FABIANO BALDO

CONFIGURAÇÃO SEMI-AUTOMÁTICA DE EMPRESAS VIRTUAIS – UMA ABORDAGEM MULTIAGENTE

FLORIANÓPOLIS
2003

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO
EM ENGENHARIA ELÉTRICA

CONFIGURAÇÃO SEMI-AUTOMÁTICA DE EMPRESAS VIRTUAIS – UMA ABORDAGEM MULTIAGENTE

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica.

FABIANO BALDO

Florianópolis, Outubro de 2003.

CONFIGURAÇÃO SEMI-AUTOMÁTICA DE EMPRESAS VIRTUAIS – UMA ABORDAGEM MULTIAGENTE

Fabiano Baldo

'Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em Controle, Automação e Informática Industrial, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.'

_____
Prof. Ricardo José Rabelo, Dr.
Orientador

_____
Prof. Edson Roberto de Pieri, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

_____
Prof. Ricardo José Rabelo, Dr.
Presidente

_____
Alexandra Augusta Pereira Klen, Dra. Eng.

_____
Prof. Carlos Barros Montez, Dr. Eng.

_____
Prof. Rômulo Silva de Oliveira, Dr. Eng.

Aos meus pais Sebastião Baldo

e Ercília Olímpia Carlesso Baldo

Agradecimentos

Antes de agradecer a todos que direta ou indiretamente contribuíram na realização deste trabalho, quero expressar minha satisfação em ter realizado mais um sonho. Apesar de todas as dificuldades encontradas no caminho, posso dizer que valeu a pena todo o sacrifício pelo qual passei para ao final ser recompensado com a vitória.

Primeiramente quero agradecer a Deus por ter me dado forças nos momentos de maior dificuldade, e assim conseguir superar noites e finais de semana no laboratório longe da família, namorada e amigos.

A meus pais, por acreditarem em meu potencial e sempre me estimularem a buscar meus objetivos me estendendo a mão por menores que fossem as dificuldades... mãe, pai eu amo vocês! A Minha irmã Marisa Baldo, por quem tenho um carinho especial, pois nunca demonstrou inveja ou rancor por ter que dividir comigo o amor de nossos pais.

A minha namorada Ana Paula Costa, pela compreensão nos momentos em que não pudemos estar juntos, pelo seu carinho e atenção quando estou ao seu lado, que me faziam relaxar e esquecer por alguns momentos as pressões do trabalho.

Agradeço também meu orientador e amigo professor Ricardo José Rabelo, por aceitar o convite em ser meu orientador e assim fazer parte como peça fundamental da conquista de mais um de meus objetivos.

A todos do GSIGMA, principalmente a Alexandra Pereira Klen e Edmilson Rampazzo Klen, por me proporcionarem a chance de trabalhar com o grupo. Em especial a meus amigos e companheiros de todos os dias Leandro Loss, Rui Jorge Tramontin Júnior e Carlos Eduardo Gesser pelas dicas e auxílios que contribuíram na realização do trabalho, e por todos os momentos que me aturaram mal humorado e estressado.

Aos professores e funcionários do Programa de Pós-Graduação em Engenharia Elétrica, pelo pronto atendimento e ótimos serviços prestados.

Finalmente agradeço a meus amigos de infância Eduardo, Rafael, Germano, Patrick, Alvino e Vinícius que mesmo distantes sempre me passaram pensamentos positivos e palavras de otimismo.

Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

CONFIGURAÇÃO SEMI-AUTOMÁTICA DE EMPRESAS VIRTUAIS – UMA ABORDAGEM MULTIAGENTE

Fabiano Baldo

Outubro, 2003

Orientador: Prof. Ricardo José Rabelo, Dr.
Área de Concentração: Controle, Automação e Informática Industrial.
Palavras-chave: Empresas Virtuais, Configuração de Empresas Virtuais, Sistemas Multiagente, XML, *Workflow* e CORBA.
Número de Páginas: 168.

Tendo em vista a dinâmica das relações comerciais realizadas entre as empresas atualmente, tecnologias de informação que supram suas necessidades e dêem suporte a esta nova tendência de negócios são necessárias. Como uma das conseqüências desta nova tendência de realização de negócios surge o conceito de empresa virtual (EV). Uma EV pode ser considerada, em termos gerais, como sendo a união temporária de várias empresas para atender a uma oportunidade de negócios e que, para tal, partilham recursos e habilidades pela rede de computadores.

Este conceito é relativamente novo, existindo vários pontos importantes associados a ele que ainda estão em aberto ou pouco desenvolvidos. Um destes pontos está relacionado ao processo de configuração da EV. Este processo consiste em definir e ajustar os parâmetros necessários para o correto funcionamento / operação de uma EV, e assim contribuir para a realização do objetivo pelo qual a mesma foi criada. Atualmente, a configuração de EVs tem sido realizada de forma manual, freqüentemente incorrendo em erros, dificuldades de atualização ao longo do funcionamento da mesma, e uma conseqüente pouca agilidade da EV como um todo.

O foco principal deste trabalho é propor uma solução para o problema de configuração de EVs, aqui basicamente visto como um processo constituído por duas

etapas: configuração da topologia da EV, e configuração do acesso às informações dos parceiros a ela pertencentes. Na direção de se contribuir para a solução deste problema, uma proposta de configuração semi-automática de EVs foi concebida. A arquitetura do modelo concebido é composta por uma infraestrutura multiagente e um modelo de informação. Além do modelo, uma ferramenta baseada nesta arquitetura foi desenvolvida para validar a viabilidade da proposta. Esta ferramenta foi implementada sobre uma plataforma de suporte a gestão de EVs elaborada no âmbito de um projeto de cooperação internacional.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

A MULTIAGENT APPROACH FOR SEMI-AUTOMATIC CONFIGURATION OF VIRTUAL ENTERPRISE

Fabiano Baldo

October, 2003

Advisor: Prof. Ricardo José Rabelo, Dr.
Area of Concentration: Control, Automation and Industrial Informatics.
Keywords: Virtual Enterprises, Virtual Enterprises Configuration, Multiagent Systems, XML, Workflow and CORBA.
Number of Pages: 168.

The dynamic of the business and the commercial relations that take place nowadays among enterprises calls for the necessity of information technologies to cope with the current enterprises' needs. The concept of Virtual Enterprise (EV) emerges as a consequence of this new way of doing business. The VE can be defined as a temporary alliance of enterprises that come together in order to attend to a new business opportunity. To this end, the enterprises operate in a collaborative way sharing resources and skills by means of a computer network.

This concept is relatively new and therefore there are still many relevant issues that are till now either open questions or subject of current research. One of the open questions is related to the VE configuration process. This process consists of defining and adjusting the necessary parameters for an adequate VE operation, contributing for the accomplishment of the main envisaged goal. The VE configuration has – till now – been performed in a manual way, incurring frequently in errors as well as in the difficulty to keep the information updated along the VE operation, reflecting directly in the less agility of the EV.

The main focus of this work is to propose an alternative solution to the VE configuration problem, here basically divided into two stages: VE topology configuration and access configuration of the partners' information. The contribution of this work relies

on the proposal of a semi-automatic EV configuration. The architecture of the conceived model is composed by a multiagent infrastructure and by an information model. Besides conceiving the model, a tool was also developed based on the architecture in order to validate the proposal. This tool was implemented as part of an EV management platform within the scope of an international cooperation project.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1: Introdução

Na economia atual podemos observar duas importantes tendências. A primeira mostra que as empresas devem integrar seus processos de negócios dentro do processo de negócio global para serem hábeis a operar e sobreviver no mercado. Do ponto de vista funcional, integrações podem ser necessárias pois certos produtos ou serviços podem se tornar muito complexos para serem realizados por uma única empresa. A integração também pode ser necessária do ponto de vista da eficiência, pois a empresa concentra suas atividades na sua área de competência e contrata outras empresas para executar as atividades secundárias no intuito de prover produtos ou serviços mais baratos e de melhor qualidade. A segunda tendência indica que a maior integração entre as empresas exige que seus processos de negócios se transformem em processos mais dinâmicos e também mais eficientes para lidar com as mudanças freqüentes do mercado (HOFFNER *et al.*, 2001).

Para satisfazer estas duas novas tendências em processos de negócios é que o conceito de empresa virtual (EV) surgiu. Assim, quando uma determinada empresa percebe uma boa oportunidade de negócio e não tem capacidade produtiva, técnica ou temporal suficiente para executá-la autonomamente, a mesma pode lançar mão de uma alternativa para solucionar esse problema com a contratação e / ou parceria com outras empresas que possam vir a somar e com isso compartilhar o trabalho para que ao final do processo o objetivo comum seja alcançado.

Portanto, empresas virtuais são agrupamentos de empresas que se unem temporária e dinamicamente para atacar uma oportunidade de negócio iminente. Se esta tarefa for realizada de forma eficiente, menores serão os custos e maiores serão os benefícios para todos os envolvidos, tanto produtor quanto consumidor. Para se alcançar um alto grau de eficiência na realização desta tarefa, os mais recentes e avançados conceitos de tecnologia de informação e ferramentas computacionais devem ser utilizados.

Entretanto, existem vários pontos importantes associados ao conceito de empresas virtuais que ainda estão em aberto ou pouco desenvolvidos. Um destes pontos está relacionado à questão da configuração. O processo de configuração é de suma importância dentro do ciclo de vida da EV. É neste processo que são definidos e ajustados os parâmetros necessários para o correto funcionamento da EV. Além disto, este processo

também define os papéis dos membros participantes e seus relacionamentos – a chamada topologia da EV –, bem como quais informações de um determinado membro os outros membros da EV terão privilégio para acessar.

Atualmente a configuração de EVs tem sido realizada de forma manual, freqüentemente incorrendo em erros, dificuldades de atualização ao longo do funcionamento da mesma, e uma conseqüente pouca agilidade da EV como um todo.

Tendo este como o estado atual do processo de configuração de EVs, o foco principal deste trabalho é propor uma alternativa mais avançada para a solução deste problema. De um modo geral, o problema de configuração é aqui basicamente visto como um processo constituído por duas etapas: a configuração da topologia da EV, de maneira eficiente e garantindo os níveis de privilégio de visibilidade de cada membro, e a configuração dos direitos de acesso às informações, mantendo assim a privacidade das informações de cada membro. Contudo, deve-se levar em consideração que o intuito é criar uma solução que mantenha a EV sempre configurada e suas informações sempre atualizadas e confiáveis em ambientes dinâmicos. Estes ambientes dinâmicos são ambientes onde acontece grande variação de parceiros.

Na direção de se contribuir para a solução deste problema, esta dissertação tem como objetivo conceber uma proposta para a configuração semi-automática de empresas virtuais. A arquitetura do modelo concebido é composta por uma infraestrutura multiagente e um modelo de informação que servem de base para a construção de outras funcionalidades relacionadas à gestão de EVs. Cada empresa participante da EV terá um sistema multiagente instalado em seus domínios. Assim, o sistema multiagente instalado na empresa coordenadora terá a possibilidade de interagir com os sistemas multiagente das outras empresas participantes para, de forma semi-automática, configurar e manter atualizadas as informações sobre a topologia da EV, bem como, das restrições de acesso à informação dos parceiros, tentando desta forma garantir um melhor resultado dos sistemas de gestão que utilizam essas informações. Além deste modelo, uma ferramenta baseada nesta arquitetura foi implementada para validar a viabilidade da proposta.

Esta dissertação foi desenvolvida como parte do projeto IST (*Information Society Technologies*) de cooperação internacional chamado DAMASCOS (*Dynamic Forecast for Master Production Planning with Stock and Capacity Constraints*), número 11850, desenvolvido entre 2000 e 2002 em parceria com países da União Européia

(http://www.inescporto.pt/~damascos/). O consórcio incluiu parceiros da Alemanha, Portugal, Itália, Suécia e Brasil. O parceiro brasileiro do consórcio foi o Grupo de Sistemas Inteligentes de Manufatura (GSIGMA) da Universidade Federal de Santa Catarina (http://www.gsigma-grucon.ufsc.br/). O projeto DAMASCOS visou o desenvolvimento de uma plataforma aberta com a implementação e posterior disponibilização de adequados módulos e mecanismos baseados em tecnologia da informação, utilizados para gerenciar redes de fornecimento personalizadas usando a filosofia de Empresa Virtual, e ao mesmo tempo integrando o consumidor e suas demandas (FERREIRA *et al.*, 2000a). Seu foco foi voltado principalmente para pequenas e médias empresas do setor calçadista, de forma a fornecer um ambiente de gerenciamento e interoperação de empresas virtuais dinâmicas.

No âmbito do projeto DAMASCOS o laboratório GSIGMA foi o responsável pela concepção do módulo de software chamado SC^2 (*Supply Chain Smart Coordination*). O SC^2 foi o sistema multiagente de suporte à decisão desenvolvido com o intuito de gerenciar o fluxo de informações de uma EV. O gerenciamento é realizado por meio do monitoramento e supervisão das atividades executadas pela EV ao longo de seu funcionamento. Este monitoramento facilita a detecção e análise dos conflitos e auxilia na busca pela solução dos mesmos. A localização da melhor solução para os conflitos é suportada por um protocolo de apoio à decisão flexível e configurável que provê uma coordenação inteligente sobre toda a EV (RABELO *et al.*, 2002, KLEN *et al.*, 2002).

O texto desta dissertação foi organizado de forma a refletir as etapas percorridas durante as pesquisas realizadas no decorrer do projeto DAMASCOS. Esta dissertação está dividida em sete capítulos, sendo os capítulos dois e três relacionados à revisão bibliográfica, o capítulo quatro apresenta o modelo proposto para a resolução do problema de configuração de EVs, o capítulo cinco descreve o protótipo desenvolvido para validar o modelo, o capítulo seis apresenta uma simulação do sistema e a análise dos resultados obtidos e, por fim, o capítulo sete apresenta as conclusões que foram obtidas com esta dissertação.

O capítulo 2 introduz os conceitos mais importantes relacionados a EVs, explicando suas principais terminologias, seu ciclo de vida, seus diferentes tipos de classificação, além de apresentar uma série de trabalhos previamente realizados na área de configuração de EVs.

O capítulo 3 mostra as principais tecnologias de informação utilizadas com suporte ao desenvolvimento do protótipo construído para validar o modelo proposto. São relatadas, de forma resumida, as tecnologias de agentes, *workflow*, XML e CORBA.

No capítulo 4 é apresentado o modelo para a configuração semi-automática de EVs. Este modelo é composto por duas alternativas de configuração da topologia e uma para a configuração das restrições de acesso às informações.

A primeira alternativa de configuração da topologia utiliza o resultado de uma ferramenta de seleção de parceiros como entrada para o processo. A segunda alternativa realiza o processo através de métodos internos que fazem a busca pelos parceiros previamente escolhidos.

A parte do modelo proposto dedicado a restringir o acesso às informações produzidas pelos membros da EV, utiliza o conceito de "cláusulas de supervisão" como forma de especificar quais informações de um determinado parceiro estão visíveis para cada um dos outros membros de uma EV.

No capítulo 5 são descritos os principais aspectos relacionados ao protótipo implementado para validar o modelo. Nesta descrição são utilizados os conceitos e tecnologias apresentadas no capítulo 3. Estas tecnologias serviram de base para a especificação de modelo e posterior desenvolvimento do protótipo.

No capítulo 6 é apresentada uma simulação do funcionamento do protótipo no processo de configuração de uma dada EV. Por fim, são feitas avaliações dos resultados obtidos nesta simulação.

No capítulo 7 são apresentadas as principais conclusões e perspectivas de trabalhos futuros.

Ao final do documento são apresentados dois anexos: o primeiro contendo o protocolo de mensagens trocadas entre os sistemas multiagente; o segundo exemplificando alguns detalhes da implementação do protótipo e, por fim, um glossário de termos e siglas.

CAPÍTULO 2: Empresas Virtuais

Uma dificuldade encontrada nas pesquisas envolvendo empresas virtuais (EVs) é a falta de uma definição "padrão" usada por todos os pesquisadores em suas publicações acerca deste conceito. Entretanto, existem esforços para se chegar a uma definição comum, e para isso, vários pesquisadores têm proposto definições sobre o assunto. BYRNE (1993) afirma que "Corporação Virtual é uma rede temporária de companhias independentes – fornecedores, clientes, até mesmo rivais – unidos por tecnologia de informação (TI) para compartilhar habilidades, reduzir custos e acessar o mercado um do outro". Para WALTON et al. (1996) "A Empresa Virtual consiste de uma série de nós cooperantes de excelência em determinada função, que formam uma cadeia de fornecimento para suprir uma necessidade específica de mercado". Já para PARK et al. (1999), "Empresas Virtuais são criadas para atingir uma necessidade de mercado específica, sendo formadas por partes de duas ou mais empresas diferentes, e projetadas para facilitar a agregação de recursos rápida, ampla e concorrentemente".

A definição de empresas virtuais adotada como base para a escrita dessa dissertação foi formulada por CAMARINHA-MATOS et al. (1999a), e diz que "Uma Empresa Virtual é uma aliança temporária de empresas, que se unem para compartilhar habilidades e recursos para melhor responder a uma oportunidade de negócios. Essa cooperação é suportada por redes de computadores". Esta definição foi escolhida por apresentar uma das mais completas caracterizações do conceito em questão, além do fato de ser largamente utilizada e conhecida por pesquisadores no mundo tudo.

A próxima seção apresenta as terminologias mais utilizadas para caracterizar alianças de empresas nos seus mais diversos níveis de integração. As seções seguintes mostram conceitos relacionados especificamente a empresas virtuais, que é o foco principal desta dissertação. Deve-se levar em consideração que vários dos conceitos referentes a empresas virtuais podem, em muitos casos, ser empregados também a outros tipos de associações de empresas.

2.1 Terminologias

Atualmente, o crescimento da quantidade de organizações conectadas em rede e o surgimento de novos paradigmas de gerenciamento da produção têm levado à criação de vários termos relacionados a alianças de empresas. Dentre estes, os que merecem maior destaque são os seguintes: organização em rede (*networked organization*), organização virtual (*virtual organization*), cadeia de fornecimento (*supply chain*), empresa estendida (*extended enterprise*), agrupamento de empresas (*cluster of enterprises*) e ambiente de incubação (*breeding environment*). Alguns autores tratam esses termos como se não existissem diferenças, entretanto, essas diferenças serão representadas em detalhes a seguir e a Figura 1 (CAMARINHA-MATOS *et al*., 1999a) representa o relacionamento entre alguns destes diferentes tipos de organizações.

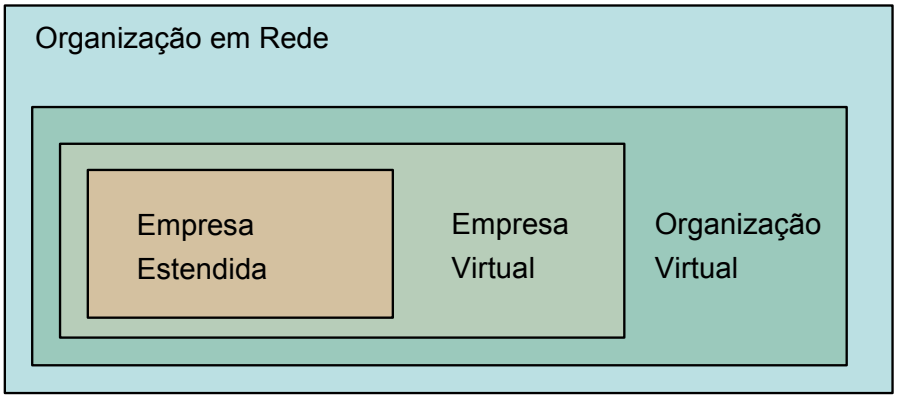


Figura 1 – Tipos de organizações.

Organização em rede é talvez o termo mais geral no que se refere a grupos de organizações ligados por rede de computadores, entretanto sem necessariamente compartilhar recursos, habilidades ou ter um objetivo comum (OUZOUNIS, 2001). Ou seja, são empresas preparadas e disponíveis para realizarem negócios via rede.

Uma organização virtual pode ser vista como um conceito que engloba o conceito de empresa virtual. Porém, uma organização virtual é uma coleção geograficamente distribuída de funcionalidades e entidades culturalmente diversas, conectadas através de uma infraestrutura de telecomunicações, e que estão comprometidas a alcançar um objetivo coletivo através do compartilhamento de suas competências e recursos (BULTJE *et al.*, 1998, FILOS *et al.*, 2000, LIMA, 2001).

Uma cadeia de fornecimento compreende um conjunto de fornecedores, fabricantes, distribuidores, clientes e etc., associado à transformação de matéria-prima em

mercadorias com valor agregado. Ela pode ser referenciada como sendo uma rede integrada que sincroniza uma série de processos de negócios para: adquirir matéria-prima ou partes; transformar esta matéria-prima em partes de produtos finais; adicionar valor a estes produtos; distribuir estes produtos a seus armazéns ou clientes; e facilitar a troca de informações entre as várias entidades do negócio (MIN *et al.*, 2002).

O termo gerenciamento de cadeias de fornecimento, refere-se às regras e mecanismos de suporte ao gerenciamento do fluxo de material e informação em uma cadeia de valor. Normalmente este gerenciamento cobre vários aspectos do produto , desde a matéria-prima do fornecedor ao consumidor, envolvendo os fabricantes, os transportadores, os distribuidores, os armazéns, etc., e suportando o fluxo de informação trocado entre os participantes (CAMARINHA-MATOS *et al.*, 1999a). Este conceito é tradicionalmente aplicado a organizações que são relativamente estáveis, por exemplo, onde a área de atuação permanece a mesma por um longo período de tempo.

O conceito de empresa estendida é aplicado a uma organização em que a empresa dominante estende seus limites a todos ou a alguns de seus fornecedores, e estes têm um grau limitado de autonomia. Uma empresa virtual pode ser vista como um conceito mais geral incluindo outros tipos de organizações, isto é, tem uma estrutura mais democrática em que a coordenação é feita par a par. Neste caso, uma empresa estendida pode ser vista como um caso particular de empresa virtual (CAMARINHA-MATOS *et al.*, 1999a, LIMA, 2001).

O termo agrupamento de empresas representa um conjunto de empresas que têm potencial e estão dispostas a cooperar para o estabelecimento de uma aliança de longa duração. Em muitos casos os agrupamentos de empresas são formados em torno de uma tecnologia específica ou tipo de produto. As situações mais freqüentes são os casos em que os agrupamentos são formados por empresas localizadas em uma região comum.

Por último, o conceito de ambiente de incubação expressa a idéia de um ambiente onde organizações possam desenvolver-se, adquirir conhecimento e potencialmente cooperar (OLLUS *et al.*, 2003). O objetivo é desenvolver um protocolo de preparação para a cooperação entre estas organizações para o caso de uma tarefa específica ou para atender a demanda dos clientes (KÜRÜMLÜOGLU *et al.*, 2003).

2.2 Classificação de Empresas Virtuais

Levando em consideração que o foco deste trabalho está relacionado ao conceito de empresas virtuais, a primeira seção deste capítulo teve apenas o objetivo de dar uma visão geral da localização deste conceito dentro do contexto das alianças entre empresas. Portanto, esta e as próximas seções deste capítulo têm o intuito de apresentar conceitos e classificações especificamente relacionadas a empresas virtuais. Embora em alguns casos os conceitos aqui apresentados possam ser utilizados em outros tipos de alianças de empresas, este tipo de analogia foge do escopo desta dissertação.

Para dar início ao detalhamento dos conceitos relacionados a EVs, deve-se destacar que as mesmas podem ser classificadas por diversas características que são usadas como base para distingui-las umas das outras. Estas classificações facilitam a escolha das características necessárias a uma EV para um tipo de negócio específico. Neste estudo será adotada a classificação que leva em consideração as seguintes características (CAMARINHA-MATOS *et al.*, 1999a, OUZOUNIS, 2001):

- **Duração**: Algumas alianças entre empresas são estabelecidas para uma única oportunidade de negócios e são dissolvidas ao final do processo. Esta situação corresponde ao tipo considerado "padrão" de empresas virtuais. Entretanto, há também alianças de longa duração que são estabelecidas para executar um número indeterminado de processos de negócios, ou para executar um processo de negócio longo.
- **Flexibilidade**: Levando-se em consideração a flexibilidade, existem dois tipos de empresas virtuais. Quando uma empresa pode dinamicamente entrar ou sair da aliança enquanto o processo de negócio está sendo executado e sem afetar seu funcionamento, é chamada de "variável" ou "dinâmica". Entretanto, existe outro tipo de empresa virtual no qual a variação de parceiros é muito pequena ou inexistente, onde se dá o nome de "fixa" ou "estática". Este assunto será abordado novamente com maiores detalhes na próxima seção (seção 2.3), pois o mesmo é de grande importância no escopo deste trabalho.
- **Participação**: Uma outra característica importante a ser considerada é a possibilidade de uma empresa participar simultaneamente em vários negócios,

sendo chamada “aliança múltipla”, ou se dedicar exclusivamente a uma aliança, denominada “aliança única”.

- **Autonomia**: Esta característica determina se os parceiros da empresa virtual são fortemente acoplados ao processo de negócio da EV, ou podem mudar um de seus processos internos e ainda assim manter-se dentro do processo de negócio da aliança.

- **Evolução e Escalabilidade**: A evolução é a característica que identifica se uma determinada empresa virtual pode evoluir em termos de especialização e / ou refinamento do produto, nem que para isso seja necessária a contratação de novos parceiros ou estabelecimento de novos relacionamentos. Já a escalabilidade identifica se o modelo de negócios é escalar, ou seja, se ele pode ser expandido em número de parceiros ou processos sem que com isso seu rendimento diminua.

- **Coordenação**: Em relação à coordenação vários modelos podem ser considerados. Em alguns tipos de EVs há uma companhia dominante para uma rede relativamente fixa de fornecedores. Essa companhia define as regras e propõe seu padrão, tanto em termos do processo de negócios, como nas informações trocadas entre os parceiros. A este tipo atribui-se o nome de “coordenação estruturada em estrela”. Em outros casos, existem companhias que participam de diferentes empresas virtuais sem existir uma companhia dominante, formando assim uma “aliança democrática”. Neste tipo de rede, todos os nós cooperam em igualdade, prevalecendo sua autonomia e agregando suas competências. Por último, existe um tipo de aliança que é formada por parceiros que utilizam os benefícios mútuos da junção do gerenciamento dos recursos e habilidades, e assim podem criar um novo tipo de coordenação conjunta chamada “coordenação federada”.

- **Visibilidade**: Esta característica está relacionada ao escopo de visualização de um nó da rede. Em muitos casos, um determinado nó só enxerga seus vizinhos diretos (fornecedores, armazéns, clientes). Isto é chamado de “visibilidade de nível único” (*one tier*). Em outras situações, um nó pode ter o direito de visualizar outras empresas que não são seus vizinhos diretos. A este tipo de visibilidade é chamada de “visibilidade multi-nível” (*multi-tiers*).

- **Topologia**: A topologia consiste em classificar as empresas virtuais levando em consideração o arranjo das ligações entre as empresas envolvidas. Estas ligações podem representar produtos, informações ou fluxo de trabalho. Quando as ligações são feitas em camadas, onde cada parceiro se relaciona apenas com seus vizinhos superiores e inferiores, a topologia é chamada "topologia cadeia de fornecimento" (*supply chain topology*). Entretanto, se as empresas estão arranjadas na forma de estrela, onde existe um parceiro central e todas as outras empresas estão ligadas a este, dá-se o nome de "topologia estrela" (*star topology*). Finalmente, se as empresas estão ligadas com quaisquer outros parceiros sem uma hierarquia, a mesma é chamada "topologia par-a-par" (*peer-to-peer topology*) (OLLUS *et al.*, 2003). A Figura 2 (OLLUS *et al.*, 2003) apresenta estes três tipos de topologias de empresas virtuais.

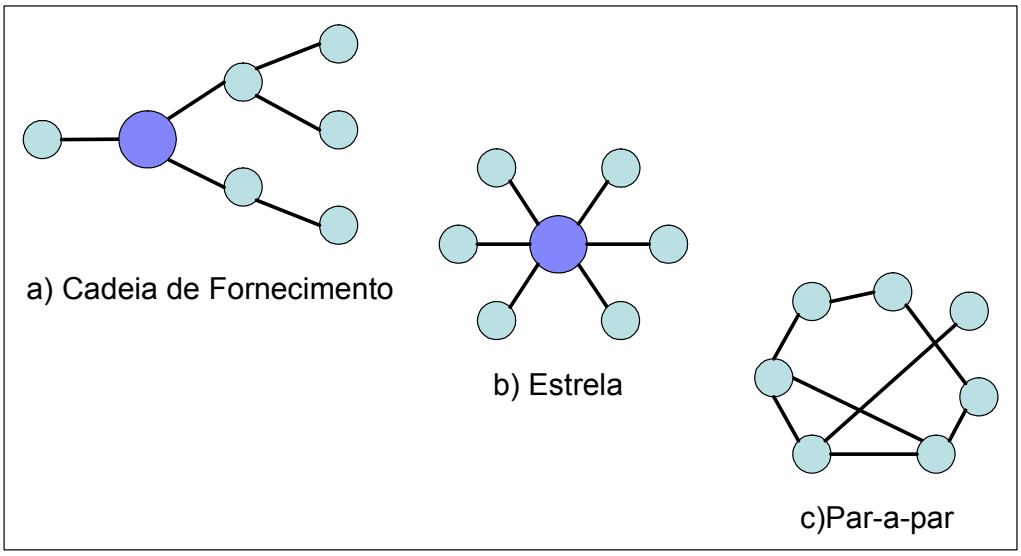


Figura 2 – Topologias de Empresas Virtuais.

2.3 Categorias de Empresas Virtuais

Além dos critérios de classificação apresentados na seção 2.2, empresas virtuais podem ser caracterizadas através de duas categorias bem definidas (OUZOUNIS, 2001). São elas:

- Empresas Virtuais Estáticas;
- Empresas Virtuais Dinâmicas.

Este tipo de classificação merece um enfoque especial dada sua importância para o trabalho em questão. Portanto, uma seção foi criada para abordar especificamente este assunto.

Nas próximas subseções as duas categorias de empresas virtuais serão descritas em maiores detalhes para esclarecer suas diferenças, vantagens e desvantagens.

2.3.1 Empresas Virtuais Estáticas

As empresas virtuais estáticas têm o conjunto de parceiros definido antes de sua criação e não muda durante toda a sua execução. Desta forma, os processos de negócios compartilhados podem se tornar fortemente integrados e os relacionamentos entre os parceiros podem conter procedimentos bem definidos e personalizados. Baseado no estilo de distribuição e gerenciamento da rede, dois tipos de EV estáticas podem ser identificadas: centralizada e descentralizada (OUZOUNIS, 1999).

Nas empresas virtuais estáticas centralizadas, uma empresa chamada coordenador gerencia os relacionamentos de negócios entre os membros da rede, integra os processos dos parceiros criando processos compartilhados, gerencia a infraestrutura técnica subjacente e os processos de negócios compartilhados dos parceiros de um modo estático e centralizado (OUZOUNIS, 2001). Os parceiros formam um relacionamento de longa duração focado no retorno do investimento sobre o tempo de vida do relacionamento. Assim, o estabelecimento da empresa virtual é feito manualmente, de uma forma personalizada e sobre o controle total da empresa dominante. As necessidades de integrações, desenvolvimentos, e os custos de reengenharia são altos para todos os membros (WOGNUM, 1999).

Nas empresas virtuais estáticas descentralizadas, diferentes parceiros são ligados conjuntamente de uma forma autônoma e descentralizada. Este tipo de rede é similar a anterior com a exceção de que não há uma empresa central que execute o gerenciamento, e assim cada membro da rede pode cooperar com várias outras alianças (OUZOUNIS, 2001). Nenhum dos parceiros têm controle total sobre a rede e a infraestrutura subjacente. Entretanto, a integração entre os processos de negócios dos membros é executada de um modo conjunto, coordenado e incremental. O estabelecimento da empresa virtual é executado manualmente e de um modo personalizado, focando os requisitos técnicos

específicos dos parceiros. Os custos de desenvolvimento e integração são altos enquanto a evolução da rede é praticamente inexistente (ZARLI, 1999).

Uma abordagem mais recente para automatizar o processo de formação de uma empresa virtual estática é através da utilização de mercados virtuais ou diretórios de serviços. Estes últimos são locais onde membros em potencial de empresas virtuais mantêm registrados seus recursos e habilidades. Os mercados virtuais provêem serviços de casamento para domínios de negócios que desejam localizar parceiros (CAMARINHA-MATOS *et al.*, 1999b, OUZOUNIS, 2001). Os negociadores, conhecidos com *brokers*, procuram no mercado e localizam os parceiros em potencial que podem prover processos específicos. Então, manualmente, um negociador humano inicia o processo de negociação para selecionar o candidato mais apropriado para a empresa virtual. Com essa abordagem, o tempo necessário para localizar parceiros e estabelecer os relacionamentos é melhorado. Os processos de seleção de parceiros e *brokerage* serão abordados posteriormente.

2.3.2 Empresas Virtuais Dinâmicas

Em empresas virtuais dinâmicas, o conjunto de parceiros é ligado dinamicamente, sob demanda, e de acordo com as necessidades do cliente. Empresas virtuais dinâmicas não apresentam, na maioria dos casos, relacionamentos fixos entre as empresas, podendo mudar / evoluir continuamente para atender as novas exigências do mercado (DOZ *et al.*, 1998). Baseado no estilo de distribuição e gerenciamento da rede, dois tipos de empresas virtuais dinâmicas podem ser identificadas (ALONSO *et al.*, 1999, OUZOUNIS, 1999):

Centralizada, acontece quando o coordenador é um parceiro da EV. Este gerencia e administra a EV e impõe seus modelos específicos de processos, sendo também o responsável por escolher os parceiros bem como substitui-los, se necessário. Este é o tipo mais comum de empresas virtuais dinâmicas encontradas atualmente.

Descentralizado, acontece quando o coordenador é um provedor de serviços e não exerce poder de gerenciamento sob seus parceiros. O controle das trocas e subcontratações de parceiros acontece através de um conselho onde todos decidem de forma democrática que caminho tomar. Este é provavelmente o modelo mais avançado e flexível e contém as características de maior benefício.

2.4 Estágios do Ciclo de Vida

Uma Empresa virtual é um relacionamento temporário entre dois ou mais participantes, que é formado, operado e dissolvido para realizar um objetivo específico em um curto espaço de tempo (REID *et al.* 1996). Desta forma, pode-se perceber que no processo que compreende desde o início até o fim de uma empresa virtual, ela passa por vários estágios que são agrupados na seqüência em que acontecem e formam assim as fases de seu ciclo de vida. Existem várias formas de se definir as fases que compõem o ciclo de vida de uma empresa virtual. Entretanto, o que mais difere uma da outra é o escopo que cada uma compreende, ou seja, alguns autores definem poucas fases que agregam um grande número de funções, enquanto outros criam um maior número de fases cada uma contendo funções mais específicas, porém todos tratam praticamente das mesmas etapas (KANET *et al.*, 1999).

É importante considerar que na execução destas fases, assume-se que as empresas envolvidas possuam uma infraestrutura de comunicação e serviços – uma plataforma instalada e configurada – para suportar suas participações em EVs.

A seguir são identificadas as fases da empresa virtual (CAMARINHA-MATOS *et al.*, 1999a, KANET *et al.*, 1999, SPINOSA *et al.*, 1998, OUZOUNIS, 2001) que podem ser visualizadas na Figura 3 (CAMARINHA-MATOS *et al.*, 1999a).

Figura 3 – Fases do ciclo de vida da empresa virtual.

Fase de Criação: Na fase de criação são identificados os parceiros que satisfaçam os requisitos que foram especificados pelo processo de negócio. A procura por parceiros em potencial pode ser facilitada pelo uso de ferramentas como bases de dados *on-line*

(diretório externo de fornecedores), diretórios de *cluster* e serviços de procura na Internet (CAMARINHA-MATOS *et al.*, 1999b).

Além da identificação dos parceiros, é nesta fase que são especificadas detalhadamente as regras e ajustados os parâmetros para o processo ao qual a EV se destina. Durante esta fase, ferramentas de especificação de projetos podem ser úteis para projetar o fluxo de informação e material. Como uma das atividades realizadas nesta fase, tem-se a criação de um repositório de dados. Tal repositório conterá todas as informações intercambiadas entre as empresas participantes e será acessível a todos os parceiros, levando-se em consideração o nível de direito de acesso de cada um. As principais funcionalidades desta fase são: seleção de parceiros, negociação dos contratos, definição dos direitos de acesso à informação, configuração da topologia e da plataforma de gestão da EV.

Fase de Operação: Esta é a fase em que a empresa virtual executa seu processo de negócio. Ou seja, é neste estágio do ciclo de vida da EV que são atendidos os pedidos de clientes através da execução das respectivas ordens de produção. Todo este processo precisa ser constantemente verificado para garantir que a EV possa alcançar seu objetivo. A verificação é feita através do acompanhamento das informações de produção, vendas e distribuição. A qualquer sinal de uma possível falha na condução normal do processo de negócio uma modificação no contrato pode ser feita. Desta forma, o processo de negócio pode ter suas especificações alteradas durante a execução da EV, tais como: reescalonamento de tarefas de produção, mudança de datas de vencimento dos pedidos, alterações de algumas especificações técnicas e / ou procedimentos de transporte, etc. (RABELO *et al.*, 1997).

Fase de Evolução: Refinamentos podem ser necessários durante a operação da empresa virtual. Esta situação acontece quando é necessária a inclusão, exclusão ou substituição de um parceiro ou ainda uma alteração nos direitos de acesso / visibilidade de informação. Isto pode ocorrer devido a algum evento excepcional, tal como a incapacidade de um parceiro na execução de sua tarefa ou da necessidade de incrementar a carga de trabalho.

Fase de Dissolução: A fase de dissolução é a fase onde a EV acaba seu processo de negócio e se extingue. Dois motivos podem ser a causa da dissolução da empresa virtual. O primeiro se dá quando todos os objetivos da EV são realizados, ou seja, o negócio é

executado plenamente pelos parceiros e os mesmos não vêem mais a necessidade de trabalhar cooperativamente. O segundo é uma situação extrema, que ocorre quando da observância de uma grave situação na execução de alguma parte do processo de negócio da EV e que, por conseqüência, impeça que a mesma prossiga sua execução como um todo. É importante observar que um negócio não termina necessariamente quando o produto é entregue ao cliente final. Consoante ao tipo de produto e leis vigentes, pode ocorrer que a EV tenha que se manter formalmente conectada para fins de garantir serviços pós-venda (garantia, devolução, etc.) e eventualmente de desmontagem / reciclagem.

2.5 Papéis dos Participantes

Cada empresa executa um papel bem determinado dentro de uma empresa virtual. Assim, vários tipos de atores podem ser encontrados em uma EV. Os dois principais papéis são o coordenador e o membro (CAMARINHA-MATOS *et al.*, 1999a). Tipicamente, uma empresa virtual tem apenas um coordenador sendo os outros participantes chamados de membros. Além destes dois papéis principais, chamados de "papéis estratégicos", onde cada empresa deve se encaixar em um destes perfis, cada empresa envolvida em uma EV pode desempenhar também outros papéis, chamados de "papéis operacionais", tais como: Provedor de Dados / Serviços, Gerente de Projeto, Auditor, Técnico da rede, etc. Estes últimos são papéis definidos dependendo do ramo de negócios da EV, ou através de uma nomenclatura muitas vezes proposta pelo coordenador da EV no momento de sua concepção. Para este trabalho foi definido um conjunto particular de papéis operacionais para seus participantes que será apresentado na seção 4.1.1.

2.5.1 Coordenador da EV

Este é o papel desempenhado pelo componente regulador das atividades da empresa virtual (KATZY *et al*., 2003). É normalmente exercido por uma das empresas da EV que é também, na maioria dos casos, a empresa que recebe o pedido do cliente, ou seja, é o elo de ligação entre o cliente final e a EV. Contudo, eventualmente o coordenador pode ser uma empresa externa à EV, inserida à mesma especialmente para a coordenação. Isto ocorre quando nenhuma das empresas participantes tem capacidade suficiente para exercer esta tarefa. O coordenador tem tipicamente as seguintes atribuições (CAMARINHA-MATOS *et al.*, 1999a):

- Registra uma nova empresa na rede;
- Mantém o diretório externo de informação;
- Provê assistência a uma nova empresa para a instalação e configuração da infraestrutura de suporte;
- Reconfigura a EV, se necessário, e difunde as informações sobre a evolução da mesma;
- Supervisiona e coordena as atividades de diferentes empresas para o objetivo comum da empresa virtual;
- Supervisiona e assiste as empresas na dissolução da empresa virtual.

