

## **PROCESSAMENTO DISTRIBUÍDO ABERTO E O MODELO RM - ODP / ISO**

Volker Tschammer, FOKUS/GMD, Berlin  
Manuel de J. Mendes, Unicamp-FEE-DCA, Campinas  
Wanderley L. de Souza, UFPb-CCT-DCS, Campina Grande  
Edmundo R. M. Madeira, Unicamp-IMECC-DCC, Campinas  
Waldomiro P. de Loyolla, Unesp-FE-DEE, Baurú

### **RESUMO:**

Este artigo tem por objetivo a discussão dos principais conceitos e características do modelo de referência para Processamento Distribuído Aberto (ODP) da ISO. Após apresentação de conceitos básicos, consideram-se os modelos descritivo e prescritivo do RM-ODP/ISO, além da sua semântica arquitetônica. Como exemplo do atual estágio dos trabalhos de padronização são vistos alguns detalhes das funções de "Trading". O artigo termina com alguns experimentos e projetos na área.

### **ABSTRACT:**

This paper discusses some important concepts of the Reference Model for Open Distributed Processing from ISO. Initially, some fundamental concepts and the Descriptive and Prescriptive Models of RM-ODP/ISO are presented, besides their Architectural Semantics. As an example of the current standardization work the functions of Trading are discussed in more detail.

## **1. Introdução.**

Informação é uma das necessidades básicas da sociedade humana. Informação implica em conhecimento, que por sua vez é necessário ao desempenho de qualquer tarefa. A fim de realizar uma tarefa complexa, normalmente é preciso adquirir conhecimento de outros e, para tal, faz-se necessária a troca de informações. Para que essa troca possa ser realizada, a informação deve ser representada numa forma comunicável: linguagem falada ou escrita, dados, sinais, gráficos, imagens, etc.

Informação também é fundamental para a operação bem sucedida de qualquer empresa. Para atender aos requisitos organizacionais de uma empresa, é necessário que a informação possua mobilidade, ou seja, possa ser alcançada no local onde está disponível e possa ser levada ao local onde é necessária à tomada de decisões.

A localização da informação é fortemente influenciada pela localização dos recursos de armazenamento e processamento, que fazem parte dos sistemas de informação que dão suporte a uma empresa. As restrições impostas pelo mundo real levam normalmente à distribuição de tais sistemas e conseqüentemente à distribuição do conjunto das informações necessárias a uma empresa.

Além da característica acima mencionada, essas informações (ou parte delas) precisam frequentemente ser:

- duplicadas, para prevenir falhas;
- distribuídas, para permitir o acesso às mesmas a tempo;

- processadas em paralelo, para possibilitar uma solução rápida para problemas de grande porte;
- copiadas para os diferentes grupos que gerenciam uma empresa.

A tecnologia atual dispõe de redes complexas que fornecem facilidades para o acesso e o processamento da informação. Essas facilidades permitem a interação entre os usuários de uma forma aparentemente independente. Entretanto, a todo instante, a informação presente nessas redes deve refletir uma descrição verídica dos elementos e negócios relevantes para uma empresa. Portanto, essa informação deve estar disponível somente às pessoas autorizadas, que devem ter confiança na mesma no momento de utilizá-la.

O objetivo principal da padronização do processamento distribuído aberto é permitir que os procedimentos de utilização de um conjunto de recursos (hardware e software) heterogêneos e distribuídos, que possibilitam o acesso e o processamento da informação, possam ser realizados de uma forma consistente, confiável e flexível.

Este trabalho segue a estrutura básica da referência [1]. Na seção 2 são apresentados os conceitos fundamentais de Sistemas Distribuídos Abertos. Aspectos da padronização ODP, em particular os modelos descritivo e prescritivo, a sua semântica arquitetônica e a funcionalidade do Trader, são discutidos na seção 3. Finalmente apresentam-se, na seção 4, alguns aspectos de projetos experimentais na área, seguidos das conclusões do trabalho, na seção 5.

## 2. Conceitos de Sistemas Distribuídos Abertos

Ambientes Abertos podem ser desenvolvidos para o aproveitamento de sistemas computacionais e de comunicação, com o propósito de obter-se a utilização de recursos e serviços externos, a oferta de recursos e serviços locais para utilização externa, e a participação em atividades comuns e aplicações cooperativas.

### *CARACTERÍSTICAS DOS AMBIENTES ABERTOS*

Da mesma forma que em grandes sistemas distribuídos, as principais características, físicas e funcionais, de um ambiente aberto são: **ilimitado**, quer dizer, público e livre de restrições tanto em caráter geográfico como organizacional; **de livre acesso**, estando sempre pronto a admitir qualquer tipo de usuário, componente ou aplicação; **heterogêneo**, com cada usuário decidindo sobre as próprias características de existência, projeto, implementação, uso e gerenciamento de seus componentes; **autônomo**, quer dizer, cada usuário ou servidor de aplicações livre de regulamentação para determinar as propriedades, comportamento e evolução de seus elementos; **descentralizado**, com funcionalidade física e lógica de cada elemento gerenciada a nível local, sem qualquer influência ou controle externos.

As características físicas e funcionais dos Ambientes Abertos levam à ocorrência de certas consequências quanto à observação, controle, integridade e privacidade de elementos destes ambientes.

Quanto à **observação**, nenhum elemento (usuário, administrador ou servidor) é capaz de ter uma visão completa ou capaz de influenciá-lo como um todo. Cada observador é apenas capaz de ter uma vista parcial do ambiente global, provocando que cada observação tenha valor local e restrita a certa área.

Em relação ao **controle**, a área de influência de cada elemento gerenciador também é restrita. Assim, fatos, eventos e atividades podem passar despercebidos e fora de sua área de controle. Ainda mais, cada informação é válida apenas por um certo intervalo de tempo, já que seu ambiente local pode variar continuamente.

Quanto à **integridade** do ambiente, deve-se considerar a necessidade de adoção de certos procedimentos de controle de acesso, como medida de segurança, com o propósito de reconhecer e evitar perturbações devido a operações indevidas de entidades não habilitadas ou indesejáveis: a operação local deve permanecer sem perturbações. Por outro lado, a informação sobre os componentes locais é incerta, atrasada, incompleta e duvidosa. Segundo [2], trata-se de um verdadeiro "Heisenberg World".

Em relação à **privacidade**, deve-se estabelecer certos limites à observação e influência externa, de forma a evitar-se não só influências indesejáveis como também a garantir a integridade de recursos locais. A região interna destes limites é denominada de ambiente local e a externa de ambiente global.

### **COOPERAÇÃO EM AMBIENTES ABERTOS**

Os conceitos de utilização de serviços externos, de privacidade e de autonomia podem apresentar-se como conflitantes, ao ser considerado um ambiente aberto, principalmente quando se leva em conta a intenção de realização de operação cooperativa entre entidades participantes deste ambiente.

Uma estrutura de Ambiente de Serviços Abertos pode contornar este aparente conflito. Neste ambiente, entidades podem oferecer e utilizar serviços que são apresentados na rede de comunicações. Os serviços são oferecidos de maneira distribuída e desregulamentada, sendo gerenciados em um contexto local (ambiente local) de forma a preservar a autonomia da entidade que o oferece. Em tal ambiente, elementos locais podem operar de forma cooperativa e coerente com elementos remotos, através do compartilhamento de serviços.

A cooperação em Ambientes de Serviços Abertos corresponde à junção de esforços e operações para a obtenção de propósitos comuns. Esta cooperação deve ser coerente, isto é, operar de forma logicamente integrada, consistente e inteligível pelos seus componentes e compreenderá o estabelecimento de um contexto e controle globais, além do estabelecimento de princípios homogêneos e de relações coordenadas entre os componentes.

Cooperações em um Ambiente de Serviços Abertos podem estabelecer-se de maneira permanente ou através da criação de federações limitadas, para o desenvolvimento de ações coordenadas na obtenção de metas comuns.

Diferentes pontos de vista podem ser considerados na avaliação de cooperações em Ambientes de Serviços Abertos. A utilização de pontos de vista distintos facilita o entendimento de sistemas complexos, permitindo focalizar-se a atenção em importantes aspectos, abstraindo-se os detalhes irrelevantes do ponto de vista específico. Para os Ambientes de Serviços Abertos pode-se citar como importantes a análise dos pontos de vista organizacional, operacional, de projeto, de engenharia e desenvolvimento e técnico.

### **ARQUITETURA PARA AMBIENTES DE SERVIÇOS ABERTOS**

Uma arquitetura para Ambientes de Serviços Abertos deve comportar tanto as características locais como as globais de seus componentes. Os principais aspectos descritivos de uma arquitetura destas podem ser citados como: ambiente operacional local, ambiente de aplicações locais, terminologia local e global, modelos e estruturas do empreendimento, formas e mecanismos de cooperação.

O **ambiente operacional local** corresponde aos elementos técnicos locais (estações de trabalho, sistemas operacionais, linguagens e bibliotecas) e aos serviços, ferramentas e características de suporte a aplicações (serviços de processamento e transmissão de arquivos, qualidade de serviços, capacidade de recursos, taxa de erro, etc.).

