante em ambientes distribuídos e abertos, pois em um Sistema Distribuído Aberto é altamente desejável que existam meios de fazer a seleção dinâmica (em tempo de execução) de serviços computacionais que satisfaçam a determinadas propriedades. É através do trader que

importadores de serviços (clientes) podem entrar em contato com exportadores de serviços (servidores) no modelo ODP.

O presente trabalho propôs um modelo de implementação do trader dando atenção especial aos aspectos que dizem respeito à "Federação de Traders" apresentando-se algoritmos e detalhes de implementação do protótipo construído.

A maior dificuldade encontrada na fase de implementação foi com relação à parte de comunicação entre objetos através da rede. Foi necessária a construção de uma camada (o "BROKER", figura 7) que toma os aspectos de comunicação transparentes para o trader e especialmente para o cliente do trader. Com o futuro transporte do protótipo do trader para o ambiente *Multiware* (1), a camada "BROKER" (aqui, de uso específico do trader) será substituída por outra camada que fornecerá todas as facilidades de comunicação necessárias para atender não só ao trader, mas também a um objeto computacional genérico. Para tanto, o "BROKER" do trader será o resultado da compilação do conjunto de interfaces do trader.

Também, a inexistência de "binding" dinâmico no ambiente em que nos propusemos a desenvolver o protótipo, impôs algumas restrições na implementação do modelo criado. Este problema foi parcialmente contornado com a introdução do código que simula o conceito de interfaces e que faz parte do "BROKER".

AGRADECIMENTOS

Agradecemos à CAPES e à FAPESP pelo apoio financeiro durante o desenvolvimento do projeto.

REFERÊNCIAS

- (1) *Madeira, E. R. M.; Mendes, M.* "Plataforma Multiware: Projeto e Desenvolvimento da Camada Multiware", XII SBRC - Curitiba, maio, 1994.
- (2) *Mendes, M. J.; Loyola, W. P. D. C.; Madeira, E. R. M.* "Demos: A Distributed Decision-Making Open Support System" -4th IEEE Workshop on Future Trends in Distributed Computing Systems, Lisboa, Portugal, pp 208-214, Sept, 1994.
- (3) *Bearman, M.* "ODP Trader" - Proceedings of the ICODP, pp 19-23, Sept, 1993.
- (4) *Vogt, F.; Andrae, C.* "Middleware for Distributed Applications Support : ODP and/or CORBA, Proceedings of the ICODP, pp 405-430, Sept., 1993.
- (5) "Broadening the User Environment with Implicit Trading" Proceedings of the ICODP, pp 129-140, Sept., 1993.

- (6) Meyer, B.; Popien, C.; "Object Configuration by ODP Traders" Proceedings of the ICODP, pp 425-430, Sept.,1993.
- (7) Tschammer, V., "CODE - Cooperating Open System, Open Distributed Processing, Distributed Computing Environment Experiments and Projects", Nov.,1992.
- (8) Tschammer, V.; Mendes, M. J.; Souza, W. L.; Madeira, E. R. M.; Loyolla, W., "Processamento Distribuído Aberto e o Modelo RM-ODP/ISO", XI SBRC,1993.
- (9) "OSF Distributed Computing Environment" - Sept 1990.
- (10) Fauth, Dr. Dietmar; Gossels, Jonathan; at all, "Distributed Computing Environment Evaluation Team-OSF", May, 1990.
- (11) "Distributed Computing Environment Overview" - OSF.
- (12) "Directory Services for a Distributed Computing Environment", OSF, Sept.,1990.
- (13) "File Systems in a Distributed Computing Environment", OSF, Sept.,1990.
- (14) "Remote Procedure Call in a Distributed Computing Environment", OSF, Oct.,1990.
- (15) "ISO/IEC JTC 1/SC 21/N 7053" Basic Reference Model of ODP - Part 1: Overview and Guide to Use.
- (16) "ISO/IEC DIS 10746-2 - ITU-T Draft Rec. X. 902 - Feb. 1994" Basic Reference Model of ODP - Part 2: Descriptive Model.
- (17) "ISO/IEC JTC 1/SC 21/N 1076-3.2- ITU-T Draft Rec. X. 903 - Feb.,1994" Basic Reference Model of ODP - Part 3: Prescriptive Model.
- (18) "ISO/IEC 21/N 7056" (Recomendation X. 905) Basic Reference Model of ODP - Part 5: Architectural Semantics.
- (19) "ISO/IEC JTC 1/SC 21/N 7047, 1992-06-30" Working Document on Topic 9.1 - ODP Trader
- (20) "ISO/IEC JTC 1/SC 21 - Nov. 1993" Information Technology - ODP Trading Function
- (21) HP Company, Sun Microsystems, inc, "The Object Management Group, Object Request Broker, RFP joint response",1991.
- (22) HP Company, Sun Microsystems, inc, "Distributed Object Management Facility Core Especification",1991.
- (23) "ANSA: An Engineer's Introduction to the Architecture" -Release TR.03.02-Architecture Projects Management Limited Nov., 1991.
- (24) "ANSA Work-Programme - Phase III" - Issue I - Architecture Projects Management Limited, June, 1991.
- (25) "The Open Systems Newsletter - ANSA: Objects Pioneers" - Technology Appraisals Ltd, V. 7, Issue 2, Feb., 1993..