2.5.2 Membro da EV

As empresas com diferentes habilidades que participam de uma empresa virtual constituem os nós membros da empresa virtual (KATZY *et al*., 2003). As principais funções executadas por um membro são (CAMARINHA-MATOS *et al.*, 1999a):

- Executar suas atividades associadas ao processo de negócio de acordo como as cláusulas contratuais;
- Gerenciar os direitos de visibilidade das informações locais para proteger seus interesses e os da empresa virtual;
- Trocar informações visando uma melhor cooperação e escalonamento da produção de uma ordem específica em outros membros.

2.5.3 Outros Papéis

Além dos dois tipos principais de papéis apresentados acima, uma empresa pertencente a uma EV pode desempenhar também outros papéis "secundários". Os tipos de papéis apresentados nesta seção são chamados de "papéis operacionais", pois qualificam o tipo de função exercida por uma empresa em uma determinada EV. Dentre a vasta quantidade de possíveis papéis operacionais existentes, os mais importantes são (CAMARINHA-MATOS *et al.*, 1999a, OUZOUNIS, 2001):

- **Provedor de Dados**: Em uma rede de empresas um ou mais nós podem atuar como fonte de dados. Tipicamente, um nó provedor de dados é um ambiente somente de leitura que pode autorizar as empresas a receber suas informações.
- ***Broker***: Este é o papel desempenhado por uma empresa (não necessariamente o coordenador), que através de um primeiro contato feito por um cliente, procura os parceiros e inicializa / cria a empresa virtual. Devido a sua importância para o presente trabalho, este papel será apresentado de forma detalhada na seção 2.6.
- **Gerente de Projeto**: É o responsável pela engenharia e processamento das ordens, mantendo as restrições de tempo, orçamento e sendo hábil a reescalonar o processo de maneira a continuar satisfazendo as necessidades do negócio. Seu papel é por vezes confundido com o papel do coordenador, porém suas funções se restringem a uma pequena parte das atribuições do mesmo. Por exemplo, o gerente de projetos não tem a capacidade de retirar ou substituir um determinado parceiro da EV.
- **Auditor**: É o responsável pelo controle financeiro da EV, ou seja, controla as receitas e despesas da mesma, tentando manter um equilíbrio entre estas duas contas.
- **Técnico da rede**: É o encarregado da construção e manutenção da infraestrutura de comunicação.
- **Gerente de recursos**: É responsável pela coordenação e alocação dos recursos tecnológicos.

2.6 Brokerage

Dentre os papéis apresentados na seção 2.5.3, um merece destaque especial pela função que desempenha dentro da EV. Este papel é chamado *Broker* e sua função na EV é de fundamental importância. O conjunto de atividades desempenhadas pelo *broker* é denominado *Brokerage*. O *broker* atua na seleção de novos parceiros na fase de criação, quando é necessário decidir o conjunto de habilidades e recursos apropriados para a oportunidade de negócio vigente. Atua também durante a fase de operação, quando da necessidade de substituir um parceiro da EV (ÁVILA *et al.*, 2002). O propósito do *broker*

é encontrar o grupo mais adequado de empresas que respondam a uma oportunidade de negócios. No contexto de EV, o conceito de *broker* também é visto como a entidade projetada para procurar por oportunidades de negócios e trazê-las para o conjunto de empresas que ele representa (RABELO *et al*., 2000). Apesar de sua importância, em alguns casos este papel pode não estar explicitamente representado por uma empresa da EV. A falta de um *broker* pode acarretar em algumas dificuldades na condução da EV, entretanto, não inviabiliza sua execução. Na maioria dos casos em que o *broker* não está presente, as funções do mesmo acabam sendo desempenhadas separadamente pelo conjunto de empresas que fazem parte da EV.

O *broker* é em primeiro lugar um membro da EV, que atua como gerenciador de informações e iniciador de negócios. Na maioria dos casos, o *broker* se transforma no coordenador da EV e também no moderador durante a fase de operação. Ele é o primeiro ponto de contato quando o cliente pretende interagir com a EV (MEJÍA *et al.*, 2002).

O uso do *broker* em EVs pode ser justificado como resposta às seguintes questões: como o cliente localiza o fornecedor?; como ele efetua uma compra?; como ele encontra o requerido produto ou serviço com preço justo? etc. De acordo com HANDS et al. (2000), o *broker* serve como solução ideal para a mediação entre os fornecedores e os clientes. Para RESNICK et al. (1994), a importância do *broker* é justificada através da redução dos custos, maior disponibilidade de informação para o cliente (sobre a qualidade do produto e a satisfação do mercado), diminuição dos riscos pelo não cumprimento dos prazos por alguma parte envolvida, e o melhoramento da eficiência do preço pela criação de mecanismos que envolvem somente as vendas adequadas. Assim, o *broker* contribui decisivamente para um melhor desempenho da EV, agilizando seu projeto e operação.

2.7 Configuração de Empresas Virtuais

Como a ênfase do trabalho diz respeito à configuração de empresas virtuais, uma seção dedicada a esse assunto foi introduzida. Esta seção tem por objetivo detalhar os passos envolvidos no processo de configuração de uma EV ressaltando seus principais aspectos. Para BERG et al., (2000), uma empresa virtual deve ser configurada exatamente para atender a um processo de negócio específico. O processo de configuração é muito importante no contexto de empresas virtuais. Se o mesmo não for levado em consideração pode comprometer o sucesso do objetivo para o qual a mesma foi criada.

As atividades que compreendem o processo de configuração têm sua aplicação destacada na fase de criação da EV. Entretanto, várias das funcionalidades envolvidas neste processo podem ser acionadas em outras fases de seu ciclo de vida ou até mesmo antes de seu início (pré-configurações). Uma destas situações ocorre na fase de evolução, onde, para a adição de um novo parceiro, algumas funcionalidades de configuração podem ser executadas para especificar as regras e parâmetros envolvidos com a inclusão do novo parceiro no ambiente de operação da EV.

As funcionalidades de suporte à configuração são importantes tanto no caso de oportunidades de negócio simples, quanto em alianças de longo tempo de duração. No último caso, embora parte da rede permaneça relativamente estável, nós não estratégicos podem ser dinamicamente adicionados ou removidos de acordo com a oportunidade de negócio corrente e o estado atual de cada processo de negócio (CAMARINHA-MATOS *et al.*, 1997).

Uma vez que uma oportunidade de negócio é detectada, há a necessidade de rápida e eficientemente se planejar uma EV, identificando parceiros e estabelecendo acordos de cooperação para regular a sua operação. Para ajudar o criador da EV (coordenador da EV ou *broker*), o processo de configuração deve prover um conjunto específico de funcionalidades que facilite a tarefa e minimize o tempo gasto em sua execução, tornando-a assim mais eficiente (CAMARINHA-MATOS *et al.*, 2000). A seguir serão descritos os passos de configuração de uma EV a partir da identificação de uma oportunidade de negócios (CAMARINHA-MATOS *et al.*, 1999a, CAMARINHA-MATOS *et al.*, 1999c, SOARES *et al.*, 2002).

Procura por Parceiros

O processo de criação da EV envolve a procura e seleção de parceiros para fazer parte do consórcio. O problema de localizar parceiros pode ser decomposto em dois grupos:

- Parceiros principais, responsáveis por serviços e componentes críticos da EV;
- Parceiros secundários, que fornecem serviços e componentes menos importantes.

Para o primeiro grupo, as empresas têm usualmente suas listas proprietárias de potenciais fornecedores. Para a procura de parceiros secundários, entretanto, pode ser útil

uma consulta a fontes externas, como o uso de ferramentas normalmente baseadas na Internet.

A procura por parceiros pode ser baseada em um número diferente de fontes de informação, sendo elas privadas, públicas, ou independentes. A lista privada de cadastro de fornecedores da empresa é um repositório de dados que contém informações sobre as empresas que têm relacionamentos comerciais com a mesma. Estas informações são usualmente mantidas pelo *ERP*[1], e podem ser vistas como pertencentes a um diretório interno de fornecedores (ISD)[2] (CAMARINHA-MATOS *et al.*, 1999b). As fontes públicas de parceiros incluem diretórios mantidos por associações industriais, canais de comércio ou serviços de Internet. Estes serviços já estão disponíveis em vários países e oferecem interfaces WWW para o repositório de dados que contém as informações das empresas. Fontes públicas de fornecedores são normalmente chamadas de diretórios externos de fornecedores (ESD)[3] (CAMARINHA-MATOS *et al.*, 1999b). Outra solução emergente é a criação de *clusters* de empresas que concordam em cooperar e cuja habilidade e disponibilidade de recursos são registrados em um diretório de *cluster* (CD)[4] comum / próprio (MARTIN, 1999).

Negociação e Seleção de Parceiros

Neste passo da configuração todos os parceiros em potencial devem mandar suas propostas para o provedor da oferta, que é normalmente o *broker* ou o coordenador da EV. Adicionalmente, algumas interações complementares entre os parceiros podem ser necessárias para alcançar o acordo final para ambas as partes. O projeto de um protocolo de negociação adequado e o uso de ferramentas de suporte à decisão podem ser utilizados para facilitar o processo (HOFFNER *et al.,* 2002).

Formalização do Contrato

Na formalização do contrato os parceiros selecionados no passo anterior irão cooperar e escrever o contrato final conjuntamente. No contrato, as regras, responsabilidades, e condições financeiras para cada parceiro são acordadas entre os

[1] ERP – *Enterprise Resource Planning*.
[2] ISD – *Internal Suppliers Directory*.
[3] ESD – *External Suppliers Directory*.
[4] CD – *Cluster Directory*.

membros e o coordenador. Para gerenciar este processo podem ser utilizados ambientes de trabalho cooperativo suportados por computador (CSCW)[5] entre o coordenador e os membros da EV (BARATA *et al.*, 2002).

Configuração dos Recursos e Instalação da Plataforma de Gestão

Neste passo acontece a preparação dos parceiros selecionados para torná-los aptos a participar da EV. Esta preparação ocorre do ponto de vista da infraestrutura, onde são configurados os recursos de rede e sistemas para que cada membro da EV possa interoperar e trocar informações. Portanto, é neste passo da configuração que ocorre a instalação da plataforma em cada um dos parceiros da EV, plataforma esta que fará a gestão integrada da mesma. Após a instalação, testes envolvendo os recursos de suporte à comunicação entre as empresas envolvidas são realizados. Estes testes servem para garantir o correto funcionamento da plataforma na fase de operação.

Integração dos Repositórios Locais aos Sistemas de Gestão da EV

Neste passo são feitas as integrações das fontes locais de dados dos parceiros ao repositório de dados da plataforma de gestão da EV. Devido à heterogeneidade dos modelos de dados utilizados pelas fontes de dados dos sistemas que gerenciam as atividades locais de cada empresas, e levando-se em consideração que são a partir destas fontes de dados que a plataforma de gestão da EV gerencia todo o processo, há a necessidade de se integrar estas fontes aos sistemas de gestão da EV. Para haver esta integração um pré-processamento destas informações de ser feito.

Este processo consiste em extrair os modelos de dados dos repositórios locais e fazer o relacionamento entre cada uma destas informações a seu correspondente no modelo de dados da plataforma de gestão da EV. Assim, quando a plataforma necessitar alguma informação armazenada em algum repositório local, a sua localização acontece através da ponte construída pelo relacionamento estabelecido previamente (CARVALHO *et al.*, 2000, FRAYRET *et al.*, 2000).

Embora estejam sendo desenvolvidas ferramentas para automatizar este processo, na maioria dos casos o relacionamento entre os modelos de dados é feito manualmente pelo usuário configurador da plataforma de gestão da EV.

---

[5] CSCW – *Computer Supported Cooperative Work*.

Configuração da Topologia da EV

A topologia da empresa virtual tem por objetivo dar ao usuário da plataforma de gestão da EV uma visão global das empresas participantes, bem como dos relacionamentos existentes entre elas. A representação é feita de forma gráfica e cada empresa é caracterizada por um ícone. O relacionamento comercial entre duas empresas é representado por uma conexão interligando as mesmas. Esta exibição gráfica facilita o entendimento e gerenciamento da EV por parte dos usuários da plataforma (RABELO *et al.*, 2002). A Figura 4 apresenta um exemplo de configuração da topologia para uma determinada EV.

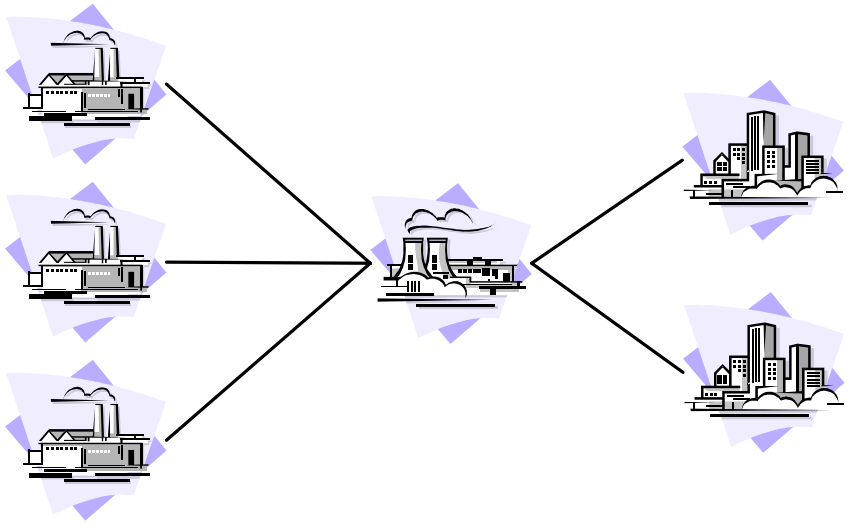


Figura 4 – Exemplo de topologia de EV.

Para configurar a topologia da empresa virtual, a plataforma de gestão deve prover uma ferramenta visual que dê suporte à associação das empresas aos respectivos ícones que representam seus papéis no consórcio, além de possibilitar a criação de seus relacionamentos. Quanto mais automático for o processo de configuração da topologia, mais rápido a mesma pode entrar em operação e melhores serão os resultados desta configuração em ambientes dinâmicos.

Especificação dos Direitos de Acesso às Informações Intercambiadas

Neste passo são feitas as restrições de acesso às informações dos parceiros envolvidos na EV. Levando-se em consideração a importância da privacidade das informações intercambiadas entre os membros de uma empresa virtual, a plataforma de gestão da EV deve prover uma forma de restringir o acesso a estas informações. Através

desta restrição as empresas envolvidas poderão ter confiança e segurança para liberar suas informações ao consórcio.

Uma das formas para manter a privacidade das informações da EV consiste em liberar ou restringir o acesso à parte da informação intercambiada entre os parceiros, tornando o coordenador o único provedor de informações para os membros do consórcio. Sempre que um membro necessitar de uma informação da EV, o mesmo fará essa requisição ao coordenador que analisará o pedido e dependendo de seu nível de acesso enviará ou não a informação pedida. A esta forma de restringir o acesso à informação dá-se o nome de "Cláusulas de Supervisão" (RABELO *et al*., 1999b).

2.8 Contratos e Cláusulas de Supervisão

Assim como em contratos tradicionais entre empresas, EVs também devem ter contratos que especifiquem os direitos e obrigações que uma empresa deve ter em relação à outra. Um contrato é feito da promessa de uma entidade fazer uma certa coisa em troca da promessa de outra entidade fazer outra coisa (BARATA *et al.*, 2002). Para REINECKE et al. (1989) "Um contrato é um acordo legal em que duas ou mais partes se comprometem a cumprir certas obrigações tendo como retorno certos direitos". Estes direitos e obrigações são representados comumente na forma de cláusulas. Em geral, esta especificação inclui diferentes tipos de cláusulas compreendendo informações legais, técnicas e financeiras.

Entretanto, do ponto de vista da empresa virtual, se faz necessário um controle mais rígido sobre o que está acontecendo no local de cada um dos parceiros bem como na EV por completo. Para minimizar este problema foi usado um tipo adicional de cláusula, chamada "Cláusula de Supervisão" (RABELO *et al*., 1999b).

Cláusulas de supervisão são usadas para especificar os direitos de acesso às informações tendo em vista o propósito de monitoramento da EV. Em geral, elas especificam qual, como, quando e onde um dado conjunto de informações de um determinado membro pode ser acessada. Isso significa que o coordenador e os demais membros da EV só terão acesso às informações previamente acordadas com cada um dos parceiros (FRENKEL *et al*., 2000).

Utilizando cláusulas de supervisão os membros da EV terão acesso apenas às informações autorizadas. Desta forma, empresas que são concorrentes em potencial fora do

âmbito da EV, podem restringir o acesso a suas informações mais sigilosas. Portanto, as cláusulas de supervisão auxiliam na manutenção da privacidade entre os parceiros da EV, sem que com isso haja uma perda da qualidade da informação necessária para seu funcionamento e gerenciamento.

2.9 Trabalhos Relacionados

A atividade de configuração de empresas virtuais – foco deste trabalho – vem sendo pesquisada nos últimos anos no âmbito de alguns projetos de pesquisa. Entretanto, este ainda é um assunto relativamente novo em termos de trabalhos já realizados. Isso se deve ao fato de que em um primeiro momento as pesquisas relacionadas com empresas virtuais destacavam o processo de realização da tarefa, ou seja, a fase de operação, que é a etapa na qual os objetivos da EV são alcançados. O enfoque dado a esta fase teve como propósito mostrar à comunidade científica e empresarial que o conceito de empresas virtuais é relevante e aplicável.

Porém, após um bom tempo dedicado a pesquisas direcionadas à operação da EV (levando em consideração tarefas tais como: escalonamento de processo, controle distribuído da produção, logística e etc.), agora uma atenção maior está sendo dada a outras fases de seu ciclo de vida, dentre as quais a fase de criação e o processo de configuração. Este processo começou com o desenvolvimento de métodos e ferramentas para a procura / seleção de parceiros, *brokerage*, e mais recentemente com algumas pesquisas sendo realizadas em outras etapas do processo de configuração.

Por ser um assunto ainda pouco pesquisado, a localização de projetos que abordassem esse assunto foi pequena. A procura foi realizada através da análise de artigos publicados em conferências nacionais e internacionais que abordaram o assunto, e também através de relatórios técnicos de projetos disponibilizados nas páginas Internet dos mesmos. Na Tabela 1 é apresentada a lista de trabalhos relacionados no âmbito de empresas virtuais que foram localizados. Entretanto, os projetos presentes nesta lista não apresentaram em seus documentos, disponíveis ao público, informações relevantes ao processo de configuração de empresas virtuais. A falta de projetos nacionais na Tabela 1 indica que esse é um assunto ainda pouco difundido nos grupos de pesquisa brasileiros. Em muitos casos, grupos nacionais necessitam se juntar a projetos internacionais para poder desenvolver suas pesquisas e trocar experiências.

Abreviatura	Nome	Duração	Programa
VOSTER	Virtual Organisations Cluster http://cic.vtt.fi/projects/voster/public.html	2001-2004	IST
ePROCON	eProduct Information in Construction http://cic.vtt.fi/projects/eProCon/public.html	2002-2004	–
SYMPHONY	A Dynamic Management Methodology with Modular and Integrated Methods and Tools for Knowledge Based, Adaptive SMEs http://www.smartlink.net.au/symphony.htm	2001-2004	IST - IMS
VOmap	Roadmap Design for Collaborative Virtual Organizations in Dynamic Business Ecosystems http://cic.vtt.fi/projects/vomap/index.html	2002-2003	IST
E-COLLEG	Advanced Infrastructure for Pan-European Collaborative Engineering http://www.ecolleg.org/	2000-2003	IST
CE-NET	Concurrent Enterprising Network of Excellence http://www.ce-net.org/	2001-2003	IST
ICCI	Innovation Co-ordination, Transfer and Deployment through Networked Co-operation in the Construction Industry http://cic.vtt.fi/projects/icci/public.html	2001-2003	IST
e-COGNOS	Methodology, Tools and Architectures for Electronic Consistent knowledge Management across Projects and between Enterprises in the Construction Domain http://cic.vtt.fi/projects/ecognos/index.html	2001-2003	IST
ROADCON	Strategic Roadmap towards Knowledge-driven "sustainable" Construction http://cic.vtt.fi/projects/roadcon/public.html	2002-2003	IST
PRODCHAIN	Development of a decision support methodology to improve logistics performance in production networks http://complingua.net/prodchain/	2000-2003	IST - IMS
prodAEC	European Network for Product and Project Data	2001-2003	IST

	Exchange, e-Work and e-Business in Architecture, Engineering and Construction http://cic.vtt.fi/projects/prodaec/index.html		
EXPIDE	Extended Products in Dynamic Enterprises http://www.biba.uni-bremen.de/projects/expide/	2001-2003	IST
BIDSAVER	Business Integrator Dynamic Support Agents for Virtual Enterprise http://www.ceconsulting.it/ve/bidsaver.html	2001-2002	IST
ALIVE	Working group on Advanced Legal Issues in Virtual Enterprise http://www.vive-ig.net/projects/alive/	2001-2002	IST
eLEGAL	Specifying Legal Terms of Contract in ICT Environment http://cic.vtt.fi/projects/elegal/public.html	2000-2002	IST
EXTERNAL	Extended Enterprise Resources, Network Architectures and Learning http://research.dnv.com/external/	2000-2002	IST
COVE	CO-operation infrastructure for Virtual Enterprises and electronic business http://www.uninova.pt/~cove/	2001-2002	IFIP
ICSS	Integrated Client-Server System for a Virtual Enterprise in the Building Industry http://cib.bau.tu-dresden.de/icss/	2000-2002	BMBF
GENESIS	Global Enterprise Network Support for the Innovation Process http://www.cetim.org/genesis.html	2000-2002	IST
eConstruct	Electronic Business in the Building and Construction Industry: Preparing for the new Internet http://www.econstruct.org	2000-2002	IST
ISTforCE	Intelligent Services and Tools for Concurrent Engineering http://cic.vtt.fi/projects/voster/public.html	2000-2002	IST
GNOSIS	GNOSIS Virtual Factory	1998-2001	ESPRIT

	http://www.vtt.fi/aut/tau/network/ims/web/ims_gnos.htm		
I-SEEC	Information Services to Enable European Construction Enterprises http://cic.vtt.fi/projects/i-seec/index.html	2000-2001	ETTN
MASSYVE	Multi-Agent Agile Manufacturing Scheduling Systems in Virtual Enterprise Industry http://www.gsigma-grucon.ufsc.br/massyve/	1997-2000	INCO-DC
CONNET	Construction Information Service Network http://cic.vtt.fi/projects/connet/index.html	1999-2000	ETTN
COWORK	Concurrent Project Development IT Tools for Small-medium Enterprises Networks http://www.tekniker.es/cowork/home.htm	1997-2000	ESPRIT
ABROSE	Based Brokerage Service in Electronic Commerce http://www.cordis.lu/infowin/acts/rus/projects/ac316.htm	1998-2000	ACTS
MIAMI	Mobile Intelligent Agent for Managing the Information Infrastructure http://www.cordis.lu/infowin/acts/analysys/products/thematic/agents/ch3/miami.htm	1998-2000	ACTS
Globeman21	Global Manufacturing towards the 21st Century http://cic.vtt.fi/projects/gm21/index.html	1997-1999	IMS
VENICE	Virtual Enterprise Nurtured using Intelligent Collaborative Environments http://event-net.fi/venice/	1997-1999	–
VEGA	Virtual Entreprises using Groupware tools and distibuted Architecture http://cic.cstb.fr/ILC/ecprojec/vega/home.htm	1996-1998	ESPRIT

Tabela 1 – Lista de projetos pesquisados.

Os projetos a seguir apresentam trabalhos relevantes em termos de configuração de empresas virtuais. Cada um destes projetos tem uma abordagem específica deste problema, trazendo um maior embasamento para compreender suas dimensões.

2.9.1 Projeto PRODNET - II

PRODNET (*Production Planning and Management in an Extended Enterprise*) foi um projeto realizado entre outubro de 1996 e setembro de 1999, financiado pela Comissão Européia e que fez parte da iniciativa ESPRIT. O projeto PRODNET visou o desenvolvimento de uma plataforma aberta para o suporte a empresas virtuais industriais. Seu foco principal foi as pequenas e médias empresas, ajudando no suporte à intercooperação em rede. O consórcio PRODNET não teve a pretensão de abordar todos os tópicos relacionados à coordenação global da EV. Isto ocorreu devido ao fato de que para várias funcionalidades já existiam várias ferramentas no mercado. Portanto, levando em consideração esta disponibilidade de recursos, um subconjunto de funcionalidades foi desenvolvido. A plataforma básica inclui (CAMARINHA-MATOS *et al.*, 1997):

- Troca de dados comerciais (EDIFACT[6]);
- Troca de dados técnicos do produto (STEP[7]);
- Monitoramento do estado das ordens;
- Suporte ao gerenciamento de informação administrativa sobre a EV, e das informações que os parceiros decidem tornar disponíveis na rede;
- Módulo de coordenação que executa os eventos relacionados à coordenação;
- Módulo de configuração que permite a definição e parametrização da EV e do comportamento de cada um dos parceiros;
- Sistema estendido de controle da produção, adaptado para interagir com o ambiente de EV, e inclui o gerenciamento de ordens incompletas ou imprecisas.

Considerando que o cenário alvo foram as pequenas e médias empresas, deve-se notar que nem todas as empresas estariam interessadas em todas as funcionalidades disponíveis pela plataforma PRODNET. Portanto, as várias funcionalidades poderiam ser habilidades ou não, dependendo dos valores do conjunto de parâmetros de configuração. A Figura 5 (CAMARINHA-MATOS *et al.*, 1997) apresenta a arquitetura geral para cada nó da EV. A arquitetura é composta pelos seguintes componentes:

---

[6] EDIFACT – *Electronic Data Interchange for Administration, Commerce and Ttransport*.
[7] STEP – *Standard for the Exchange of Product Model Data.*

PPC – Sistema de controle e planejamento da produção;

PCL – Camada de cooperação PRODNET;

DIMS – Sistema de gerenciamento de informação distribuído;

LCM – Módulo de coordenação local;

EDI – Módulo de troca de dados eletrônicos;

STEP – Módulo para a troca padrão de dados técnico de produto;

PCI – Infraestrutura de comunicação PRODNET;

PICP – Protocolo interno de comunicação PRODNET;

PECP – Protocolo externo de comunicação PRODNET;

AFC – Funcionalidades de coordenação avançada.

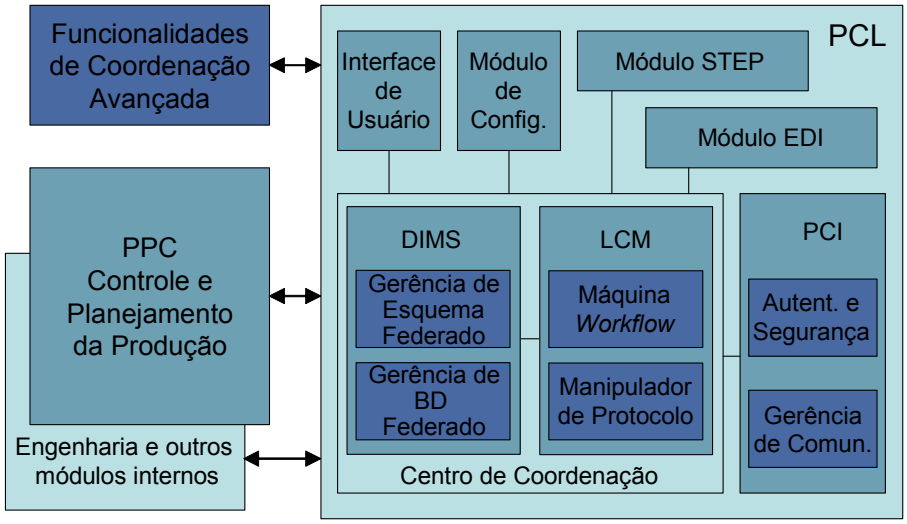


Figura 5 – Arquitetura geral PRODNET.

No desenvolvimento da arquitetura PRODNET uma grande ênfase foi dada para o desenvolvimento de um ambiente flexível e configurável. Ou seja, a infraestrutura PRODNET deveria satisfazer as necessidades das empresas em dois diferentes estágios. Primeiramente, na inicialização da camada de cooperação PRODNET para a especificação do ambiente da empresa e do comportamento de seus processos de negócios. Em segundo lugar, na configuração da camada de cooperação PRODNET para a especificação e regulamentação da EV para o envolvimento das empresas em diferentes empresas virtuais.

O PRODNET identificou alguns dos principais passos para o estágio de inicialização e configuração da EV. Estes passos são apresentados abaixo (CAMARINHA-MATOS *et al*., 1999c):

- **Preparação dos Recursos de Infraestrutura**: Este passo identifica e instala os recursos computacionais para a empresa. Isto inclui a preparação: do sistema operacional, do sistema de gerenciamento do banco de dados, da conexão à Internet.
- **Criação e Evolução da EV**: Quando uma oportunidade de negócio é identificada, os parceiros apropriados são procurados, identificados e selecionados. Alguns acordos preliminares precisam ser formalizados e então a topologia da EV e o papel de seus parceiros são definidos manualmente através de uma interface amigável. Este mesmo processo deve ser feito caso ocorra alguma evolução na EV. Por exemplo, caso encontre uma alteração no plano de produção da mesma, o coordenador necessita selecionar um ou mais novos parceiros, modificar a topologia e refazer os acordos dentro da EV.
- **Configuração da EV nos membros**: É neste passo da configuração que cada parceiro da EV carrega a definição da topologia da rede, as regulamentações e as cláusulas de supervisão recebidas do coordenador. Baseada nestas definições a empresa irá configurar o direito de acesso e visibilidade de suas informações para cada um dos outros membros da EV. Além disso, são configurados os componentes que compõem a camada de cooperação do PRODNET para operar em conformidade às especificações previamente acordadas entre os parceiros.

2.9.2 Projeto GLOBEMEN

O GLOBEMEN (*Global Engineering and Manufacturing in Enterprise Networks*) foi um projeto com duração de três anos, iniciado em janeiro de 2000 e encerrado em janeiro de 2003. Este projeto fez parte do programa IMS (*Intelligent Manufacturing Systems*) com o número (IMS99004). O objetivo do GLOBEMEN foi definir e desenvolver métodos, ferramentas e arquiteturas para suportar o relacionamento entre as empresas em todas as diferentes fases do ciclo de vida da EV. Estes modelos foram concebidos da forma mais genérica possível para serem usados em todos os tipos de indústrias que fazem produtos personalizados (TOLLE *et al.*, 2003).

Vários problemas podem ocorrer durante o projeto de uma empresa virtual. Por definição uma empresa virtual é complexa e a configuração das empresas participantes é freqüentemente custosa e demorada. As atividades executadas durante cada fase do ciclo de vida da EV são essencialmente derivadas de procedimentos especializados e específicos. Assim, a qualidade da empresa virtual dependerá consideravelmente da experiência do especialista envolvido. O problema é acentuado devido ao pobre formalismo com o qual estas atividades são usualmente executadas. Isto leva freqüentemente a soluções que não são adequadas às necessidades do negócio.

O objetivo da arquitetura de referência para empresas virtuais (VERA)[8] é oferecer um arcabouço que resolva o problema mencionado acima. A idéia principal por trás da arquitetura de referência é que uma grande parte dos processos de integração são de fato similares em vários projetos de plataformas de EVs. Desta forma, estas partes similares podem ser padronizadas e utilizadas por todos os projetos ao invés de serem desenvolvidas novamente para cada novo projeto. Uma vez padronizadas, estas arquiteturas de referências podem ser suportadas por ferramentas, metodologias e uma gama de produtos compatíveis, propiciando assim a completa integração entre os projetos de uma forma mais eficiente em relação ao tempo e aos custos (BERG *et al.*, 2000).

Na arquitetura de referência proposta pelo projeto GLOBEMEN, o processo de preparação e configuração da empresa virtual consiste das seguintes tarefas (TOLLE *et al.*, 2003):

- **Seleção de parceiros**: Na seleção de parceiros para uma EV devem ser levados em consideração dois aspectos: se o candidato tem a competência necessária para desempenhar a função; ou, se o candidato tem capacidade de produção suficiente. As fontes de informações utilizadas para localizar os candidatos a membros da EV são os diretórios internos de fornecedores, os diretórios externos de fornecedores e os diretórios de *cluster* (CAMARINHA-MATOS *et al.*, 1999b) apresentados na seção 2.7.
- **Divisão do trabalho**: Consiste na decomposição da tarefa que a EV deve desempenhar em sub-tarefas atribuídas a cada uma das empresas participantes. Além da decomposição da tarefa são estabelecidos, se necessário, os

[8] VERA – *Virtual Enterprise Reference Architecture*.

relacionamentos temporais entre cada uma delas. Esta tarefa é executada praticamente em paralelo com a tarefa de seleção de parceiros.

- **Preparação da EV**: Uma vez que os parceiros tenham sido selecionados e as tarefas tenham sido divididas, o próximo passo da configuração é a preparação da infraestrutura da empresa virtual. Isto significa:
    - Instalar a plataforma da EV em cada um dos membros;
    - Definir os direitos de acesso às informações;
    - Atribuir um papel a cada um dos membros da EV;
    - Configurar a interface entre os ERP locais à plataforma de gestão da EV para cada parceiro;
    - Acertar assuntos contratuais.

2.9.3 Projeto OSMOS

O projeto OSMOS (*Open System for Inter-enterprise Information Management in Dynamic Virtual Environments*), foi realizado entre janeiro de 2000 e março de 2002, sendo financiado pela Comissão Européia sub o escopo do programa IST com o número IST-10491. Este projeto contou com a participação das seguintes instituições: DERBi (França), CSTB (França), Granlund (Finlândia), JM (Suécia), ISI - University of Salford (Reino Unido), VTT (Finlândia). Seu objetivo foi aumentar a capacidade dos consórcios de empresas, especialmente pequenas e médias, para atuarem e colaborarem eficientemente em EVs através da utilização de serviços flexíveis baseados na Internet (*web services*). Estes serviços tinham como característica o suporte ao trabalho em grupo em redes dinâmicas. Este projeto teve como foco principal o desenvolvimento de soluções para empresas virtuais voltadas ao ramo da construção civil. O objetivo geral do projeto OSMOS pode ser subdividido da seguinte forma (REZGUI *et al*., 2000):

- Especificação de serviços baseados na Internet para a colaboração entre aplicações de consórcios diferentes, e uma semântica comum para referenciar as informações que ambos os consórcios manipulam.
- Especificação de serviços baseados na Internet que permitam a coordenação das interações entre os indivíduos em uma empresa virtual dinâmica.

- Especificação de um modelo de ambiente onde o acesso a qualquer informação compartilhada produzida por algum membro seja feita de forma segura.
- Disponibilização de ferramentas baratas e amigáveis para pequenas empresas, para que as mesmas possam mais facilmente participar de empresas virtuais.
- Implementação de serviços específicos para permitir que usuários utilizem suas aplicações proprietárias ou outras aplicações comerciais.

ZARLI et al. (2001) apresenta o processo de configuração dos serviços disponibilizados pela plataforma OSMOS como sendo a seguinte seqüência de etapas. Uma vez detectados os parceiros para uma determinada empresa virtual, a configuração da infraestrutura de hardware e software do mesmo deve ser feita. Isto significa a instalação e configuração, se necessário, de um servidor WWW, uma base de dados no coordenador, conexões de rede, e navegadores WWW nos parceiros. Após a instalação da infraestrutura, é necessária a configuração das permissões de acesso aos serviços disponibilizados pela plataforma a cada um dos parceiros. Esta configuração leva em consideração os acordos especificados previamente no contrato. Além das permissões de acesso aos serviços é definido o papel de cada uma das empresas que fazem parte do consórcio. Após a execução estes passos mencionados acima a empresa virtual está pronta para entrar em operação.

2.9.4 Considerações Sobre os Projetos Analisados

Considerando os projetos descritos nas seções anteriores, algumas análises podem ser feitas levando em consideração as características de cada um. Os três projetos analisados apresentaram vários avanços no que diz respeito às pesquisas envolvendo empresas virtuais. O projeto PRODNET teve como contribuição, na questão da configuração, a utilização de cláusulas de supervisão como técnica para o controle de acesso aos dados dos parceiros. Entretanto, este controle foi realizado ao nível de banco de dados e não sobre o protocolo comum de comunicação. Isto torna esta tarefa mais complexa, pois necessita de um administrador de banco de dados para executá-la. Outra contribuição importante do projeto PRODNET foi a concepção de uma ferramenta para a configuração da topologia da EV. A topologia é de grande valor para a compreensão da EV como um todo. Apesar de configurar a topologia, a ferramenta desenvolvida para executar esta tarefa não tem nenhuma espécie de automatização para facilitar a sua utilização por

parte do coordenador. Todo o processo é efetuado manualmente, o que torna o mesmo lento e passível de erros.

O projeto GLOBEMEN trouxe uma contribuição importante com a especificação de uma arquitetura de referência para empresas virtuais (VERA). Esta arquitetura de referência serviu para consolidar vários conceitos e também delinear as novas pesquisas relacionadas a EVs. Em relação à configuração de EVs este projeto não agregou inovações significativas. Fazendo uma comparação entre este projeto e o projeto PRODNET, ambos são bastante similares no contexto da configuração. A única diferença encontrada entre eles diz respeito à configuração da topologia da EV. O projeto GLOBEMEN não menciona como esta tarefa é realizada em sua plataforma.

Para finalizar a análise dos projetos relacionados à configuração de EVs, tem-se o projeto OSMOS. Este projeto teve como um de seus principais avanços a utilização de *web services* para facilitar o suporte ao trabalho em grupo. O suporte a este tipo de trabalho é de grande importância em se tratando de empresas virtuais. Porém, apesar de adicionar esta importante inovação ao âmbito das EVs, na questão da configuração não foram encontradas grandes contribuições em relação aos projetos analisados anteriormente. Além da já conhecida configuração dos papéis dos parceiros, o que chama mais atenção neste projeto é a configuração da permissão de acesso aos serviços utilizados pelos parceiros. Neste projeto nenhuma menção é feita em relação à configuração da topologia da EV.

Os projetos analisados nesta seção podem ser considerados como uma primeira iniciativa à configuração de EVs. Levando-se em consideração a natureza superficial de suas abordagens, estes projetos não destacaram a real importância da etapa de configuração da EV. Os processos de configuração apresentados são limitados, ou seja, não abordam o problema com a devida dinâmica com que o mesmo deve ser considerado. Portanto, o intuito desta dissertação é aprimorar parte do processo de configuração, basicamente a configuração da topologia e dos direitos de acesso às informações, introduzindo maior dinamismo ao processo e tornando o mesmo mais próximo às expectativas das empresas virtuais dinâmicas.

CAPÍTULO 3: Tecnologias Utilizadas

Este capítulo tem o intuito de descrever de forma sucinta as principais tecnologias de informação que foram utilizadas no desenvolvimento do trabalho ao qual essa dissertação se propõe, são elas: agentes, *workflow*, XML e CORBA. Além das tecnologias citadas, também foi utilizado um sistema de gerenciamento de banco de dados. Entretanto, este não foi aqui descrito, de forma conceitual, devido à sua grande disseminação e conhecimento por parte dos analistas e programadores de aplicações em sistemas de informação. Alguns dos conceitos e tecnologias aqui descritos são utilizados na modelagem da solução do problema, enquanto outros são utilizados na implementação do protótipo. O modelo proposto e a implementação do protótipo são apresentados respectivamente nos capítulos 4 e 5.

O modelo de configuração de EVs proposto utiliza a tecnologia de agentes para projetar as entidades de software que representam cada uma das empresas parceiras da EV. O grupo de agentes que representa uma empresa é chamado de sistema multiagente. Os sistemas multiagente das empresas devem cooperar entre si para realizar a configuração da EV. Para haver esta cooperação são necessárias trocas de mensagens entre as empresas. Esta troca de mensagens é suportada através da utilização do *middleware*[9] CORBA. O CORBA é uma das mais recentes e robustas tecnologias utilizadas para propiciar a troca de mensagens em ambientes constituídos por aplicações distribuídas. As mensagens trocadas entre as empresas devem estar em conformidade com o protocolo de mensagens previamente acordado. Este protocolo foi especificado utilizando a linguagem de marcadores chamada XML. As mensagens especificadas em XML e trocadas entre as empresas através do CORBA, necessitam de um sistema que controle o envio / recepção e gerencie o fluxo das mesmas para que não haja perda ou inversão da ordenação. Este tipo de serviço é oferecido pelas ferramentas de *workflow* e sua utilização é de primordial importância para a manutenção da seqüência lógica do processo. Todas as informações contidas nas mensagens trocadas entre os parceiros são armazenadas no banco de dados do parceiro que fez a requisição da informação.