O ambiente de aplicações locais refere-se à estrutura local para projeto, implementação, operação, gerenciamento e administração de aplicações, além das considerações de confiabilidade, disponibilidade e desempenho operacional.

A Terminologia Local e Global corresponde à compreensão de termos, nomes, serviços e protocolos pelos componentes e interfaces.

Os mecanismos e formas de cooperação dizem respeito ao conjunto de metodologias para o desenvolvimento de negociações e estabelecimentos de concordâncias sobre atribuições, recursos comuns, atividades e decisões conjuntas, níveis de autoridade.

Estes aspectos podem compor uma estrutura com níveis hierárquicos crescentes ("bottom-up") para a composição de entidades componentes do Ambiente de Serviços Abertos. Nesta estrutura, os níveis mais baixos encarregam-se das atribuições do ambiente local, enquanto os outros efetuam as negociações, estabelecem e desenvolvem trabalho cooperativo.

### **TRABALHO COOPERATIVO**

O desenvolvimento de trabalho cooperativo leva à formação de grupos compostos por entidades ou elementos pertencentes a diferentes organizações.

A definição de grupos, e a conseqüente admissão de outros elementos ao grupo, leva ao estabelecimento de um conjunto de aspectos que devem ser compartilhados pelos seus elementos, tais como: propriedades (atributos técnicos), comportamento (propriedades dinâmicas), conhecimento e pontos de vista (compreensão de protocolos e serviços), autoridade e direitos (atribuição de áreas de influência e classificação de operações).

A descrição de conjunção de grupos e de suas formas de interação leva ao conceito de Domínios. A composição de Domínios requer a descrição dos elementos dessa união e de métodos para identificação e localização de grupos e elementos, administração de grupos, integração de serviços e para gerenciamento de interações.

A inter-relação cooperativa e coerente entre componentes, entidades, grupos e domínios requer o estabelecimento de uma infraestrutura de serviço para a integração de todos os elementos apesar da autonomia e privacidade dos elementos. Assim são definidos serviços de administração, informação e negociação. Os serviços de administração gerenciam o estabelecimento e inter-relação entre os elementos ou composições destes. Os serviços de informação encarregam-se de obter e difundir as informações sobre diferentes serviços dentro de certo domínio. Os serviços de negociação encarregam-se das tratativas necessárias ao reconhecimento de serviços desejados e ao estabelecimento de parâmetros de operação de serviços interessantes para cliente e servidor.

### **NOVAS APLICAÇÕES E FORMAS DE PROCESSAMENTO DISTRIBUÍDO**

Na Tabela 1 mostram-se diferenças entre as hipóteses consideradas no projeto de sistemas tradicionais (distribuídos ou "stand-alone") e de sistemas ODP.

Um sistema distribuído tradicional em geral é fechado, caracterizado por um número limitado de componentes, pertencentes a uma única organização que controla o projeto, instalação, operação e gerenciamento do sistema. A distribuição resulta de decisões de projeto (por ex. tolerância a falhas). Os elementos são fortemente acoplados, através de conceitos e mecanismos comuns e a sua autonomia é limitada. Os sistemas distribuídos abertos, pelo contrário, podem espalhar-se por diversas autoridades de gerenciamento e controle, além da necessidade de esquemas de nomes relativos a contexto. A construção de uma memória virtual compartilhada não é prática e a consistência de dados e estados converge mais lentamente que em sistemas tradicionais, uma vez que programas e dados podem movimentar-se entre os diversos locais. A distribuição é

**TABELA 1**  
COMPARAÇÃO DE SISTEMAS TRADICIONAIS E SISTEMAS ODP

CARACTERÍSTICAS	SIST. TRADICIONAIS	SIST. ABERTOS DISTRIBUÍDOS
CONTROLE	CENTRAL	AUTONOMIA
ESPAÇO DE NOMES	NOMES GLOBAIS	NOMES DE FEDERAÇÃO
MEMÓRIA COMPARTILHADA	GLOBAL	ESTADOS LOCAIS ENCAPSULADOS
CONSISTÊNCIA	GLOBAL	FRACA
EXECUÇÃO	SEQUENCIAL	CONCORRENTE
FALHAS	VULNERÁVEL	TOLERANTE
LOCALIDADE DE INTERAÇÃO	LOCAL	REMOTO
LOCAÇÃO	FIXA	MIGRÁVEL
CONFIGURAÇÃO	FIXA	INCREMENTÁVEL
AMBIENTE	HOMOGÉNEO	HETEROGÉNEO

consequência de condições do ambiente e não de decisões de projeto. A cooperação é limitada no tempo e espaço, permanecendo os elementos sob controle local. Por isso é mais apropriado caracterizá-los como federações fracamente acopladas.

Por outro lado, os campos de aplicação de processamento distribuído aberto são virtualmente ilimitados, desde aplicações residenciais de lazer, até às redes globais de comunicação (por ex. IBCN = "Integra-

ted Broadband Communication Systems"), incluindo todos os tipos de sistemas de voz, texto e imagem (CAD cooperativo, sistemas integrados de engenharia de software, vídeo-telefone, correio multimídia, telemedicina, tele-diagnose., CSCW = "Computer Supported Cooperative Work", etc).

### 3. Aspectos da padronização ODP

Nesta secção descrevem-se os principais conceitos desenvolvidos até hoje na ISO/IEC JTC1/ SC1(ODP) e no CCITT SGVII.Q19 (DAF). Consideram-se os seguintes documentos, ainda a nível de "Working Drafts" do "Basic Reference Model of ODP":

- Generalidades e Guia de Uso ,ISO 10746-1, X.901, [3] ;
- Modelo Descritivo( ISO 10746-2, X.902) [4] ;
- Modelo Prescritivo ( ISO 10746-3, X.903) [5];
- Semântica Arquitetônica ( ISO 10746-5, X.905) [6];
- Documento de trabalho no Trader- ODP [7];
- Lista de aspectos do ODP em aberto ( Junho de 1992) [8].

#### 3.1. Conceitos e definições

##### REQUISITOS E ORGANIZAÇÃO DO RM- ODP

Os componentes de um Sistema ODP, tendo em vista a distribuição da informação na empresa, o carácter necessário de abertura, e a diversidade de equipamentos, sistemas operacionais e computacionais, de autoridades e aplicações, podem ser:

- heterogêneos;
- capazes de operação concorrente;
- distribuídos física/logicamente;
- portáteis de forma estática e dinâmica;
- interoperáveis.

Os requisitos de usuários ODP são inúmeros, tais como disponibilidade, confiabilidade, tolerância a falhas, integração e compartilhamento de recursos, melhoria de desempenho, balanceamento de carga, distribuição de gerenciamento, segurança, integridade e

consistência de dados e transparência de distribuição. Desta forma, os requisitos básicos do Modelo de Referência ODP devem :

- prover um modelo consistente para o desenvolvimento em separado de diversos padrões;
- definir áreas que necessitem de padrões ODP e o seu relacionamento com outros padrões;
- descrever um conjunto básico de ferramentas a serem usadas na definição dos padrões;
- prover um conjunto consistente de conceitos e terminologia comuns.

Os padrões ODP devem especificar a comunicação e a coordenação da informação distribuída relevante para uma empresa. As principais categorias previstas são:

- um modelo geral de referência (RM-ODP) com a definição dos conceitos e identificação de funções comuns;
- modelos de referência específicos que cubram tipos especiais de empresas, usando os conceitos e funções do RM-ODP, e definindo detalhes conceituais e funções específicas adicionais ( por ex. TINA = "Telecom Intelligent Network Architecture");
- padrões para a realização de funções comuns e genéricas para um espectro amplo de aplicações ( por ex. Trader);
- padrões para a realização de funções específicas de aplicações particulares ( por ex. interfaces de chamadas de conexões telefônicas).

## VIEWPOINTS E ESTRUTURA DO SISTEMA

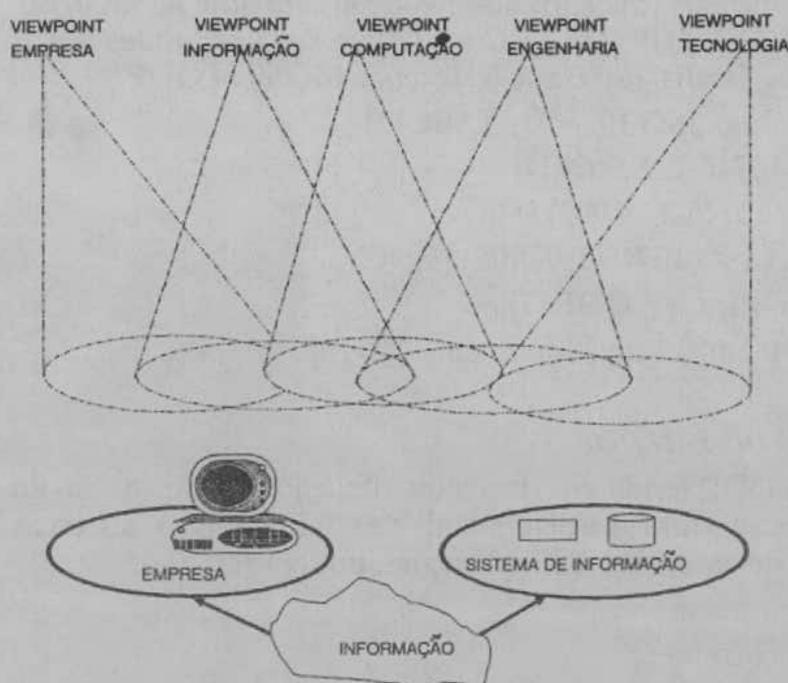


FIGURA 1  
VIEWPOINTS EM ODP

Para lidar com a complexidade do sistema ODP, ele deverá ser descrito segundo diversos pontos de vista (Figura 1) representando cada um deles uma abstração diferente do sistema original. A descrição do sistema ODP por cada ponto de vista permite a verificação da completude da descrição e da consistência entre diferentes pontos. Atualmente reconhecem-se cinco pontos de vista do modelo ODP:

- **empresarial:** fornece uma descrição de onde e como o sistema de informação está colocado dentro da empresa ( ou parte dela). Procura-se caracterizar os requisitos empresariais

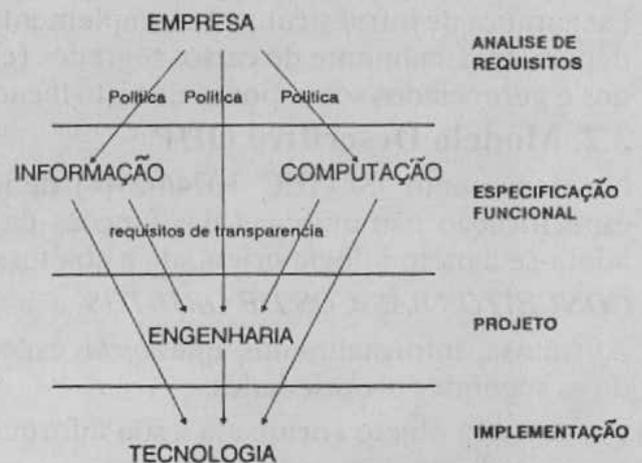
para o sistema IT que deverá ser expresso em termos de objetos que representem papéis, políticas de negócio e de gerenciamento;

- **de informação:** modelam-se as estruturas e o fluxo de informação, além das regras e restrições que devem governar a sua manipulação;

- **computacional:** ponto de vista dos projetistas da aplicação preocupados com os algoritmos e tipos de dados e com a definição dos requisitos de transparência de distribuição;
- **de engenharia:** considerado pelos especialistas de sistemas operacionais e sistemas de comunicação que lidam com processadores, memórias e redes de comunicação e que devem definir características de desempenho, interconectividade e distribuição;
- **tecnológico:** os principais usuários são aqui os responsáveis pela configuração, instalação e manutenção do hardware e do software.

Especificadores de padrões ODP são uma classe especial de usuários, que têm como objetivo o desenvolvimento de especificações e que deverão dispor para tal de linguagens apropriadas. Como se mostra na Figura 2, essas especificações deverão ser usadas

nas diversas fases do ciclo de vida de um sistema ODP. Por exemplo a linguagem empresarial levará à análise dos requisitos e políticas do sistema; as linguagens de informação e computação fornecerão documentos válidos para a fase de especificação funcional, por ex., especificações dos requisitos da informação (sua organização, aquisição, processamento, armazenamento e representação no sistema ODP) e os modelos de interação (acesso a todas as interfaces) e de construção (infraestrutura distribuída para programação de funções). A linguagem de engenharia permite a especificação das funções de processamento, armazenamento e comunicação necessárias à implementação do sistema: em particular deverão considerar-se aqui todos os requisitos de transparência de distribuição, de portabilidade e de interconectividade. Finalmente, através da linguagem tecnológica, deverão ser formulados requisitos de implementação, identificadas tecnologias a serem usadas, esclarecidos testes, etc.



**FIGURA 2**  
USO DE VIEWPOINTS NO CICLO DE VIDA

## CONCEITOS DE GERENCIAMENTO E SEGURANÇA ODP

Um objeto no ambiente ODP terá, além da sua funcionalidade própria (serviços que providencia), funções de gerenciamento. Os objetos gerenciados (hardware, software, estrutura de dados, pessoas,...) são entidades às quais se aplica uma política de gerenciamento e cujo comportamento será monitorado/modificado pelo gerenciador (ser humano ou computador). As interações entre eles seguem os princípios definidos no Gerenciamento OSI, com o uso do protocolo CMIP.

As *funções de gerenciamento ODP* são: configuração, qualidade de serviço, contabilidade, monitoramento e definição de política de gerenciamento. Não existe uma correspondência direta com as funções de gerenciamento OSI. Por exemplo, o gerenciamento de qualidade é uma generalização dos gerenciamentos de desempenho e de falhas e as funções de definição de política de gerenciamento deverão incluir funções de segurança.

Definem-se *domínios*, no âmbito de gerenciamento ODP, como conjuntos de objetos controlados por uma política comum, estabelecida pelo objeto de controle de domínio:

qualquer objeto terá que pertencer a, pelo menos, um domínio, para poder ser gerenciado. A política de gerenciamento deverá, entre outras coisas, especificar as operações de gerenciadores sobre os objetos. Domínios não são necessariamente disjuntos, podendo um objeto pertencer a vários (por ex. um roteador).

As *funções de segurança* providenciam serviços de proteção para os objetos. Existirão funções de segurança visíveis nos diversos Viewpoints apresentados. Por exemplo, do ponto de vista empresarial, serão definidos domínios, políticas de segurança para controle de acesso ao domínio e as restrições de interação entre eles. Do ponto de vista de informação consideram-se os mais diversos tipos de controle de acesso à informação, de auditorias e de criptografia. Do ponto de vista computacional as funções de segurança providenciam controle ativo de invocações de objetos. Do ponto de vista de engenharia serão considerados vários aspectos de implementação de segurança: por exemplo deverão existir funções de segurança a nível básico dos sistemas operacionais (segurança de infraestrutura). A implementação da política de segurança num domínio dependerá finalmente de certos segredos (chaves) e mecanismos criptográficos, trocados e gerenciados sob o ponto de vista tecnológico.

### 3.2. Modelo Descritivo ODP

No documento ISO/IEC 10746-2 [4] definem-se todos os conceitos necessários à especificação não ambígua das funções da parte 3 (Modelo Prescriptivo). Para tal, adota-se a metodologia orientada a objetos.

#### CONCEITOS BÁSICOS DE OBJETOS

Afirma-se, informalmente, que *objetos* executam funções e oferecem serviços, possuindo as seguintes propriedades:

- cada objeto **encapsula** a sua informação e interage com outros objetos de uma coleção;
- a informação contida num objeto não é diretamente acessível por outros, que só poderão manipulá-la a partir de **interações**;
- o **comportamento** de um objeto é descrito em termos das suas interações com outros objetos, e tipicamente subdividido em **interfaces**. A interação ocorre nos pontos de interação (também chamados de **operações**);
- o comportamento do objeto é definido por um **"template"**;
- objetos diferentes de uma coleção podem ser identificados e endereçados explicitamente.

*Ações* representam qualquer coisa que aconteça referente a um ou mais objetos. As ações não são necessariamente observáveis (no caso, internas de um objeto). O termo evento foi substituído em ODP por **ação atômica**. *Interações* são ações compartilhadas por vários objetos e modeladas como **sincronizações**.

O **Comportamento** de um objeto é definido como o conjunto de sequências de ações que podem ocorrer. O **estado inicial** de um objeto é dual do seu comportamento, sendo este determinado pelo seu valor. O comportamento pode ser complexo, incluindo atividades paralelas, com ações sobrepostas ou entrelaçadas.

Os objetos podem ter várias **Interfaces**, sendo cada uma a projeção do comportamento visível do objeto, em termos de um conjunto predefinido de ações observáveis. Em ODP chama-se de **assinatura** a esse conjunto de ações observáveis na interface. A assinatura descreve os nomes das operações, tipos de parâmetros de requisição e término e a ordem da sua listagem. As diversas interfaces de um objeto são classificadas, normalmente, em interfaces funcionais (serviços) e de gerenciamento.

### CONCEITOS DE ESPECIFICAÇÃO

**Template** é a especificação das características comuns de uma coleção de objetos. **Tipos** são predicados e implicitamente classificam objetos. **Classe** é um conjunto de objetos prescritos por um tipo. Uma **subclasse** é um subconjunto de outra classe ( **superclasse** desta). Um tipo de subclasse é **subtipo** do tipo da superclasse correspondente ( o seu **supertipo**).