---

[9] *Middleware* – Camada de software que provê serviços para aplicações distribuídas.

3.1 Agentes

A tecnologia de agentes é um campo emergente e que está sendo considerado a próxima geração de paradigma de programação, chamada "orientada a agente" (PANG, 2000). O termo agente tem vários significados na literatura e até hoje não existe uma definição aceita por todos de maneira unânime. Cada autor expõe sua conceituação sobre o tema levando em consideração sua área de atuação, desta forma alguns conceitos podem destoar uns dos outros. Aqui são vistas algumas destas definições.

Para XIANG (2002), um agente inteligente é um sistema computacional que percebe seu ambiente e executa ações inteligentes de acordo com seus objetivos.

KENDALL et al. (1997) divide agentes em duas categorias: um agente fraco é autônomo, sociável, reativo e pró-ativo, já um agente forte, além das características acima, também pode conter, noções mentais (crenças, objetivos, planos e intenções), racionalidade, veracidade e adaptabilidade.

Já WOOLDRIDGE et al. (1998) define um agente como sendo um sistema computacional, posicionado em algum ambiente, capaz de agir com autonomia flexível visando atingir os objetivos para o qual foi projetado.

Entretanto, para o presente trabalho será adotada a definição genérica proposta por FERBER et al. (1999), onde um agente é uma entidade real ou virtual, que está imersa em um ambiente dentro do qual ela pode executar certas ações, aprender e representar parcialmente esse ambiente, se comunicar com outros agentes e possuir um comportamento autônomo que é conseqüência das suas observações, do seu conhecimento e das suas interações com os outros agentes.

Levando em consideração as definições mostradas acima, algumas características e propriedades dos agentes podem ser definidas. Estas características servem de base para a distinção de um sistema formado por agentes de outros sistemas computacionais tradicionais. São estas características que fazem com que os agentes cada dia ampliem mais sua utilização nas mais diversas áreas da TI.

3.1.1 Propriedades dos Agentes

Quando se fala em agentes existem algumas propriedades que devem ser levadas em consideração. Para que o sistema seja considerado um agente, ele não precisa

necessariamente apresentar todas as propriedades atribuídas ao conceito. Entretanto, conforme visto nas definições, algumas destas propriedades são essenciais. Estas propriedades são importantes também na hora de classificá-los, pois é através delas que esse processo é feito.

WOOLDRIDGE et al. (1995) identifica oito características para sistemas de hardware ou software que juntas dão a noção de agente, são elas:

Autonomia: Os agentes podem operar sem a intervenção direta do usuário ou de outros agentes, possuindo algum tipo de controle em cima de suas ações e de seu estado interno. Uma analogia com esta propriedade pode ser feita à percepção de um relógio. Caso seu proprietário viajasse para o exterior, o mecanismo interno do relógio, se o mesmo tivesse autonomia, o ajustaria automaticamente à hora local do destino.

Habilidade Social: É a característica que os agentes possuem de interagir uns com os outros, e possivelmente com os humanos, através de algum tipo de linguagem de comunicação.

Reatividade: Os agentes percebem seu ambiente (que pode ser o mundo físico, um usuário através de uma interface gráfica ou outros agentes) e respondem de forma oportuna a estes estímulos.

Pró-Atividade: Agentes não atuam simplesmente em resposta a estímulos do ambiente, eles também são capazes de exibir um comportamento orientado ao objetivo, através da tomada de iniciativa própria.

Mobilidade: É a capacidade que um agente tem de se mover pela rede, transportar-se de uma máquina para outra preservando seu estado interno. Desta forma, o agente pode ocupar diferentes nós e recursos da rede ao longo do tempo.

Veracidade: Assume que um agente não comunicará conscientemente informações falsas para os outros agentes que fazem parte da comunidade.

Benevolência: Supõe que os agentes não têm objetivos conflitantes, e que todo agente conseqüentemente sempre tentará desempenhar a tarefa que lhe foi solicitada.

Racionalidade: Hipótese de que um agente deve agir de forma a atingir seus objetivos e não ir contra eles, pelo menos dentro do alcance de suas regras.

FRANKLIN et al. (1996) em sua proposta utiliza algumas das propriedades exibidas acima, entretanto, ele estende esse conjunto inclui cinco outras propriedades:

Orientação a Objetivo: Também chamado de pró-atividade intencional, define que os agentes devem ser capazes de lidar com tarefas complexas de alto nível. A decisão de como uma tarefa deve ser subdividida em tarefas menores, e em qual ordem e modo devem ser executadas deve ser feita pelo próprio agente.

Continuidade Temporal: Os agentes executam continuamente processos que podem tanto estar em primeiro plano, *foreground*, quanto em segundo plano, *background*.

Comunicabilidade: Também denominada habilidade social, é a capacidade que o agente tem no curso da realização de seus objetivos, de acessar informações sobre o estado atual do ambiente externo. Isso requer a habilidade de se comunicar com os repositórios destas informações. Essa comunicação pode ser na forma de um pedido simples, com um conjunto conciso de respostas possíveis, ou pode ser uma comunicação complexa com respostas variáveis.

Uma questão a ser tratada quando se fala de comunicação entre agentes é a interpretação do significado das declarações dos mesmos. Um termo pode ter diferentes significados para diferentes agentes. Assim, comunicabilidade é a capacidade que os agentes têm de se comunicar com outros agentes ou outras entidades, com o intuito de oferecer e obter informações. Para alcançar a comunicabilidade desejada, os agentes devem se comunicar através da mesma linguagem.

Aprendizagem: Também chamada de adaptabilidade, é a característica que define um agente como sendo capaz de se adaptar às rotinas, métodos de trabalho e preferências de seus usuários. Estas mudanças devem ser baseadas em suas experiências prévias. Para um agente ser considerado autônomo e demonstrar raciocínio, é necessário que o mesmo avalie o estado atual do ambiente no qual está imerso, e o incorpore para que em situações posteriores que conduzam a situações similares, o mesmo adapte suas ações com o intuito de melhorar a eficiência na realização de suas tarefas.

Flexibilidade: É a capacidade que o agente pode possuir de agir de forma não determinística para atender a uma condição adversa.

Caráter: O agente deve ser honesto para que desta forma os outros agentes possam acreditar nas suas informações. É através desta propriedade que a comunidade de agentes consegue distinguir se um agente é ou não malicioso[10].

3.1.2 Tipos de Agentes

Os agentes são divididos em categorias de acordo com suas características, estratégias de processamento, funções que os mesmos realizam ou pelo ambiente onde são executados. NWANA (1996) identificou alguns pontos como base para classificação de agentes:

- Por sua mobilidade, ele pode ser um agente estacionário ou móvel;
- Por sua ação, como cognitivo ou reativo;
- Por alguns de seus atributos: autonomia, aprendizado e cooperação;
- Por seus papéis, por exemplo, agente de coleta de informação, etc.

A seguir serão apresentados os tipos mais importantes de agentes relatados pela literatura.

Agentes de Hardware

Como o próprio nome já diz, estes agentes são implementados em componentes eletrônicos ou dispositivos físicos, e toda sua lógica de funcionamento está expressa dentro de circuitos integrados ou em softwares embutidos. Este tipo de agente está posicionado em um ambiente normalmente fechado e é guiado por uma função de satisfação, também chamada função de sobrevivência (HORN *et al.*, 1998). Possui recursos próprios, tais como: ferramentas, sensores, atuadores e etc., assim são capazes de perceber o ambiente onde estão inseridos. Por estarem imersos em um ambiente fechado, sua percepção se torna normalmente limitada. Seu comportamento segue estritamente sua função de satisfação, isso ocorre pelo fato de possuir poucos recursos e também um certo número limitado de habilidades.

[10] Agente Malicioso – É um agente que se comporta de forma inadequada, se passando por outro tipo de agente ou enviando informações incorretas, para desta maneira obter vantagens sobre os outros agentes.

Agentes de Software

Agentes de software estão quase sempre posicionados em um ambiente aberto, e são guiados por um conjunto dinâmico de objetivos (HORN *et al.*, 1998). Por serem tipicamente implementados utilizando linguagens de programação de alto nível, são mais flexíveis e adaptáveis que os agentes de hardware. Possuem uma visão parcial do ambiente onde estão inseridos. Suas habilidades são oferecidas aos demais agentes na forma de serviços. Seu comportamento segue um objeto considerando não apenas os recursos que possui, mas também considerando os recursos dos outros agentes que fazem parte do sistema.

Agentes Estacionários

Agentes estacionários são agentes que executam somente no ambiente onde são lançados. Se um agente precisar de informações que não estão neste ambiente ou necessitar interagir com agentes que estão em outros sistemas, são tipicamente usados mecanismos de comunicação para isto (LANGE, 1998).

Agentes Móveis

Os agentes móveis não estão limitados aos ambientes onde foram lançados. Estes agentes têm a habilidade de se transportar de um sistema para outro através da rede. Esta habilidade permite que o agente móvel viaje até o sistema que contém o objeto que ele deseja interagir, e assim obtenha vantagens por ambos estarem no mesmo local (LANGE, 1998).

Agentes Persistentes

Agentes persistentes são agentes que uma vez executados em um determinado ambiente computacional não podem ser excluídos do sistema (HORN *et al.*, 1998).

Agentes Temporários

Agentes temporários são agentes que têm uma vida finita, normalmente de duração igual ao tempo da tarefa a ser executada. Portanto, tão logo finalizem suas missões são automaticamente excluídos do sistema. Esta exclusão na maioria das vezes é executada por ele próprio (HORN *et al.*, 1998).

Agentes Reativos

Os agentes reativos têm como objetivo sobreviver. Seu modelo de funcionamento é baseado em estímulo-resposta. Geralmente estes agentes não apresentam uma memória das ações tomadas e também não planejam suas ações. Todo o conhecimento a respeito das ações e do comportamento dos demais membros da sociedade de agentes é percebido através das modificações sofridas pelo ambiente. Estes agentes não possuem um modelo de comunicação de alto nível, nem uma representação explícita do ambiente ou dos membros da sociedade (COSTA, 1997). Normalmente são de baixa complexidade e aparecem em grande quantidade nos sistemas.

Agentes Cognitivos

O objetivo dos agentes cognitivos é cooperar. Estes agentes possuem uma representação explícita do ambiente e dos membros da comunidade, e podem raciocinar sobre as ações tomadas no passado e planejar as ações futuras. Os agentes cognitivos podem ainda interagir com os demais membros da comunidade através de: linguagens e protocolos de comunicação complexos, sofisticadas estratégias de negociação, etc. Normalmente apresentam elevada complexidade computacional, e se caracterizam por apresentar um comportamento inteligente tanto em uma comunidade de agentes como isoladamente. Estas comunidades geralmente são compostas por um pequeno número de participantes (COSTA, 1997).

3.1.3 Tipos de Agentes de Software

Há várias maneiras de se classificar os agentes de software. Nesta seção serão apresentadas as categorias de agentes definidas por NWANA (1996). São identificados sete tipos diferentes de agentes de software, que podem ser vistos na Figura 6 (NWANA, 1996).

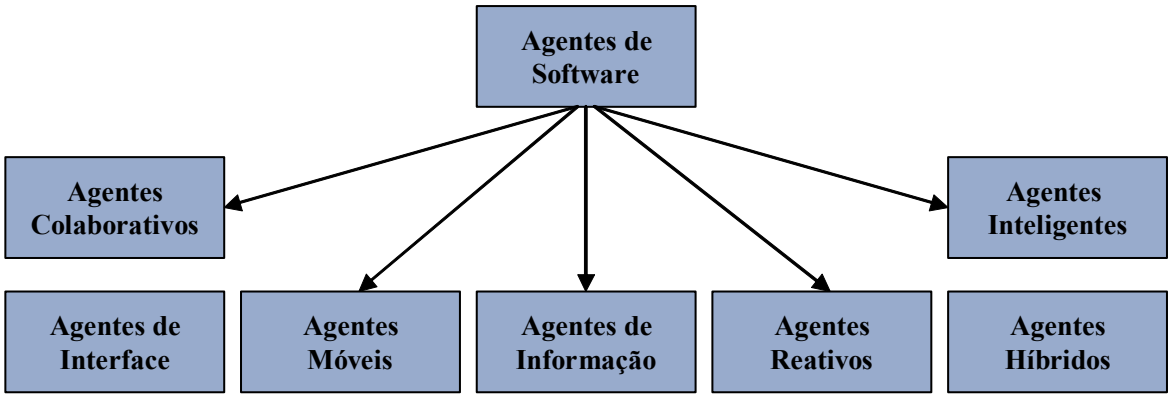


Figura 6 – Tipos de agentes de software.

Agentes Colaborativos

Agentes colaborativos são úteis na execução de tarefas que envolvam solução de problemas. Estes agentes executam várias tarefas e interagem com os outros para disseminar informação e resolver conflitos em ambientes multiagente. Eles podem aprender, entretanto, esse aspecto não é a maior ênfase da sua operação. Para coordenar suas atividades, eles podem ter que negociar para alcançar mutuamente um acordo aceitável NWANA (1996). Agentes colaborativos tendem a ser estáticos e de grandes complexidade. Este tipo de agente visa:

- Resolver problemas que são muito grande para um único agente centralizado, devido a limitações de recursos ou pelo risco de se ter um sistema centralizado;
- Habilitar a interconexão e interoperação com sistemas legados existentes, por exemplo, sistemas especialistas, sistemas de suporte à decisão e programas convencionais;
- Prover solução para problemas inerentemente distribuídos, por exemplo, distribuição on-line de recursos de informações ou redes de sensores distribuídos.

Agentes de Interface

Agentes de interface servem ao usuário como um assistente pessoal. O foco principal destes agentes é suprir as necessidades do usuário, aprendendo suas necessidades e hábitos, e ajustando seu conhecimento através de decisões prévias (WHITAKER, 1999).

Agentes de interface suportam e provêem assistência pró-ativa, tipicamente para um usuário aprendendo a usar uma aplicação em particular, tal como uma planilha eletrônica ou um sistema operacional. O agente do usuário observa e monitora as ações tomadas na interface pelo usuário, aprende novos atalhos e identifica as melhores formas para fazer as tarefas. Agentes de interface aprendem a assistir seus usuários da seguinte maneira:

- Observando e imitando o usuário;
- Recebendo retornos positivos ou negativos do usuário;
- Recebendo instruções implícitas do usuário.

Agentes Móveis

Este tipo de agente pode ser definido como um processo de software, que pode migrar entre sistemas computacionais em uma rede, com o intuito de realizar tarefas para seus usuários (BIGUS *et al.*, 1998). Estes agentes têm a capacidade de cruzar os limites da rede e acessar outros computadores em redes de longo alcance WAN[11], ou até mesmo na rede mundial de computadores Internet. Eles interagem com o sítio estrangeiro, recolhendo informações e voltando para seu local de origem, tendo executado seu conjunto de obrigações especificado pelo usuário (NWANA, 1996). Os agentes móveis trazem como principais benefícios:

Redução nos custos de comunicação: Pode haver muita informação a ser examinada para determinar quais são relevantes. Transmitir toda essa informação pode ter um custo de tempo e alocação de recurso muito elevado. Por exemplo, transmitir milhares de imagens para escolher somente uma. É muito mais natural enviar um agente móvel para executar a procura no local e transferir somente a imagem escolhida.

Recurso local limitado: O potencial de processamento e armazenagem da máquina local pode ser limitado, assim, o agente móvel executaria as tarefas na máquina remota apenas enviando o resultado.

Fácil coordenação: Pode ser mais simples coordenar um número de requisições remotas independentes e somente colecionar seus resultados localmente.

[11] WAN – *Wide Area Network*.

Computação assíncrona: Pode-se despachar os agentes móveis e ocupar o processamento com outras coisas, não havendo obrigatoriamente a necessidade de ficar a espera de seus resultados.

Uma arquitetura computacional flexível e distribuída: Agentes móveis provêem uma arquitetura computacional distribuída que funciona diferentemente dos agentes estáticos.

Agentes de Informação

Estes agentes são úteis em situações de sobrecarga de informação. Eles executam o papel de gerenciamento, manipulação e coleta de informação de várias fontes distribuídas. Agentes de informação são capazes também de filtrar grande quantidade de informação e selecionar os dados relevantes. O principal problema encontrado em agentes de informação é manter seus índices atualizados em um ambiente que vive em constante mudança (NWANA, 1996). Os agentes de informação são projetados para aliviar a sobrecarga de informação causada pela disponibilidade de uma ampla quantidade de informação mal catalogada.

Agentes Reativos

Agentes reativos representam uma categoria especial de agentes que não processam modelos internos de seu ambiente. Esles atuam / respondem de maneira simultânea aos estímulos do ambiente em que estão inseridos (NWANA, 1996). O ponto mais importante é o fato de que os agentes são relativamente simples e interagem uns com os outros de forma básica. Não obstante, padrões mais complexos de comportamento emergem desta interação quando o conjunto de agentes é visto de uma forma global.

MAES (1991) destaca três idéias que suportam agentes reativos. Primeiramente, a "complexidade das funcionalidades", devido à dinâmica das interações que conduz a uma maior exigência na concepção de cada função. Em segundo, a "decomposição das tarefas", pois um agente reativo é visto como uma coleção de módulos que operam autonomamente e são responsáveis por tarefas específicas. Por último, agentes reativos tendem a ser "muito específicos", devido à sua proximidade com os sensores e atuadores, em contraste com a representação mais alto nível dos outros tipos de agentes.

Agentes Híbridos

A arquitetura híbrida de agentes é o casamento de dois ou mais tipos de agentes para construir um modelo mais robusto e adaptado. Ou seja, agentes híbridos são a combinação de duas ou mais filosofias de agentes dentro de um único agente (NWANA, 1996). Como cada tipo de agente tem suas vantagens e deficiências, busca-se maximizar as vantagens e minimizar as deficiências para um propósito em particular.

Agentes Inteligentes

Para um agente ser considerado inteligente, o mesmo deve ser capaz de aprender como reagir e interagir com seu ambiente externo. LOGAN (1998) propõe que para um agente ser considerado inteligente o mesmo deve possuir as seguintes características:

- Ter habilidade para explorar e recuperar resultados significativos dentre do domínio do conhecimento;
- Ser tolerante a erros, entradas erradas ou inesperadas;
- Ter habilidade para usar abstrações e símbolos;
- Ter capacidade de se adaptar aos objetivos do ambiente;
- Ter habilidade para aprender com o ambiente.

3.1.4 Sistemas Multiagente

Sistemas multiagente são sistemas computacionais em que dois ou mais agentes interagem ou trabalham juntos para executar um conjunto de tarefas ou satisfazer um objetivo comum (LESSER, 1999). Estes sistemas emergiram de um subcampo da inteligência artificial (IA), chamado inteligência artificial distribuída (IAD). A IAD tem por objetivo prover princípios para a construção de sistemas complexos, envolvendo múltiplos agentes e mecanismos para a coordenação do comportamento destes agentes independentes (STONE *et al.*, 2000).

A IAD é uma das áreas da IA que mais se desenvolveram nos últimos anos e apresenta um enorme potencial de aplicações (BITTENCOURT, 1998). Ela está relacionada à solução cooperativa de problemas dentro de um certo ambiente por intermédio de agentes distribuídos (LESSER, 1992). Quando se fala em problema distribuído, deve-se ressaltar os dois tipos de problemas que a IAD trata: solução de

problemas distribuídos (SPD) e solução distribuída de problemas (SDP). A primeira trata de problemas que são inerentemente distribuídos, como por exemplo: controle de tráfego aéreo, distribuição de energia elétrica, etc. (BITTENCOURT, 1998). Já a segunda tem como ponto principal à questão da decomposição das tarefas em sub-tarefas para que o conjunto alcance a solução. Por exemplo, para satisfazer as restrições de um determinado problema, o mesmo pode ser decomposto em vários subproblemas, não inteiramente independentes, que são resolvidos por diferentes processos. Assim, as soluções dos subproblemas podem ser sintetizadas e transformadas na solução do problema original (STONE *et al.*, 2000).

Com o passar do tempo a IAD se dividiu tendo dois domínios principais: A solução distribuída de problemas (SDP) e os sistemas multiagente (SMA) (BITTENCOURT, 1998). A SDP tem como foco principal o problema, conforme a tradição na IA simbólica, da qual este enfoque é diretamente derivado. Seus objetivos são utilizar a capacidade de processamento e a robustez oferecida pela tecnologia de rede para atacar problemas naturalmente distribuídos ou excessivamente complexos. Para a SDP, os agentes são pré-programados para cooperar e seus métodos visam garantir que esta cooperação ocorra de maneira coerente, robusta e eficiente. Os agentes partilham um objetivo comum global, utilizando uma mesma linguagem e semântica de interação.

Já SMA tem como foco o agente, que coopera em um ambiente aberto onde cada um tem autonomia e pode, eventualmente, participar na solução de um dado problema. Estes agentes competem entre si por recursos, tendo assim que saber lidar com conflitos e coordenar suas atividades para aumentar a eficiência na solução do problema. Eles não necessitam utilizar uma mesma linguagem, o que implica na necessidade da tradução e mapeamento da informação para que haja comunicação com outros agentes. A seguir serão apresentadas algumas outras características importantes de um SMA:

- Cada agente tem informação incompleta, ou seja, apenas uma visão parcial do ambiente;
- Não há um sistema global de controle, portanto o controle é descentralizado;
- As informações estão distribuídas;
- O Processamento é assíncrono.

Como mencionado, agentes são utilizados para melhor resolver problemas complexos, e devem trabalhar cooperativamente com outros agentes em um ambiente heterogêneo (DELOACH, 1999). Em um SMA se deve coordenar o comportamento de um agente individualmente para prover um comportamento ao nível de sistema. SYCARA (1998) lista seis aspectos a serem considerados no projeto de um SMA:

- Como decompor o problema e alocar as tarefas aos agentes individuais;
- Como coordenar o controle e a comunicação entre os agentes;
- Como fazer com que múltiplos agentes atuem de uma maneira coerente;
- Como fazer com que um agente argumente sobre outro agente através do estabelecimento de um protocolo de comunicação;
- Como reconciliar objetivos conflitantes entre agentes coordenadores;
- Como criar sistemas multiagente práticos e eficientes.

3.1.5 Vantagens da Abordagem Multiagente

A razão mais importante para usar SMA quando se projeta um sistema, é o fato de alguns domínios de aplicações necessitá-lo. Isto acontece particularmente se houver diferentes pessoas ou organizações com diferentes, e possivelmente conflitantes, objetivos e informações proprietárias. Nestes casos, SMAs são uma boa alternativa para possibilitar suas interações (STONE *et al.*, 2000).

Mesmo em domínios onde não há a necessidade de se usar sistemas distribuídos, existem várias razões para utilizar SMAs. Tendo múltiplos agentes, pode-se aumentar a velocidade de operação do sistema provendo métodos de computação paralela. Por exemplo, um domínio que é facilmente dividido em componentes, várias tarefas independentes que podem ser executadas por agentes separados, pode ser beneficiado por SMAs.

Enquanto o paralelismo é alcançado associando diferentes tarefas ou problemas a agentes diferentes, a robustez é um benefício dos sistemas multiagente que pode ser obtida através da redundância de agentes. Se o controle e as responsabilidades forem compartilhados entre agentes diferentes, o sistema pode tolerar faltas usando um ou mais destes agentes (STONE *et al.*, 2000).

Outro benefício de sistemas multiagente é sua escalabilidade. Desde que ele seja inerentemente modular, é muito mais fácil adicionar um novo agente em um sistema multiagente que adicionar uma nova capacidade a um sistema monolítico. Sistemas nos quais seus parâmetros ou capacidades estão facilmente mudando com o passar do tempo podem também ser beneficiados por estas vantagens dos SMAs.

Em suma, as principais vantagens dos SMAs são:

- Tolerância à faltas;
- Solução de problemas complexos;
- Rapidez;
- Robustez;
- Paralelismo;
- Escalabilidade;
- Distribuição;
- Eficiência nos custos.

3.2 Workflow

O *Workflow* ou "fluxo de trabalho" se preocupa com a automação dos procedimentos, onde informações e tarefas são passadas entre participantes de acordo com um conjunto de regras pré-definidas para contribuir ou alcançar totalmente o objetivo global (OMG, 1998). Alguns processos *workflow* ainda podem ser encontrados sendo executados manualmente, entretanto, a maioria já foi transformada em sistemas que provêem um suporte computadorizado a estes processos.

Em termos gerais, o *workflow* pode ser definido como uma automação ou facilitação computadorizada de um processo de negócios, ao todo ou em partes (WFMC, 1995).

Portanto, *workflow* é um conjunto de atividades coordenadas que trabalham para alcançar um objetivo comum. Assim, um *workflow* separa as várias atividades de um dado processo organizacional em um conjunto de tarefas bem definidas. Uma atividade ou tarefa é um passo lógico ou descrição de uma parte do trabalho que contribui para a realização do processo. Tarefas que compõem o *workflow* são usualmente relacionadas e dependentes umas das outras. Estes relacionamentos são especificados por um conjunto de restrições de

execução chamado “dependências de tarefas”. Estas dependências de tarefas têm um papel chave no suporte às várias especificações de *workflow*, tais como concorrência, serialização, execução, compensação e etc. (ATLURI, 2001).

A idéia básica por trás da tecnologia *workflow* é separar os processos de negócios e os componentes de gerenciamento do *workflow* das aplicações, para assim aumentar a flexibilidade e o grau de manutenção dos sistemas (DU, 1997). *Workflow* é freqüentemente associado com a reengenharia de processo de negócio (RPN), que consiste da avaliação, análise, modelagem, definição e subseqüente implementação operacional do núcleo do processo de negócio de uma organização (WFMC, 1995).

O processo de reengenharia tem como proposta aumentar a produtividade, reduzir os custos e responder às mudanças do ambiente mais rapidamente (DU, 1997). Embora nem toda a atividade de RPN resulte em implementações, a tecnologia *workflow* é na maioria das vezes uma solução apropriada para prover a separação entre a lógica de procedimentos de negócios e seu suporte operacional, habilitando assim subseqüentes mudanças a serem incorporadas dentro das regras procedurais que definem o processo de negócio. Reciprocamente, nem toda a implementação de *workflow* necessariamente parte de um exercício de reengenharia, por exemplo, implementações para automatizar um processo de negócio já existente e consolidado (WFMC, 1995, OMG, 1998).

Workflow é uma tecnologia em crescente evolução que está sendo sucessivamente explorada por empresas que atuam em variados tipos de negócios. Sua característica principal é a automação de processos, envolvendo a combinação de atividades desenvolvidas por homens e máquinas, particularmente das que envolvem interações com TI. Embora seja mais usada dentro de ambientes de escritórios (tais como, seguradoras, bancos, administradoras, etc.), ele é também aplicável a algumas classes de processos industriais.

A *Workflow Management Coalition* (WfMC) foi fundada em agosto de 1993, com a missão é desenvolver e promover o uso do *workflow* para o estabelecimento de um padrão de software com terminologia, interoperabilidade e conectividade bem definidas. Um modelo de referência para *workflow* tem sido desenvolvido, para conceber uma estrutura para aplicações genéricas, através da identificação das interfaces necessárias para que os produtos implementados nessa especificação cooperem entre si em vários níveis

(MENG, 2002). Nas próximas seções este modelo de referência será exposto em maiores detalhes, para que se possa compreender melhor o seu funcionamento.

3.2.1 Sistema de Gerenciamento de *Workflow*

O sistema de gerenciamento de *workflow* (SGWF) provê a automação de um processo de negócio, através do gerenciamento da seqüência das atividades e a invocação apropriada de recursos humanos e computacionais, associados com os vários passos da atividade (OMG, 1998).

Por definição, o sistema de gerenciamento de *workflow* é um sistema que define, gerencia e executa atividades, por intermédio de representações computadorizadas da lógica do fluxo de trabalho das mesmas (WFMC, 1995).

Um SGWF consiste de componentes de software, utilizado para armazenar e interpretar definições de processos, criar e gerenciar instâncias de *workflow*, bem como controlar suas interações com os participantes e as aplicações. Tais sistemas também provêem funções de administração e supervisão, por exemplo, para permitir a realocação ou escalonamento do trabalho, adicionado informações de auditoria e gerenciamento sobre todo o sistema ou relativo a cada instância individual de processo (WFMC, 1996). Uma vantagem do sistema *workflow* é que ele suporta a alocação racional de recursos, e pode então se adaptar dinamicamente às mudanças na carga de trabalho. Outra grande vantagem é que ele simplifica o desenvolvimento das aplicações, não somente porque os componentes das aplicações podem ser reutilizados, mas também porque funções comuns para várias aplicações já estão à disposição dos desenvolvedores de sistemas, em especificações fornecidas pela WfMC (DU, 1997).

Nos últimos anos os SGWFs têm sido usados principalmente para prover automação dentro de ambientes organizacionais, especialmente para coordenar as atividades humanas, facilitar o gerenciamento dos dados e o roteamento dos documentos e relatórios (ZHONGQIAO, 2000).

A automação de um processo de negócio é definida dentro de uma *definição de processo*, que identifica as várias atividades do processo, regras procedurais e dados de controle usados para gerenciar o *workflow* durante o processo de representação (WFMC, 1996). Um processo de negócio pode ter um tempo de vida variando de alguns minutos a

vários dias, podendo inclusive chegar até mesmo há anos. Isso depende da sua complexidade e da duração das várias atividades constituintes.

Em um nível mais alto de abstração, todo SGWF pode ser caracterizado por prover suporte a três áreas funcionais (OMG, 1998):

- **Funções de configuração**: Compreende a definição e modelagem de processos de *workflow* e suas atividades constituintes;
- **Funções de Controle**: São responsáveis pelo gerenciamento dos processos de *workflow* no ambiente operacional, e também do escalonamento das várias atividades que fazem parte de cada processo;
- **Funções de interação com o usuário**: Estas interações são freqüentemente realizadas através do uso de interfaces de usuário como, por exemplo, formulários com campos para preenchimento.

3.2.2 Terminologia Básica

Esta seção contém definições técnicas para os termos usados pela *Workflow Management Coalition* em suas especificações. Serão identificadas as terminologias usadas para descrever os conceitos e as estruturas gerais dos sistemas de gerenciamento de *workflow* (WFMC, 1996). A Figura 7 (WFMC, 1996) mostra estas definições e seus relacionamentos.

Processos de negócios: Um conjunto de uma ou mais atividades ou procedimentos interligados, que coletivamente percebe uma boa oportunidade de negócio. Isso normalmente acontece dentro do contexto de uma estrutura organizacional que define regras funcionais e de relacionamentos.

Definição de processo: É a representação de um processo de negócios em uma forma que suporte a manipulação automatizada, tal como modelagem ou representação por um sistema de gerenciamento de *workflow*. A definição de processo, consiste de uma rede de atividades e seus relacionamentos, critérios para indicar o início e fim do processo, e informações sobre as atividades individuais, tais como: participantes, dados e tecnologias associadas.

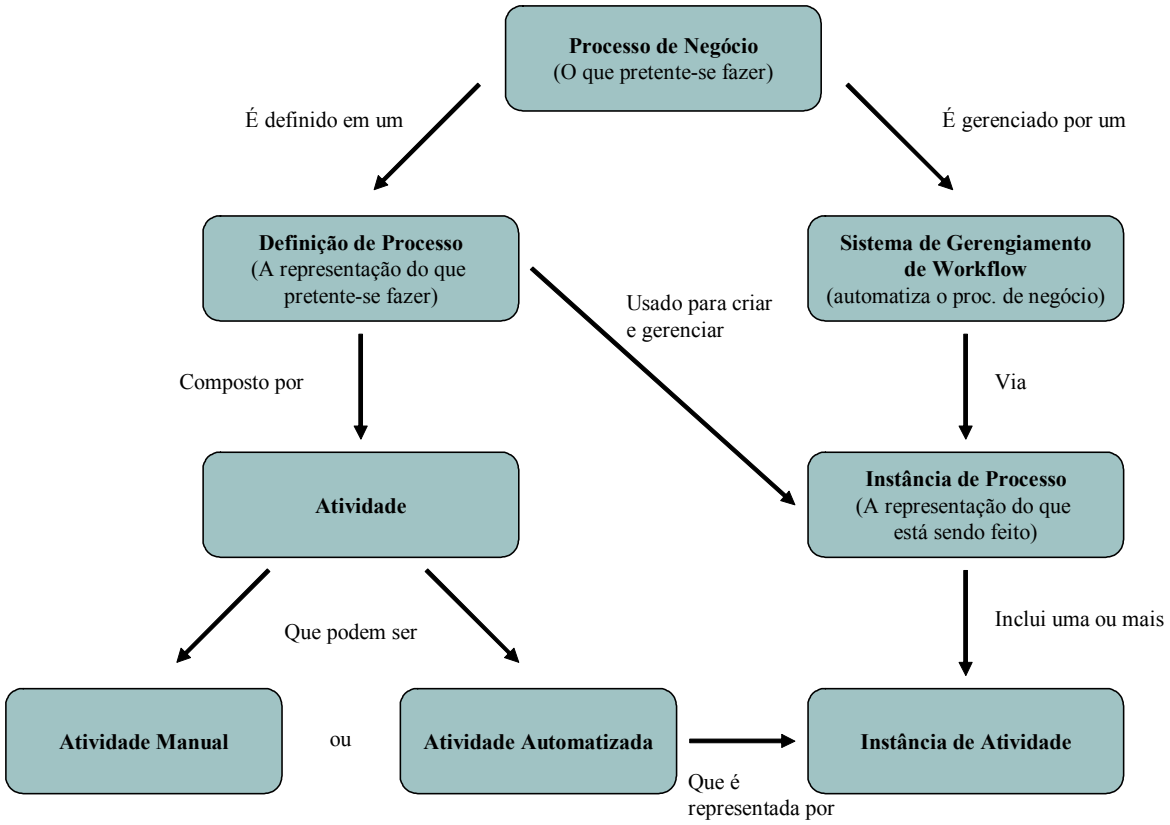


Figura 7 – Relacionamento entre as terminologias de *workflow*.

Atividade: É a descrição de uma parte do trabalho que forma um passo lógico dentro do processo. Uma atividade pode ser manual (neste caso não suporta automação computacional) ou uma atividade *workflow* (automatizada). Uma atividade *workflow* requer recursos de homem e máquina para suportar a execução do processo. Uma atividade é, na maioria dos casos, a menor unidade de trabalho que é escalonada por uma máquina *workflow* (ver seção 3.2.4) durante o processo.

Atividade Automatizada: É uma atividade computacional capaz de usar um SGWF para gerenciar as atividades durante a execução do processo de negócio. Uma atividade automatizada pode ser ativada diretamente pelo sistema *workflow* ou por intermédio de um dos participantes.

Atividade Manual: É uma atividade dentro do processo de negócio que não é capaz de ser automatizada, e conseqüentemente fica fora do escopo de um SGWF. Tal atividade pode ser incluída dentro da definição do processo, por exemplo, para suportar a modelagem do processo, mas não faz parte do resultado do *workflow*.

Instância: É a representação de um processo ou atividade dentro do sistema de *workflow*, incluindo seus dados associados. Cada instância representa uma nova execução

separada de uma entidade de processo ou atividade, que pode ser independentemente controlada e terá seu próprio estado interno visível externamente. Pode ser utilizada como um manipulador, por exemplo, para gravar e recuperar dados de auditoria relacionados a sua representação individual. Cada instância de processo ou atividade é criada e gerenciada individualmente pelo SGWF, assim, seus estados e informações relevantes são preservados.

3.2.3 Modelo de Referência de *Workflow*

O modelo de referência de *workflow* tem sido desenvolvido na forma de uma estrutura genérica, identificando suas interfaces dentro da estrutura que habilitará os produtos a interoperarem em vários níveis. Todo sistema *workflow* contém um número de componentes genérico, que interagem entre si através de um conjunto definido de maneiras diferentes. Estes produtos tipicamente exibem diferentes níveis de capacidade dentro de cada um desses componentes genéricos (WFMC, 1995).

Levando em consideração os aspectos apresentados acima, podemos definir modelo de referência de *workflow* como sendo uma arquitetura de representação de um sistema de gerenciamento de *workflow*, que identifica as interfaces mais importantes do sistema (WFMC, 1996).

Para alcançar a interoperabilidade entre os produtos de *workflow*, um conjunto padronizado de interfaces e de formatos de intercâmbio de dados entre estes componentes é necessário (WFMC, 1995). O modelo de referência identifica cinco áreas funcionais que são apresentadas abaixo e podem ser visualizadas na Figura 8 (OMG, 1998).

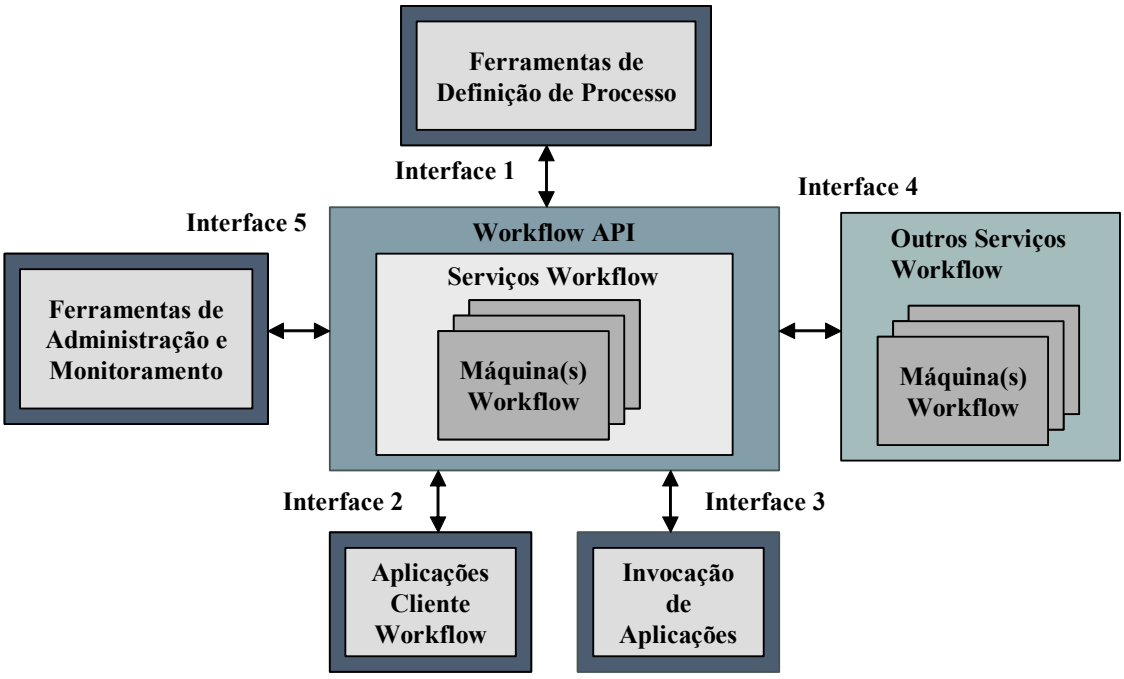


Figura 8 – Modelo de referência de *workflow*.

- **Definição de Processo (interface 1)**: Interface para a definição dos dados do processo e de sua inter-relação com o ambiente de execução do *workflow*;
- **Aplicações *Workflow* cliente (interface 2)**: Interface para suportar interações com interfaces de usuário;
- **Invocação de Aplicações (interface 3)**: Interface para suportar interações com vários tipos de aplicações;
- **Interoperabilidade do *Workflow* (interface 4)**: Interface para suportar a interoperabilidade entre diferentes sistemas de *workflow*;
- **Administração e Monitoramento (interface 5)**: Interface para prover ao sistema a capacidade de monitoramento e métricas para facilitar o gerenciamento do ambiente de aplicações *workflow*.

A interface em torno dos serviços de *workflow* é designada WAPI[12], e pode ser considerada como sendo um conjunto de construções pela qual os serviços do sistema *workflow* podem ser acessados. A WAPI regula as interações entre o software de controle do *workflow* e outros componentes do sistema.

[12] WAPI – *Workflow API and Interchange* é apresentada na seção 3.2.5.