Objetos podem ser gerados por **instanciação** de um template. Contudo, outros objetos poderão existir que satisfaçam o tipo de template considerado . Estes objetos são designados de **instâncias** do template. Na figura 3 estão ilustradas as relações entre esses

### OBJETOS E CLASSES

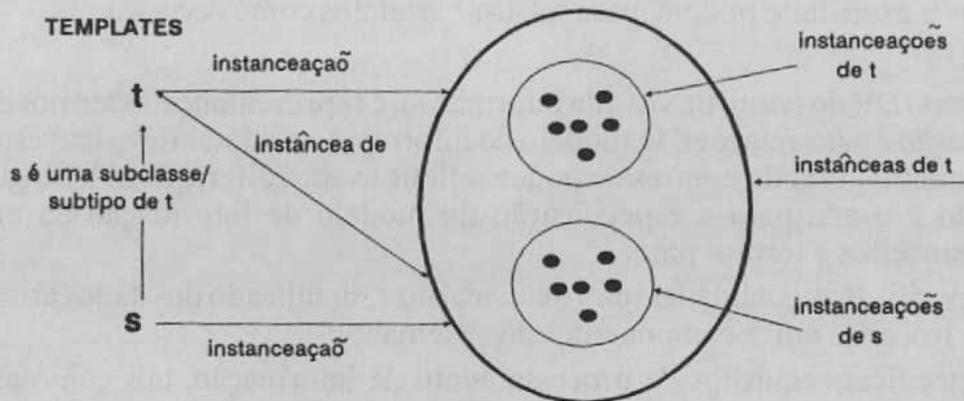


FIGURA 3  
RELAÇÕES ENTRE TEMPLATES, CLASSES E OBJETOS

conceitos.

O conceito de **herança**, conhecido na comunidade de programação orientada a objetos, pode( em ODP), tomar as formas de subtipagem ou herança incremental. Nesta última, deriva-se um novo template modificando um já existente.

### CONCEITOS DE ARQUITETURA

Uma série de conceitos são ainda introduzidos mais formalmente, como grupos, domínios e configuração de objetos. Os sistemas ODP satisfazem diversas propriedades :

- **transparência**, que consiste em ocultar de um usuário o comportamento potencial de partes do sistema;
- **obrigações**, termo genérico para propriedades relacionadas à cooperação entre objetos. Algumas formas de obrigação são **contratos** e **conjuntos de requisitos** de qualidade de serviços.

O comportamento será caracterizado em relação à sua localização ( temporal e espacial), ao contrato estabelecido, à causalidade e à confiabilidade. Uma interação pode ser estabelecida entre duas ou mais interfaces através do processo de ligação ("binding"). **Trading** ( negociação ) é um tipo especial de interação em que a informação sobre contratos novos ou potenciais é trocada com uma terceira parte ( objeto). Trading envolve as atividades de **exportação** e de **importação** que serão esclarecidas nas próximas secções.

### 3.3 O Modelo Prescriptivo ODP

Na parte 3 do Modelo Básico de Referência [5] definem-se, em maior detalhe, linguagens para cada viewpoint, destinadas a justificar a necessidade de funções ODP, e a

descrever seu papel e relação com outras funções e as funções necessárias para o Processamento Aberto Distribuído.

## LINGUAGENS DE VIEWPOINTS

### Empresa

Uma especificação de um sistema ODP, sob o ponto de vista da empresa, está limitada aos aspectos ODP e não deve ser confundida com a especificação da empresa como um todo. A linguagem é usada pelo especificador para exprimir requisitos das funções e sistemas, bem como as políticas. No modelo da linguagem definem-se novos conceitos (agentes, artefatos, comunidade, permissão, proibição, privilégio,...). Por ex. um agente é um objeto que executa ações que alteram obrigações, permissões e proibições, enquanto que artefatos são objetos que não têm essa capacidade. Agentes têm responsabilidades a executar e podem, para tal, usar artefatos como recursos

### Informação

Um sistema ODP, do ponto de vista da informação, é representado em termos de objetos de informação e suas relações. O modelo de informação a adotar deve corresponder ao comportamento geral da empresa e poder refletir as suas alterações. A Linguagem de Informação é usada para a especificação do modelo de informação da empresa e introduz conceitos e termos para:

- especificar as qualidades informativas, isto é, significado dos dados armazenados ou trocados entre componentes do sistema;
- especificar requisitos de processamento de informação, tais como armazenamento, aquisição, organização, processamento e apresentação.

Os conceitos introduzidos são diversos (esquema, relação, regra de integridade, restrição de cardinalidade, partição, etc). Assim, **esquemas** são templates de objetos compostos de informação, satisfazendo certa proposição (a regra de integridade). Esquemas dinâmicos podem ser criados a partir de invariantes, através de várias ações atômicas (criação, destruição, reclassificação, classificação e desclassificação).

### Computação

A linguagem de especificação, do ponto de vista computacional, lida com conceitos usados para especificar a estruturação e interação dos componentes das aplicações distribuídas. Contém ainda regras de interação para acesso consistente, transparente e distribuído a todas as interfaces, bem como regras de construção de objetos computacionais para a infraestrutura de implementação de funções.

Existem dois aspectos fundamentais no modelo computacional. O primeiro é o modelo de interação, que introduz os conceitos de interrogação e anúncio como formas de invocação (instanciação de templates computacionais). O modelo estabelece a parametrização e a semântica de falhas na interação. Por outro lado, definem-se os requisitos de interfaces e seus tipos em relação às assinaturas, qualidade de serviços e atributos de transparência, comportamento e papéis (objeto cliente e servidor). O segundo aspecto diz respeito à construção de configurações de objetos, suportando a montagem e o desmantelamento de redes complexas de objetos interoperáveis. Estes mecanismos formam a base para a identificação de funções genéricas, necessárias para suportar a especificação funcional, tais como "objeto fábrica" e "trader".

### Engenharia

A linguagem de especificação, do ponto de vista de engenharia, permite descrever a organização de uma infraestrutura abstrata para a execução das aplicações distribuídas especificadas, identificar componentes para o gerenciamento da distribuição física e dos

recursos locais, assim como definir o papel dos diversos objetos de suporte às funções ODP. O modelo de engenharia é representado na figura 4.

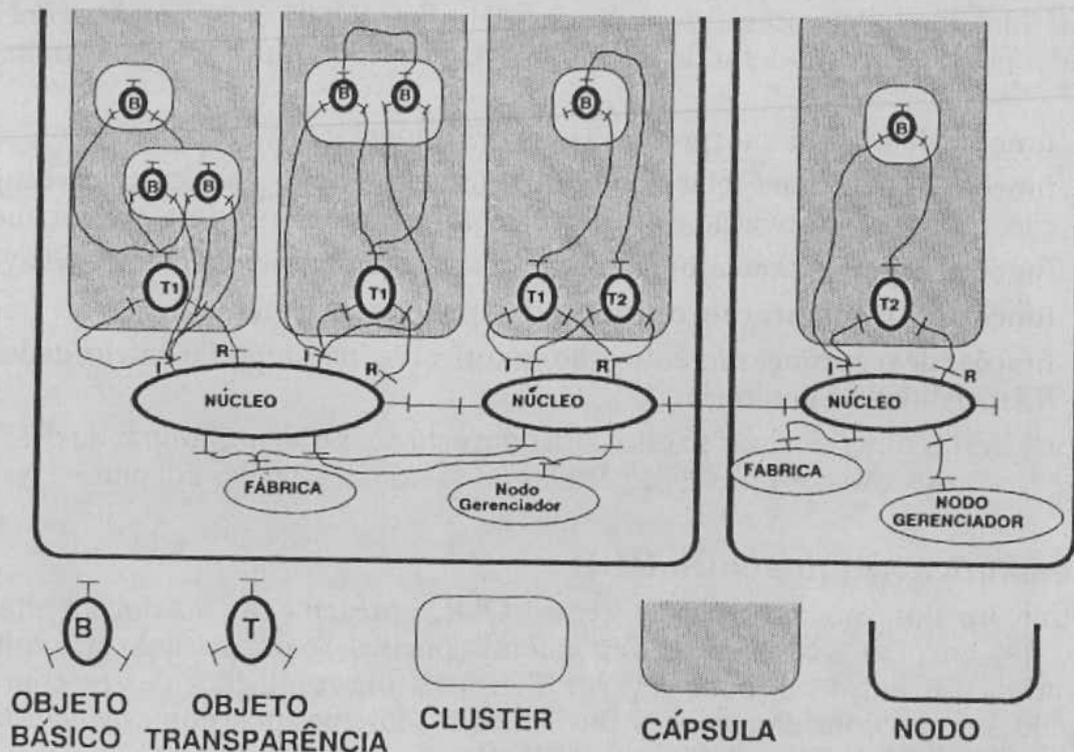


FIGURA 4  
MODELO DEFINIDO NA LINGUAGEM DE ENGENHARIA

**Nodo** é um domínio de gerenciamento constituído de diversos tipos de objetos:

- **núcleo**, que providencia funções de processamento, armazenamento e comunicação e possui as **interfaces I de interpretação** (para gerenciamento de threads dos objetos básicos de engenharia) e **R de recursos** (colocando à disposição as funções de recursos do núcleo);
- objetos de **transparência**  $T_i$  (de acesso, de localização, de migração, de concorrência, de recursos, de falhas e de réplicas) destinados a tornar transparentes, para os objetos básicos de engenharia, os diversos aspectos de comunicação;
- objetos **básicos de engenharia**  $B_i$ , correspondentes à funcionalidade propriamente dita de aplicações ODP;
- **Nodo Fábrica**, com as funções de criação de novas cápsulas de nodos  $B_i$ ;
- **Nodo gerenciador**.

Os objetos básicos agrupam-se em **Clusters** que representam as unidades de engenharia a serem ativadas e passíveis de migração. Uma **cápsula** é um conjunto de clusters e de objetos de transparência, residentes num certo nodo e sujeitos a um núcleo comum: formam a unidade de proteção e de atomicidade de falha. Outros componentes do modelo são ainda definidos como **Canal**, abstração de comunicação entre um "**plug**" (interface do canal no objeto iniciador) e "**socket**" (interface do canal do objeto respondedor).

#### Tecnologia

Esta linguagem é usada para exprimir a forma de implementação das especificações ODP, identificar especificações tecnológicas relevantes para a construção do sistema,

construir a taxonomia correspondente e exprimir informações relevantes para testes do sistema. Não existem ainda detalhes para esta linguagem.

## FUNÇÕES GENÉRICAS

A seção final do modelo prescritivo procura identificar e definir as funções genéricas, chamadas nos diversos modelos de viewpoints. As principais funções hoje reconhecidas são :

- **funções do núcleo:** interpretação e recursos funcionais;
- **funções de transparência:** relocação, transação, acesso, checkpoint e recuperação, réplicas, desativação e ativação, migração, interceptação e armazenamento;
- **funções de gerenciamento:** nodo, cápsula, cluster, fábrica e coleta "garbage";
- **funções de configuração:** descrição de tipos, gerenciamento de tipos e trading;
- **funções de segurança:** autenticação, autorização, não-repúdio, integridade, confidencialidade e auditoria.

A maioria destas funções ainda se encontra num estágio muito preliminar de desenvolvimento, com exceção das funções do Trader que serão detalhadas adiante.

## 3.4. Semântica Arquitetônica ODP

As definições dos conceitos para sistemas ODP, apresentadas no documento ISO 10746-2, [4], em muitos casos são extremamente concisas. Se por um lado essa concisão é desejável, por outro ela pode levar a diferentes interpretações desses conceitos, principalmente se considerarmos a formalização dos mesmos por especialistas de diferentes Técnicas de Descrição Formal (TDFs).

Por exemplo, se considerarmos os conceitos **ação** e **interação** e considerarmos especialistas em diferentes TDFs, muito provavelmente interpretações diferentes surgirão para esses dois conceitos.

Para resolver esse tipo de problema, uma metodologia, que vise uma tradução dos conceitos ODP para as construções das diferentes TDFs, faz-se necessária (Figura 5).

Numa primeira etapa dessa metodologia, deve ser definido um conjunto de conceitos arquitetônicos ODP. Esses conceitos devem ser básicos e as suas definições devem ser precisas a tal ponto que não possam ser questionadas.

Por exemplo, em [18] há definições mais detalhadas dos conceitos ação e interação:

uma **ação** é o estabelecimento, por um objeto, de um item de informação que possa ser referenciado por esse objeto. Ao estabelecer essa informação, outras informações, estabelecidas por ações anteriores, podem ser referenciadas pelo objeto. Associada a cada ação, há um momento a partir do qual a

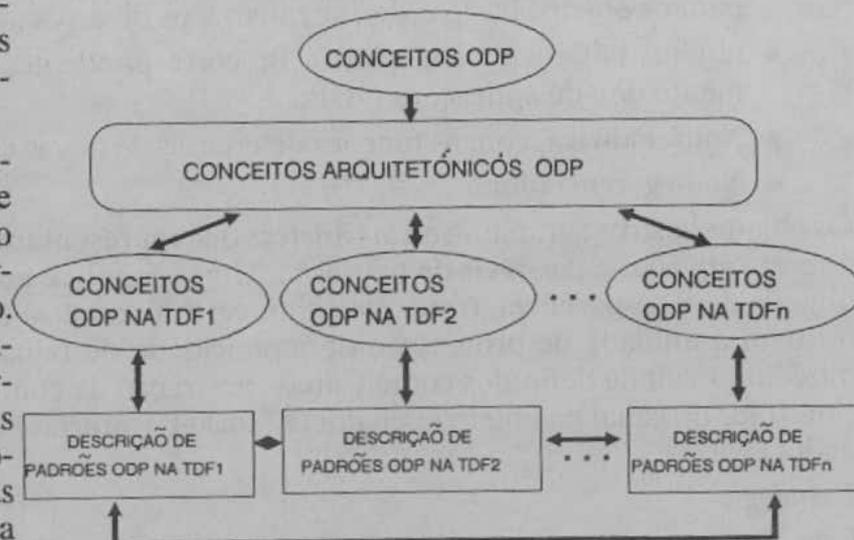


FIGURA 5  
SEMÂNTICA ARQUITETÔNICA ODP

informação estabelecida pela ação pode ser referenciada pelo objeto;

Uma **interação** é uma ação na qual o ítem de informação é estabelecido por dois ou mais objetos, sendo que essa informação pode ser referenciada por todos os objetos envolvidos.

Numa segunda etapa dessa metodologia, os conceitos arquitetônicos ODP devem ser traduzidos, por especialistas, para os conceitos arquitetônicos de diferentes TDFs.

Por exemplo, os conceitos ODP acima definidos poderiam ser traduzidos na TDF "Language of Temporal Ordering Specification (LOTOS)" [15] da seguinte forma:

- um **evento observável** ou um **evento interno** LOTOS podem ser considerados ações;
- um evento observável LOTOS pode ser considerado uma interação (mas não um evento interno).

Na TDF "Extended State Transition Language" (Estelle) [16], esses mesmos conceitos poderiam ser traduzidos da seguinte forma:

- uma **transição** de estado Estelle pode ser considerada uma ação.
- uma **entrada** numa fila Estelle pode ser considerada uma interação.

Essa metodologia, que estabelece uma base semântica para os conceitos ODP para as diferentes semânticas das TDFs, é denominada Semântica Arquitetônica ODP. A existência de tal semântica permite o estabelecimento de relações entre expressões em diferentes TDFs, que são referentes aos mesmos conceitos ODP. Consequentemente, permite o estabelecimento de relações entre descrições em diferentes TDFs do mesmo padrão ODP.

De acordo com o documento ISO 10746-5,[6], os objetivos de uma Semântica Arquitetônica ODP são:

- possibilitar a formalização dos conceitos de modelagem ODP;
- permitir um desenvolvimento prudente e uniforme das descrições formais dos padrões ODP;
- agir como um elo de ligação entre os conceitos de modelagem ODP e os modelos semânticos das TDFs;
- fornecer uma base para uma comparação uniforme e consistente entre descrições formais, realizadas em diferentes TDFs, do mesmo padrão ODP.

Esse documento apresenta os resultados de um primeiro trabalho de desenvolvimento de uma Semântica Arquitetônica ODP. Inicialmente, o conjunto de conceitos arquitetônicos ODP foi formado através da união do conjunto de conceitos básicos de modelagem com o conjunto de conceitos de especificação

O conjunto de conceitos básicos de modelagem é constituído de 15 elementos:

ação; ação atômica; objeto; localização (de uma ação); localização no espaço; localização no tempo; comportamento (de um objeto); interação; experimento ("observation run"); observação; ação observável; ponto de interação; comunicação; estado (de um objeto); interface.

O conjunto de conceitos de especificação é constituído de 29 elementos:

composição (de objetos e de ações); objeto composto; decomposição; ambiente (de um objeto); compatibilidade comportamental; refinamento; traço; "template"; template-ação; template-interface; assinatura (de uma interface); instanciação (de um template); criação (de um objeto); destruição (de um objeto); introdução (de um objeto); criação (de uma interface); destruição (de uma interface); tipo (de um objeto); classe (de objetos); subclasse/superclasse; instância (de um template); tipo-template; classe-

template (de objetos); subtipo/supertipo; tipo-template-interface; instância (de um template-interface); classe-derivada/classe-genitora; herança incremental; polimorfismo incluído.

Posteriormente, os conceitos arquitetônicos ODP são traduzidos para os conceitos arquitetônicos das TDFs LOTOS, "Specification and Description Language (SDL)" [14] e Z [17].

Esse é todo o trabalho relativo à Semântica Arquitetônica ODP, que foi publicado pela ISO, na forma de "Working Draft", até o presente momento. Portanto, em função dos objetivos traçados pela própria ISO, resta ainda muito trabalho a ser realizado para o desenvolvimento desse tema.