3.2.4 Serviços *Workflow*

Os serviços *workflow* provêem um ambiente de execução em que ocorre instanciação e ativação de processos, utilizando uma ou mais máquinas *workflow*. Estas máquinas são responsáveis por interpretar e ativar parcial ou totalmente as definições de processo, e interagir com os recursos externos necessários para processar as várias atividades (WFMC, 1995).

Uma máquina *workflow* (*workflow engine*) é responsável pelo controle do ambiente de execução dentro do sistema *workflow*. Por definição, uma máquina *workflow* é um software ou "máquina" que provê o ambiente de execução para uma instância de *workflow*. Tipicamente essa máquina tem as seguintes características:

- Interoperação com a definição de processos;
- Controle de instâncias de processos – criação, ativação, suspensão e término;
- Navegação entre processos ativos, que podem envolver operações seqüenciais ou paralelas, escalonamento, *deadlines*, interoperação de dados relevantes do *workflow*;
- Identificação da lista de trabalho para cada usuário e uma interface para suporte as interações do usuário;
- Manutenção do controle do fluxo de dados, passando dados relevantes do usuário para a aplicação ou vice-versa;
- Uma interface para invocar aplicações externas e ligar outros fluxos de dados relevantes;
- Ações de supervisão para controle, administração e auditoria.

3.2.5 WAPI

WAPI é a abreviação do formato chamado ***W**orkflow **API** and **I**nterchange*, que é um formato de protocolo para interação e acesso que foi publicado pela WfMC. Este protocolo incorpora especificações que facilitam a interoperabilidade entre os diferentes componentes do sistema *workflow* e as aplicações (WFMC, 1996).

O WAPI é encarado como um conjunto comum de chamadas de APIs[13] e de formatos de intercâmbio de dados relacionados, que podem ser agrupados para suportar cada uma das cinco áreas funcionais de interface do modelo de referência (WFMC, 1995). Em suas especificações concebidas pela WfMC a WAPI inclui:

- Um conjunto de chamadas de API para suportar as funções entre a máquina de *workflow* e as aplicações ou entre outros componentes do sistema;
- Formatos para intercâmbio e protocolos para suportar a interoperabilidade entre diferentes máquinas de *workflow*;
- Formatos para troca de informações, tais como, definição de processo e auditoria de dados entre uma máquina *workflow* e outros repositórios externos.

As operações que já foram identificadas sobre as cinco interfaces funcionais do modelo de referência são mostradas abaixo e estão divididas em dois grupos:

Chamadas de API

- Estabelecimento de seção;
- Operações de definição de *workflow* e de seus objetos;
- Funções de controle de processo;
- Funções supervisórias de controle de processo;
- Funções de estado de processo;
- Funções de gerenciamento de atividades;
- Operações de gerenciamento de usuários;
- Operações de gerenciamento de papéis;
- Operações de gerenciamento de auditoria;
- Operações de controle de recursos.

Funções de intercâmbio de dados

- Protocolo para transferência de definições de processos;
- Protocolo para transferência de dados relevantes do *workflow*.

[13] API – *Application Programming Interface*.

3.2.6 WPDL

A WPDL, que significa *Workflow Process Definition Language*, é definida como sendo uma interface comum para intercâmbio de definições de processos de *workflow*. Esta interface é uma linguagem padrão que pode ser suportada por vários fabricantes de produtos *workflow*, e permite a troca de documentos de definição de processo (WFMC, 1999).

Uma variedade de diferentes ferramentas pode ser usada para analisar, modelar, descrever e documentar um processo de negócios. A interface de definição de processos define um formato comum de intercâmbio que suporta a transferência de definições de processos entre produtos separados (GUIMARÃES, 1997). A interface também define uma separação formal entre a tarefa de desenvolvimento e o ambiente de execução, possibilitando que uma definição de processo gerada por uma ferramenta de modelagem seja usada como entrada em um grande número de produtos *workflow* diferentes. Isto confere com a necessidade freqüente dos usuários na questão da independência entre a modelagem e os produtos *workflow* (DAS, 1997).

Para prover um método comum para acessar e descrever definições *workflow*, um modelo de metadados para definição de processos foi estabelecido. Este modelo de metadados identifica entidades comumente usadas dentro da definição de processo. Uma variedade de atributos descreve as características deste conjunto limitado de entidades. Baseados neste modelo, ferramentas específicas de cada fabricante podem transferir modelos via um formato comum de intercâmbio de dados (WFMC, 1999). Na Figura 9 pode ser visto um exemplo escrito em WPDL, e abaixo é apresentada uma breve descrição dos principais elementos de sua sintaxe:

- ***WORKFLOW***: É o elemento que define o início da descrição de um processo *workflow*;
- **ACTIVITY**: Descreve a lista de atividades realizadas pelo *workflow*, além de definir seus relacionamentos;
- **PARTICIPANT**: Representa os elementos que são parte ativa de um processo *workflow*;
- **APPLICATION**: Declara a lista de aplicações ou ferramentas necessárias, invocadas pelas atividades definidas no WPDL;

- **DATA**: Representa os tipos de dados definidos pelo usuário que são utilizados pelas atividades do *workflow*.

```
WORKFLOW              "At the Sales Department"
   WPDL_VERSION       1.0
   VENDOR             Vendor:Product:Release
   CREATED            1995-12-06

   ACTIVITY           "act01"
      NAME            "Order Status"
      DESCRIPTION     "Verify order status"
      LIMIT           10
      ROTE
   END_ACTIVITY

   PARTICIPANT        "Tim White"
      TYPE            HUMAN
      USERID          "tw456"
      SURNAME         "White"
      FORENAME        "Tim"
      DESCRIPTION     "Mail Room Clerk"
   END_PARTICIPANT

   APPLICATION        scan_document
      TOOLNAME        winscan.exe
      OUT_PARAMETERS  scanned_document
   END_APPLICATION

   DATA               document_type
      TYPE            string
      DEFAULT_VALUE   "Sales Order"
   END_DATA
END_WORKFLOW
```

Figura 9 – Exemplo escrito em WPDL.

3.3 XML

Extensible Markup Language (XML) é uma linguagem simples e flexível criada em 1998, derivada da linguagem SGML[14]. O SGML foi desenvolvido há vinte anos e surgiu como uma forma de estruturar documentos em um formato padrão para facilitar a troca e manipulação de dados. O padrão XML foi criado e é mantido até hoje pela W3C[15]. XML é uma especificação para projeto de linguagens de marcadores, ou seja, XML é uma metalinguagem. Por ser um subconjunto simplificado do SGML, ela incorpora algumas

[14] SGML – *Standard Generalized Markup Language*.
[15] W3C – *World Wide Web Consortium*.

características desta linguagem, de onde se destacam as três mais importantes: extensibilidade, estruturação e validação (MORRISON *et al*., 2000).

O XML foi criado para suprir as deficiências do HTML[16]. O HTML também é uma linguagem de marcadores derivada do SGML, concebido para codificar informações a serem apresentadas em *browsers web*. Entretanto, o HTML foi originalmente projetado com o intuito de exibir apenas informações estáticas. Desta forma, seu conjunto de identificadores, também chamados de *tags*, é fixo e limitado. Assim, o XML teve como primeiro objetivo introduzir flexibilidade e robustez ao HTML, adicionar contexto e dar maior estruturação às informações na *web*. É importante salientar que o XML não veio substituir o HTML, o XML é utilizado para estruturar e descrever os dados, enquanto o HMTL é usado para formatar e exibir estes mesmos dados. Atualmente, o XML tem um objetivo ainda maior do que aquele para o qual foi concebido originalmente. Por ser uma metalinguagem ele pode ser usado para diversos fins, tais como:

- Manter os dados separados do HTML;
- Especificar protocolos de comunicação;
- Formatar os dados em arquivos ou bancos de dados.

O XML é uma tecnologia valiosa. Documentos bem projetados em XML podem prover pesquisas mais precisas e robustas, pelo fato da informação estar bem contextualizada e estruturada. O XML pode auxiliar o HTML nas situações em que o segundo é insuficiente para atender as necessidades. Além disso, softwares que utilizam XML são mais fáceis de serem desenvolvidos quando comparados a softwares que usam SGML, e são mais apropriados para o ambiente *web* (USDIN *et al*., 1998).

A filosofia XML está baseada em quatro princípios fundamentais (USDIN *et al.*, 1998):

Separação entre o contexto e o formato: O conceito mais importante por trás do XML é a separação da informação da forma como ela é apresentada ou processada. Informações devem ser identificadas da forma mais abstrata possível. Assim, as informações podem ser usadas de modo diferente por diferentes aplicações, e serem

[16] HTML – *Hypertext Markup Language*.

processadas através de métodos que não tenham sido calculados no momento em que os dados foram criados.

Estrutura de dados hierárquica: O XML assume que os dados são estruturados hierarquicamente, ou seja, vários tipos de informações úteis e identificáveis contêm outras úteis e identificáveis informações. Um tipo de informação em XML é chamado de elemento. Dados XML são empacotados dentro de "instâncias de documentos". Cada instância de documento consiste de um elemento chamado "*root*", que indica o início do documento e contém todos os outros elementos. É comum serem encontrados elementos dentro do elemento *root* contendo outros elementos, e assim por diante. Uma estrutura hierárquica é muito comum na escrita de documentos texto; uma seção sempre inicia antes do cabeçalho desta seção e termina logo antes do cabeçalho da próxima seção. Uma instância de documento XML tem basicamente três princípios: cada instância tem um e somente um *root*, não há interseção entre seus ramos, e a estrutura do documento não possui laços.

***Tags* embutidas**: Em XML as *tags* estão embutidas nos dados para identificar onde cada elemento inicia e termina. *Tags* são palavras incluídas dentro dos símbolos de menor e maior, semelhante as *tags* usadas em HTML. Muito da flexibilidade do XML vem do fato dos elementos de informação serem identificados através de *tags* embutidas nos dados. Isto significa que não há a necessidade de especificar de antemão qual o tamanho e conteúdo, ou precisamente qual a seqüência em que o dado irá ocorrer. Assim, aplicações podem tolerar uma considerável variação nos dados que recebem por estarem aptas a identificar as informações de seu interesse.

Estrutura definida pelo usuário: O XML define um modo de criar *tags* para identificar informações que são de interesse de um usuário ou um grupo de usuários. XML não é um conjunto de *tags* pré-definido; assume-se que o usuário criará novas *tags* à medida que ele cria e trabalha com documentos. Talvez a maior surpresa sobre XML, especialmente para usuários de HTML, seja o fato dele não prover nem recomendar *tags*. Isto é uma premissa fundamental do XML, pois não é possível listar todos os tipos de informações que qualquer grupo de usuários possa querer identificar. Assim, o XML provê mecanismos para o usuário criar suas próprias *tags* para as informações de seu interesse.

3.3.1 Sintaxe XML

O XML tem sua sintaxe muito parecida com a sintaxe HTML justamente pelo fato de ambas terem sido derivadas do SGML. Basicamente, um documento XML é constituído por entidades que podem conter dados analisáveis (*parsed*) ou não (*unparsed*) gramaticalmente. Dados analisáveis consistem de caracteres que são considerados "dados de caracteres" ou "marcadores" e são processados por um "processador XML"[17]. Já dados não analisáveis são vistos como um texto "bruto" e não são processados.

A sintaxe XML descreve essencialmente as construções usadas para definir a estrutura e a forma do documento, bem como as restrições envolvidas (MORRISON *et al*., 2000).

Os documentos XML são projetados para serem processados por processadores XML. Um processador XML é um módulo de software que lê um documento XML e provê acesso a seu conteúdo e estrutura. Tipicamente, um processador XML processa um documento XML em nome de uma aplicação de propósito específico. Em outras palavras, uma aplicação XML emprega um processador XML para ter acesso ao conteúdo do documento.

A estrutura que dá acesso ao conteúdo do documento, que é gerada pelo processador XML, tem o nome de DOM (*Document Object Model*). O DOM provê acesso às informações armazenadas em documentos XML como um modelo de objetos hierárquico. Ele cria uma árvore de nós baseado na estrutura e informação do documento XML. Utilizando esta estrutura se pode acessar as informações do documento através da interação com os nós da árvore (IDRIS, 1999). Na Figura 10 pode ser visto um exemplo de documento XML e seu respectivo DOM.

[17] Processador XML é a tradução da expressão inglesa XML processor.

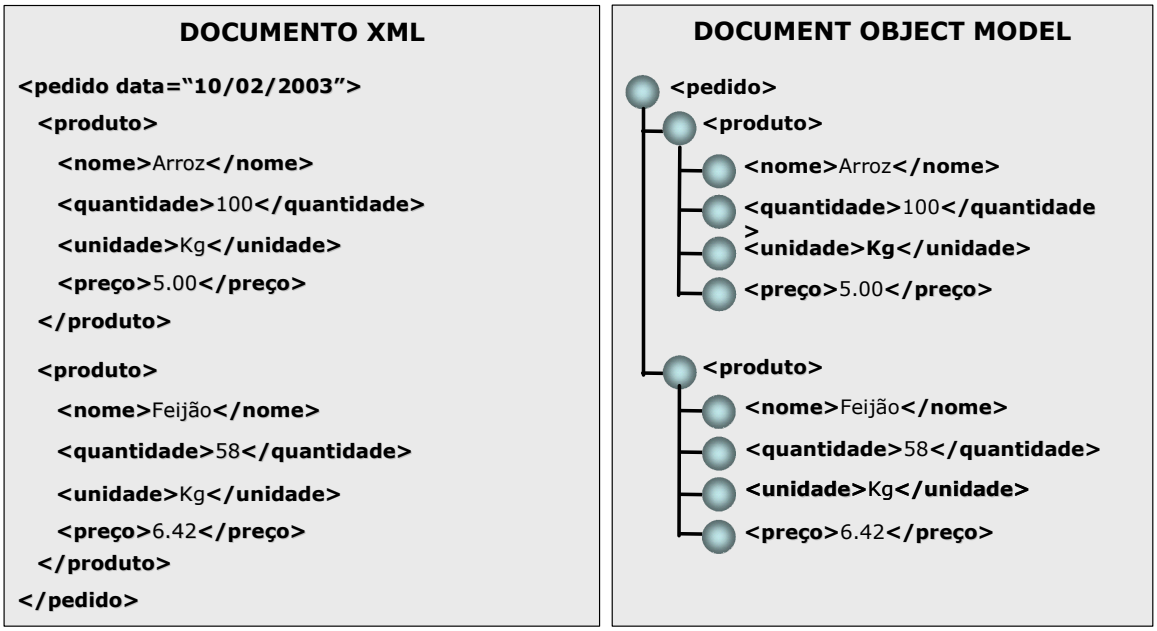


Figura 10 – Exemplo XML e DOM.

Como mencionado anteriormente, um documento XML consiste de caracteres de dados e marcadores. Entretanto, não são apenas estes dois componentes que formam os documentos XML. A seguir são apresentados os diferentes componentes suportados pelo XML versão 1.0 (MORRISON *et al*., 2000), ilustrados através da Figura 11.

Elementos *tags*: As *tags* são os componentes mais importantes na sintaxe XML e são usadas para descrever elementos. Por exemplo, na Figura 11 linha 12 o elemento cidade é compreendido pelas *tags* <*cidade*> e </*cidade*>. Uma *tag* referencia um texto *string* específico usado para representar um elemento em um documento XML. Documentos XML podem conter elementos vazios, o que significa que certos elementos não contêm dados caracteres analisáveis. Para identificar um elemento com sendo vazio, deve-se adicionar uma barra (/) ao final do nome da *tag* em questão. Por exemplo, a *tag* cidade se estivesse vazia seria representada assim <*cidade/*>.

Referências de entidades: Referências de entidades são utilizadas em XML para associar um pseudônimo a uma parte de um dado. Basicamente, uma referência de entidade serve como um nome único para uma parte do dado XML. No exemplo da Figura 11 linha 21, uma referência de entidade é usada para representar um apostrofe (‘) (*Silva's*), e uma outra é usada para representar um “e comercial” (&) (*Shoes & Glasses*) como sendo dados analisáveis.

```
1 <?xml version="1.0"?>
2 <!DOCTYPE agenda SYSTEM "Agenda.dtd" [
3 <!ENTITY amp "&#38;#38;">
4 <!ENTITY apos "&#39;">
5 ]>
6
7   <agenda>
8   <!-- Meu amigo Eduardo. -->
9      <contato numero="15">
10        <nome>Eduardo da Silva</nome>
11        <endereco>Rua Tiradentes</endereco>
12        <cidade>Rio de Janeiro</cidade>
13        <estado>Rio de Janeiro</estado>
14        <cep>88532-600</cep>
15        <fone>
16           <residencial>21-555-1818</residencial>
17           <fax>21-555-1819</fax>
18        </fone>
19        <email>edu@fruit.com</email>
20        <web>http://www.fruit.com/edu</web>
21        <companhia>Silva&amp;apos;s Shoes &amp;amp; Glasses</companhia>
22     </contato>
23  </agenda>
```

Figura 11 – Exemplo de um documento XML completo.

Comentários: Comentários são usados em documentos XML para apresentar informações que não são tecnicamente parte do conteúdo do documento. Tais como os comentários em linguagem de programação, os comentários em documentos XML são usados puramente para prover descrições aos dados do documento, e geralmente são ignorados pelos processadores XML. Os comentários são iniciados com (<*!*--) e terminados com (-->). A única limitação em comentários é o fato de não se poder incluir dois hífens consecutivamente (--), pois ocorrem conflitos com a sintaxe de comentários do XML. A Figura 11 mostra um exemplo deste tipo de construção na linha 8.

Instruções de processamento: As instruções de processamento são instruções especiais usadas pela aplicação que processa o documento XML. Estas instruções sempre iniciam com um sinal de "menor-que" e uma "interrogação" (<*?*), e terminam com um símbolo de "interrogação" e um "maior-que" (*?*>). O exemplo mais comum de instrução de processamento é a versão da especificação na qual o documento foi gerado. Este tipo de construção pode ser visto na Figura 11 linha 1.

Declaração do tipo de documento: Declarações do tipo de documento são usadas em XML para especificar informações sobre o documento, incluindo o elemento raiz (*root*) e a definição do tipo do documento (DTD). A declaração do tipo do documento é

importante para estabelecer se o documento é bem formado[18] ou válido[19]. As três principais tarefas executadas pela declaração do tipo do documento são: especificar o elemento raiz do documento, definir os elementos, atributos e entidades específicas do documento (DTD interno) e identificar o DTD externo para o documento, se for o caso. Na Figura 11 a linha 2 mostra um exemplo de declaração de tipo de documento.

Seção CDATA: Seções CDATA são usadas em documentos XML para marcar blocos de texto que devem ser ignorados pelo *parser* XML[20]. Os delimitadores de abertura e fechamento da seção são, respectivamente, (*<![CDATA[*) e (*]]>*). A única seqüência de caracteres que não pode aparecer em uma seção *CDATA* é (*]]>*). As seções CDATA são úteis quando se deseja que todos os caracteres de um texto sejam interpretados como caracteres e não como elementos de marcação. Exemplos são textos contendo os caracteres <, >, &, etc., comuns em trechos de código de programas ou em textos descrevendo XML. Exemplo: *<![CDATA[if A > B then MAIOR = A else MAIOR = B;]]>*.

3.3.2 Definição do Tipo do Documento (DTD)

Um documento XML é basicamente uma estrutura para armazenar informações. Para avaliar a validade de um documento XML, é necessário estabelecer exatamente qual a estrutura em que a informação dentro do documento deve estar. Isto é realizado através do chamado DTD, que é um dos modelos usados para descrever a estrutura da informação em um documento XML. Um DTD descreve o arranjo dos marcadores e dados caracteres em um documento válido. O termo válido é importante porque um documento pode perfeitamente ser criado sem um DTD, neste caso o documento é bem formado, mas não válido. Assim, para um documento ser válido ele deve ser bem formado e seguir as especificações de um DTD (MORRISON *et al*., 2000).

Um DTD descreve um vocabulário XML específico, definindo claramente os relacionamentos entre os elementos deste vocabulário. Sem o DTD o XML fica sub-utilizado, pois se torna apenas uma metalinguagem, o que na maioria dos casos é insuficiente para as aplicações em geral. O DTD possibilita criar vocabulários específicos em que os documentos devem seguir sua sintaxe para ser considerados válidos. Um

[18] Um documento bem formado é aquele que está dentro das regras básicas da sintaxe XML.
[19] Um documento válido é um documento bem formado e de acordo com a estrutura definida por um DTD.

documento XML criado por um vocabulário específico, é conferido e validado através da sua comparação com o DTD que especifica seu vocabulário. Um DTD estabelece restrições na estrutura do conteúdo do documento. Isto pode acontecer de duas formas:

- Estabelecendo o modelo do conteúdo para documentos, que define a ordem e o aninhamento dos elementos;
- Identificando os tipos dos dados do documento.

Em um DTD são definidos todos os aspectos de como um novo conjunto de *tags* pode ser usado. Também se define que elementos estão disponíveis, em que seqüência eles devem aparecer, com que freqüência eles podem ser usados, a possibilidade de aninhamento da estrutura dos elementos, e os atributos que cada um dos elementos pode conter.

Os componentes básicos descritos em um DTD são os elementos e os atributos. Estes componentes são responsáveis pelo estabelecimento da estrutura lógica do documento XML. Um elemento pode ser visto com sendo a unidade lógica da informação, enquanto um atributo é a característica desta informação. Portanto, um elemento representa um objeto de informação, enquanto um atributo representa uma propriedade deste objeto. A seguir serão descritas estas estruturas básicas do DTD.

Declaração de Elementos

Um elemento para ser usado em um documento XML válido deve ser declarado em um DTD. Os elementos são declarados no DTD da seguinte forma: <!ELEMENT *ElementName Type*>. O nome do elemento determina o nome da *tag* usada para marcar o elemento no documento XML, e corresponde ao *ElementName* na declaração de elemento. O tipo do elemento é especificado em *Type*, que também é conhecido como o conteúdo específico do elemento. O XML suporta quatro tipos de elementos:

- **Vazio** – O elemento não detém nenhum conteúdo, mas pode conter atributos;
- **Somente Elemento** – O elemento só contém elementos filhos;

[20] O parser XML é implementado dentro do processador XML. Ele recebe um documento XML e verifica se ele é bem formado ou válido. Se o documento for válido o parser gera o *Document Object Model* (DOM).

- **Misto** – O elemento contém uma combinação de elementos filhos e dados caracteres;
- **Qualquer** – O elemento contém qualquer conteúdo permitido pelo DTD.

Exemplos de declarações de elementos podem ser vistos na Figura 12 em todas as linhas que começam com a palavra *<!ELEMENT*.

Declaração de Atributos

Como mencionado, atributos são usados para especificar informações adicionais sobre um elemento. Dentro de um elemento, atributos são usados para formar um par nome/valor que descreve uma propriedade particular de um elemento. Atributos são declarados no DTD através da lista de declaração de atributos, que tem a seguinte forma: <!ATTLIST *ElementName AttrName AttrType Default*>. Um atributo tem um nome (*AttrName*) e um tipo (*AttrType*), bem como um valor padrão (*Default*). O valor padrão para um atributo referencia um valor ou um símbolo que indica a utilização do atributo. Existem quatro tipos diferentes de valores padrão para o campo *Default*:

- *#REQUIRED* – Quando o atributo é requerido;
- *#IMPLIED* – Quando o atributo é opcional;
- *#FIXED value* – Quando o atributo é um valor fixo;
- *Default* – Quanto existe um valor predefinido para o atributo.

Um exemplo de declaração de atributo pode ser vistos na Figura 12 linha 6.

```
1 <!ELEMENT agenda (contato)+>
2
3 <!ELEMENT contato (nome, endereco+, cidade, estado,
4                          cep, fone, email, web, companhia)>
5
6   <!ATTLIST contato numero CDATA #REQUIRED>
7   <!ELEMENT nome (#PCDATA)>
8   <!ELEMENT endereco (#PCDATA)>
9   <!ELEMENT cidade (#PCDATA)>
10  <!ELEMENT estado (#PCDATA)>
11  <!ELEMENT cep (#PCDATA)>
12  <!ELEMENT fone (residencial, fax?)>
13      <!ELEMENT residencial (#PCDATA)>
14      <!ELEMENT fax (#PCDATA)>
15  <!ELEMENT email (#PCDATA)>
16  <!ELEMENT web (#PCDATA)>
17  <!ELEMENT companhia (#PCDATA)>
```

Figura 12 – DTD do exemplo da Figura 11.

3.3.3 Vantagens e desvantagens no uso de XML

Para melhor determinar as implicações positivas e negativas do uso do XML para resolver necessidades particulares, esta seção traz uma série de pontos a favor e contra o uso do mesmo (SCHMELZER, 2001, ARMSTRONG, 2001, HAROLD *et al*., 2002).

Vantagens

XML é um formato de texto estruturado: O XML é um formato de documento estruturado que não representa apenas as informações trocadas, mas também os metadados que encapsulam seu significado e a estrutura da informação. Por exemplo, informações sobre um livro contêm dados sobre o título, autor, capítulos, corpo do texto e índice. Estas informações são estruturadas de uma forma que uma pessoa ou software possa entender.

XML é uma tecnologia acessível e barata: A primeira dificuldade enfrentada utilizando arquivos formatados, é que ferramentas para seu processamento são específicas, proprietárias e caras. Quando ferramentas são concebidas, elas são usualmente específicas para um formato particular de arquivo em questão. Uma das grandes forças do XML é que suas ferramentas de processamento se tornaram relativamente difundidas e baratas, quando não totalmente gratuitas.

XML é legível para humanos: Outro benefício do XML é a facilidade de ser lido e escrito por seres humanos, ao contrário de ser criado por aplicações em formato legível apenas por softwares proprietários.

XML é altamente flexível: O XML é uma linguagem extensível, ou seja, esta característica habilita a linguagem a ser usada para definir vocabulários específicos e metadados, ao contrário de ser fixa e descrever um conjunto particular de dados. O XML em conjunto com os seus DTDs definem documentos que juntos formam sua própria linguagem.

XML separa processos do conteúdo: O XML representa as informações e os metadados sobre a informação, ou seja, não especifica qualquer maneira em particular de como os dados devam ser processados, ou provê alguma restrição para os mecanismos que executam estas informações. Este é o contraste do XML com os outros formatos, tais como, arquivos de texto e bancos de dados, que explicitamente informam a maneira específica para acessar o documento.

XML é um padrão aberto: Enquanto certos tipos de formatos de arquivos e bancos de dados são proprietários, e seu desenvolvimento é controlado sobre uma única companhia. O XML é um produto de padrão aberto, que continua a ser desenvolvido com tal. Além disso, vários vocabulários e linguagens derivadas do XML estão também sendo desenvolvidos como padrão aberto.

XML não tem licença, é neutro de plataforma e largamente suportado: O XML não é uma tecnologia com um único dono ou uma licença comercial. Assim, ele pode ser livremente implementado em diversas aplicações dentro das organizações, sem a necessidade de gastos com licenças. Devido à separação entre o processamento e o conteúdo, o XML é também um bom exemplo de um formato de dados neutro de plataforma.

Desvantagens

Um Processo XML é espaçoso e monopoliza a largura de banda: O XML toma um grande espaço para representar dados que poderiam ser modelados usando um formato binário ou um arquivo de texto em formato simplificado. A razão para isto é simples, este é o preço que se paga pela: legibilidade humana, neutrabilidade de plataforma, separação do processamento, estruturação e validação do código.

XML é somente uma sintaxe de documento: O XML não é uma linguagem de programação, portanto, para manipular informações contidas dentro de documentos XML, se faz necessária a utilização de alguma linguagem de programação que possibilite esse

acesso. O XML é apenas um formato de documento que tem características que o torna adequado ao envio de informações estruturadas contendo metadados fáceis de ser validados.

XML é ótimo para texto, mas inadequado para dados binários: O XML é derivado do SGML e compartilha várias características do HTML. Estas especificações fornecem um excelente formato de documento texto baseado em marcadores, entretanto, XML é uma linguagem inadequada para se representar informações binárias ou de códigos de máquina.

XML processa arquivos: XML não utiliza os eficientes sistemas de banco de dados e sim o processamento ineficiente de arquivos. Existem argumentos que comprovam esta afirmação: Bancos de dados relacionais provêem maior eficiência no armazenamento e representação dos dados que estruturas hierárquicas. XML é uma regressão dos eficientes mecanismos de armazenamento dados para o processamento baseado em arquivos.

3.3.4 ebXML

Electronic Business Extensible Markup Language (ebXML) é uma iniciativa internacional estabelecida entre a *United Nations Center for Trade Facilitation and Electronic Business* (UN/CEFACT) e a *Organization for the Advancement of Structured Information Standards* (OASIS) iniciada em setembro de 1999. O propósito da iniciativa ebXML é pesquisar e identificar as bases técnicas em que a implementação global do XML possa ser padronizada. O seu objetivo é prover um modelo de referência aberto, baseado em XML, para ser utilizado de maneira uniforme e consistente na troca de dados de negócios eletrônicos entre aplicações, criando assim um único mercado eletrônico global (EBXML, 2001).

Apesar de não se utilizar ebXML neste trabalho, optou-se por comentá-lo dada à importância que o mesmo terá à medida que os primeiros resultados aplicáveis de sua iniciativa começarem a ser utilizados por empresas de desenvolvimento de software.

O ebXML não define simplesmente uma gramática e vocabulário XML, ele define também uma nova forma de pensar sobre negócios (IBM, 2001). Ele é baseado em padrões internacionais e tem como meta se transformar em um padrão. O ponto principal da iniciativa ebXML é a utilização do padrão XML especificado pela W3C como alicerce para sua construção. Embora essa especificação possa não prover uma ótima solução

técnica em alguns casos, a aceitação do ebXML pela comunidade de negócios e pela comunidade técnica está sendo maior que as expectativas.

Através do ebXML, pode-se definir como as companhias conduzem os negócios usando um vocabulário próprio na especificação de processos de negócios. As mensagens são enviadas utilizando formatos e protocolos padrões, e todas as informações são armazenadas em registros ebXML, o que facilita sua localização.

O escopo do ebXML é dirigido a praticamente todos os setores da comunidade de negócios, desde conglomerados internacionais até pequenas e médias empresas engajadas em comércio *business-to-business* ou *business-to-consumer*. Com esse público em mente, o ebXML se propõem a desenvolver e distribuir uma especificação que será usada por todos os parceiros de negócios, interessados em maximizar interoperabilidade usando XML dentro da comunidade de negócios (UN/CEFACT, 2002). Semelhante iniciativa tem sido lavada a cabo por instituições / empresas privadas, tal como a organização Rosettanet. Rosettanet é um consórcio formado por mais de 400 empresas de alta tecnologia, que estão trabalhando para criar, implementar e prover padrões abertos de processos para *e-business* (www.rosettanet.org).

Portanto, o ebXML vem a padronizar também a parte semântica das informações, particularizando cada tipo de transação (processo de negócio) para os diferentes setores / áreas empresariais. A idéia é análoga aos esforços efetuados pela comunidade de desenvolvimento de produtos quando da criação da norma genérica ISO 10303 (STEP), particularizável para cada setor empresarial através dos chamados protocolos de aplicação, e que atualmente estão passando a ser representados em XMI. O XMI é uma linguagem baseada em XML utilizada para representação de modelos UML (OMG, 2003).

O objetivo da especificação ebXML é definir os processos de negócios (*Business Processes*) e as informações de negócios (*Business Information*) (IBM, 2001). Processos de negócios são coisas que o processo faz e envolvem a troca de informações entre os parceiros de mercado. Informações de negócios são informações que um negócio usa para executar o processo. Normalmente estas informações são expressas na forma de documentos de negócios (*Business Documents*). Documentos de negócios são compostos por objetos de informação de negócio, ou por pequenas partes de informação previamente identificadas. Estas partes ou componentes são estruturas, tais como, um DTD / XML, que definem as informações e como elas devem ser apresentadas. O resultado final é uma

estrutura predefinida em que as informações são colocadas, podendo o receptor do documento final interpretá-lo para extrair suas informações (IBM, 2001).

Um dos objetivos do ebXML é criar componentes (*Core Components*), que serão armazenados em uma biblioteca (*Core Library*), e estarão disponíveis para a construção de documentos de negócios. Entretanto, a especificação dos componentes ainda não está completa, porém, já foram descritos alguns componentes, tais como: o *Collaboration Protocol Profiles* (CPP) e o *Collaboration Protocol Agreements* (CPA). Na Figura 13 é apresentado um exemplo de um documento CPP.

```
<?xml version="1.0" encoding="UTF-8"?>
<tp:CollaborationProtocolProfile
  xmlns:tp="http://www.ebxml.org/namespaces/tradePartner"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  tp:version="1.1">
  <tp:PartyInfo>
    <tp:PartyId tp:type="DUNS">123456789</tp:PartyId>
    <tp:PartyRef tp:href="http://example.com/about.html"/>
    <tp:Certificate tp:certId="N03">
      <ds:KeyInfo/>
    </tp:Certificate>
    <tp:Transport tp:transportId="N05">
      <tp:SendingProtocol tp:version="1.1">HTTP</tp:SendingProtocol>
      <tp:ReceivingProtocol tp:version="1.1">HTTP</tp:ReceivingProtocol>
      <tp:Endpoint tp:uri="https://www.example.com/servlets/ebxmlhandler" tp:type="allPurpose"/>
      <tp:TransportSecurity>
        <tp:Protocol tp:version="3.0">SSL</tp:Protocol>
        <tp:CertificateRef tp:certId="N03"/>
      </tp:TransportSecurity>
    </tp:Transport>
    <tp:Transport tp:transportId="N08">
      <tp:SendingProtocol tp:version="1.1">HTTP</tp:SendingProtocol>
      <tp:ReceivingProtocol tp:version="1.1">SMTP</tp:ReceivingProtocol>
      <tp:Endpoint tp:uri="mailto:ebxmlhandler@example.com" tp:type="allPurpose"/>
    </tp:Transport>
  </tp:PartyInfo>
  <tp:Packaging tp:id="N0402">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
      <tp:Composite tp:id="N42" tp:mimetype="multipart/related" tp:mimeparameters="type=text/xml;">
        <tp:Constituent tp:idref="N40"/>
        <tp:Constituent tp:idref="N41"/>
      </tp:Composite>
    </tp:CompositeList>
  </tp:Packaging>
  <tp:Comment tp:xml_lang="en-us">buy/sell agreement between example.com and contrived-example.com
  </tp:Comment>
</tp:CollaborationProtocolProfile>
```

Figura 13 – Exemplo de um documento ebXML.

3.4 CORBA

A arquitetura CORBA (*Common Object Request Broker Architecture*) é uma plataforma aberta definida através de um conjunto de especificações e de conceitos para

que objetos distribuídos, em um ambiente de rede heterogêneo, possam ser acessíveis através de uma interface bem definida (LUNG, 2001). A OMG (*Object Management Group*) é a organização responsável pela especificação do padrão aberto CORBA para programação distribuída. Esta organização foi estabelecida em 1989 e é constituída por membros de diversas áreas, desde vendedores, usuários até desenvolvedores de software. Seu objetivo é criar um modelo de arquitetura orientada a objeto, padronizada para aplicações distribuídas, baseado em especificações que habilitam e suportam objetos distribuídos.

O CORBA segue o paradigma de computação distribuída chamado cliente-servidor. Este modelo é constituído basicamente por trocas de mensagens entre o cliente e o servidor, ou seja, o cliente faz uma requisição ao servidor que ao receber a mensagem realiza o seu processamento e envia uma resposta como o resultado. Baseado neste modelo, os objetos distribuídos encapsulam seu estado interno e se fazem acessíveis através de uma interface bem definida. Segundo a arquitetura CORBA, métodos de objetos remotos podem ser invocados de forma transparente em ambientes distribuídos heterogêneos através de um ORB[21]. O ORB, em um sentido mais genérico, pode ser considerado como um canal de comunicação para objetos distribuídos (VINOSKI, 1998).

3.4.1 Modelo de Objetos

O modelo de objetos provê uma representação organizada dos conceitos de objetos e terminologias. Ele define um modelo parcial para computação que engloba as características principais do objeto tal como qual ele é percebido (OMG, 2002).

Um sistema de objetos é uma coleção de objetos, onde os requerentes do serviço (os clientes) são isolados dos provedores dos serviços (os servidores), através de interfaces bem definidas. Particularmente, estes clientes são isolados da implementação do servidor, tanto da representação dos dados quanto do código executável.

Primeiramente, o modelo de objetos descreve conceitos significativos para os clientes, incluindo: criação e identificação de objetos, requisições e operações, tipos e assinaturas. Além disso, descreve conceitos relacionados às implementações de objetos, incluindo conceitos, tais como: métodos, máquina de execução e ativação. Este modelo se

[21] ORB – *Object Request Broker*.

preocupa mais especificamente com a definição dos conceitos relacionados com o cliente, deixando de lado algumas questões relacionadas aos servidores. Isto ocorre pelo fato de que discussões sobre implementação de objetos são subjetivas, e levam a considerações sobre a tecnologia de implementação adotada.

A implementação de um sistema de objetos leva em consideração as atividades computacionais necessárias para efetivar o comportamento desejado do serviço. Estas atividades podem incluir computação dos resultados da requisição e a atualização do estado do sistema. O modelo de implementação consiste de duas partes: o modelo de execução e o modelo de construção. O modelo de execução descreve com os serviços são executados. Já o modelo de construção descreve como os serviços são definidos (OMG, 2002).

Os objetos são descritos através de suas interfaces que descrevem o conjunto de operações que podem ser requisitadas pelos seus respectivos clientes. O uso da interface permite ocultar do cliente aspectos relacionados à implementação do serviço (LUNG, 2001).

3.4.2 Arquitetura OMA

A OMA (*Object Management Architecture*), descreve os objetivos técnicos e terminologias da OMG e provê a infraestrutura conceitual em que as especificações de suporte estão baseadas. A OMA inclui o modelo de objetos OMG, que foi apresentado anteriormente, e define a semântica comum para especificar as características externamente visíveis de objetos em um modo padrão de implementação, e também seu modelo de referência.

O modelo de referência identifica e caracteriza os componentes, interfaces e protocolos que compõem a OMA. Este modelo inclui o ORB, que habilita os clientes e objetos a se comunicarem em um ambiente distribuído, e quatro categorias de interfaces de objetos: os objetos de serviço, as facilidades comuns, as interfaces de domínio e os objetos de aplicação (OMG, 1996). A arquitetura do modelo de referência OMA com seus respectivos componentes pode ser vista da Figura 14 (OMG, 1996).

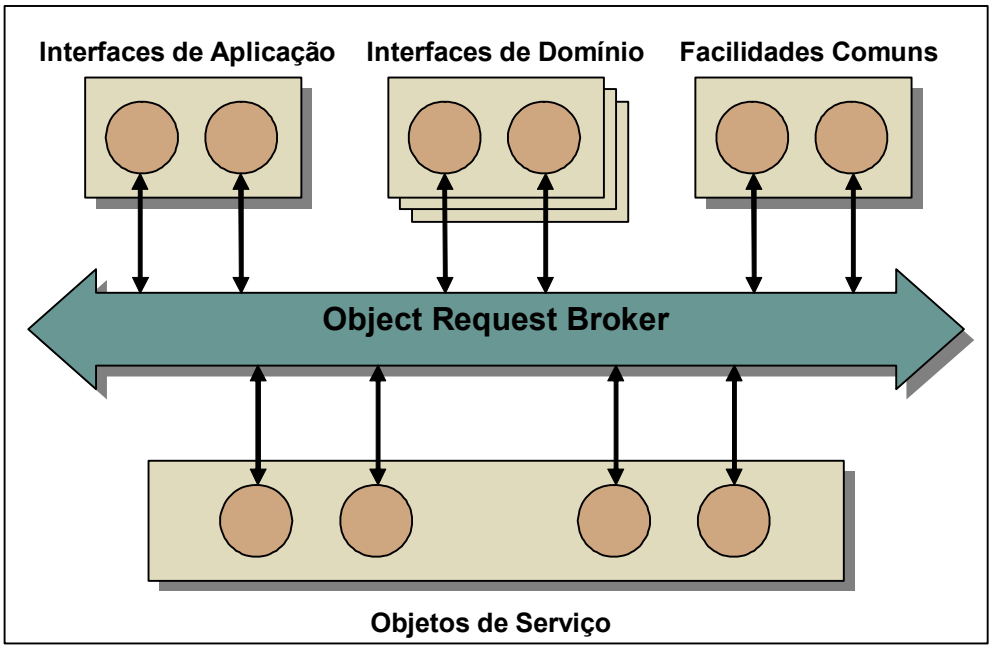


Figura 14 – Modelo de referência OMA.