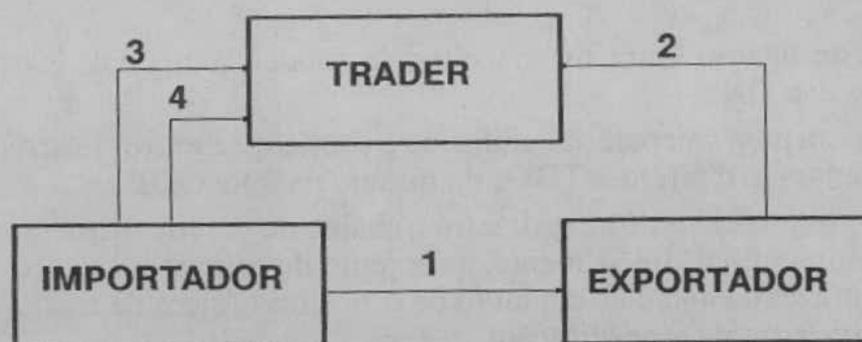
### 3.4. Especificação da função TRADER

Os comentários apresentados nesta seção baseiam-se em documento preliminar divulgado em Junho de 1992, e que servirá de base para o futuro padrão TRADER ODP. O Trader é a primeira componente ODP a ser padronizada, devendo servir de referência para as outras.

#### DEFINIÇÃO

Um Trader pode ser visto como um objeto a partir do qual um outro pode comprar (IMPORTAR) ou vender (EXPORTAR) serviços de/para outros objetos. Para tal, um certo objeto informa ao Trader que existe e providencia informação (exporta) sobre si mesmo. Um outro objeto contacta o trader, solicita e recebe dele (importa) o nome de um objeto com as características desejadas. O objeto exportador pode, a qualquer instante, retirar a sua oferta (Withdraw). Por outro lado o trader seleciona um provedor adequado de serviços, no conjunto de objetos exportadores, mapeando-o com as características expressas pelo importador.

Na figura 6 representa-se o relacionamento do trader com os seus clientes. Além destes



1. OPERAÇÕES QUE INVOCAM INTERAÇÃO
2. OFERTAS DE SERVIÇOS DO EXPORTADOR
3. SOLICITAÇÃO DE SERVIÇOS DO IMPORTADOR
4. OPERAÇÕES DE INICIAÇÃO

**Figura 6**  
INTERAÇÃO ENTRE TRADER E SEUS CLIENTES

aspectos as funções do Trader incluem a partição de serviços exportados, segundo alguma política e o estabelecimento de cooperação com outros Traders.

#### ESPECIFICAÇÃO SEGUNDO O PONTO DE VISTA DA EMPRESA

São diversos os agentes envolvidos na função Trading: exportadores, importadores, administradores, controladores de política e os traders propriamente ditos. O domínio- empresa do

trading é um conjunto de objetos com papéis de exportação e importação em relação ao trader, controlado por objetos administradores. A política de trading terá diversos requisitos para a oferta de serviços, particionamento das ofertas, estrutura organizacional, posição geográfica, custos de controle, segurança, etc. Uma Federação de Traders é uma configuração com políticas que permitem acesso aos serviços oferecidos por cada

um. Existem alguns princípios que caracterizam uma federação como, por exemplo, a liberdade dos componentes de sair ou entrar nela.

Quando se define uma política de trading, uma série de **requisitos de qualidade** de serviços podem ser impostos em relação à disponibilidade da empresa trading (objetos trader devem ser altamente transparentes a falhas), ao desempenho (por ex. podem ser incluídas aplicações de tempo-real), à remoção de ofertas de serviços não mais disponíveis e em relação à consistência dos serviços exportados pelo trader com os serviços reais existentes.

## **PONTO DE VISTA DE INFORMAÇÃO**

Para que se realize trading é necessário um método que descreva:

- Referências de objetos e de interfaces;
- Tipos e qualidade de ofertas de serviços requisitados ( especificação do importador);
- Características e capacidades de objetos ( propriedades estáticas de serviços de exportadores);
- Disponibilidade de objetos ( propriedades dinâmicas de serviços de exportadores);
- Seleção ( comparação e ligação de importadores com exportadores);
- Partição dos domínios de informação de trading;
- Efeitos da federação na informação de trading do domínio.

As **ofertas de serviços** contêm propriedades de serviços, podem designar uma política de exportação e nomeiam uma interface de um dos tipos "Withdrawn", Exportada ou Importada. Um exportador pode fazer diversas ofertas de serviços. O domínio de informação de trading será o conjunto de serviços ofertados e disponíveis para clientes de um trader.

O **modelo de informação** descreve como o conjunto de ofertas de serviços pode ser particionado em subconjuntos e quais são as relações entre eles. Em geral, os usuários dos serviços particionam o domínio em conjuntos específicos do usuário, chamados de **contextos de trading**. Contextos são assim conjuntos de ofertas de serviços, dando apoio à sua busca e colocação. A partição do domínio poderá ser feita através da estrutura organizacional ( refletindo estruturas administrativas e organizações que podem evoluir dinamicamente) ou pelo tipo de serviços ( globais, sem referência a estrutura de contexto; globais, mas restritos a instâncias específicas de tipos ou definidos num contexto). Em termos do padrão Trader ODP não há motivos para restringir a definição de estruturas locais de contexto, permitindo-se assim que elas reflitam requisitos de usuários, políticas da empresa e realidades físicas existentes.

Cada serviço oferecido pelo trader deve ser uma instância de um tipo, que especifica as características comuns de uma coleção de objetos, chamadas de **propriedades do serviço**: cada propriedade consiste de um nome para identificação e de um tipo de dados ( um conjunto de possíveis valores, incluindo defaults). Por exemplo na Tabela 2, PRINTER pode ser o nome de um tipo de serviços, onde são mostradas as suas propriedades. Uma exportação de serviços é uma instância do tipo, exportada pelo exportador. As propriedades podem ser estáticas ou dinâmicas. As estáticas (armazenadas pelo trader) descrevem a capacidade do serviço e as dinâmicas (calculadas pelo trader quando da requisição pelo importador) a sua disponibilidade. No exemplo da Tabela 2, as propriedades são todas estáticas. Uma propriedade dinâmica poderia ser "QueueLength : Integer".

**TABELA 2**  
Propriedades de Serviços

TIPO DE SERVIÇO	INSTÂNCIA DE SERVIÇO EXPORTADO
PrinterType : DotMatrix, Laser Identifier : String Localização: String PaperSize : A4, A3 ( default A4) CostPerPage: Integer	PrinterType = Laser Identificador: "L5" Localização:" Lab DCA" PaperSize = A4 CostPerPage = 0,5

O principal objetivo do trader é o de conseguir a importação da oferta de serviço de exportação mais adequada. Poderão existir diversos exportadores que satisfaçam os requisitos gerais do importador: este terá ainda a possibilidade de especificar requisitos adicionais com uma expressão de propriedades de serviços. A expressão será formada por valores das propriedades, operadores de comparação (=, >, <, ...) e operadores lógicos ("and", "or", ...). Uma restrição de igualdade ("matching") é uma expressão de propriedade usada para comparar ofertas de serviços exportados com os requisitos do importador. Por exemplo:

*PrinterType = Laser AND CostPerPage < 1*

Aplicam-se **critérios de seleção** para escolher as ofertas de serviços mais apropriadas, através de sequências de expressões de propriedades, que servirão de regras de comparação entre ofertas diferentes. Por exemplo:

*IF X.Location = "UNICAMP" AND Y.Location <> "UNICAMP" AND X.CostPerPage < 1  
THEN X ELSE*

*IF X.Location <> "UNICAMP" AND Y.Location = "UNICAMP" AND Y.CostPerPage < 1  
THEN Y ELSE*

*IF X.CostPerPage < Y.CostPerPage THEN X ELSE Y*

Poderão existir os mais diversos critérios de seleção, tais como minimização de custo (tempos de espera,...), maximização de velocidade (vazão,...), controle de acesso, informação de gerenciamento (sobre a disponibilidade, confiabilidade ou desempenho de serviços), etc.

## PONTO DE VISTA COMPUTACIONAL

Um Trader providencia um conjunto de interfaces de gerenciamento e um outro de trading. Na Tabela 3 apresenta-se o atual estágio de definição das principais operações oferecidas nessa interfaces.

Exportadores usam as operações EXPORT/WITHDRAW/MODIFY. Por exemplo a operação EXPORT, executada por um exportador, anuncia ao trader as características de serviços. Essa operação pode falhar, por exemplo se o nome de contexto não está registrado, se a oferta já foi registrada antes ou se a oferta não satisfaz as restrições de tipos de serviços. A execução com sucesso da operação resultará na disponibilidade do serviço exportado para futuras importações. O trader armazenará, de forma persistente, as características do exportador, fornecendo acesso posterior à informação. A operação WITHDRAW tem o objetivo contrário e a operação MODIFY permite ao exportador mudar valores de propriedades de serviços.

Importadores podem usar as operações LIST/SEARCH/SELECT. A operação SEARCH procura exportações que satisfaçam requisitos pré-definidos. A sua execução com sucesso resultará numa lista de serviços exportados equiparáveis aos critérios do importador. A operação LIST permite a consulta dos valores das propriedades de

TABELA 3  
OPERAÇÕES DO TRADER ODP

OPERAÇÕES	PARÂMETROS	RESULTADOS E OBSERVAÇÕES
<b>OPERAÇÕES DE EXPORTADOR</b>		
EXPORT	NOME DE ENTIDADE REQUISITANTE, NOME DO TIPO DO SERVIÇO, NOME DE CONTEXTO, VALORES DAS PROPRIEDADES DE SERVIÇOS	RETORNA O IDENTIFICADOR DE EXPORT (OU INDICAÇÃO DE ERRO). TRADER ARMAZENA CARACTERÍSTICAS DE EXPORTADOR
WITHDRAW	NOME DA ENT. REQUISITANTE, DE CONTEXTO E IDENTIFICADOR DE EXPORT	RETORNA OK OU INDICAÇÃO DE ERRO. A EXECUÇÃO COM SUCESSO RETIRA O SERVIÇO NOMEADO DO CONTEXTO
MODIFY	IDEM WITHDRAW, LISTA DE NOMES DE PROPRIEDADES MODIFICADAS DE SERVIÇOS E NOVOS VALORES, LISTA DE SERVIÇOS ADICIONAIS E VALORES, LISTA DE SERVIÇOS A REMOVER	RETORNA OK OU INDICAÇÃO DE ERRO. COMO RESULTADO OBTEM-SE NOVO CONJUNTO DE VALORES DE PROPRIEDADES DE SERVIÇOS
<b>OPERAÇÕES DO IMPORTADOR</b>		
LIST	NOME DE ENTIDADE REQUISITANTE, DE CONTEXTO, IDENTIFICADOR DE EXPORTADOR, LISTA DE PROPRIEDADES DE SERVIÇO CONSULTADA	NO CASO DE SUCESSO A OPERAÇÃO RETORNA OS VALORES DAS PROPRIEDADES DE SERVIÇOS DO EXPORT NOMEADO ( PODERÃO EXISTIR DIVERSAS OFERTAS DE SERVIÇOS PARA ESSE EXPORT)
SEARCH	NOME DE ENTIDADE REQUISITANTE, DE CONTEXTO, DO TIPO DE SERVIÇO, RESTRIÇÃO DE COMPARAÇÃO, LISTA DE PROPRIEDADES DE SERVIÇOS A RETORNAR COM MATCH, LIMITAÇÕES DE BUSCA	RESULTANA LISTA DE SERVIÇOS EXPORTADOS E SUAS PROPRIEDADES, NO CONTEXTO ESPECIFICADO . CADA OFERTA DE SERVIÇOS DA LISTA TEM ESPECIFICAÇÃO DE TIPO COMPATÍVEL E OS SEUS VALORES SATISFAZEM AS RESTRIÇÕES DE "MATCHING"
SELECT	NOMES DE E. REQ. DE CONTEXTO E DE TIPO DE SERVIÇO, RESTRIÇÃO DE COMPARAÇÃO, CRITÉRIOS DE SELEÇÃO, LISTAS DE PROPRIEDADES E LIMITAÇÕES DE BUSCA	RETORNA OFERTA DE SERVIÇO EXPORTADO QUE SATISFAZ AS RESTRIÇÕES DE "MATCHING" SENDO A MAIS ADEQUADA EM RELAÇÃO AOS CRITÉRIOS DE SELEÇÃO
<b>OPERAÇÕES DA ESTRUTURA DE CONTEXTO</b>		
ADD	NOME DE ENTIDADE REQ. E DE CONTEXTO, NOMES DE SUPERCONTEXTOS, CONJUNTO DE NOMES DE TIPOS DE SERVIÇOS PERMITIDOS	ADICIONA O NOME DO CONTEXTO À ESTRUTURA DE CONTEXTOS. O CONTEXTO PASSARÁ A SER USADO POR OPERAÇÕES DE IMPORT/EXPORT
DELETE	N. DE E. REQ. E DE CONTEXTO	REMOVE O NOME DE CONTEXTO
LIST	N. DE E. REQ. E DE CONTEXTO, SUB-CONTEXTOS A REPORTAR, INFORMAÇÃO DESEJADA DE CONTEÚDO DE CONTEXTO	LISTA AS OFERTAS DE SERVIÇOS E SUB-CONTEXTOS CONTIDOS NO CONTEXTO NOMEADO
<b>OPERAÇÕES EM PROPRIEDADES DE SERVIÇOS DINÂMICOS</b>		
STATUS INQUIRY	IDENTIFICAÇÃO DE EXPORTADOR, IDENTIFICAÇÃO DE RECURSOS	REQUISITA INFORMAÇÃO SOBRE O ATUAL ESTADO DE UMA OFERTA DE SERVIÇO DE EXPORTAÇÃO: EXECUTADO PELO TRADER PARA OBTER INFORMAÇÃO DINÂMICA
STATUS UPDATE	A DEFINIR	EXECUTADA PELO EXPORTADOR PARA ATUALIZAÇÃO NO TRADER DA INFORMAÇÃO SOBRE CERTA OFERTA
<b>OPERAÇÕES DE FEDERAÇÃO (A DEFINIR)</b>		
A DEFINIR	EM ABERTO	EM ABERTO

serviço de uma certa exportação. A operação SELECT escolhe a oferta de serviços mais adequada às restrições "matching" do importador.

**PONTOS DE VISTA DE ENGENHARIA E DE TECNOLOGIA**

A informação atualmente existente respeito desses pontos de vista ainda é muito incipiente. Por exemplo, discutem-se na especificação de engenharia as possíveis res-

trições da estrutura de implementação do trader. No ponto de vista tecnológico deverão ser feitas escolhas sobre a representação e troca de dados ( por ex. ASN.1 BER).

#### 4. Experimentos e projetos

Nesta seção serão comentados alguns projetos na área de Processamento Distribuído Aberto: DCE ("Distributed Computing Environment") da OSF ("Open Software Foundation"), para demonstrar os esforços no ambiente comercial [19]; PEMMON ("Performance Management and Monitoring of Open Networks in Heterogeneous Context"), para demonstrar os esforços em projetos grandes de pesquisa, tais como ESPRIT e RACE [9, 12]; BERCIM do Projeto BERKOM ("Berliner Kommunikationssystem"), para descrever um cenário de aplicações distribuídas de grande porte, que é construído e operado no topo de uma plataforma, que utiliza os conceitos de Arquitetura de Serviços Abertos [10, 11 e 13].

##### DCE: DISTRIBUTED COMPUTING ENVIRONMENT

O DCE é um conjunto de serviços integrados, que suporta o desenvolvimento, o uso e a manutenção de aplicações distribuídas. O DCE é independente de sistema operacional e de rede, providenciando compatibilidade com ambientes de usuários já existentes. A arquitetura DCE é baseada em camadas, onde os níveis mais baixos oferecem os serviços básicos (sistema operacional) enquanto que os mais altos são consumidores de serviços (aplicações). Segurança e gerenciamento são essenciais a todos os níveis. A Figura 7 apresenta a arquitetura DCE.

O DCE oferece um conjunto de serviços que podem ser utilizados de maneira separada ou combinada, formando um ambiente de computação distribuída. Estes serviços são

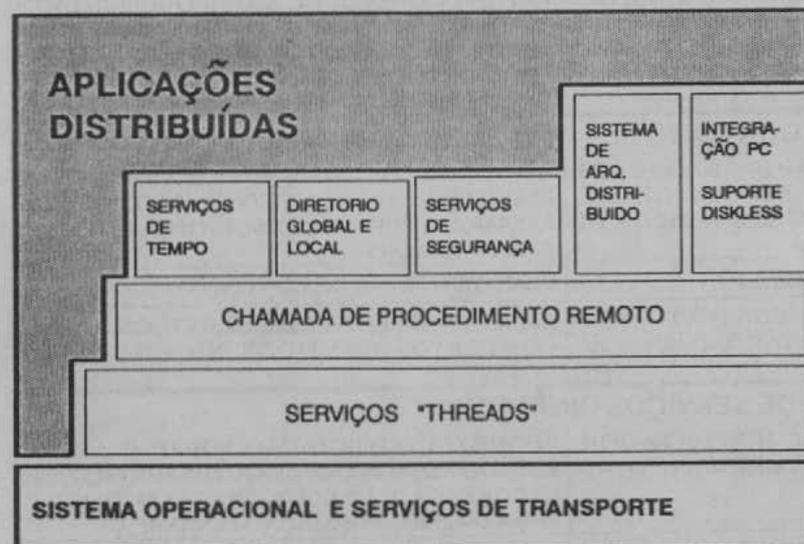


FIGURA 7  
Estrutura da plataforma DCE/OSF

organizados em duas categorias:

- **Serviços Distribuídos Fundamentais:** providenciam as ferramentas para o desenvolvimento de software, permitindo a criação de serviços para os usuários finais. As ferramentas incluem:

- **Serviço de Threads:** provê uma implementação portátil de processos "light-weight", chamados threads (compartilham um único espaço de endereçamento e permitem processamento concorrente);

- **Serviço de Temporização Distribuída:** permite a sincronização dos relógios do sistema pela troca periódica do tempo do sistema;
- **Chamada de Procedimento Remoto:** suporta comunicação entre entidades de aplicação distribuídas de acordo com o paradigma cliente/servidor;
- **Serviço de Diretório:** administra os nomes lógicos e mapeia estes nomes em endereços de rede para permitir acesso remoto;

- **Serviço de Segurança:** permite a autenticação, a autorização e a criptografia;
- **Serviços de Compartilhamento de Dados:** providenciam aos usuários finais capacidades construídas sobre os Serviços Distribuídos Fundamentais. Estes serviços não requerem programação por parte do usuário final e são os seguintes:
  - **Serviço de Diretório Global:** incrementa o Serviço de Diretório, providenciando a administração do espaço de nomes globais para sistemas de grande porte. Este serviço segue o padrão X-500. Um domínio Internet de Serviço de Nomes pode ser usado como alternativa;
  - **Sistema de Arquivos Distribuídos:** para a administração de arquivos em rede: implementado por servidores múltiplos; acesso transparente à localização para arquivos distribuídos dentro de um domínio administrativo; interoperabilidade com o NFS da SUN;
  - **Suporte Diskless:** para integração de estações de trabalho sem disco;
  - **Integração PC:** permite a PCs do DOS, Unix, OS/2 o acesso a servidores DCE (serviços de arquivos distribuídos e servidores de impressão dedicados).