Estrutura do ORB

O ORB é responsável por todo o mecanismo de localização da implementação do objeto para uma determinada requisição. Ele prepara a implementação do objeto para receber a requisição do cliente. O cliente enxerga a interface completamente independente de onde o objeto está localizado, qual linguagem de programação foi implementado e outros aspectos que não refletem na interface do objeto (OMG, 2002). A Figura 15 (OMG, 2002) mostra a estrutura do ORB.

As interfaces dos objetos podem ser definidas de duas formas. Estaticamente, através de uma linguagem de definição de interfaces chamada de OMG *Interface Definition Language* (OMG IDL). Esta linguagem define os atributos e operações que o objeto pode executar. Dinamicamente, através de "repositórios de interfaces". Este serviço representa os componentes de uma interface com o objeto, permitindo o acesso destes componentes em tempo de execução. Na implementação do ORB, tanto a IDL quanto o repositório de interfaces têm equivalente poder de expressão (OMG, 2002).

Para invocações estáticas, quando um cliente faz uma requisição, esta é transformada em uma mensagem através de *stubs* particulares do CORBA (gerados por um compilador de linguagem IDL). A mensagem serializada (*marshalling*) é transmitida na rede, usando as estruturas do ORB que localiza o objeto servidor e transporta os dados de

acordo com a sintaxe de transferência. No servidor, a mensagem que chega passa o controle para o Adaptador de Objetos Portável (*Portable Object Adaptor* - POA) que, por sua vez, tem a responsabilidade de ativar a implementação de objeto correspondente. Os processos de deserialização (*unmarshalling*), e a conseqüente chamada de método de implementação do objeto, são realizados através do correspondente *skeleton* do objeto servidor (LUNG, 2001).

Para invocações dinâmicas, as interfaces de objetos CORBA devem estar disponíveis, armazenadas no repositório de interfaces. Uma invocação dinâmica permite ao cliente dinamicamente construir e invocar métodos do servidor usando interfaces DII[22]. Este tipo de chamada oferece uma grande flexibilidade para sistemas altamente dinâmicos.

O repositório de interfaces contém informações de interfaces IDL em uma forma disponível em tempo de execução para invocação dinâmica. Usando a informação disponível no repositório, é possível a um programa encontrar um objeto remoto mesmo desconhecendo sua interface, seus métodos, parâmetros e forma de ativação (LUNG, 2001).

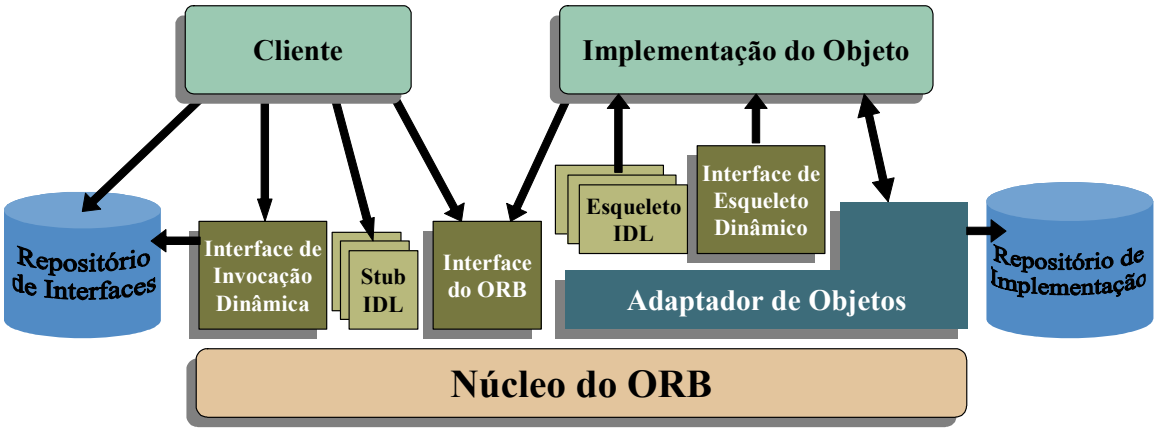


Figura 15 – Arquitetura CORBA.

Objetos de Serviço

Objetos de serviço são serviços de propósito geral, que têm fundamental importância para o desenvolvimento de implementações baseadas em CORBA, compostas de objetos distribuídos, ou que provê uma base universal, independente do domínio da aplicação, para interoperabilidade entre aplicações (OMG, 1996).

[22] DII – *Dynamic Invocation Interface*.

Os objetos de serviço, chamados de *CORBAservice*, formam uma coleção de serviços (interfaces e objetos) que são padronizados pela OMG dentro da arquitetura OMA. Múltiplos objetos de serviço podem cooperar no sistema. Por exemplo, um serviço de suporte a grupos pode utilizar o serviço de recuperação de erros para restabelecer objetos servidores após uma falha. Além de estar de acordo com a norma CORBA, um serviço é muito mais fácil de ser padronizado do que uma extensão ao ORB, ou até mesmo um novo adaptador de objetos.

A OMG tem vários serviços definidos, tais como: serviço de nomes, serviço de notificação de eventos, serviço de ciclo de vida, serviço de persistência, serviço de relacionamento, serviço de externalização, serviço de transação, serviço de controle de concorrência, serviço de autorização, serviço de segurança e serviço de tempo.

Facilidades Comuns

As facilidades comuns, também chamadas *CORBAfacilities*, são coleções de interfaces aplicáveis a um grande número de domínios de aplicações. As facilidades comuns são divididas segundo dois aspectos: as facilidades comuns horizontais e as facilidades comuns verticais (OMG, 1996, LUNG, 2001). As facilidades horizontais podem ser utilizadas por diferentes aplicações, independente de seu domínio. São divididas segundo quatro categorias: interface de usuário, gerenciamento de informação, gerenciamento de sistema e gerenciamento de tarefa. As facilidades verticais são utilizadas em áreas de aplicação específicas, ou seja, domínios específicos. Como exemplo tem-se: gerenciamento e controle de imagens, supervias de informação, manufatura integrada por computador, simulação distribuída, etc.

Interfaces de Domínio

Interfaces de domínio são interfaces para domínios específicos de aplicações, tais como: financeira, saúde, telecomunicações, comércio eletrônico e transporte. O seu propósito é separar o modelo conceitual da representação de objetos, e também prover o máximo de independência possível entre os objetos envolvidos. Estas interfaces são projetadas de uma forma genérica para cobrir as necessidades típicas de um determinado domínio de aplicação. O termo genérico significa que as interfaces não são baseadas em objetos específicos de uma única aplicação, suas definições IDL são completamente independentes de qualquer sistema em particular.

Objetos de Aplicação

Os objetos de aplicação são objetos não padronizados utilizados em aplicações específicas. Portanto, são produtos de um único grupo de desenvolvedores que controla suas interfaces. Objetos de aplicação correspondem à noção tradicional de aplicações, portanto, não são padronizados pela OMG. Entretanto, os objetos de aplicação constituem a camada superior do modelo de referência.

3.4.3 Interoperabilidade

Com o passar do tempo foram surgindo vários produtos ORB disponíveis no mercado. Esta diversidade é muito importante, pois permite que os fabricantes construam seus produtos para atender às necessidades específicas de seus clientes. Entretanto, existe a necessidade destes diferentes ORBs interoperarem. Para responder a esta necessidade a OMG formulou a "arquitetura de interoperabilidade OMG".

Diferenças de implementação não são a única barreira que separa os objetos. Desta forma, para prover um ambiente interoperável, todas estas diferenças devem ser levadas em consideração. Para isso, a versão CORBA 2.0 introduziu o conceito de domínio, que define conjuntos de objetos que, por alguma razão, de implementação ou administrativa, são separados uns dos outros. Assim, objetos de diferentes domínios precisam de mecanismos de ponte (*Bridge*) para fazer o mapeamento entre os domínios (LUNG, 2001).

Estas *bridges* podem ser implementadas internamente ou em um nível acima do ORB. Quando são implementadas dentro do ORB, tem-se o nome de *in-line bridges*. Entretanto, quando são implementadas fora do ORB, são chamadas *request-level bridges* (OMG, 2002). Para tornar as *bridges* possíveis, faz-se necessária a especificação de alguma sintaxe de transferência padrão. A OMG, desde o padrão CORBA 2.0, definiu os seguintes protocolos:

Protocolo Inter-ORB Geral (GIOP)[23]: Este protocolo especifica um conjunto de formatos para mensagens e dados que são transportados nas comunicações entre ORBs;

[23] GIOP – *General Inter-ORB Protocol*.

Protocolo Inter-ORB Internet (IIOP)[24]: Este protocolo especifica como mensagens GIOP são transmitidas em uma rede TCP/IP (GIOP + TCP/IP = IIOP);

Protocolo Inter-ORB para Ambientes Específicos (ESIOPs)[25]: É uma especificação feita para permitir a interoperabilidade de um ORB com outros ambientes.

[24] IIOP – *Internet Inter-ORB Protocol*.
[25] ESIOP – *Environment-Specific Inter-ORB Protocol*.

CAPÍTULO 4: Modelo Semi-automático de Configuração de Empresas Virtuais

Em uma empresa virtual existem várias etapas de configuração pela qual a mesma deve passar para estar apta a operar ou se manter em operação. Estas etapas estão distribuídas ao longo das fases do ciclo de vida da EV. Além destas atividades, existem algumas outras chamadas "preparatórias", que acontecem antes do início do ciclo de vida da EV. As etapas preparatórias são as seguintes: a concepção da plataforma, e a instalação e integração da mesma com os sistemas legados das empresas participantes. É importante ressaltar que estas etapas preparatórias de configuração são executadas apenas uma vez em cada empresa participante. Portanto, deste momento em diante a empresa pode participar em quantas EVs desejar, desde que todas as empresas participantes utilizem a mesma plataforma ou outra interoperável.

Após o término deste conjunto preliminar de configurações a empresa está apta a fazer parte de uma empresa virtual. A partir deste momento começa o segundo conjunto de configurações, que engloba três etapas: 1) localização dos parceiros mais adequados para a referida oportunidade de negócios; 2) configuração da topologia para especificar os papéis dos parceiros e suas relações; 3) restrição de acesso às informações dos parceiros. Após o término deste conjunto de configurações a empresa virtual pode entrar em operação. Estas etapas do processo de configuração não são utilizadas apenas na fase de criação da EV, ou seja, algumas destas etapas podem ser requisitadas no momento que a EV está operando. Neste momento a EV muda para o estado de evolução e executa as etapas de configuração necessárias. Um exemplo de evolução é a inclusão de um novo parceiro quando a EV já está em operação. Assim que o parceiro é escolhido, reconfigurações na topologia e no direito de acesso às informações desse novo membro devem ser feitas. A Figura 16 apresenta as etapas de configuração descritas acima, posicionando-as no ciclo de vida da EV. Além das etapas de configuração, a Figura 16 também apresenta os subsistemas que compõem o sistema SC^2 e suas relações com as etapas de configuração e gestão da empresa virtual. O sistema SC^2 (*Supply Chain Smart Coordination*), já mencionado no capítulo 1, foi o sistema concebido pelo laboratório GSIGMA no âmbito do projeto DAMASCOS e serviu de base para o modelo proposto neste trabalho.

A proposta desta dissertação é formular um modelo que auxilie no processo de configuração da topologia da EV e das restrições de acesso às informações dos parceiros. Na Figura 16 é apresentada em destaque a parte do problema de configuração de EVs abordada pelo modelo concebido nesta dissertação. Este modelo tem por objetivo aumentar a eficiência e a agilidade do processo de configuração. Estas são características de suma importância no contexto de empresas virtuais dinâmicas, foco principal deste trabalho.

O modelo concebido neste trabalho é denominado semi-automático, pois uma parte do processo é automática enquanto a outra é manual. Ou seja, a parte de configuração da topologia não necessita da intervenção do usuário no decorrer de seu processo, portanto é automática. Entretanto, para o processo de configuração das restrições de acesso às informações o usuário participa do processo, tornando-o manual. Devido a esta particularidade no processo de configuração da restrição de acesso às informações intercambiadas entre os membros, o processo como um todo é considerado semi-automático. Nas próximas seções as duas partes do modelo de configuração proposto nesta dissertação serão apresentadas em detalhes.

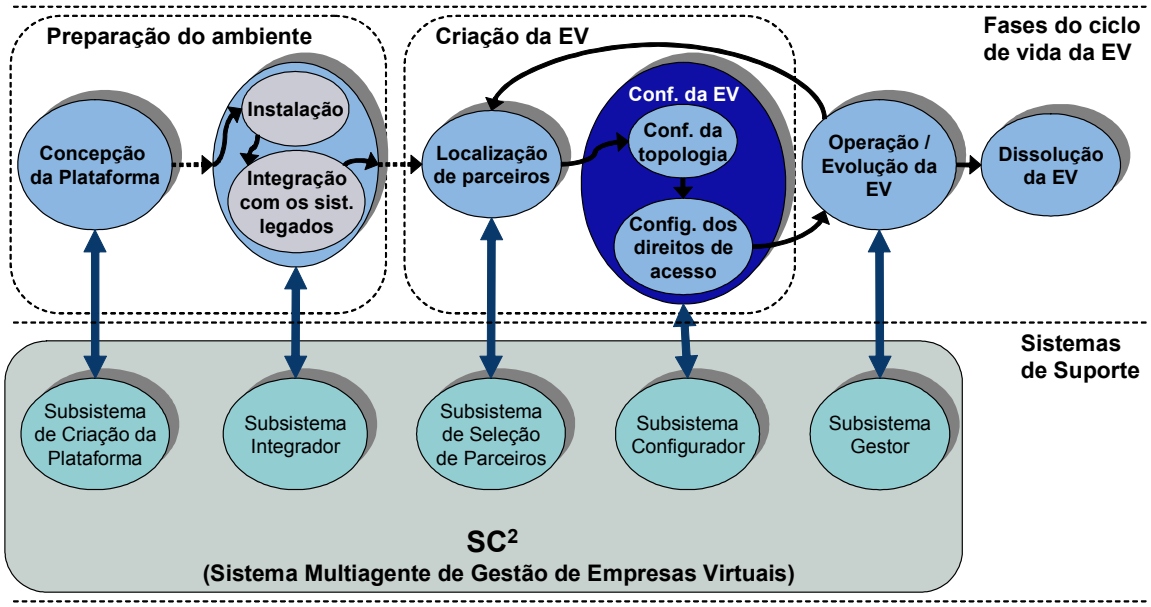


Figura 16 – Enquadramento da configuração no contexto de EVs.

4.1 Configuração da Topologia

A tarefa de configuração da topologia de uma EV consiste no processo de, dado o conjunto de parceiros selecionados para fazer parte da EV, arranjá-los de forma gráfica

para facilitar sua visualização, tanto por parte do coordenador como dos membros, que ao final da configuração receberão uma cópia deste "mapa" mostrando quais empresas fazem parte da EV. Nesta topologia, além de serem mostrados os parceiros, também são apresentados seus relacionamentos, mais especificamente quais produtos uma empresa comercializa com a outra. Na Figura 17 é mostrada uma forma conceitual de representação da topologia da empresa virtual. A título de exemplo, supõem-se que a EV da Figura 17 seja fabricante de sapatos. Assim, o fornecedor F1 fornece cadarço e linha, o fornecedor F2 fornece sola e palmilha, o fornecedor F3 fornece tecido e o fornecedor F4 fornece couro. A função do produtor principal P1 é produzir os sapatos, as sandálias ou os tênis que serão entregues aos distribuidores D1 e D2, que repassarão aos clientes C1, C2 e C3 interessados nos produtos.

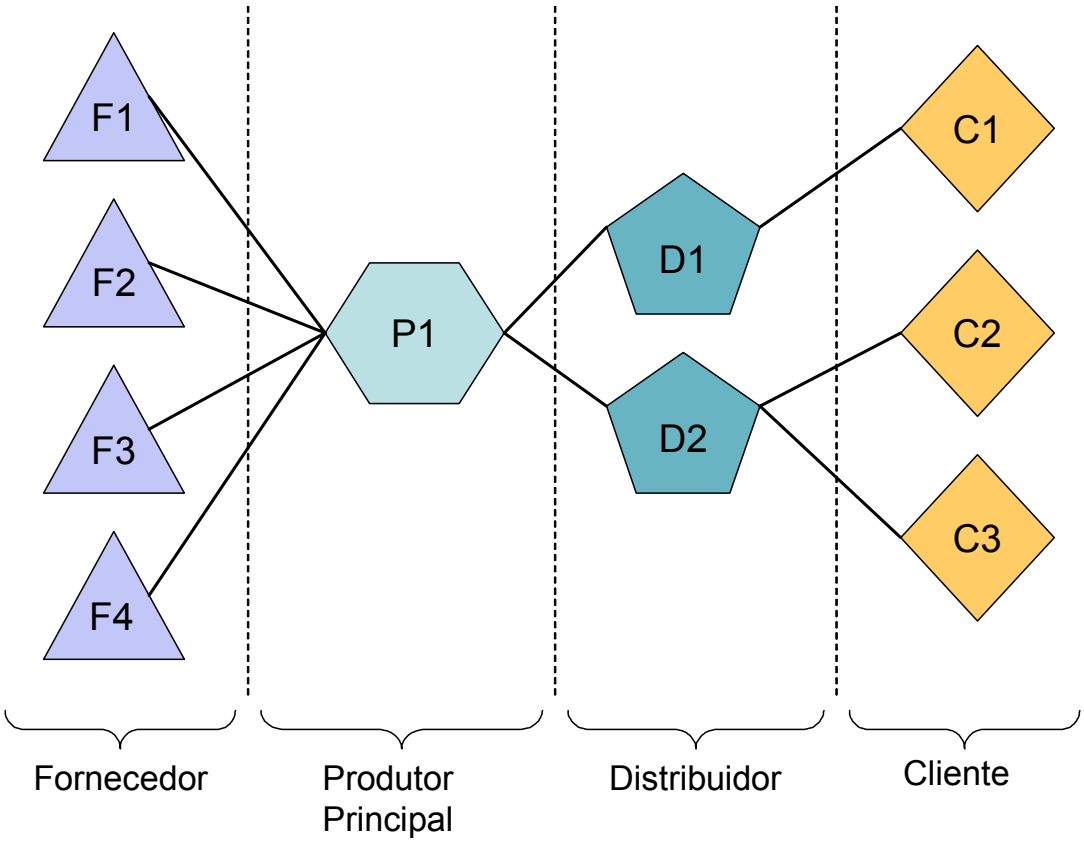


Figura 17 – Exemplo conceitual da topologia da EV.

Em um primeiro momento a tarefa de configuração parece intuitiva para o coordenador da EV. Entretanto, esta tarefa é de vital importância quando do gerenciamento efetivo da mesma na fase de operação. Sem essa visão global de como a EV está montada, de quais são os relacionamentos entres as empresas, torna-se mais difícil achar uma solução para possíveis problemas que venham a ocorrer. Também, sem essa visão global,

pior será a compreensão do funcionamento da EV por parte de todas as empresas envolvidas.

Para tentar solucionar o problema da configuração da topologia da EV, um modelo de configuração manual foi criado. O coordenador de posse da lista de parceiros poderia montar a EV especificando os relacionamentos entre os membros um a um. Entretanto, essa forma de construção da topologia ainda não atendia aos requisitos para EV dinâmicas, onde as trocas de parceiros acontecem com maior freqüência no momento que a mesma está em operação. Portanto, para tornar essa ferramenta apta a trabalhar com esse tipo de EVs, um modelo automático de configuração da topologia foi concebido.

No intuito de tornar este modelo mais abrangente e fazer frente aos requisitos exigidos pelas empresas virtuais dinâmicas, o modelo proposto contempla duas formas de configuração automática da topologia baseada em sistemas multiagente:

- Com o auxílio de uma ferramenta de procura e seleção de parceiros;
- Sem o auxílio de uma ferramenta de procura e seleção de parceiros.

O modelo foi concebido desta forma para que empresas que já possuam ferramentas de seleção de parceiros possam utilizá-las de forma integrada ao modelo de configuração, sem a necessidade de descartá-las ou substituí-las. Entretanto, para empresas que não tenham este tipo de ferramenta, uma alternativa para a localização dos parceiros previamente selecionados é fornecida através da segunda alternativa.

A configuração da topologia é considerada como sendo automática, pois se todas as informações necessárias ao processo estiverem disponíveis, a mesma é montada de forma completa, sem a necessidade da intervenção de um usuário no decorrer do processo. Porém, se parte das informações necessárias ao processo não estiver disponível, o usuário pode interagir com o processo e completar a configuração manualmente. Como as informações necessárias para o processo são repassadas pelos membros da EV, se algum problema ocorrer com algum determinado membro, suas informações não serão enviadas, o que acarreta em uma falha na construção da topologia. Neste momento, o usuário que está monitorando o processo de configuração pode lançar mão da configuração manual para intervir e completar o processo. Nas próximas subseções serão apresentados respectivamente: os papéis das empresas na EV, o modelo multiagente, e os dois tipos de configuração de EVs ressaltando as características mais importantes de cada um.

4.1.1 Papéis Operacionais das Empresas

Em uma empresa virtual cada empresa participante desempenha funções predefinidas pelo seu papel operacional. Os papéis operacionais são atribuídos às empresas pelo coordenador em conformidade com suas características. Por exemplo, na Figura 17 as empresas representadas por retângulos têm o papel de fornecedores e são responsáveis por fornecer matéria-prima para a confecção dos calçados. No modelo proposto foram definidos alguns tipos de papéis operacionais possíveis que podem ser representados por uma empresa em uma EV do ramo calçadista. São eles:

- Pré-fornecedor: Fornece a matéria-prima que é beneficiada pelos fornecedores;
- Fornecedor: Transforma a matéria-prima em insumos;
- Produtor Principal: Monta o produto final através da junção dos vários insumos que o constitui;
- Distribuidor: Concentra a distribuição dos produtos produzidos pelo produtor principal;
- *Broker*: Faz a intermediação entre o produtor principal e o cliente final;
- Varejista: Atende ao consumidor com produtos a pronta entrega.

É importante observar que nem sempre todos estes papéis precisam necessariamente estar presentes na mesma EV. O conjunto de empresas que formam uma EV pode representar apenas um subconjunto dos tipos de papéis possíveis.

4.1.2 Configuração Baseada em Agentes

Antes de entrar em profundidade na descrição dos modelos de configuração da topologia de EVs mencionados acima, algumas características em comum de ambos os modelos serão apresentadas. Como uma das características mais importantes podemos destacar a iniciativa de conceber um modelo tendo por base as técnicas para a modelagem de sistemas multiagente (SMA). A escolha de sistemas multiagente foi feita principalmente pelo fato do problema ser inerentemente distribuído. Ou seja, as informações necessárias para a solução do problema estão distribuídas em cada uma das empresas envolvidas. Outra característica importante que serve de justificativa para a utilização de agentes é a questão da divisão das tarefas. Desta forma, cada agente concentra determinadas

habilidades necessárias para a resolução de parte do problema, tendo um papel bem definido na solução do problema como um todo. A concepção do modelo tomada desta maneira, torna-o mais modularizado e flexível em relação a modelos gerados de forma monolítica. Assim, sistemas multiagente são a solução mais adequada para a resolução deste tipo de problema. Para a solução específica do problema de configuração de EVs foram construídos os seguintes agentes: o "agente Configurador", o "agente Empresa" e o "agente XML". Estes agentes são descritos detalhadamente a seguir e podem ser vistos na Figura 18.

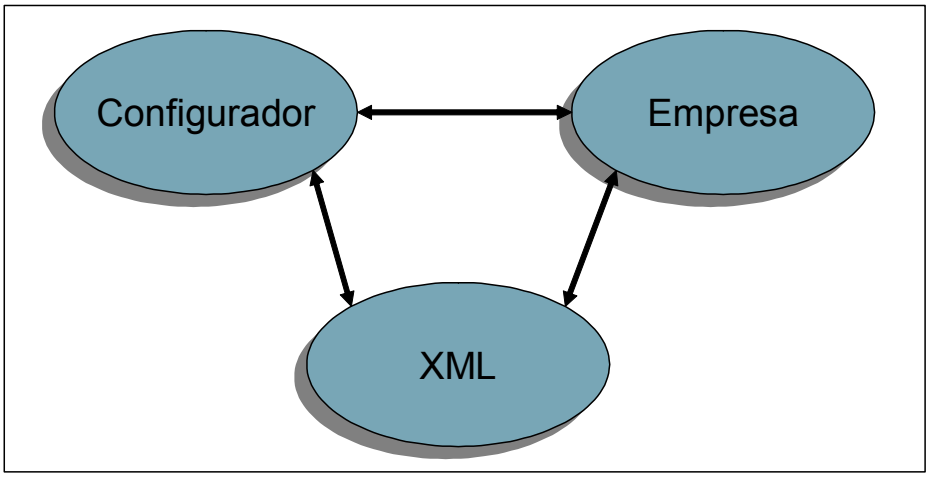


Figura 18 – Agentes do sistema.

Outra característica importante do modelo é a estrutura da informação que está por trás da ferramenta de configuração. Para dar suporte à ferramenta de configuração e servir como base para todas as funcionalidades de gerenciamento e coordenação da EV foi construído um modelo relacional de informação, onde todas as informações coletadas entre as empresas participantes ou produzidas pela própria EV são armazenadas de forma persistente. Este modelo de informações foi projetado como sendo a espinha dorsal para a estrutura da informação dos sistemas de gestão da EV. Portanto, tomando por base esse modelo, toda a informação necessária em virtude da criação de novas funcionalidades será facilmente agregada ao mesmo de forma distinta e estruturada.

Agente Configurador

O agente Configurador é o agente responsável pela configuração da EV. Este agente está ativo somente na empresa coordenadora, pois é a ela que a realização desta tarefa é delegada. Entretanto, deve-se levar em consideração que todo membro de uma EV

pode, em outra ocasião, se tornar um coordenador de uma outra EV. Por esse motivo, o sistema multiagente do modelo proposto é um pacote com três agentes independentemente do papel ao qual a empresa esteja desempenhando no consórcio.

Este agente, entre outras características, possui a habilidade de se conectar com ferramentas de procura e seleção de parceiros. Esta conexão acontece da seguinte maneira: o resultado da ferramenta de seleção de parceiros serve como entrada para o agente configurador que pega estas informações e monta a topologia da EV. Entretanto, este agente possui uma outra alternativa caso não haja uma ferramenta de seleção de parceiros disponível. Assim, o configurador faz o papel do localizador de parceiros, porém, o agente configurador não tem a pretensão de substituir as ferramentas de seleção de parceiros. Ele não faz a procura por parceiro, apenas localiza na rede de potenciais parceiros quais foram previamente selecionados para fazer parte da EV. Maiores explicações sobre os métodos de configuração da topologia da EV serão detalhadas nas próximas subseções.

Além da configuração da topologia, o agente Configurador também se preocupa com o controle de acesso às informações compartilhadas entre os membros participantes. Este controle acontece através da restrição da visibilidade da topologia e do direito de acesso à informação de cada membro. A restrição de visibilidade diz respeito a que parte da topologia da EV cada um dos membros terá privilégio para visualizar. Já o direito de acesso à informação está relacionado a quais de suas informações cada parceiro disponibiliza para os outros membros da EV. Deve-se notar que cada membro pode selecionar um conjunto diferente de informações disponíveis para cada parceiro dependendo das necessidades de cada um.

Agente Empresa

O agente Empresa tem como funções principais manter atualizadas as informações sobre a empresa onde está situado, e responder às requisições feitas pelo agente Configurador. Quando uma nova EV começa seu processo de configuração, o agente Configurador da empresa coordenadora difunde uma mensagem para saber quem são efetivamente os parceiros selecionados para a referida EV. Do lado receptor, quem analisa esta requisição são os agentes Empresa dos parceiros que respondem com as informações pedidas pelo coordenador.

Esta não é a única função do agente Empresa, ele também é responsável pela visualização da topologia da EV, que é enviada para cada empresa participante logo após o término do processo de configuração por parte do coordenador. Este agente está ativo em todos os parceiros da EV diferentemente do agente Configurador que está ativo apenas no coordenador.

Além das funções descritas acima, o agente Empresa é o responsável pelas funcionalidades relacionadas à gestão da empresa virtual, funcionalidades estas que fogem do escopo do trabalho em questão. No coordenador, o agente Empresa serve como meio para alterar os parâmetros envolvidos no bom funcionamento da EV. Porém, no lado dos membros o mesmo serve como porta para visualizar estas alterações. Ou seja, é através dele que o coordenador controla a EV e os membros observam este controle. Dentre outras funções incorporadas a este agente podemos destacar funcionalidades tais como (RABELO *et al.*, 2002): *Ad Hoc Report*, *Diagnostic Report*, *Position Paper*, *Reactive Inventory*, etc.

Agente XML

A este agente é delegada a tarefa de manter a comunicação entre os SMAs. Ele serve como porta de entrada e saída do sistema multiagente para o exterior. Seu papel é fundamental do processo de configuração semi-automática de empresas virtuais. Basicamente sua função é receber e enviar mensagens. Cada mensagem que chega ao agente é processada e suas informações são extraídas para posteriormente serem armazenadas no repositório de dados. Quando uma informação precisa ser enviada para algum parceiro, esta é recuperada do repositório de dados e transformada em uma mensagem. Esta mensagem segue um formato específico definido dentro que um conjunto padrão de tipos de mensagens que podem ser trocadas entre os elementos participantes. Após passar por esse processo de codificação, a mensagem pode ser enviada para seu destino onde passará pelo processo inverso.

O agente XML concentra todo o trabalho de codificação e decodificação das informações trocadas entre os sistemas multiagente. Com isso, os outros agentes ficam livres para atuarem apenas nas funções de importância para a resolução do problema. Suas mensagens são codificadas utilizando um padrão reconhecido internacionalmente, o que facilita a interconexão com sistemas de terceiros (externos ao modelo). Estes sistemas externos podem ser desde uma ferramenta de *workflow* até serviços para a *web*.

4.1.3 Usando Ferramenta de Seleção de Parceiros

Uma alternativa para a configuração da topologia da EV é a utilização de uma ferramenta de seleção de parceiros como suporte a esta tarefa. Estas ferramentas são importantes no processo de configuração tendo em vista que são elas as responsáveis em localizar e escolher os melhores parceiros para uma determinada oportunidade de negócio. Para otimizar a procura, as ferramentas de busca e seleção de parceiros utilizam vários critérios que servem de apoio para uma melhor escolha. Estes critérios vão desde custos e prazos até qualidade e confiança.

O processo de configuração utilizando uma ferramenta de seleção é relativamente simples se comparado com a segunda abordagem onde não se tem esse tipo de auxílio. Este processo se inicia com o resultado da tarefa de seleção, ou seja, a saída da ferramenta de seleção, sendo transformada na entrada do módulo do agente Configurador que faz a configuração. Entretanto, existe uma pequena restrição a respeito do formato das informações que são fornecidas para o agente Configurador. As informações fornecidas pela ferramenta de seleção devem estar de acordo com o formato pré-definido pelo agente Configurador. Este formato utiliza uma metalinguagem padrão na qual foi definida uma estrutura que especifica todas as informações necessárias para efetuar a configuração da topologia. Portanto, uma contribuição importante do modelo apresentado neste trabalho é a proposta de uma forma de integração entre duas etapas do processo de configuração, de um lado a seleção de parceiros e do outro a configuração da topologia. Através da utilização dessa metalinguagem padrão, tentou-se deixar aberta a possibilidade de se utilizar "qualquer" ferramenta de seleção de parceiros disponível. Entretanto, se não houver a possibilidade de tornar o resultado da ferramenta de seleção de parceiros compatível com a entrada esperada pelo módulo de configuração, o usuário configurador pode inserir manualmente estas informações e completar a configuração.

As informações necessárias para a configuração são facilmente fornecidas por uma ferramenta de seleção. Estas informações são basicamente: informações básicas das empresas participantes (nome, endereço, telefone), função desempenhada por cada uma na EV, produtos intercambiados entre elas, e com quais outros parceiros estão relacionadas. Através deste conjunto de informações o agente Configurador monta a topologia da EV, envia uma mensagem para cada parceiro confirmado que ele faz parte da mesma, e por fim salva todas as informações relativas à configuração no repositório de dados. O conjunto de

informações necessárias para a configuração da topologia da EV é apresentado na Figura 19. Nesta estrutura pode ser visto que uma empresa virtual é representada por um código e um nome, e contém uma lista de membros (empresas participantes), uma lista de conexões entre estes membros e uma lista com as informações básicas de cada empresa participante da EV. Sua forma aninhada indica a hierarquia em que está estruturada a informação.

```
Empresa Virtual
    código,
    nome;
    lista membros
        membro
            código membro,
            código empresa,
            papel;
    lista conexões
        conexão
            membro fonte,
            membro destino;
            lista itens
                item
                    empresa fonte,
                    código produto;
    lista empresas
        empresa
            código empresa,
            nome empresa,
            telefone,
            endereço,
            cidade,
            estado,
            país;
            lista produtos
                produto
                    código produto,
                    descrição,
                    tipo,
                    cor;
```

Figura 19 – Informações para a configuração da topologia da EV.

Apesar das informações sobre os parceiros serem fornecidas pela ferramenta de seleção, o usuário configurador pode, se necessário, fazer alterações na topologia. Estas alterações incluem adicionar ou excluir parceiros bem como mudar seu papel de atuação. A partir do momento que uma alteração é feita, automaticamente o parceiro afetado será informado. Por exemplo, se o usuário configurador resolver adicionar um outro parceiro de sua lista pré-cadastrada de possíveis parceiros, o parceiro que for adicionado à EV receberá uma mensagem informando sobre essa alteração.

Esta alternativa de configuração da topologia foi concebida no intuito de integrar duas partes do projeto DAMASCOS, de um lado a ferramenta de busca e seleção de

parceiros (SCHMIDT, 2003) e do outro o modelo de configuração concebido neste trabalho. A ferramenta de busca e seleção de parceiros desenvolvida no projeto DAMASCOS teve como proposta utilizar um sistema híbrido de agente (móveis e estacionários) no suporte a criação de EVs. Esta abordagem visou oferecer uma solução flexível, eficiente e segura para os problemas encontrados no processo de busca e seleção de parceiros, procurando incorporar os benefícios da utilização de agentes móveis e estacionários no auxílio à decisão do especialista humano (SCHMIDT, 2003).

Como o auxílio de ferramentas de seleção de parceiros o processo de configuração de EVs se torna mais rápido e com isso ganha em agilidade. Porém, a restrição do formato das informações de entrada pode levar a pré-processamentos na informação, deixando assim o processo menos eficiente do que o esperado. A Figura 20 esboça o modelo multiagente com a inserção da ferramenta de busca e seleção de parceiros.

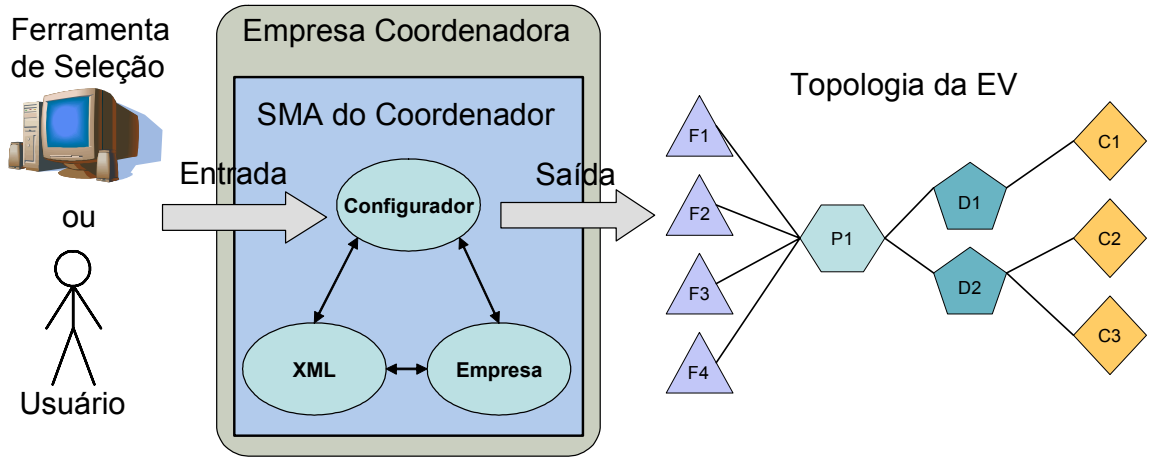


Figura 20 – Configuração com auxílio de ferramenta de seleção de parceiros.

4.1.4 Usando Métodos Internos

A segunda alternativa de configuração serve como forma de suprir a ausência de uma ferramenta de seleção de parceiros. A idéia é fazer com que cada membro em potencial utilize seu SMA para entrar em conversação como o SMA do coordenador, e assim cada um consiga informá-lo qual seu papel e posição na EV, bem como descrever quais os produtos o mesmo intercambia com o consórcio.

O processo acontece da seguinte maneira: inicialmente o agente Configurador da empresa coordenadora difunde uma mensagem na sua rede de potenciais parceiros para saber quais são efetivamente os parceiros eleitos. Nesta etapa, o agente Empresa de cada

empresa que recebeu a mensagem tem duas alternativas de resposta. Se o parceiro não participar da EV, sua resposta será negativa. Por outro lado, se este tiver sido definido previamente como participante, sua resposta se torna mais refinada. Nesta resposta são repassadas as informações básicas da empresa (nome, endereço, telefone), seu papel na EV, seus relacionamentos com os outros membros, e por fim os produtos que ela fornece, no caso de uma fábrica, ou que ela consome, no caso de um distribuidor. Ao final desta etapa do processo de configuração, um esboço da topologia da EV com seus membros corretamente posicionados e os relacionamentos entre eles estabelecidos é entregue ao usuário configurador. A Figura 21 ilustra como o SMA de cada uma das empresas envolvidas interagem com o SMA da empresa coordenadora para realizar a configuração da topologia.

Nesta abordagem o agente Empresa é de suma importância, pois é através dele que o usuário cadastra as informações que serão posteriormente consultadas pelo agente Configurador para saber se a referida empresa faz ou não parte da EV que está sendo configurada.

Tal como na abordagem anterior, se alguma alteração posterior ao processo inicial de configuração precisar ser feita pelo usuário configurador, estas serão realizadas e informadas aos parceiros que forem afetados. Assim, se uma empresa que faz parte da EV não mandar suas informações no processo de localização dos parceiros, o usuário configurador pode de forma manual provocar sua inclusão. Para encerrar o processo de inclusão manual, uma mensagem informando que a empresa agora faz parte da EV é enviada ao novo membro.

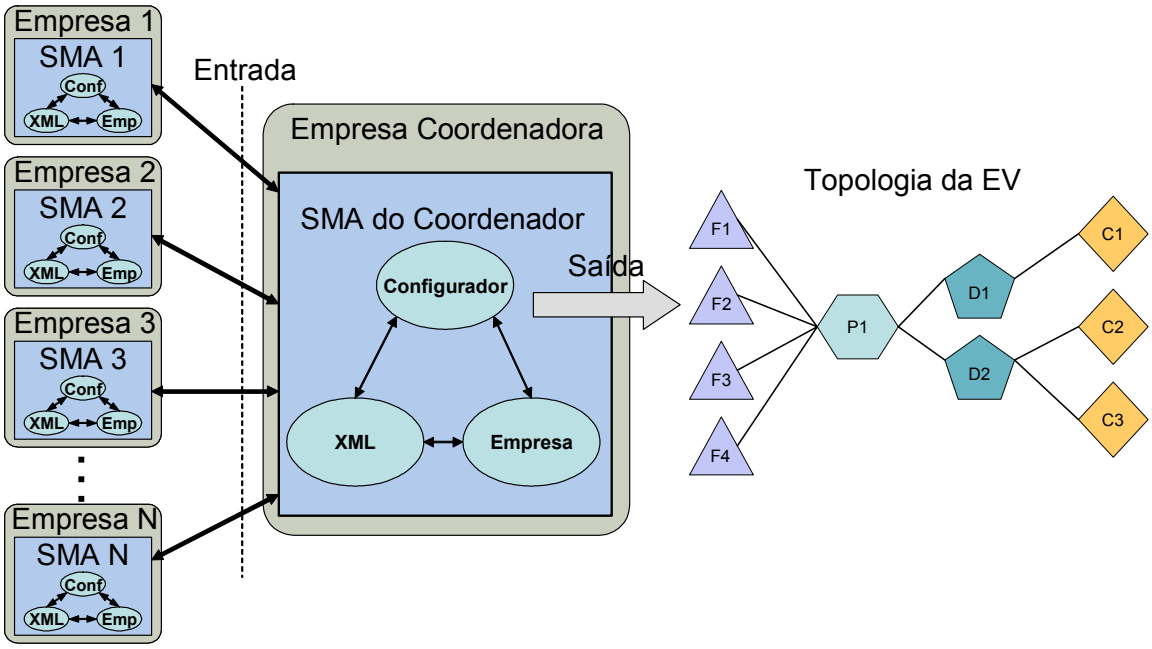
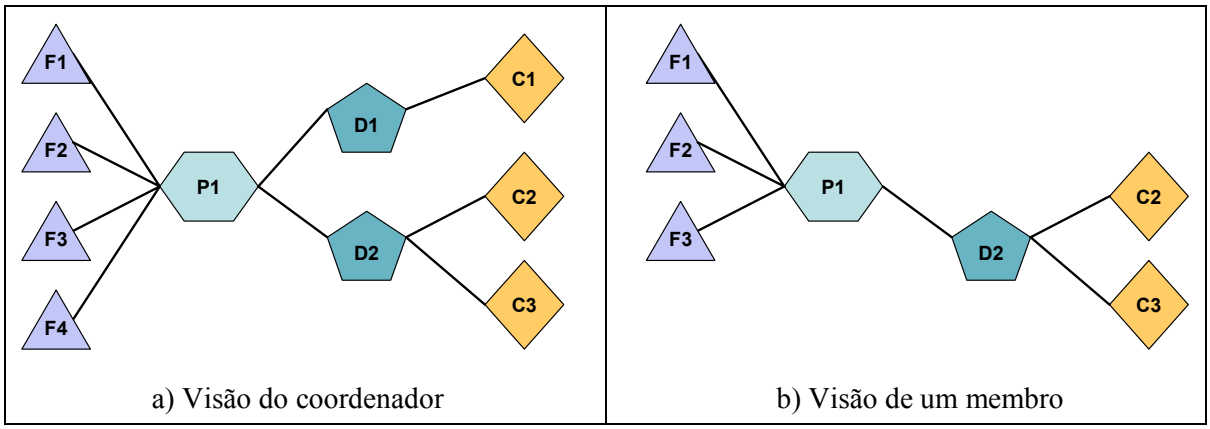


Figura 21 – Sistemas Multiagente realizando a configuração.

4.2 Configuração do Acesso à Informação dos Parceiros

Após o término da configuração da topologia começa a segunda fase da configuração da EV. Ela consiste basicamente de duas tarefas: especificar a visibilidade da topologia para cada membro da EV, e selecionar quais informações um determinado membro libera para cada um dos outros membros que fazem parte da EV.

Por definição, uma EV é composta pela união de diferentes empresas. Entretanto, algumas destas empresas, por motivos diversos (por exemplo, por serem concorrentes) podem não querer que outras empresas da mesma EV tenham acesso às suas informações. Assim, antes que cada um dos membros da EV receba uma cópia da topologia, deve haver uma configuração de qual parte da topologia cada um deles pode visualizar. Este processo deve ser idealmente realizado através de uma interface amigável onde, para cada membro da EV, o usuário configurador possa selecionar quais outros parceiros o referido membro pode ver. A Figura 22 ilustra um exemplo conceitual da visão geral do coordenador (a) e da visão restrita de um de seus membros (b).



a) Visão do coordenador b) Visão de um membro

Figura 22 – Visões dos membros da EV.

A segunda tarefa que completa a questão de direitos de acesso à informação, consiste da especificação de quais informações cada um dos membros da EV pode ver de seus respectivos parceiros. Pelos mesmos motivos que levam às restrições de visibilidade da topologia, as informações que cada uma das empresas produzem também devem ser restringidas. Estas informações são representadas na forma de um modelo de referência (MR), que é previamente acordado entre os parceiros no momento da criação ou instalação da plataforma de gestão de EVs (CAMARINHA-MATOS *et al*., 2001). A escolha das informações que compõem o MR pode acontecer de duas maneiras: através da compilação das "meta-informações" extraídas dos repositórios de informações de cada um dos membros da EV, ou adotando um dos modelos de referência para comércio eletrônico existentes no mercado. Estes duas formas de concepção do MR são abordadas em detalhe na seção 4.2.1.

Para restringir o acesso à informação o modelo de referência foi transformado em "cláusulas de supervisão" (RABELO *et al.*, 1999b). Estas cláusulas de supervisão mapeiam cada uma das informações do MR e atribuem a cada uma delas um valor "visível" ou "não visível" para cada membro da EV. O coordenador é quem mantém as cláusulas de supervisão, podendo facilmente consultá-las para saber se pode ou não liberar certas informações requeridas por um dado membro. É importante salientar que a comunicação entre os membros é mediada pelo coordenador, ou seja, não existe comunicação direta entre eles. Assim, para que um membro possa acessar uma informação de outro membro, o primeiro faz uma requisição ao coordenador. Quando a requisição chega ao coordenador, ele verifica nas cláusulas de supervisão se o membro que fez o pedido tem privilégio para receber essa informação. Se o membro requerente tiver privilégio, o coordenador, se não estiver de posse da informação, requisita a mesma para o membro ao qual ela pertence e a

envia para o membro que fez o pedido. O processo de configuração do acesso às informações deve ser feito através de uma interface amigável, onde para cada informação do MR o usuário configurador escolhe se determinado membro pode ou não ter acesso a ela.

Ao final de todo o processo de configuração, incluindo a configuração da topologia e o acesso às informações, o coordenador repassa para cada um dos membros da EV uma cópia da topologia com suas devidas restrições. A partir deste momento a empresa virtual está apta a operar e desenvolver as atividades a que se propõe.

4.2.1 Modelo de Referência da Informação e Cláusulas de Supervisão

Toda a empresa virtual antes de entrar em operação precisa, dentre outras coisas, ter bem definido o conjunto de informações que serão trocadas entre os parceiros. Este conjunto de informações é chamado de modelo de referência das informações. Ele é único para uma plataforma de gestão de EVs, e todos os membros que quiserem trocar informações devem levá-lo em consideração. Como não existe um MR padrão "ISO" das informações necessárias em uma EV, os parceiros – usuários de uma dada plataforma – precisam definir e chegar a um acordo sobre quais são as informações relevantes para fazer parte do modelo. Para chegar a este modelo de referência muitas interações com os membros do consórcio precisam ser feitas. O tipo de negócio realizado pela EV influencia o conteúdo do MR. Por exemplo, se o consórcio fabricar roupas, o modelo terá meta-informações, tais como: tamanho, cor, tipo de tecido, coleção, etc. Entretanto, se outro consórcio fabricar peças para carros, o conjunto de meta-informações será totalmente diferente e incompatível com o modelo anterior. Como possíveis meta-informações têm-se: tipo da peça, tipo do material, pressão máxima suportada, vazão máxima fornecida, etc. Além das diferenças entre os negócios, as informações podem ser diferentes para negócios em um mesmo ramo de atuação. Voltando ao exemplo da indústria automotiva, dependendo do tipo de peças que a EV produza, irão variar as informações intercambiadas entre as empresas. Este fato fica evidente quando são comparados consórcios que fabricam motores com outros que fabricam pneus.

As meta-informações que a plataforma de EVs utiliza para conceber o seu MR são extraídas dos repositórios de dados legados de cada uma das empresas participantes (RABELO *et al*., 2002). Ou seja, são retiradas dos seus sistemas de gerenciamento interno.

Este tipo de sistema é chamado ERP e tem por responsabilidade controlar todos os processos dentro de uma empresa, processos tais como: produção, finanças, vendas, etc. Entretanto, como existem vários sistemas ERPs disponíveis no mercado, empresas que fazem parte da mesma EV podem estar utilizando sistemas ERPs diferentes, o que leva a trabalharem com diferentes modelos de dados. Para resolver o problema da semântica da informação em repositórios de dados diferentes é que surgiu a figura do MR. Assim, todas as meta-informações de todos os diferentes repositórios envolvidos são analisadas e uma síntese das meta-informações compatíveis é feita. Por exemplo, se forem tomados dois repositórios de dados diferentes e comparado o campo "prodNome" do repositório 1 com o campo "nomeProduto" do repositório 2, pode-se concluir que sintaticamente os dois são diferentes, mas semanticamente eles tratam da mesma informação, "nome do produto". Portanto, essa meta-informação deve fazer parte do MR. Este processo deve ser feito para todas as meta-informações presentes em todos os repositórios de dados envolvidos. Ao final dessa análise, o modelo de referência das informações para a plataforma específica de gestão de EVs é concluído.

Outra alternativa, além da apresentada acima, é a adoção de um MR de informação para comércio eletrônico, tal como: ebXML ou EDIFACT. Uma experiência na utilização do EDIFACT como MR das informações intercambiadas entre os parceiros foi feita no projeto PRODNET (CAMARINHA-MATOS *et al*., 1999d). No caso da utilização desse tipo de MR, a tarefa não é construir o modelo propriamente dito, mas sim adequar os repositórios de dados dos sistemas legados das empresas a trabalharem com ele. Sistemas auxiliares para traduzir uma meta-informação do sistema legado para o MR adotado poderiam fazer este processo.

É importante salientar que a problemática envolvida na construção do modelo de referência das informações está além do escopo deste trabalho. Desta forma, partimos do pré-suposto que o MR está definido e a questão é como restringir o acesso às informações dos parceiros pertencentes a uma EV. Através das cláusulas de supervisão estas restrições são feitas e a privacidade das informações é garantida. As cláusulas de supervisão são estruturas mantidas pelo coordenador e organizadas em forma de árvore. A EV é o nó raiz, os seus membros são os nós do primeiro nível, os parceiros de cada membro (que nada mais são do que os outros membros da EV) são os nós do segundo nível, e o terceiro e último nível contém as informações do modelo de referência com um atributo para cada

uma dizendo se a mesma está ou não visível para o referido parceiro. Esta estrutura auxilia o coordenador a saber se o membro que requisitou uma informação tem ou não privilégio para acessá-la. Ela fica armazenada no coordenador que, antes de disponibilizar uma informação de um determinado membro, acessa essa estrutura e consulta se o parceiro que fez o pedido tem ou não privilégio para recebê-la. A Figura 23 mostra a estrutura que compreende as cláusulas de supervisão.

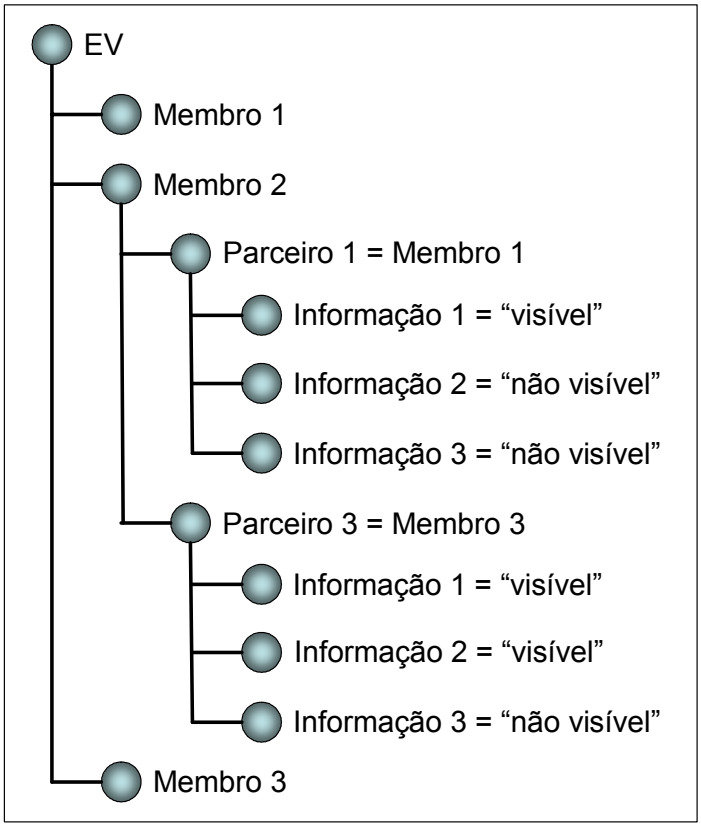


Figura 23 – Estrutura das cláusulas de supervisão.

4.3 Descrição Completa do Processo de Configuração

Após a exposição do modelo de configuração de EVs concebido nesta dissertação, esta seção apresenta uma descrição do fluxo lógico funcional entre as várias entidades que compõem este modelo. O processo será descrito desde seu início, logo após o término da tarefa de busca e seleção de parceiros, até o momento em que se encerra e torna a EV apta à operação. O cenário utilizado para esta descrição passo a passo do processo de configuração é apresentado na Figura 21. Este ambiente apresenta um número N de empresas que têm o sistema multiagente instalado em seus domínios e estão conectadas ao SMA do coordenador. No decorrer do texto são feitas referências às trocas de mensagens

realizadas entre os agentes pertencentes ao cenário. Estas trocas de mensagens estão representadas por setas numeradas apresentadas na Figura 24.

A primeira etapa do processo de configuração se inicia com o usuário configurador acessando a interface do agente Configurador do SMA coordenador, e requisitando a criação de uma nova EV. A partir deste momento, o agente Configurador envia uma mensagem para o agente XML (seta 1) pedindo para o mesmo enviar uma mensagem requisitando quem são os parceiros desta EV. Esta mensagem é enviada ao SMA de cada uma das empresas que estão conectadas ao coordenador (seta 2). A mensagem é interceptada no destino pelo agente XML do SMA de cada uma das empresas que são potenciais parceiros. Este agente por sua vez lê e identifica a mensagem como sendo uma mensagem de busca por parceiro e repassa as informações desta requisição ao agente Empresa (seta 3). Ao chegar ao agente Empresa, esta mensagem é analisada e as informações sobre se a mesma participa ou não da EV são buscadas no repositório de dados do SMA (seta 4). Se o resultado da busca no repositório de dados (seta 5) informar que a empresa não participa da EV, o agente Empresa envia uma mensagem para o agente XML (seta 6) dizendo para ele informar ao SMA coordenador que a empresa em questão não faz parte da EV (seta 7).

Entretanto, se o agente Empresa identificar que a empresa faz parte da EV, ele envia uma mensagem para o agente XML (seta 6) pedindo para que o mesmo envie ao SMA do coordenador uma mensagem informando que a empresa em questão participa da EV (seta 7), e também uma outra contendo as informações básicas da mesma (seta 7). A cada resposta afirmativa recebida pelo SMA coordenador, o agente XML armazena as informações básicas das empresas no repositório do SMA coordenador (seta 8), e envia uma mensagem para o agente configurador informando que existe uma nova empresa na EV (seta 9). O agente Configurador, ao receber esta mensagem do agente XML, busca as informações sobre a referida empresa no seu repositório (seta 10) e a insere na topologia da EV.

Após a chegada da resposta das empresas informando quais efetivamente participam da EV, o agente configurador do SMA coordenador envia uma mensagem para o agente XML (seta 1), pedindo para o mesmo enviar uma mensagem para o SMA das empresas que responderam afirmativamente ao pedido anterior, requisitando suas conexões (seta 2). O processamento desta mensagem pelo SMA das empresas é semelhante ao

ocorrido com a mensagem anterior. Ou seja, o agente XML intercepta a mensagem e envia o pedido para o agente Empresa (seta 3). Este verifica no repositório de dados quais são suas conexões para esta EV em específico (seta 4 e seta 5), as repassa para o agente XML (seta 6), que as envia para o SMA coordenador (seta 7). Chegando ao SMA coordenador, o agente XML repassa as informações das conexões da empresa ao agente Configurador (seta 9) que as insere na topologia da EV.

Terminada a etapa de configuração da topologia, inicia-se a etapa de configuração das restrições de visibilidade e acesso à informação. Este processo é executado manualmente, ou seja, o usuário configurador é quem desempenha esta tarefa. Para cada empresa representada na topologia, o usuário configurador seleciona quais empresas pertencentes à referida EV o membro em questão poderá visualizar a partir do momento que receber sua "cópia" da topologia. Além da restrição de visibilidade da topologia, o usuário configurador também deve configurar o acesso às informações dos membros por parte dos outros membros. Portando, membro a membro, o usuário configurador seleciona quais informações produzidas por este membro em específico cada um dos outros membros poderão ter acesso. O processo completo de configuração é apresentado na Figura 24.

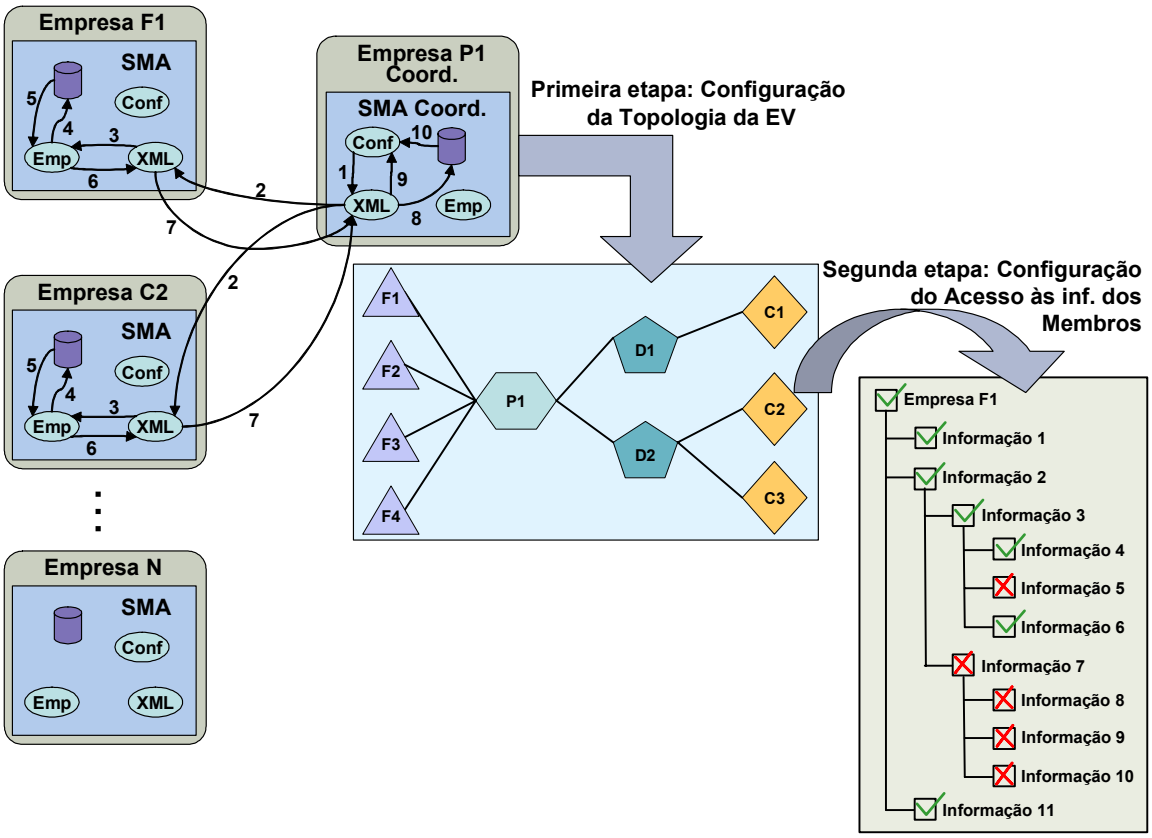


Figura 24 – Visão completa do processo de configuração de EVs.

Após o término desta segunda e última etapa da configuração, todas as informações pertencentes ao processo são armazenadas no repositório de dados do SMA do coordenador, ou seja, a topologia completa da EV e as cláusulas de supervisão. Assim que as informações forem armazenadas, o agente Configurador do SMA coordenador invoca o agente XML a carregar as informações da topologia da EV do repositório de dados, e enviar uma cópia da mesma a cada um dos SMA das empresas participantes, resguardando suas restrições de visibilidade. A partir deste momento a EV estará pronta para operar.

CAPÍTULO 5: Implementação

Este capítulo visa descrever a implementação realizada para validar o modelo conceitual proposto no capítulo 4. Utilizando as tecnologias apresentadas no capítulo 3 uma plataforma foi implementada, testada e validada. Estas tecnologias foram utilizadas como base para o desenvolvimento do projeto DAMASCOS e a escolha de cada uma delas foi feita após um longo trabalho de pesquisa e avaliação por parte de todos os parceiros do projeto. Estas escolhas foram feitas tendo em vista a utilização do que se tinha de mais avançado e robusto em termos de tecnologia da informação disponível até o momento.

De um modo geral, no desenvolvimento da plataforma e dos diferentes agentes, tentou-se utilizar o máximo possível em tecnologias abertas, interoperáveis e padronizadas. Este tipo de tecnologia facilita a interconexão com outros sistemas e arquiteturas que sigam padrões internacionalmente utilizados. Desta forma, buscou-se alcançar um estado o mais próximo do ideal no que diz respeito à utilização em larga escala deste protótipo. No decorrer deste capítulo será apresentada a plataforma na qual está compreendido o sistema e também as tecnologias e ferramentas de suporte ao seu funcionamento.

5.1 Plataforma

A plataforma concebida para validar o modelo proposto vai além do sistema multiagente descrito anteriormente. Esta plataforma agrega outras ferramentas e tecnologias que dão suporte à execução das funções essenciais à configuração da EV. Como ferramentas de suporte, tem-se a utilização de um sistema que auxilia na troca de mensagens entre os sistemas multiagente (a ferramenta WFBB), bem como um repositório de dados para manter as informações relativas à EV de forma consistente e persistente.

Nesta seção será descrito cada um dos componentes que fazem parte da plataforma apresentada na Figura 25, ressaltando suas principais funções e características mais importantes.

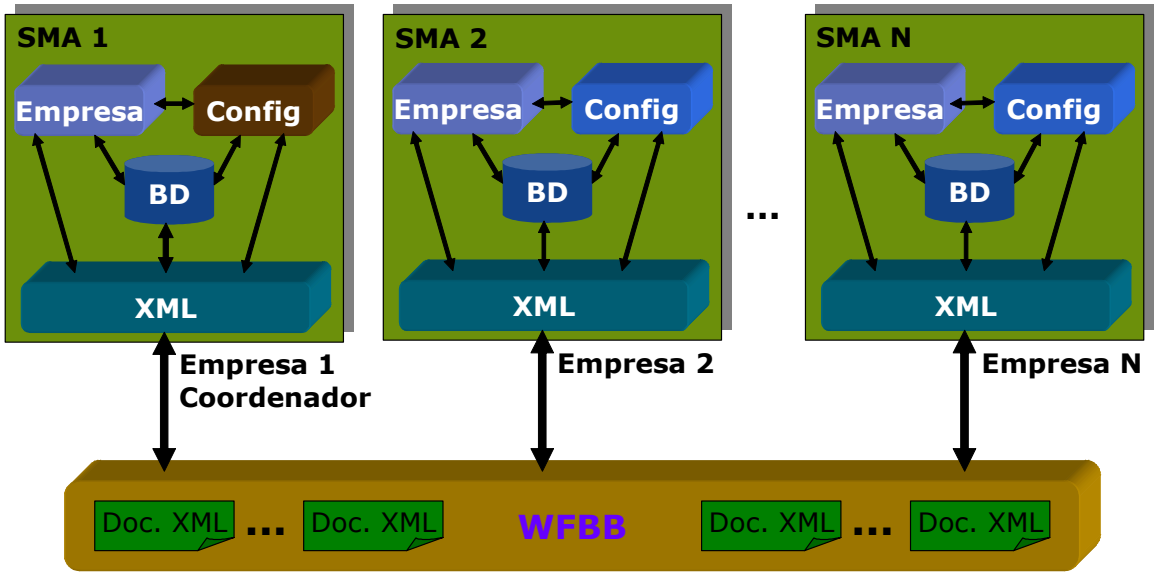


Figura 25 – Plataforma geral do sistema.

5.1.1 O Sistema Multiagente

As funcionalidades relacionadas aos diferentes tipos de tarefas realizadas pelo SMA foram agrupadas por certos critérios de afinidade e encapsuladas dentro de agentes específicos. Assim sendo, de acordo com o descrito no capítulo 4, no modelo proposto há três tipos ("classes") de agentes: Configurador, Empresa e XML. O agente Configurador é "especialista" em configurar empresas virtuais. O agente Empresa encapsula as funções de gerenciamento das informações da empresa. O agente XML é o responsável pelas funções de tradução das mensagens XML trocadas entre as empresas.

Nas próximas subseções serão apresentadas as arquiteturas de cada um dos agentes do SMA, ressaltando suas funções principais e como as informações, tanto de dados como de controle, fluem entre os módulos que compõem cada agente. Além das funções que cada agente contém, também podem ser vistas as entidades externas ao SMA com as quais ele se relaciona. Dentre estas entidades podemos destacar o meio de comunicação que envia e recebe mensagens, o banco de dados que armazena de forma persistente as informações produzidas / intercambiadas, e por último o usuário final que pode ser representado por qualquer usuário autorizado pela empresa, e tem a função de coordenar ou monitorar o processo de configuração e posteriormente gestão da EV.

Além da arquitetura dos agentes, são apresentados também os seus diagramas UML de classes com uma breve descrição das funções que cada uma delas desempenha. Nestes

diagramas foram suprimidos seus atributos e métodos pelo fato de algumas classes, devido sua à complexidade, serem muito extensas. Estas classes foram implementadas utilizando como linguagem de programação o C++ e como ambiente de desenvolvimento o *Borland® C++ Builder™ 5.0 Professional*. Este ambiente de desenvolvimento de aplicações foi escolhido por ser robusto e ter todas as características exigidas para a construção da plataforma. Outro motivo que levou à sua utilização foi o fato do *shell* utilizado para a concepção dos agentes, chamado Massyve-Kit (JACOMINO *et al.*, 1999), gerar agentes codificados nessa linguagem.

No intuito de esclarecer um pouco mais a forma como o SMA foi implementado, o anexo B apresenta alguns detalhes da implementação do mesmo. Neste anexo, pequenos trechos de código que executam uma das tarefas associadas ao processo de configuração de EVs são exibidos acompanhados de breves comentários sobre suas funções.

5.1.1.1 Agente Configurador

O agente Configurador é o responsável por todo o processo de configuração, tanto da topologia da EV, como do acesso às informações dos membros. Este agente é composto por alguns módulos dentre os quais o de maior destaque é o "Módulo de Configuração". Este módulo é o responsável por iniciar o processo de troca de mensagens entre os SMAs dos membros da EV. Este processo pode ser tanto para fazer a configuração com base no acesso às informações de cada SMA, ou com base na interpretação do arquivo de entrada fornecido por uma ferramenta de busca e seleção de parceiros. O Módulo de Comunicação tem apenas os componentes para a comunicação entre os agentes do sistema, visto que o agente XML fica responsável por todas as trocas de informações com o exterior incluindo as ocorridas com os outros sistemas multiagente. Assim, quando algum agente do sistema quiser se comunicar com outro sistema externo, esta comunicação é feita passando as informações pelo agente XML. O agente Configurador também contém as classes dos objetos comuns do SMA. Este conjunto de classes de objetos, chamado de "Modelo de Objetos" do sistema, é único e compartilhado pelos três agentes que compõem o SMA. O Modelo de Objetos, incluindo suas características e funcionalidades, será detalhado na seção 5.2. É através dos objetos pertencentes a este modelo que o agente tem acesso, entre outras coisas, ao banco de dados do sistema. Este banco de dados armazena as informações intercambiadas entre os SMAs bem com as produzidas pelo sistema legado de gestão

(ERP) local, em cada um dos membros. Maiores detalhes sobre as funções do banco de dados serão descritos na seção 5.1.4.

Outro módulo importante do agente Configurador é o "Visualizador Gráfico da EV". Através das informações repassadas pelo módulo de configuração, juntamente com os objetos do "Modelo de Objetos", este módulo responsabiliza-se por disponibilizar de forma gráfica e amigável a topologia da EV para a interface de usuário. A Figura 26 mostra um exemplo dessa interface gráfica. Este exemplo de interface ilustra uma EV com dez membros, onde o membro denominado "Kyaia" está desempenhando o papel de coordenador da EV. Este está conectado diretamente a quatro fornecedores (à esquerda da tela), a um distribuidor (acima à direita) e a três empresas varejistas (à direita). Entre suas conexões é explicitado que há operações de transporte que podem ser posteriormente especificadas para fins de logística. A forma como o usuário interage e utiliza os sistemas que compõem a ferramenta de configuração de EVs é descrita na seção 6.3.

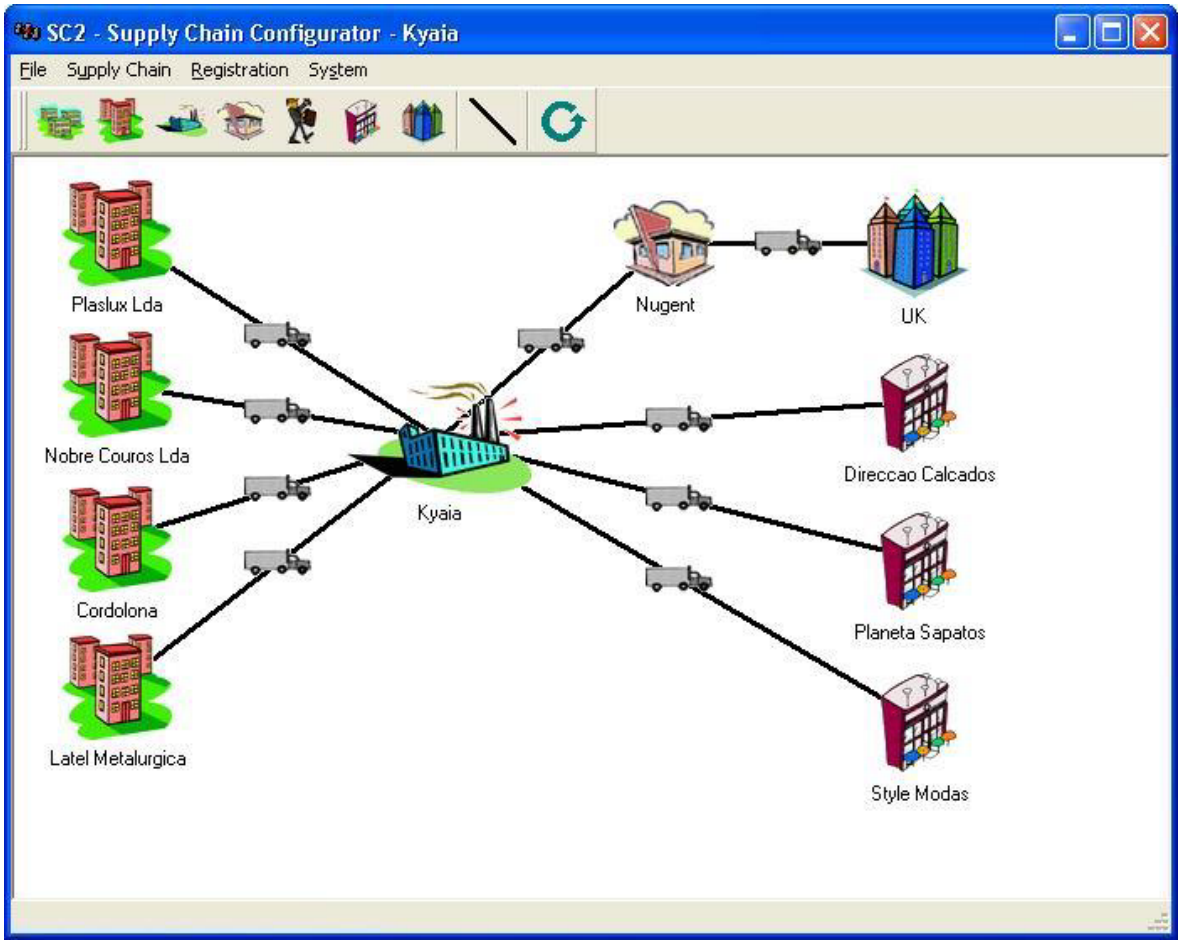


Figura 26 – Exemplo de interface do configurador.

A Figura 27 mostra a arquitetura do agente Configurador. Além da arquitetura do agente com seus componentes, faz-se necessária a visualização das classes que compõem os seus módulos mais importantes. Assim, a Figura 28 apresenta o diagrama UML de classes deste agente cuja a descrição é feita a seguir.

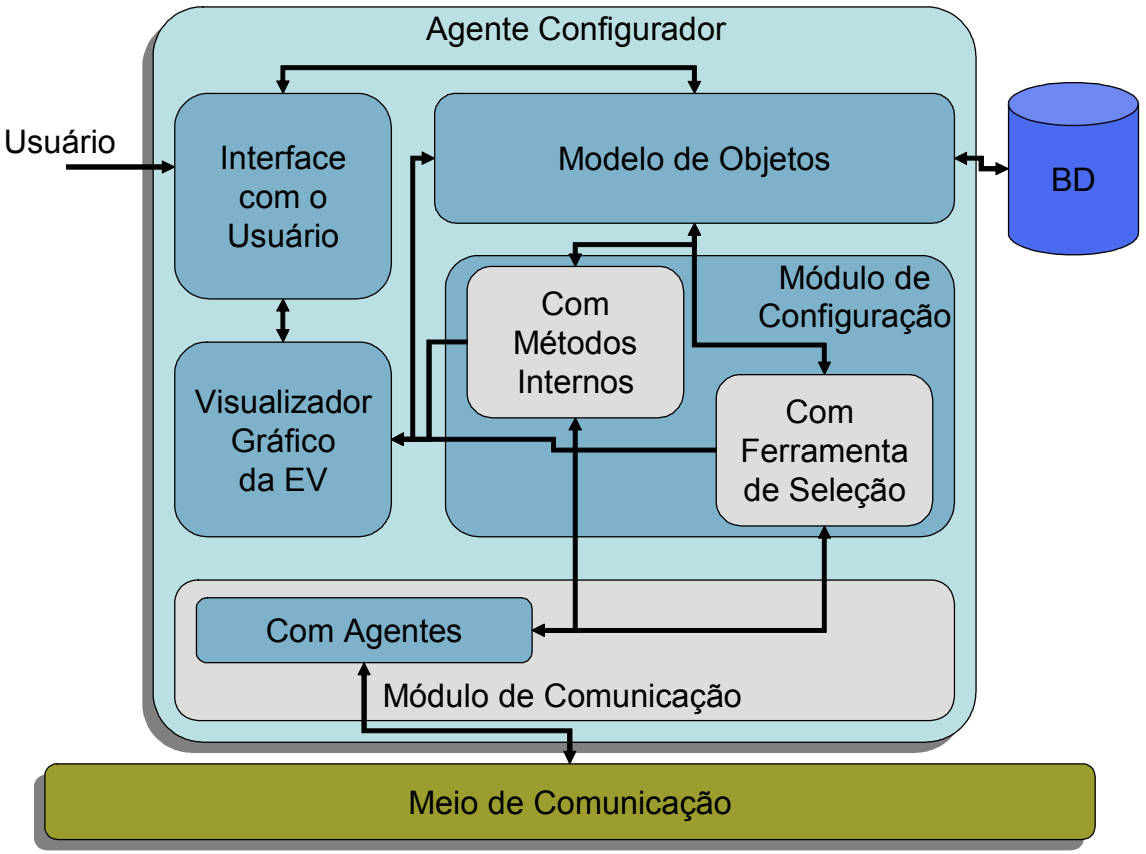


Figura 27 – Arquitetura do agente Configurador.

TSmartMap – É a principal classe relacionada à configuração da topologia da EV. Esta classe implementa as funções de ambos os processos de configuração. Cada objeto desta classe está relacionado com uma lista de imagens / ícones (membros da referida EV) e uma lista de conexões entre estes membros.

TCustomImage – Contém as funções relacionadas à disposição gráfica da imagem de um membro da EV na tela. Além destas funções, esta classe também contém funções para associar qual empresa da EV está relacionada com um dado ícone. Cada imagem / ícone representa um papel diferente dentro da EV.

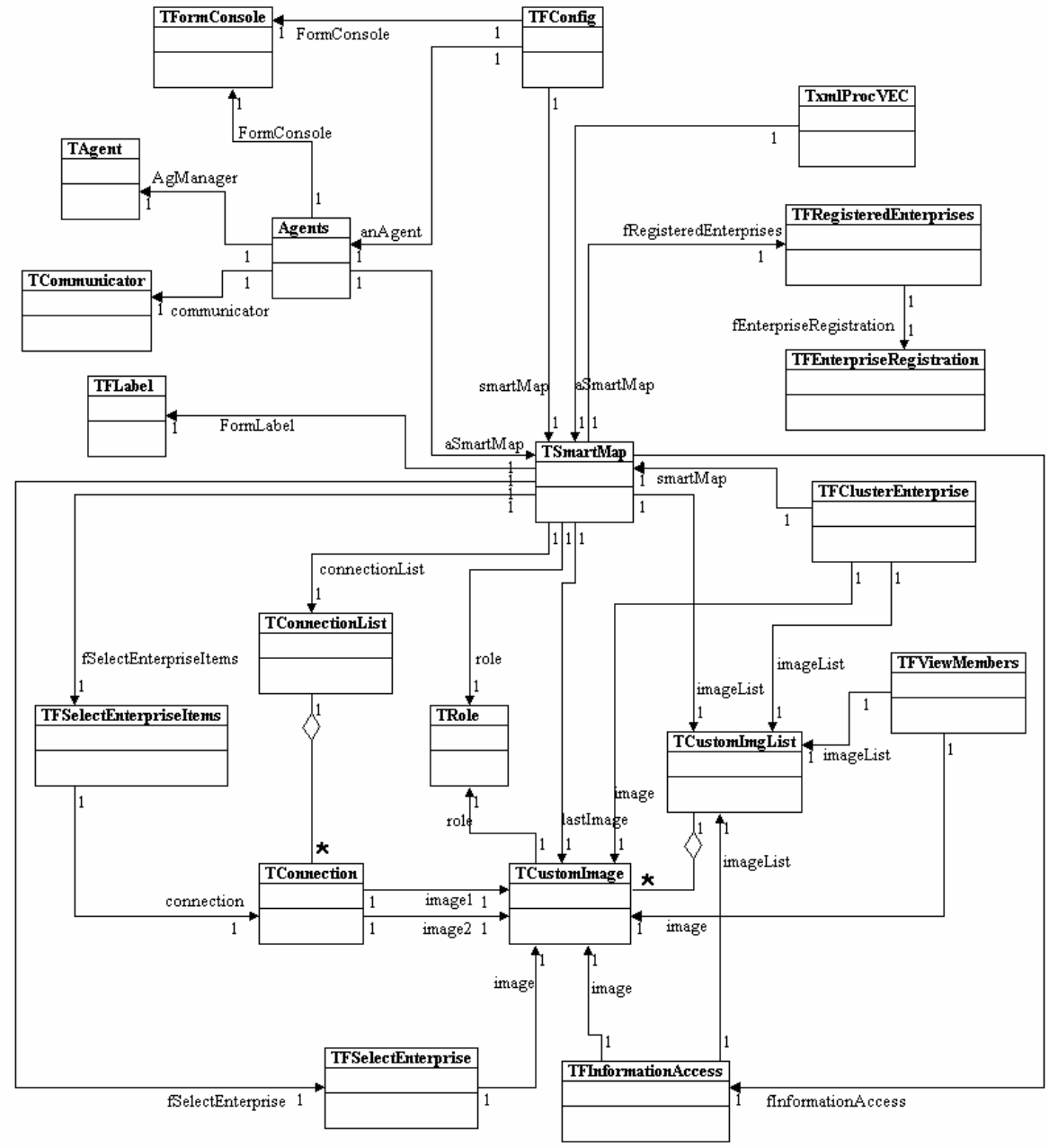


Figura 28 – Diagrama de classes do agente Configurador.

TCustomImgList – Esta classe é uma lista de objetos do tipo *TCustomImage* e está associada à classe *TSmartMap*. Portanto, como uma EV é formada por vários membros, estes membros (objetos *TCustomImage*) são organizados em uma lista de membros da EV, chamada *TCustomImgList*.

TRole – É a classe que define os seis tipos diferentes de papéis possíveis para uma empresa em uma EV. Cada um destes papéis está associado a uma imagem / ícone apresentado na seção 6.2.

TConnection – Contém as funções relacionadas ao processo de criação gráfica de uma reta conectando dois membros da EV. Além destas funções, esta classe também contém funções para configurar quais produtos então sendo intercambiados neste relacionamento.