O DCE apresenta uma vasta funcionalidade e um bom suporte à integração de componentes. O desenvolvimento de aplicações distribuídas é significativamente facilitada. Novos serviços e padrões poderão ser integrados no futuro. A desvantagem do DCE é (ainda) não apresentar suporte a orientação a objetos (C++), a transações distribuídas, a replicação específica da aplicação e a padrões ISO/OSI (exceto X-500).

## PEMMON [9]

PEMMON pertence à área de gerenciamento de redes heterogêneas, especialmente com relação ao gerenciamento de performance. A plataforma do PEMMON também é baseada em camadas (de baixo para cima):

- O Ambiente Hospedeiro: contém o sistema operacional e as Bases de Dados;
- O Ambiente de Suporte da Plataforma: é composto de:
  - Núcleo da Plataforma de Computação: provê um conjunto básico de funções de comunicação, armazenamento e processamento;
  - Suporte ao Processamento Distribuído: provê uma máquina virtual para as entidades de aplicação. Essencialmente as funções são um conjunto de mecanismos transparentes, tais como Trader, Suporte à Transação e Binder, que podem ser usados por funções de alto nível;
  - Interface da Plataforma de Computação: representa um ponto de acesso às entidades inferiores de suporte e a um número de facilidades usadas off-line;
- Ambiente de Suporte TMN ("Telecommunications Management Network"): contém os serviços específicos para aplicações distribuídas orientadas a objetos, incluindo gerenciamento de objetos e diretório;
- Funções Genéricas de Usuário: são funções de gerenciamento que podem ser compartilhadas por diversas aplicações de gerenciamento. Exemplos de funções são o configurador de rede, o gerente de relatório de erro e as funções genéricas ISO.

PEMMON é um dos exemplos de aplicações distribuídas, operando sobre uma plataforma de serviços que suporta comunicação, integração de dados e cooperação através de redes heterogêneas.

Outras áreas de aplicação dos conceitos de ODP são redes inteligentes, medicina, sistemas de controle descentralizados autônomos e CIM.

## **BERCIM [10,11,13]**

BERCIM é um projeto cooperativo que combina CIM ("Computer Integrated Manufacturing"), tolerância a falhas e comunicação broadband. É executado por três institutos de pesquisa berlinenses (IPK, DB e GMD FOKUS). Entre os objetivos do projeto está o de criar conceitos para uma Arquitetura de Serviços Abertos e uma Plataforma de Serviços.

Os cenários construídos no projeto demonstraram que estruturas CIM podem ser um exemplo complexo de aplicações cooperativas. Os seguintes requisitos têm sido satisfeitos pela rede de alta velocidade BERKOM e pela plataforma de serviço BERCIM: integração das ilhas de automação existentes, inclusão de estruturas hierárquicas e qualidades de serviços diferentes, abertura para a substituição, a evolução e a extensão, trabalho entre diferentes domínios organizacionais, uso de serviços via rede, suporte a soluções de problemas orientados a grupo e aplicações do tipo conferência e precursor para comunicação e aplicações multimídia.

BERCIM desenvolveu uma Arquitetura de Serviços Abertos e uma infraestrutura de componentes para suporte à cooperação e integração das aplicações. Esta infraestrutura é baseada em Trading, Informação e Administração.

Os Serviços de Administração permitem administração descentralizada de tipos de serviço, domínios e ofertas de serviço. Entidades de Administração podem ser autônomas e representam pessoas que possuem autoridade e dever para administrar componentes dedicados dentro do ambiente local. Os Serviços de Informação permitem a manipulação descentralizada de tipos de serviço, domínios e ofertas de serviço. Entidades de Informação manipulam informação estática e dinâmica e, de acordo com o tipo, os serviços são implementados por meio do serviço de diretório X-500 ou do serviço de suporte a gerenciamento X-700.

BERCIM é um protótipo de ODP e integra vários componentes heterogêneos, incluindo linguagens de programação (C, C++, EIFFEL), sistemas operacionais (Unix, VMS), estações de trabalho (Sun, VAX), redes (MAP, TOP, BERKOM) e usuários (programadores, cientistas da computação, engenheiros CIM). Acima da plataforma um certo grau alto de transparência pode ser obtido.

## **5. Conclusões**

O novo enfoque para a manipulação de informação, enfatizado pelo estabelecimento do conceito de processamento distribuído aberto (ODP), caracteriza o surgimento de uma nova e grande área de pesquisa, ainda não devidamente coberta pelo campo já mais tradicional dos sistemas distribuídos.

Mesmo a nível de padronização já foram alcançados alguns resultados normativos, em particular com o surgimento do RM-ODP/ISO. Nota-se contudo, pelo exposto neste trabalho, que muitos tópicos importantes do modelo e de subseqüentes padrões ainda estão por ser discutidos e conceituados.

A discussão destes tópicos deverá ser feita a partir tanto do estabelecimento de conceitos inovadores, em pesquisa acadêmica, como da realização de projetos pioneiros e da posterior migração para ambiente ODP de aplicações já existentes.

Procura-se, assim, identificar a grande necessidade do desenvolvimento de trabalhos nesta área, desde a especificação formal e modelagem, até à implementação de protótipos e sistemas-piloto.

## **6. Referências**

- [1] Tschammer, V., "CODE, Cooperating Open Systems, Open Distributed Processing, Distributed Computing Environment, Experiments and Projects", Curso intensivo da UNICAMP-FEE-DCA, Novembro de 1992
- [2] Le Lann, G., "Distributed Systems- towards a formal approach", Proc. IFIP Congress, Agosto 1977, Toronto, Canadá
- [3] ISO/IEC JTC 1/SC21/N7053, Basic Reference Model ODP - Part 1: Overview and Guide to Use
- [4] ISO/IEC JTC 1/ SC 21 / N6767, Basic Reference Model ODP - Part 2: Descriptive Model ( Fevereiro de 1992)
- [5] ISO/IEC JTC1/SC 21/ N7055, Basic Reference Model ODP - Part 3: Prescriptive Model (Junho de 1992)
- [6] ISO/IEC JTC 1/SC21/ N7056 ( Recommendation X.905), Basic Reference Model ODP - Part 5: Architectural Semantics
- [7] ISO/IEC JTC 1/SC 21 /N7074, Working Document on Topic 9.1 - ODP Trader (Junho de 1992)
- [8] ISO/IEC JTC 1 / SC 21 / N7075, WG7 Project Management Document: ODP list of open and resolved issues ( Junho de 1992)
- [9] PEMMON , Performance and Monitoring of Open Networks in Heterogeneous Contexts (PEMMON), ESPRIT Projeto No 5371, (1991),
- [10] Popescu-Zeletin, R.; Tschammer, V. e Tschichholz M. , "Y" Distributed Application Platform - Computer Communications", vol 14, 6 (Julho/Agosto1992)
- [11] Popescu-Zeletin, R.; Tschammer, V. e Tschichholz M. (1992b), "A Service Platform for Distributed Applications" Proc. 3rd IEEE Workshop on the Future Trends of Distributed Computing , Taipei, (Abril 1990)
- [12] ROSA Methodology, Internal Deliverable, RACE Ref. 1093, RACE ROSA Project (Março de 1991))
- [13] Tschammer, V. , "Domain-based Support for Service Administration and Server Selection" Proc. EurOpen Spring'91 Conference - Tronso, Noruega (Maio1991)
- [14] CCITT Z100, Blue Book, Vol. X, Fasc. X.1, ITU, 1989.
- [15] ISO IS 8807, "LOTOS: A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", 1988.
- [16] ISO IS 9074, "Estelle: A Formal Description Technique Based on an Extended State Transition Model", 1988.
- [17] J. M. Spivey, "The Z Notation: A reference Manual", Prentice Hall, 1992.
- [18] C. A. Vissers, "Trends and Proliferation of Formal Description Techniques for OSI Standards", seminário apresentado no 5o Simpósio Brasileiro de Redes de Computadores, realizado em São Paulo (SP), 1987.
- [19] OSF , Distributed Computing Environment, (Setembro 1990)