TConnectionList – Esta classe é uma lista de objetos do tipo *TConnection* e está associada à classe *TSmartMap*. Ou seja, como uma EV é formada por vários membros que estão conectados, estas conexões (objetos *TConnection*) são organizadas em uma lista de conexões chamada *TConnectionList*.

TFConfig – Está é a interface principal do agente de configuração. É através do objeto desta classe que o usuário tem acesso às funções do sistema. Ela serve de porta de entrada para todas as outras funções disponíveis pelo agente. Também é através dela que o usuário visualiza a topologia da EV no momento da configuração.

TFLabel – Tem com função requisitar ao usuário que informe o nome da empresa virtual que está começando seu processo de configuração.

TFSelectEnterprise – Serve com interface para o usuário informar qual empresa está relacionada com uma dada imagem / ícone (*TCustomImage*).

TFSelectEnterpriseItems – Serve como interface para o usuário informar quais produtos estão sendo intercambiados em uma dada conexão.

TFInformationAccess – É através desta interface que o usuário configura quais de suas informações cada um dos outros membros poderá ter acesso.

TFViewMembers – É através desta interface que o usuário configura que parte da EV cada membro terá permissão para visualizar.

TFClusterEnterprise – Serve de interface para o usuário informar quais membros fazem parte de um determinado *cluster*.

TFRegisteredEnterprises – É a interface de visualização da lista de empresas, parceiras em potencial, previamente cadastradas.

TFEnterpriseRegistration – Serve de interface para o usuário cadastrar novas empresas com as quais pretenda iniciar parcerias.

TxmlProcVEC – Esta classe implementa um *parser* XML que é utilizado para criar um documento que contenha a topologia da EV previamente configurada. Implementa também o processo de leitura do documento e posterior exibição na interface gráfica.

5.1.1.2 Agente Empresa

O agente Empresa é o agente responsável por gerenciar as informações em cada empresa que tiver o SMA instalado. Além do gerenciamento, ele também faz o processo de manutenção das informações e serve como concentrador de funcionalidades relacionadas à gestão da EV. Desta forma, novas funcionalidades que tenham como propósito auxiliar na coordenação da EV, tem como local de acoplamento este agente. No projeto DAMASCOS, as funções de coordenação inteligente de EVs do sistema SC^2 foram implementadas dentro deste agente (RABELO *et al.*, 2002).

As informações mantidas pelo agente são informações cadastrais, tais como o seu nome, descrição, endereço, telefone, contatos, lista de produtos, e todas as outras informações relacionadas ao modelo de referência de informações apresentado na seção 4.2.1. As informações cadastrais (nome, endereço e lista de produtos) são as informações que o coordenador, no momento do início da criação da EV, requisita a cada um de seus parceiros. Outra função deste agente, além de manter atualizadas as informações internas da empresa, é a visualização da topologia da EV. Após o término do processo de configuração pelo coordenador, uma cópia da topologia da EV com suas respectivas restrições de visibilidade é enviada a cada um dos parceiros, que visualizam estas informações de forma gráfica através do agente Empresa.

O principal componente deste agente é o módulo “Gerenciador de Informações”. Este módulo tem como tarefa manter as informações da empresa constantemente atualizadas em caso de uma eventual requisição de informações pelo coordenador. Outro módulo deste agente é o “Visualizador Gráfico da EV”, que, como no agente Coordenador, tem por função visualizar a topologia da EV. Este processo de montagem gráfica da EV acontece da seguinte forma: o módulo vizualizador, com o auxílio dos objetos do “Modelo de Objetos” do sistema, acessa o banco de dados e carrega as informações relativas à topologia da EV em questão. Após o processo de carga das informações, este módulo se encarrega em dispor graficamente cada uma das imagens / ícones dos membros, juntamente com suas conexões.

O “Módulo de Comunicação”, assim como no agente Configurador, tem apenas os componentes para a comunicação entre agentes, uma vez que a comunicação com o exterior fica a cargo do agente XML. Este agente utiliza o Modelo de Objetos para ter acessibilidade ao banco de dados. Na Figura 29 é mostrada a arquitetura do agente Empresa com seus módulos componentes e suas inter-relações.

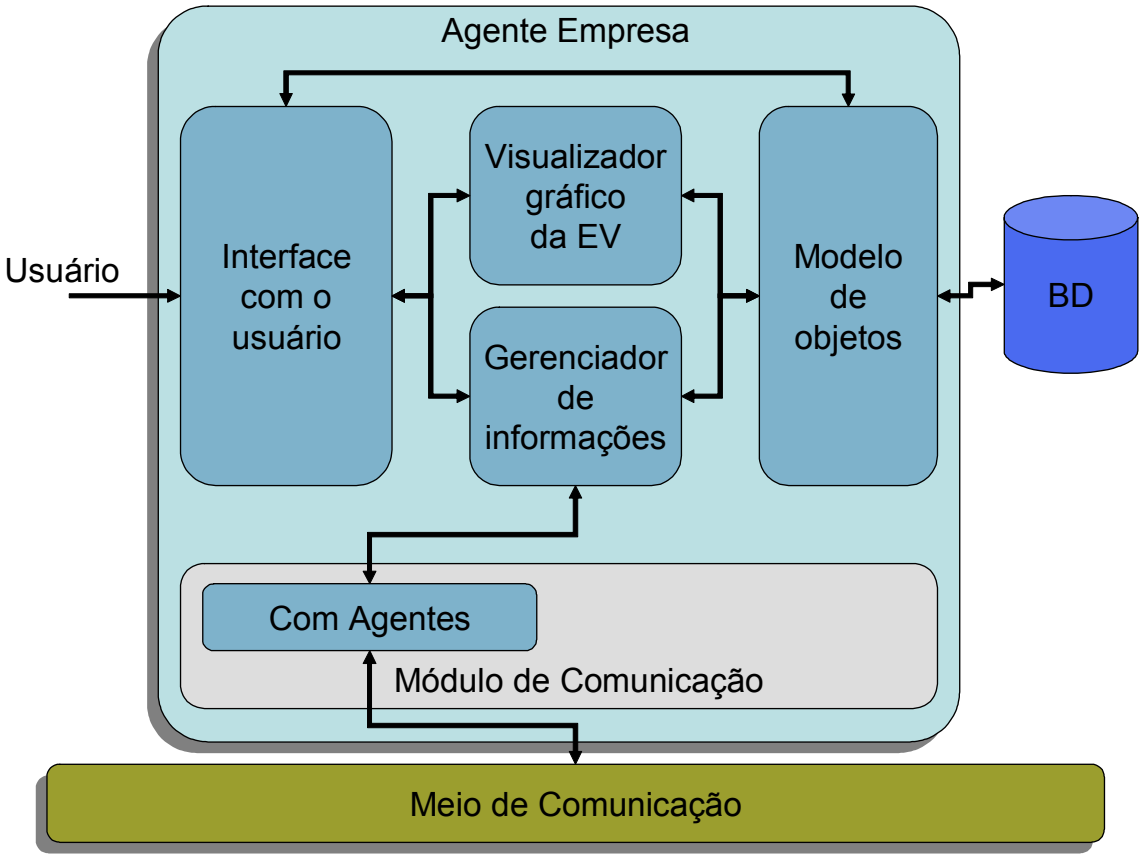


Figura 29 – Arquitetura do agente Empresa.

Os módulos que compõem o agente Empresa são constituídos por classes de objetos que implementam suas funcionalidades. A Figura 30 apresenta o digrama UML de classes do agente e a seguir é feita uma breve descrição de cada uma delas.

TSmartMap – É a principal classe relacionada à exibição da topologia da EV. Esta classe implementa as funções associadas ao processo de dispor as imagens (membros da EV) e suas respectivas conexões de forma conjunta. Cada objeto desta classe está relacionado a uma lista de imagens (*TCustomImgList*) e a uma lista de conexões (*TConnectionList*).

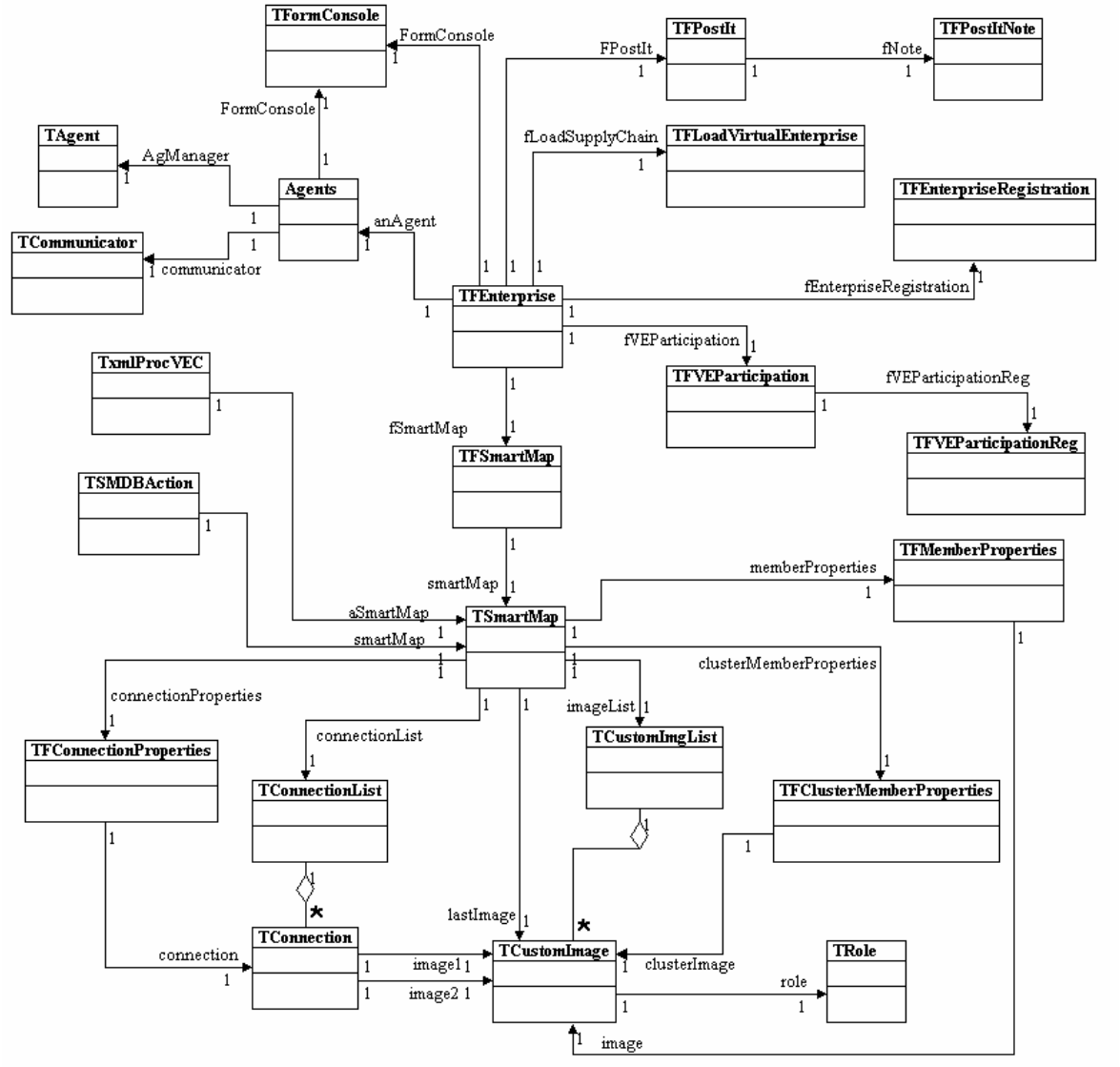


Figura 30 – Diagrama de classes do agente Empresa.

TCustomImage – Contém as funções relacionadas à disposição gráfica da imagem / ícone de um membro da EV na tela. Além destas funções, esta classe mantém atributos que contêm as informações sobre a empresa à qual ela está associada.

TCustomImgList – Esta classe é uma lista de objetos do tipo *TCustomImage* e está associada à classe *TSmartMap*.

TRole – É a classe que define os seis tipos diferentes de papéis possíveis para uma empresa em uma EV.

TConnection – Contém as funções relacionadas ao processo de exibição gráfica de uma reta conectando dois membros da EV. Além destas funções, esta classe mantém

atributos que contêm informações sobre os produtos intercambiados entre as duas empresas que fazem parte da conexão.

TConnectionList – Esta classe é uma lista de objetos do tipo *TConnection* e está associada à classe *TSmartMap*.

TFEnterprise – Está é a interface principal do agente Empresa. É através do objeto desta classe que o usuário tem acesso às funções do sistema. Ela serve de porta de entrada para todas as outras funções disponibilizadas pelo agente. Também é através dela que o usuário visualiza a topologia da EV após receber uma cópia da mesma.

TFLoadVirtualEnterprise – Serve de interface para o usuário escolher qual das EVs previamente cadastradas e armazenadas no banco de dados ele deseja visualizar.

TFSmartMap – É a interface que apresenta a topologia de cada uma das EVs carregadas do banco de dados. Cada objeto *TFSmartMap* fica responsável em mostrar uma EV. Assim, existem tantos objetos desta classe quantas forem as EV carregadas do banco de dados.

TFMemberProperties – É a interface para o usuário visualizar as informações da empresa relacionada com a imagem / ícone em questão.

TFConnectionProperties – É a interface para o usuário visualizar as informações dos produtos intercambiados entre as duas empresas as quais a conexão está ligando.

TFClusterMemberProperties – Serve de interface para o usuário visualizar os membros que fazem parte do *cluster* (seção 6.2).

TFEnterpriseRegistration – Serve de interface para o usuário cadastrar as informações da sua empresa, tais como: endereço, telefone, contato, produtos produzidos ou vendidos, etc. Estas são as informações repassadas para o coordenador da EV no momento da configuração.

TFVEParticipation – Esta é a classe que serve de interface para mostrar quais são as EVs que a empresa em questão está participando.

TFVEParticipationReg – É a classe que suporta a interface para o usuário cadastrar as EVs em que a referida empresa fará parte.

TFPostIt – Suporta a interface para o usuário cadastrar anotações sobre compromissos importantes acerca de uma dada EV.

TFPostItNote – Esta é a classe que suporta a interface para visualizar cada uma das notas cadastradas.

TxmlProcVEC – Esta classe implementa o *parser* XML que é utilizado para ler a topologia da EV de um documento XML e posteriormente exibi-la na interface gráfica.

TSMDBAction – Nesta classe são implementados os métodos que fazem a carga da topologia da EV do banco de dados.

Apesar de várias classes dos agentes Configurador e Empresa terem o mesmo nome, isso não significa que elas contenham a mesma implementação. As classes do agente Configurador têm a funcionalidade voltada à configuração da EV, enquanto as classes pertencentes ao agente Empresa contêm código dirigido à sua exibição.

5.1.1.3 Agente XML

O agente XML é a porta de comunicação do SMA com o meio externo. É através dele que os agentes do SMA recebem e enviam informações para os outros SMAs, ou outros sistemas que se comuniquem por meio de uma interface CORBA. Por concentrar a comunicação com o exterior, é nele que são implementadas as funções de tradução de mensagens. Todas as mensagens que trafegam por ele são codificadas em XML. Portanto, os *parsers* que criam e traduzem estas mensagens são implementados em seu interior, mais especificamente do módulo chamado "Tradutor de Mensagens". Para cada mensagem existe um *parser* implementado que faz o processo de leitura da mensagem para posteriormente transformá-la em um objeto que é armazenado no banco de dados. O processo acontece da seguinte maneira: quando uma mensagem chega ao Módulo de Tradução de Mensagens, o *parser* específico para o processamento desta mensagem é executado e o objeto correspondente do Modelo de Objetos é instanciado. As informações que são lidas do documento XML são associadas aos respectivos atributos do objeto instanciado. Ao final do processo de associação, o objeto é armazenado no banco de dados na forma de um registro que contém exatamente os mesmos campos que são atributos do objeto. Este processo desde a chegada do documento XML até o armazenamento das informações no banco de dados é apresentado na Figura 31.

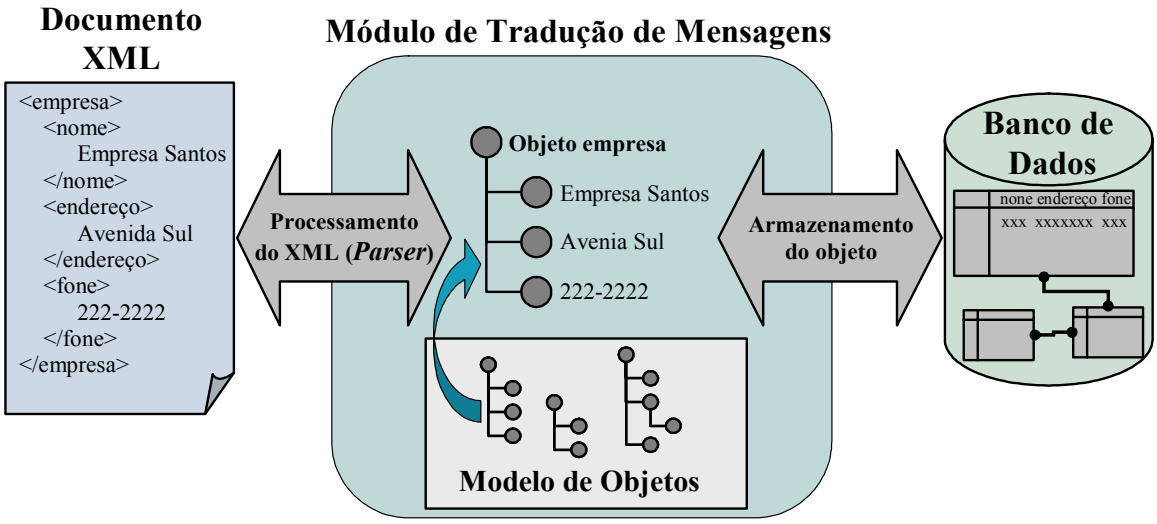


Figura 31 – Processo de tradução das mensagens.

O "Módulo de Comunicação" do agente XML, ao contrário dos módulos de comunicação dos outros agentes, tem dois subconjuntos de componentes. Além dos componentes para a comunicação entre agentes, utilizado para trocar informações entre os agentes do SMA, ele também tem componentes para a comunicação com o exterior, utilizado para trocar informações entre os SMAs através da ferramenta *workflow*. As principais diferenças entre os dois conjuntos de componentes são, respectivamente: a tecnologia de comunicação – um usando SOCKET[26] TCP/IP e o outro CORBA –, e o protocolo de comunicação, onde o primeiro utiliza um protocolo proprietário e o outro usa XML para especificar as mensagens. Nas próximas subseções serão apresentados estes dois tipos de tecnologias utilizadas na plataforma implementada. A Figura 32 apresenta a arquitetura do agente XML e seus componentes.

[26] SOCKET – É uma abstração do sistema operacional que representa o ponto de acesso a um canal de comunicação entre processos. Teve origem no UNIX BSD 4.1 (1982), mas hoje é adotado por quase todos os sistemas operacionais.

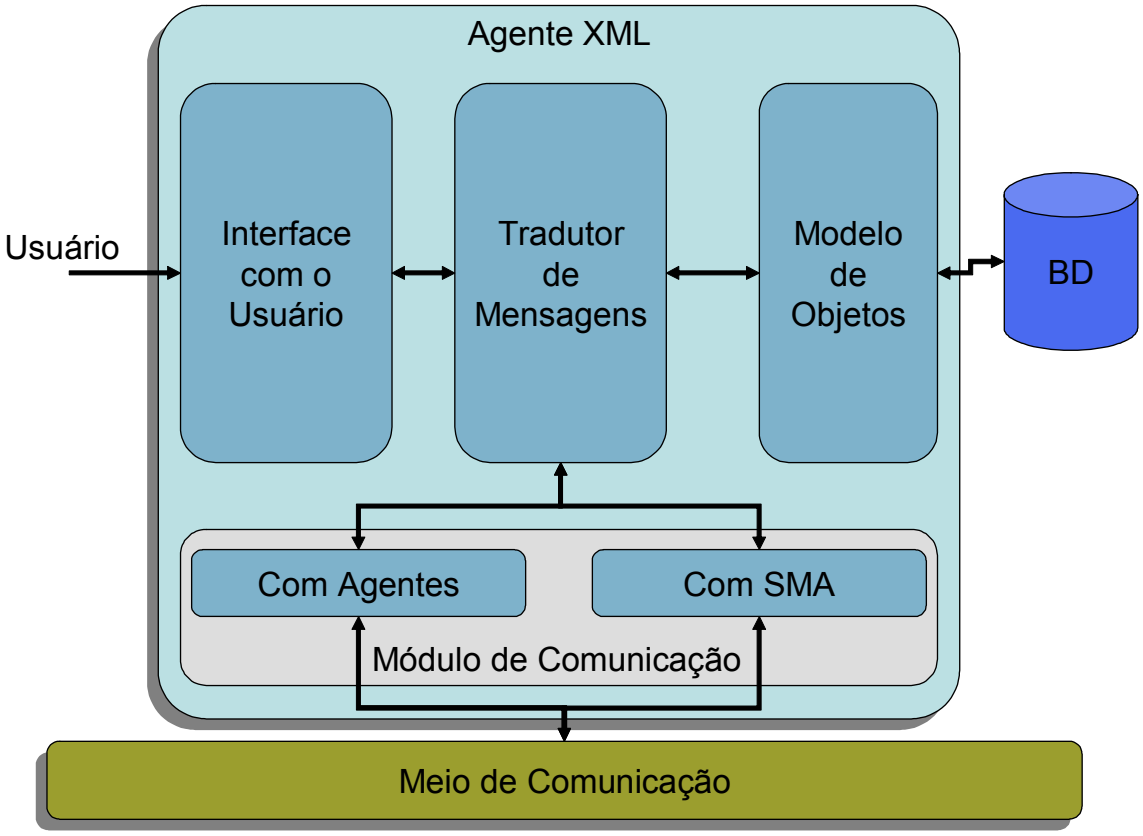


Figura 32 – Arquitetura do agente XML.

Os módulos da arquitetura apresentada acima são implementados em classes que representam o comportamento do sistema como um todo. Abaixo são descritas as classes do agente e o seu diagrama UML de classes pode ser visto na Figura 33.

TxmlProcessor – Esta é a classe principal do agente XML. Esta classe tem por responsabilidade capturar as mensagens enviadas para o SMA e instanciar o *parser* correto relacionado com a tradução de cada tipo de mensagem.

TxmlProcessorBase – Esta é a classe abstrata da qual são derivadas todas as outras classes que processam mensagens XML.

TxmlProcessorVirtualEnterprise – É a classe que implementa o *parser* XML responsável pela criação / tradução da mensagem que contém a topologia da EV.

TxmlProcessorPartnerRequest – Contém o *parser* XML responsável pela criação / tradução do pedido de procura por parceiros.

TxmlProcessorPartnerResponse – Contém o *parser* XML responsável pela criação / tradução da mensagem de resposta do pedido de procura por parceiros.

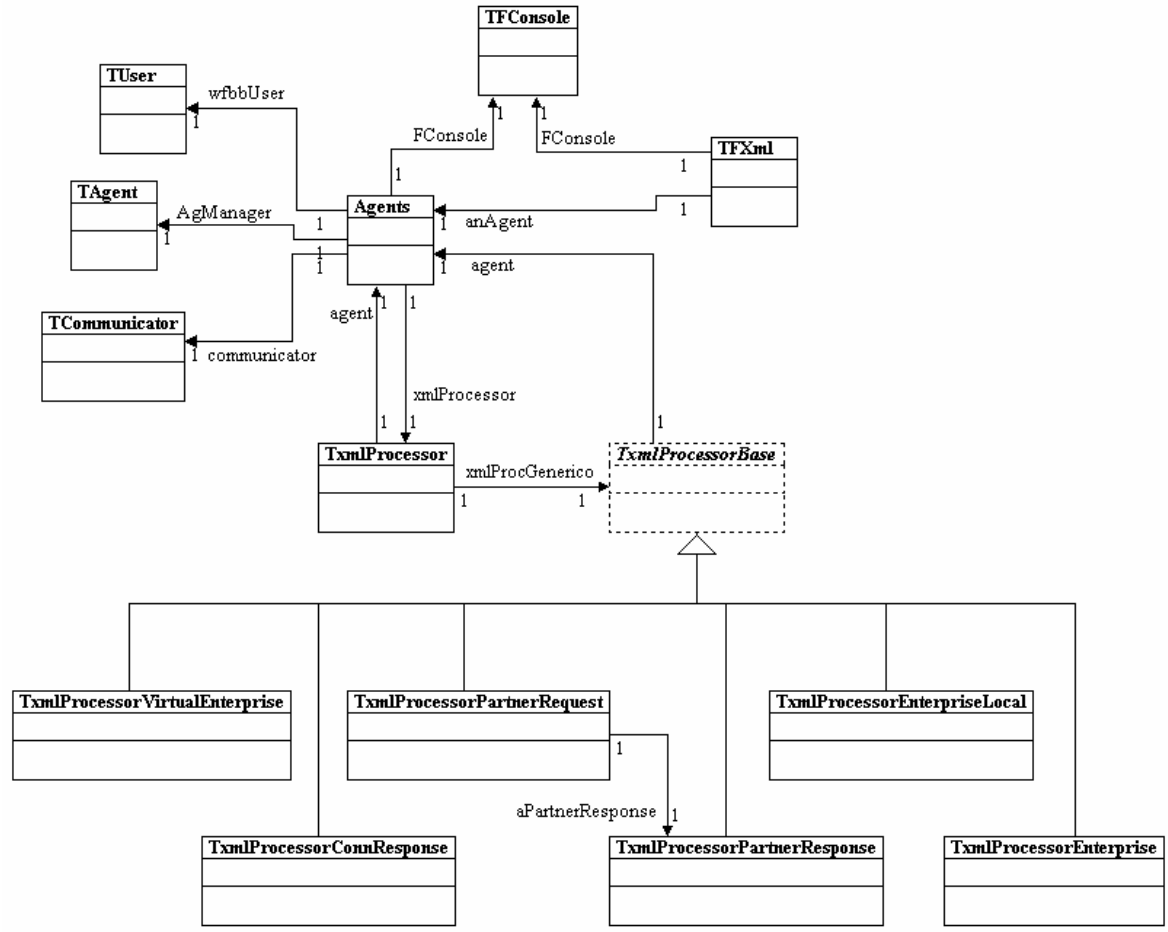


Figura 33 – Diagrama de classes do agente XML.

TxmlProcessorConnResponse – Contém o *parser* XML responsável pela criação / tradução da mensagem de resposta contendo os produtos de uma conexão.

TxmlProcessorEnterpriseLocal – Esta classe implementa o *parser* XML que cria / traduz o documento contendo as informações da empresa local. O conteúdo deste documento consiste nas informações da empresa onde o SMA está instalado, informações estas que serão repassadas ao coordenador no momento da criação de uma nova EV que a mesma faça parte.

TxmlProcessorEnterprise – Esta classe implementa o *parser* XML que cria / traduz o documento contendo as informações de uma empresa cadastrada no banco de dados das empresas potenciais ou efetivos parceiros.

TFXml – Esta é a interface principal do sistema que dá acesso às funções que o mesmo implementa, tais com: enviar ou receber uma mensagem manualmente, ou configurar parâmetros internos para o envio e recepção automáticos de mensagens.

TFConsole – Está é a interface pela qual o usuário pode visualizar o tráfego de mensagens enviadas e recebidas.

5.1.2 O Massyve-Kit

O Massyve-Kit (*Multiagent Agile Manufacturing Scheduling Systems for Virtual Enterprises*) é um ambiente computacional que permite que o usuário, gráfica e interativamente, possa facilmente construir e configurar um sistema multiagente para um domínio de aplicação em particular (RABELO *et al.*, 1999a). A ferramenta Massyse-Kit foi desenvolvida como parte do projeto Massyve (http://www.gsigma-grucon.ufsc.br/massyve/), um projeto KIT/INCO-DC coordenado pela União Européia e que teve como um de seus participantes o laboratório GSIGMA (Grupo de Sistemas Inteligentes de Manufatura) da UFSC.

A escolha desta ferramenta como suporte à criação do sistema multiagente de configuração de empresas virtuais, deu-se pelas características apresentadas acima e também devido à facilidade no acesso e utilização da mesma. Outro fator importante para esta escolha está relacionado ao fato dessa ferramenta ter sido criada pelo mesmo laboratório onde a implementação desta dissertação foi desenvolvida. Portanto, valorizando o trabalho de qualidade desenvolvido pelo laboratório.

Esta ferramenta foi desenvolvida utilizando o compilador *Borland® C++ Builder™ 5.0 Professional*. Este ambiente de programação foi escolhido por uma série de fatores, dentre eles os de maior importância são:

- É uma ferramenta largamente utilizada, o que facilita o suporte e obtenção de bibliografia especializada para solucionar os problemas encontrados, tanto no desenvolvimento, como na utilização do mesmo;
- Constitui-se num ambiente extremamente potente e confiável de programação, adequado à programação de complexos sistemas distribuídos e orientados a objetos.

Os agentes gerados pelo Massyve-Kit são agentes de software, estacionários, distribuídos em um ambiente homogêneo (PC / Windows® NT/2000/XP) e que apresentam as três propriedades primárias de um agente (autonomia, sociabilidade e independência). Estes agentes são modelados como objetos, inspirados no paradigma de programação

orientado a agentes (RABELO *et al.*, 1999a). Além disso, dado que o Massyve-Kit visa suportar a geração de sistemas distribuídos ou multiagente particulares, compete ao usuário especificar e programar, tanto o protocolo de cooperação que deva ser adotado na solução particular, assim, como os vários "comportamentos" que os agentes deverão exibir (JACOMINO *et al.*, 1999).

O Massyve-Kit oferece um ambiente de suporte à geração da infraestrutura, bem como um conjunto de funcionalidades de base para a comunicação em um sistema multiagente / distribuído em particular. Pode-se dizer que o Massyve-Kit provê, em linhas gerais, o "alicerce" do sistema e os serviços da camada de cooperação dos agentes, ficando a cargo do usuário a implementação dos métodos propriamente ditos, associados à aplicação em particular.

As funcionalidades de base para construção de agentes são fornecidas pelo Kit em forma de classes. Estas classes encapsulam todas as ações relacionadas ao processo de transformar um sistema "tradicional" em um agente Massyve. No caso específico do processo de criação dos agentes do sistema de configuração de EVs, foi feita a incorporação das classes criadas pelo Massyve-Kit ao sistema, transformado-o assim em um SMA. As classes que implementam um agente Massyve podem ser vistas a Figura 34 e a descrição de cada uma delas segue abaixo.

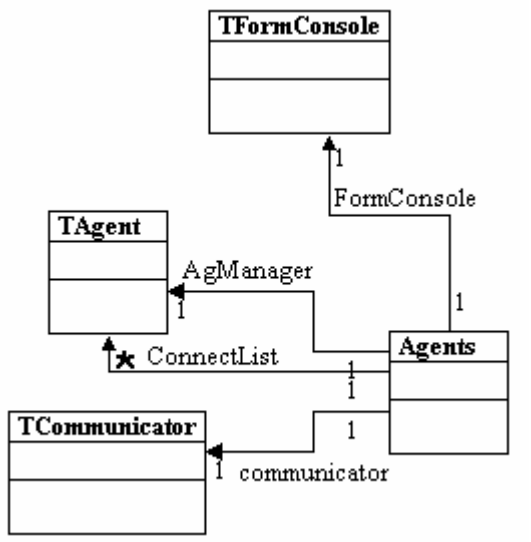


Figura 34 – Classes do agente Massyve.

Agents – Esta classe implementa um agente. É nela que estão implementadas as funções de tratamento das mensagens trocadas entre os agentes.

TAgent – Representa a relação que um agente tem com os outros agentes do SMA.

TCommunicator – Implementa a camada de comunicação entre os agentes, usando SOCKET.

TFormConsole – Esta classe implementa a interface de exibição das mensagens que estão sendo trocadas entre os agentes.

5.1.3 O Workflow Backbone

O *Workflow Backbone* (WFBB), de um modo geral, é um sistema responsável por propiciar a troca de informações entre uma rede de empresas e por coordenar o fluxo de trabalho / informação ao nível de processos (FERREIRA *et al.*, 2000a). No âmbito específico desta dissertação, o WFBB é a ferramenta responsável por transmitir, monitorar e gerenciar o tráfego de informações entre os SMAs instalados nas empresas que participam da EV. Seu objetivo principal é facilitar a integração e a interoperabilidade entre estas empresas. Este objetivo pode ser subdividido em:

- Prover a habilidade de encaminhar informações para uma empresa em específico;
- Garantir a transmissão e recepção das informações;
- Prover facilidades de gerenciamento para monitorar e controlar a transmissão e recepção das informações;
- Prover um canal seguro para a transmissão e recepção das informações;
- Prover um ambiente robusto com capacidades de tolerância à faltas.

A ferramenta WFBB foi desenvolvida pelo Instituto INESC-Porto Portugal, um dos parceiros do projeto DAMASCOS. O WFBB provê dois tipos de serviço: o de troca de mensagens (*public/subscribe service*) e o de gerenciamento de processos *workflow (workflow facility*). O primeiro fornece um mecanismo de troca de mensagens definindo uma fila de mensagens para cada usuário cadastrado, enquanto o segundo é uma ferramenta de WFMS (*workflow management system*) que define, executa e controla os processos dos usuários (FERREIRA *et al.*, 2000b). Estes serviços estão disponíveis através de interfaces especificadas em IDL/CORBA, o que facilita o acesso e utilização remota dos mesmos por parte das aplicações externas.

Esta ferramenta foi desenvolvida em *Java* e utiliza como implementação de *middleware* de comunicação o ORB ORBacus (http://www.orbacus.com). Entretanto, o agente XML que é o responsável por conectar o SMA ao WFBB, foi escrito em C++ e utiliza como implementação de *middleware* o ORB ACE-TAO (http://www.cs.wustl.edu/~schmidt/TAO.html). Uma vez que ambos os *middlewares* seguem as especificações CORBA, a integração entre a ferramenta WFBB e o SMA, proporcionada pelo protocolo de interoperabilidade entre ORBs chamado IIOP, foi simplificada. Portanto, a heterogeneidade entre as linguagens de programação e até mesmo entre os *middlewares* de comunicação utilizados no desenvolvimento de cada uma as aplicações foi uma dificuldade, mas não um empecilho para a sua interconexão.

Após uma análise das necessidades em termos de interconexão entre os SMAs da plataforma de configuração, chegou-se à conclusão que apenas o serviço de troca de mensagens da ferramenta WFBB seria suficiente para atender os requisitos do projeto.

Este sistema de troca de mensagens é constituído por um gerenciador de usuários e um gerenciador de filas de mensagens para estes usuários. O gerenciador de usuários provê funções para a criação, configuração e exclusão de usuários. Podem ser considerados como usuários quaisquer outras aplicações que desejem utilizar este sistema com ferramenta de troca de mensagens. O gerenciador de filas, além de encaminhar as mensagens recebidas para a fila correta, mantém a ordenação e consistência destas mensagens. Para cada novo usuário cadastrado no WFBB, uma fila de mensagens é criada e associada a ele. Assim, as mensagens enviadas ao WFBB endereçadas a um usuário, são armazenadas na fila pertencente àquele usuário, e ficam à disposição para serem lidas pelo mesmo. Este tipo de abordagem funciona como uma caixa postal, onde qualquer usuário pode mandar mensagens para qualquer caixa postal, mas apenas o dono da chave é quem lê as mensagens postadas para ele.

Através da utilização desta abordagem o modelo de troca de mensagens fica mais flexível, pois não há a necessidade do emissor e do receptor estarem executando ao mesmo tempo. Ou seja, o emissor envia a mensagem e o receptor lê a mesma no momento em que entra em execução, o que torna o processo de troca de mensagens assíncrono. Questões relacionadas à sincronização de mensagens devem ser tratadas pelas aplicações que utilizam a ferramenta.

No escopo do projeto, o sistema WFBB foi utilizado da seguinte maneira: cada empresa participante do consórcio tem seu nome cadastrado como um usuário do WFBB. Assim, o SMA instalado em cada parceiro se conecta ao WFBB, fazendo a autenticação de seu nome e senha, e a partir deste momento fica apto a enviar / receber informações para / de qualquer um dos parceiros da EV.

5.1.4 O Banco de Dados

Para manter de forma consistente as informações trocadas entre os SMAs e também as produzidas por cada um dos agentes do sistema, um sistema de gerenciamento de banco de dados (SGBD) foi utilizado para este propósito. Dentre os SGBDs relacionais pesquisados, o que trouxe a melhor relação custo / benefício foi o *Interbase*®.

O *Interbase*®, na sua versão 6.01, é um poderoso banco de dados Cliente/Servidor relacional compatível com SQL-ANSI-92[27], que foi desenvolvido para ser um banco de dados independente de plataformas, de sistemas operacionais, além disso seu código fonte é aberto e sua licença é pública. Seu desenvolvimento iniciou em meados de 1985 por uma equipe de engenheiros da DEC (*Digital Equipament Corporation*) tendo como nome inicial *Groton*. Esse produto veio sofrendo varias alterações até finalmente em 1986 receber o nome de *Interbase*®, iniciando na versão 2.0. Nesta época, a idéia era produzir um SGBD que oferecesse benefícios não encontrados em outros SGBDs da época.

Após um profundo estudo das informações utilizadas pelo sistema de configuração de EVs, um modelo constituído de tabelas e relacionamentos entre elas foi construído. Abaixo segue a lista de tabelas do sistema com uma breve descrição do seu conteúdo. A visão lógica do banco de dados é apresentada na Figura 35.

TABLEENTERPRISE – Esta tabela contém as informações cadastrais relativas às empresas que fazem parte das EVs configuradas pelo coordenador, e também das empresas parceiras em potencial. Quando o coordenador da EV difunde a requisição de parceiro de uma determinada EV, as empresas que responderem afirmativamente a esse pedido têm suas informações cadastrais armazenadas nesta tabela.

---

[27] SQL – *Structured Query Language*.
ANSI – *American National Standard Institute*.

TABLEPRODUCT – Contém as informações relacionadas aos produtos de cada uma das empresas cadastradas na tabela TABLEENTERPRISE.

TABLEENTERPRISELOCAL – Esta tabela contém as informações básicas da empresa onde o SMA está instalado. Portanto, cada uma das empresas que tiver a plataforma instalada terá que cadastrar suas informações nesta tabela. Estas são as informações que cada parceiro envia ao coordenador no momento em que o mesmo inicia a configuração de uma nova EV.

TABLEPRODUCTLOCAL – Contém as informações relacionadas aos produtos da empresa cadastrada na tabela TABLEENTERPRISELOCAL.

TABLEVIRTUALENTERPRISE – Contém as informações relacionadas à topologia das EVs previamente configuradas.

TABLEMEMBER – Contém as informações, relativas à topologia, de cada um dos membros da EV, tais como: com qual empresa está vinculado, qual seu papel, qual sua posição "x,y" no plano cartesiano da topologia, etc.

TABLECONNECTION – Essa tabela contém as informações relacionadas às conexões. Dentre estas informações, as mais importantes são o membro fonte e o membro destino da conexão.

TABLECONNECTIONITEM – Contém informações sobre quais produtos são intercambiados em cada conexão entre os membros da EV.

TABLEMETAINFORMATION – Contém a descrição das meta-informações pertencentes ao modelo de referência de informação da EV.

TABLEVISIBLEINFORMATION – Esta tabela indica quais informações do modelo de referência cada membro disponibiliza para cada um de seus parceiros.

TABLEVEPARTICIPATION – Esta tabela guarda as informações sobre quais EVs a referida empresa está ou vai participar. Quando o coordenador difunde a requisição por parceiros, é nesta tabela que cada empresa pesquisa para saber se ela faz parte da EV.

TABLEVEPARTICIPATIONPRODUCT – Informa quais produtos estão ou serão intercambiados pela referida empresa em uma determinada EV. Estas são as informações repassadas ao coordenador no momento em que o mesmo difunde uma mensagem de requisição de conexões.

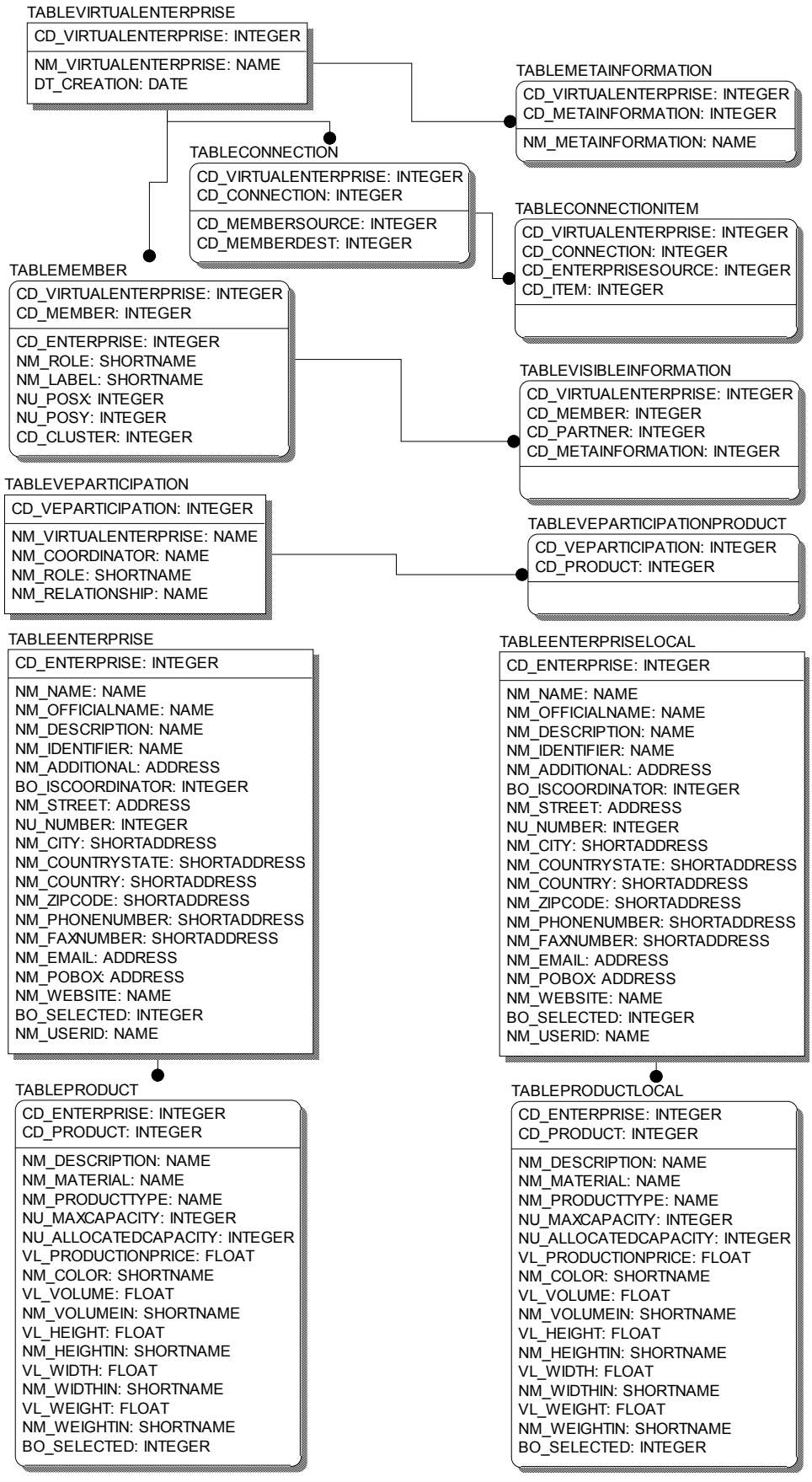


Figura 35 – Visão lógica do banco de dados.

5.2 Modelo de Objetos do Sistema

O modelo de objetos é um conjunto de classes de objetos comuns compartilhadas por todos os agentes do sistema. Seu objetivo é facilitar o acesso à informação armazenada no banco de dados, bem como na manipulação dos dados relacionados ao processo de configuração. Este modelo foi construído de forma a que todos os agentes do sistema possam ter a mesma visão das informações manipuladas pelo SMA. Por exemplo, quando um agente deseja acessar as informações de um membro da EV, ele utiliza uma instância da mesma classe que os outros agentes do sistema utilizariam para ver a mesma informação.

Estes objetos, além de representarem em seus atributos as informações relacionadas ao sistema de configuração, também têm a característica de acessar o banco de dados e as armazenar de forma persistente. Uma parte do banco de dados contém tabelas que contemplam as informações contidas no modelo de objetos do sistema. Através da utilização do *driver* correspondente, cada objeto tem a capacidade de acessar o banco de dados para carregar, gravar, atualizar ou excluir informações, mantendo assim a consistência dos dados do sistema. Como a implementação deste modelo foi concebida de maneira a dar a possibilidade de utilização de outros bancos de dados do tipo relacionais, por exemplo: *ORACLE®*, *SQL Server™*, *Postgre SQL*, *MySQL*, etc., as instruções de código relacionadas a cada banco de dados em específico são incluídas em classes derivadas da classe que contém os atributos e procedimentos comuns. Ou seja, é criada uma classe filha, derivada de cada uma das classes do modelo, com o nome da classe pai mais o nome do banco de dados sendo utilizado. Atualmente apenas o banco de dados *Interbase®* está sendo utilizado; assim só existem classes derivadas com o nome "classe pai" + Interbase.

Abaixo são descritas as classes do modelo de objetos do sistema e seu diagrama UML de classes pode ser visto na Figura 36.

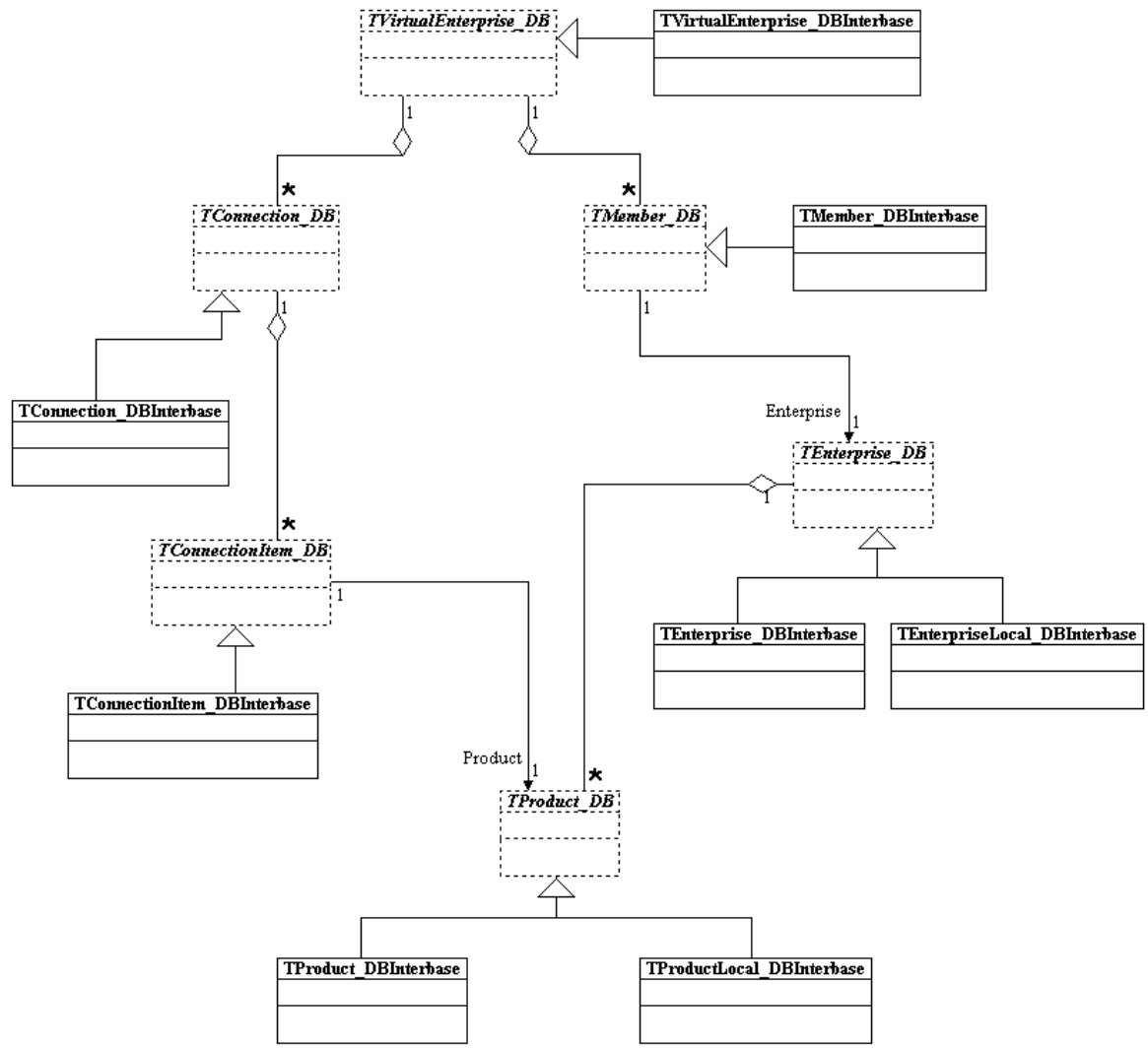


Figura 36 – Diagrama de classes do modelo de objetos.

TEnterprise_DB – Esta classe representa uma empresa e contém uma lista de produtos.

TProduct_DB – Esta classe representa um produto e contém atributos, tais como: nome, descrição, cor, tamanho, etc.

TVirtualEnterprise_DB – Esta classes está associada a uma empresa virtual e contém uma lista de membros e conexões.

TMember_DB – Esta classe representa um membro da empresa virtual com atributos, tais como: nome da empresa, papel na EV, posição "x,y" no plano cartesiano da topologia.

TConnection_DB – Representa uma conexão entre dois membros e contém a lista de produtos desta conexão.

TConnectionItem_DB – Esta classe representa um produto que faz parte de uma conexão.

5.3 Arquivo de Entrada para a Configuração

A primeira das duas formas de configuração da topologia proposta no modelo, necessita de um arquivo de entrada com as informações sobre os membros e suas conexões para iniciar o processo de configuração. Para satisfazer esta necessidade, foi concebido um arquivo que contém a especificação de todas as informações necessárias para este processo.

Para facilitar a interoperação com as ferramentas de busca e seleção de parceiros, o conjunto de informações necessárias para a configuração da topologia da EV foi codificado em XML, através da concepção de um DTD (seção 3.3.2). Este DTD especifica somente as informações estritamente pertinentes ao processo. Portanto, devido à pequena quantidade de informações exigidas, mais ferramentas de busca e seleção de parceiros poderão ser utilizadas como provedoras de informações para o processo de configuração da topologia.

A informação especificada pelo DTD está dividida em quatro partes, tendo como elemento raiz do documento o elemento *virtual_enterprise*. As partes que compõem o DTD são descritas abaixo e a especificação completa do documento é apresentada na Figura 37.

- **Informações sobre a empresa virtual**: Esta parte do documento especifica o código e o nome da empresa virtual sendo configurada;
- **Informações sobre os membros**: Especifica quais empresas fazem parte da EV e quais são seus papéis;
- **Informações sobre as conexões**: Especifica quais membros estão relacionados uns com os outros e que produtos estão sendo intercambiados;
- **Informações sobre as empresas**: Especifica as informações básicas da empresa bem como os produtos por ela comercializados. É importante ressaltar que algumas informações contidas nesse trecho da especificação são opcionais. Isto significa que o processo de configuração pode ocorre mesmo sem parte desta informação.

```
<!-- Virtual Enterprise Information -->
<!ELEMENT virtual_enterprise (code,name,members_list,
                connections_list,enterprises_list)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT name (#PCDATA)>

<!-- Member Information -->
<!ELEMENT members_list (member*)>
<!ELEMENT member (cd_member,cd_enterprise)>
<!ATTLIST member role
        (pre_supplier |supplier |principal_producer
        |distributor |broker |retailer |cluster ) #REQUIRED>
<!ELEMENT cd_member (#PCDATA)>
<!ELEMENT cd_enterprise (#PCDATA)>

<!-- Connection Information -->
<!ELEMENT connections_list (connection*)>
<!ELEMENT connection (member_src,member_dest,items_list)>
<!ELEMENT member_src (#PCDATA)>
<!ELEMENT member_dest (#PCDATA)>
<!ELEMENT items_list (item+)>
<!ELEMENT item (enterprise_src,cd_item)>
<!ELEMENT enterprise_src (#PCDATA)>
<!ELEMENT cd_item (#PCDATA)>

<!-- Enterprise Information -->
<!ELEMENT enterprises_list (enterprise*)>
<!ELEMENT enterprise (cd_enterprise,nm_name,nm_phone,
                nm_street?,nm_city?,nm_state?,nm_country?,products_list)>
<!ELEMENT nm_name (#PCDATA)>
<!ELEMENT nm_phone (#PCDATA)>
<!ELEMENT nm_street (#PCDATA)>
<!ELEMENT nm_city (#PCDATA)>
<!ELEMENT nm_state (#PCDATA)>
<!ELEMENT nm_country (#PCDATA)>
<!ELEMENT products_list (product*)>
<!ELEMENT product (cd_product,nm_description,
                nm_producttype,nm_color?)>
<!ELEMENT cd_product (#PCDATA)>
<!ELEMENT nm_description (#PCDATA)>
<!ELEMENT nm_producttype (#PCDATA)>
<!ELEMENT nm_color (#PCDATA)>
```

Figura 37 – DTD para configuração da topologia da EV.

5.4 Troca de Mensagens

Um dos objetivos deste trabalho é manter a confiabilidade na troca de informações entre as empresas participantes. Entretanto, para obter essa confiabilidade tem-se que satisfazer dois requisitos: um compreende o modo ou "como" as informações são intercambiadas; o outro é relativo à forma ou "quais" informações são efetivamente trocadas.

Para satisfazer o primeiro requisito, uma interface CORBA para a publicação e obtenção de informações do WFBB foi utilizada. Esta interface está disponível na ferramenta de *workflow* e pode ser acessada por qualquer aplicação que tenha uma camada de comunicação CORBA. Para satisfazer a segunda necessidade, a metalinguagem XML foi utilizada para caracterizar completamente a informação trocada. A interface CORBA do WFBB provê um conjunto apropriado de funções para o suporte a troca de fluxo de dados em XML. Desta forma, as informações em XML intercambiadas entre as empresas devem satisfazer os DTDs que foram definidos levando em consideração as informações relevantes ao processo de configuração.

Este tipo de mecanismo de troca de mensagens utilizado entre os SMAs, torna a plataforma mais robusta e confiável no âmbito de uma rede de computadores aberta e de larga escala como é o caso da Internet. Entretanto, quanto se fala na comunicação entre os agentes do SMA, por estarem sendo executados em um ambiente fechado, ou seja, dentro de uma rede coorporativa, todo o peso adicional atrelado à troca confiável de mensagens atribuído ao CORBA pode ser prescindido. Assim, a forma leve e rápida de troca de mensagens proporcionada pelo ambiente de desenvolvimento de sistemas multiagente Massyve-Kit foi utilizada. O Massyve-Kit utiliza um protocolo próprio de mensagens, trocando-as através de SOCKET. Este protocolo é composto por mensagens curtas onde cada informação é separada por vírgulas, e seu conteúdo pode ser facilmente expandido pelo desenvolvedor do sistema multiagente para contemplar suas necessidades de comunicação.

5.4.1 Protocolo entre SMAs

A comunicação entre os SMAs instalados nas empresas participantes é suportada pela ferramenta de troca de mensagens chamada WFBB. Como mencionado anteriormente, cada empresa deve estar cadastrada no WFBB para obter uma fila de mensagens e assim estar apta a trocar informações. Sua tecnologia de comunicação é o CORBA, o que torna o processo mais seguro e confiável, principalmente com a utilização de mecanismos como o de tolerância à faltas. Devido ao tempo relativamente curto para o desenvolvimento desta dissertação, as vantagens advindas da utilização da tolerância à faltas não puderam ser exploradas. Ao contrário da comunicação entre os agentes, a comunicação entre os sistemas multiagente requer um protocolo de mensagens que suporte de forma eficiente um conjunto maior de informações. Por exemplo, quando uma empresa responde

afirmativamente a sua participação na EV em configuração, ela necessita enviar suas informações básicas para o configurador, o que significa informar seu endereço, contatos e lista de produtos. Para assegurar que as informações estão no formato desejado, o protocolo foi escrito utilizando a metalinguagem XML, que através da utilização do seu respectivo DTD permite que um documento seja validado e assim garantir que a informação está no formato correto.

A ferramenta WFBB tem dois serviços (CORBA) para o envio de mensagens e apenas um para a recepção. O primeiro serviço de envio é o *push()*, que tem apenas um parâmetro e informa a mensagem a ser enviada. Estas mensagens são enviadas para o usuário designado como padrão (*default*) do WFBB. O segundo serviço de envio de mensagens, chamado *reply()*, tem dois parâmetros: o primeiro indica o usuário que vai receber a mensagem e o segundo a mensagem propriamente dita. O serviço de recepção *pull()* não tem parâmetros, sua função é retornar a primeira mensagem da fila do usuário de chamou o serviço. O WFBB não tem serviço para a difusão de mensagens (*broadcast*), assim, para que uma mensagem seja difundida, o emissor precisa enviá-la a cada um dos usuários cadastrados no WFBB.

As mensagens trocadas entre os SMAs são "um-para-um" ou "um-para-muitos". As mensagens um-para-um ocorrem principalmente entre um membro e o coordenador da EV. As mensagens um-para-muitos são utilizadas pelo coordenador no momento em que ele deseja difundir uma mensagem para todos os membros da EV. A Tabela 2 mostra o protocolo de mensagens de alto nível entre os SMAs e suas respectivas direções (origem, destino). A documentação completa com o formato das mensagens especificadas em XML está descrita no anexo A.

Mensagem: PartnerRequest	**De**: SMA coordenador	**Para**: SMAs membros
Descrição: Esta mensagem é enviada duas vezes para os parceiros. Na primeira vez o coordenador requisita quais são os membros escolhidos, enquanto na segunda ele requisita suas conexões. Em cada uma das mensagens ele leva "*member*" ou "*connection*" em seu conteúdo, respectivamente.		

Mensagem: PartnerResponse	**De**: SMA membro	**Para**: SMA coordenador
Descrição: Cada membro informa ao coordenador se participa ou não da EV.		
Mensagem: EnterpriseRegistration	**De**: SMA membro	**Para**: SMA coordenador
Descrição: Cada membro participante envia para o coordenador suas informações básicas para a configuração.		
Mensagem: ConnectionResponse	**De**: SMA membro	**Para**: SMA coordenador
Descrição: Cada membro informa ao coordenador seus relacionamentos com os outros parceiros.		
Mensagem: VERegistration	**De**: SMA coordenador	**Para**: SMAs membros
Descrição: O coordenador envia para os membros uma cópia da topologia da EV após o término da configuração.		

Tabela 2 – Protocolo de mensagens trocadas entre os SMAs.

5.4.2 Protocolo entre Agentes

Os agentes do SMA se comunicam através do suporte de comunicação oferecido pelo Massyve-Kit. Este suporte de comunicação utiliza, no "baixo nível", o SOCKET para o envio e recepção das mensagens usando o protocolo TCP neste processo. O Massyve-Kit tem um protocolo de "alto nível" básico, porém extensível, que contém tipos de mensagens específicas para cada uma das principais funções executadas por um agente. Por exemplo, existem mensagens para informar a localização do agente, informar as conexões entre os agentes, informar se o sistema multiagente já está executando, informar o final da execução do SMA, entre outras.

O protocolo de mensagens é simples e cada mensagem é formada por uma cadeia de caracteres, onde os seus parâmetros são separados por vírgulas, sendo que o primeiro indica qual é o tipo da mensagem. Estas mensagens são enviadas e recebidas

respectivamente pelos métodos *hsSend* e *hsReceive* implementados pelos agentes disponibilizados pelo Massyve-Kit.

Além dos tipos de mensagens padrão fornecidas pelo Massyve-Kit, foram adicionadas outras mensagens específicas para o processo de configuração da EV. Estas mensagens são apresentadas na Tabela 3 e seu diagrama UML de seqüência na Figura 38.

Mensagem: "p"	**De**: Agente configurador	**Para**: Agente XML
Formato: <p, coordCode, veName, reqInfo>		
Descrição: Invoca o agente XML a buscar por informações de parceiros. O parâmetro reqInfo indica se são informações do membro ou das conexões e seu valor é respectivamente "*member*" ou "*connection*".		
Mensagem: "m"	**De**: Agente XML	**Para**: Agente Configurador
Formato: <m, partnerCode, partnerName, partnerRole>		
Descrição: Informa ao agente Configurador que uma determinada empresa (*partnerName*) faz parte da empresa virtual com o papel específico (*partnerRole*).		
Mensagem: "o"	**De**: Agente XML	**Para**: Agente Configurador
Formato: <o, partnerSource, partnerDestination>		
Descrição: Informa ao agente Configurador que existe um relacionamento entre duas empresas que fazem parte da EV.		
Mensagem: "s"	**De**: Agente Configurador	**Para**: Agente XML
Formato: <s, veCode>		
Descrição: Requisita ao agente XML que envie uma cópia da topologia da empresa virtual configurada para seus respectivos parceiros.		
Mensagem: "v"	**De**: Agente XML	**Para**: Agente Empresa
Formato: <v, veCode>		
Descrição: Informa ao agente Empresa a chegada da topologia de uma nova empresa virtual.		

Tabela 3 – Protocolo das mensagens Massyve trocadas entre os agentes.

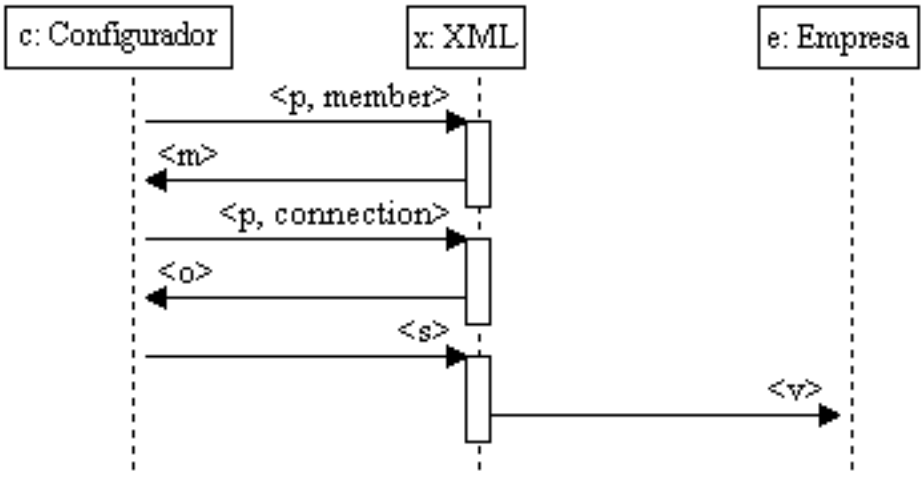


Figura 38 – Diagrama de seqüência das mensagens Massyve.

CAPÍTULO 6: Validação do Protótipo

Neste capítulo será apresentada uma simulação dirigida do protótipo implementado com o intuito de validar o modelo de configuração de EVs concebido. O protótipo desenvolvido contempla as duas alternativas de configuração da topologia da EV. Devido à similaridade dos processos de configuração, onde a diferença é encontrada apenas na forma como cada um recebe as informações de entrada, optou-se por apresentar nesta simulação apenas a segunda alternativa de configuração, chamada "Configuração da Topologia Usando Métodos Internos". O motivo desta escolha ocorre devido ao fato desta alternativa de configuração ser o foco principal do modelo proposto, sendo a outra apenas uma expansão do modelo.

Para esta simulação foi utilizada uma amostra de dados obtida de uma das empresas piloto do projeto DAMASCOS. Esta empresa, chamada Kyaia, está localizada no interior de Portugal próximo à cidade do Porto e seu ramo de atuação é o calçadista. Antes da simulação propriamente dita, são feitas algumas considerações sobre alguns aspectos importantes relacionados ao sistema e a seu ambiente de execução.

6.1 Requisitos de Sistema

Alguns requisitos, tanto de hardware quanto de software, são necessários para satisfazer as necessidades básicas do protótipo implementado. Estes requisitos estão diretamente associados ao bom desempenho da aplicação e, por conseqüência, dos resultados obtidos. Levando-se em consideração as tecnologias utilizadas na sua concepção, o protótipo deve ser executado dentro do conjunto de especificações mínimas apresentadas na Tabela 4, para que seu desempenho seja compatível com as necessidades de tempo de execução da configuração. Como explanado no capítulo 4, o tempo de configuração deve ser o menor possível a fim de que a EV possa entrar em operação no menor espaço de tempo. Minimizando-se o "tempo de montagem" a EV tem mais tempo disponível para a execução de seu processo de negócio, motivo de sua criação.

Requisito	Mínimo Aceitável
Arquitetura de hardware	PC Intel® 400 MHz ou compatível
Sistema operacional	Microsoft Windows® NT 4.0 (SP 5 ou 6)
Memória	128 MB
Rede	Ethernet 10 Mbps
Espaço em disco disponível	50 MB

Tabela 4 – Requisitos do sistema.

Este conjunto de especificações apresentado na tabela acima, garante o mínimo de recursos necessários para a instalação e execução de todas as partes que compõem o sistema de configuração da EV. O sistema de configuração desenvolvido vai além do sistema multiagente implementado. Cada empresa precisa ter instalado e configurado em seu ambiente outros sistemas externos que, conjugados com o sistema multiagente, compõem a unidade básica necessária para a operação do sistema de configuração em cada empresa. Fazem parte destas ferramentas externas, o SGBD *Interbase*®, que armazena as informações relacionadas à empresa, e o sistema BDE (*Borland Database Engine*), que possibilita a conectividade entre o banco de dados e as aplicações desenvolvidas em compiladores *Borland*®; neste caso a conexão entre o *Interbase*® e o SMA.

Ainda como parte constituinte da plataforma de configuração de EVs há a ferramenta de WFBB. Esta ferramenta precisa estar instalada e operando em uma das empresas que fazem parte da EV. É através dela que os SMA das empresas participantes da EV conseguem trocar informações entre si. Porém, para que isso ocorra, cada empresa precisa ter seu SMA cadastrado como sendo usuário do WFBB. É importante ressaltar que as especificações encontradas na Tabela 4 não contemplam as necessidades de sistema exigidas pela ferramenta WFBB. Desta forma, a empresa que for escolhida para executar o WFBB deverá ter uma carga adicional de recursos computacionais para satisfazer as suas exigências. Esta atividade não precisa ser necessariamente atribuída ao coordenador da EV. Qualquer outra empresa pertencente à EV, que tenha uma boa quantidade de recursos computacionais excedentes aos requisitos mínimos, pode vir a desempenhar esse papel. Uma alternativa mínima para solucionar esse problema seria a utilização de um computador com capacidade de processamento de 800MHz e memória de 256 MB.

6.2 Papéis Operacionais das Empresas e suas Respectivas Imagens

Para facilitar a compreensão da simulação do sistema de configuração, é importante estar familiarizado com os tipos de papéis operacionais que cada membro pode desempenhar na EV. Nesta seção serão apresentadas as imagens / ícones associadas a cada um dos papéis operacionais de uma empresa em uma EV (ver seção 4.1.1). Abaixo, na Tabela 5, podem ser vistas as imagens e uma descrição sobre o papel a que cada uma está associada. Em uma EV, tipicamente os atores (empresas participantes com seus respectivos papéis) aparecem graficamente na topologia tomando como base a direção do fluxo da produção na ordem da esquerda para a direita. Portanto, os elementos mais à esquerda são os pré-fornecedores e os elementos mais à direita são os varejistas. A Tabela 5 apresenta os tipos de papéis na ordem (de cima para baixo) em que eles aparecem na topologia (da esquerda para a direita).

O "*cluster*" (ver Tabela 5) tem um comportamento diferente dos outros papéis, ou seja, ele não é um papel propriamente dito, apenas serve para agrupar um conjunto de empresas já adicionadas a EV, que tenham o mesmo tipo de papel e de acordo com uma determinada "regra" de classificação (por país, por produto, etc.). Desta forma, utilizando o *cluster*, pode-se diminuir o número de elementos visuais da tela com o intuito de facilitar a compreensão do usuário. Esta não é a única função do *cluster*, ele serve também como uma forma de classificação os membros da EV. Por exemplo, se em uma determinada EV existirem muitas empresas da Inglaterra, o usuário que está configurando a mesma poderá agrupá-las e nomear esse *cluster* como "Empresas da Inglaterra". Entretanto, se o usuário deseja visualizar o conjunto de empresas pertencentes a um *cluster*, existe uma função associada e ele que permite a expansão do mesmo, e assim a apresentação de seu conteúdo como se o mesmo não existisse.

Imagem	Descrição
	Pré-fornecedor

	Fornecedor
	Produtor Principal
	Distribuidor
	*Broker*
	Varejista
	*Cluster*[28]

Tabela 5 – Imagens dos papéis dos membros.

6.3 Simulação

Na simulação foram utilizados cinco computadores, cada um representando respectivamente um membro da EV. Ao Computador 1 foi atribuída a incumbência de simular o coordenador. Por este motivo o agente configurador está ativo em seu ambiente de execução. Para executar o WFBB foi escolhido o Computador 2. Esta decisão foi tomada devido a maior quantidade de recursos computacionais disponíveis nesta máquina. Os outros três computadores, que representam os elementos restantes da simulação, têm em seu ambiente de execução apenas a unidade básica que compõe o sistema de configuração. A Tabela 6 apresenta uma breve descrição das empresas participantes e seus respectivos ambientes de simulação.

[28] *Cluster* – Neste contexto *cluster* significa um agrupamento visual de membros para minimizar o número de elementos gráficos na tela.

Computador	Papel	Empresa	Sistemas instalados
Computador 1	Produtor principal (Coordenador)	Kyaia	SMA e Banco de dados
Computador 2	Fornecedor (Membro)	Nobre Couro	SMA, WFBB e Banco de dados
Computador 3	Varejista (Membro)	Planeta Sapatos	SMA e Banco de dados
Computador 4	Varejista (Membro)	Style Modas	SMA e Banco de dados
Computador 5	Fornecedor (Membro)	Cordolona	SMA e Banco de dados

Tabela 6 – Empresas participantes da simulação.

A simulação será apresentada passo a passo mostrando uma captura da tela do computador no momento em que cada ação é executada. Para que a configuração ocorra com sucesso, cada empresa deverá ter cadastrado previamente no sistema suas informações básicas, bem como as informações sobre quais EVs ela está participando. As interfaces de cadastramento destas informações podem ser vistas, respectivamente, na Figura 41 e na Figura 44.

O processo de configuração inicia-se com o usuário configurador, através do agente configurador, requisitando a configuração de uma nova empresa virtual. O usuário acessa a função de criação de uma nova empresa virtual passando como informação o nome desta nova EV. Esta etapa do processo pode ser vista na Figura 39. Como segundo plano em todas as figuras pertencentes à simulação, é mostrada uma tela de console que mostra as mensagens trocadas entre os agentes do sistema, console este provido pelas funcionalidades básicas do Massyve-Kit.

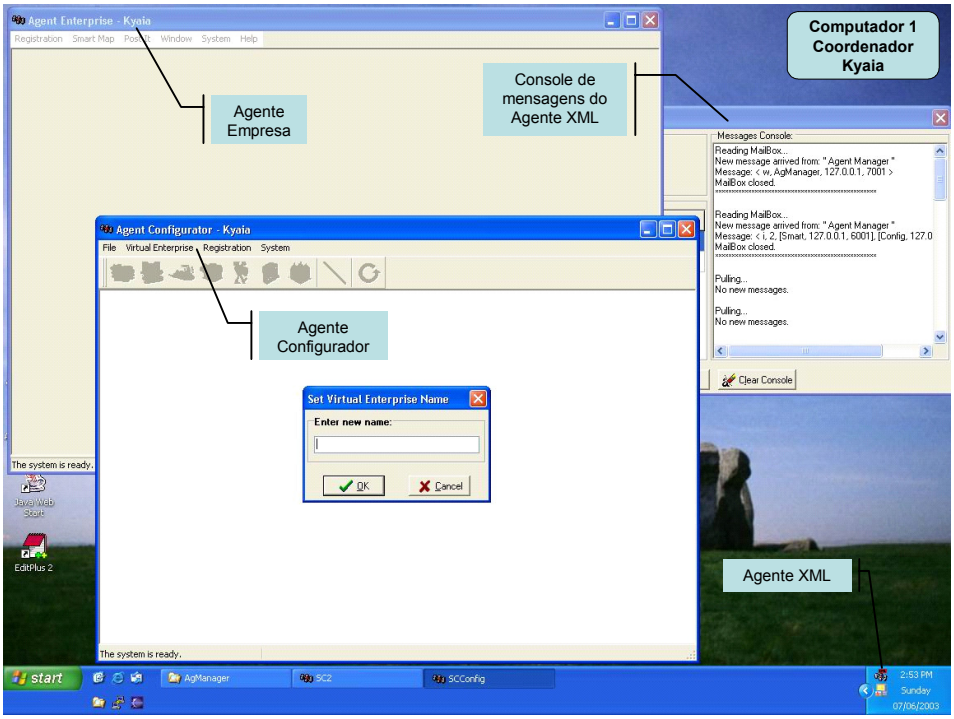


Figura 39 – Coordenador iniciando configuração.

Logo após o coordenador informar o nome da EV, uma mensagem é enviada ao SMA de cada uma das empresas cadastradas como usuários do WFBB para localizar os membros efetivamente escolhidos no processo de busca e seleção de parceiros (Figura 40).

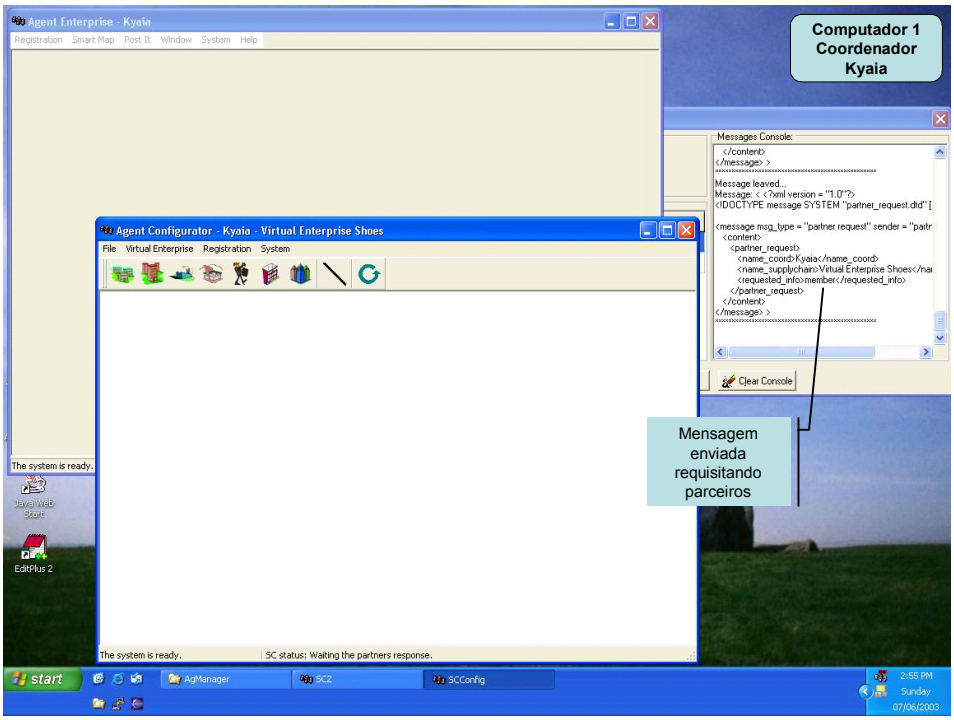


Figura 40 – Configurador enviando mensagens para localizar parceiros.

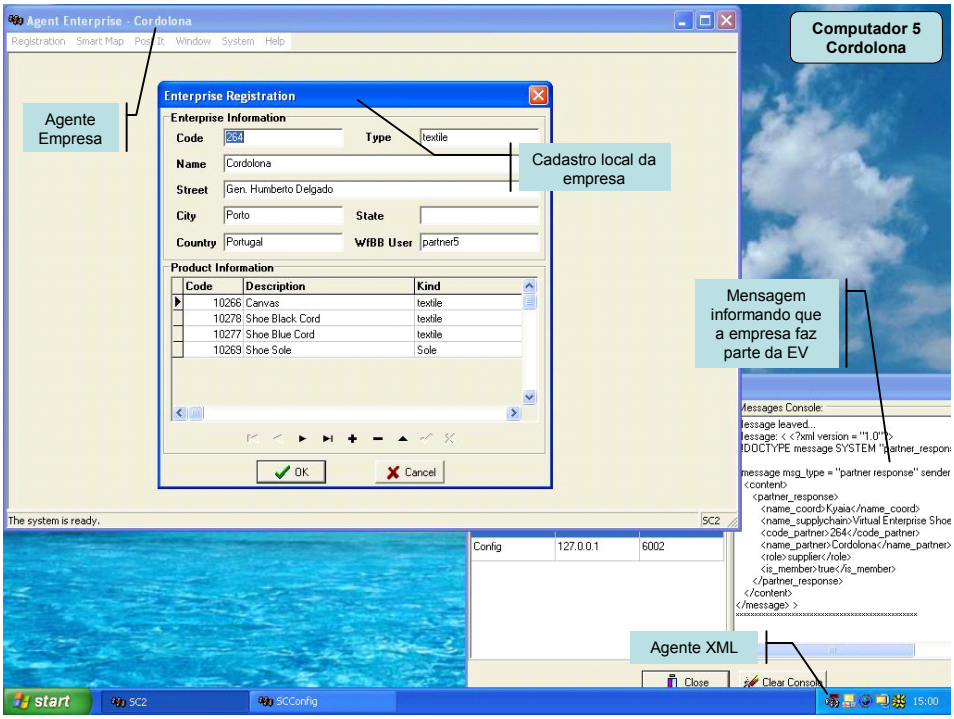


Figura 41 – Empresa informando que faz parte da EV.

Cada empresa que recebe a mensagem responde, através de seu SMA, informando se faz parte ou não da EV (Figura 41). As empresas que responderem afirmativamente à requisição são automaticamente incluídas na EV (Figura 42).

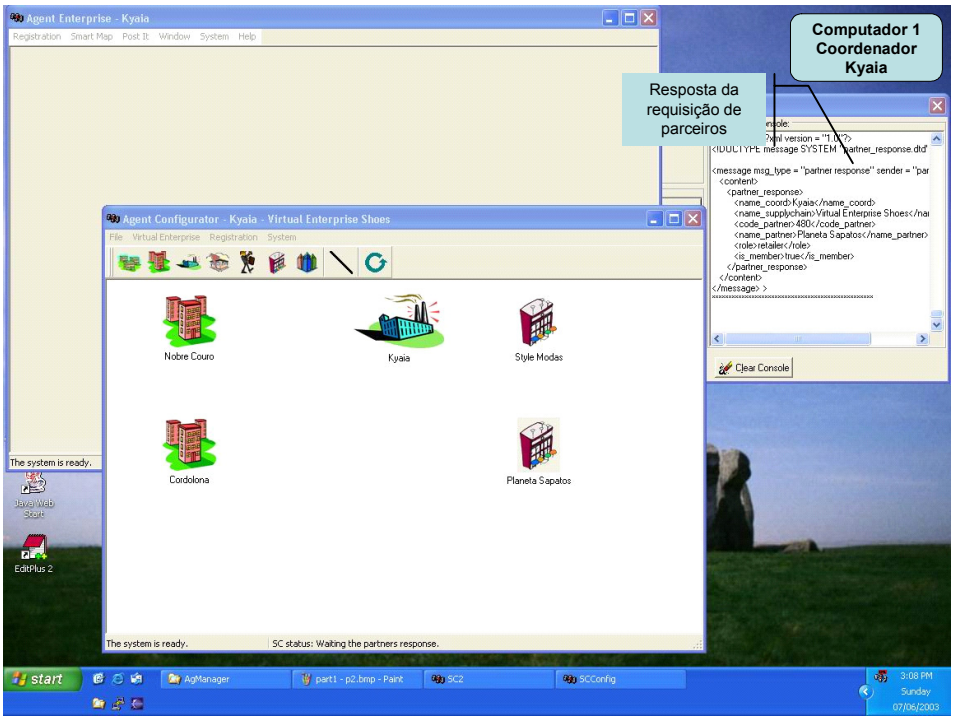


Figura 42 – Configurador incluindo membros automaticamente na EV.

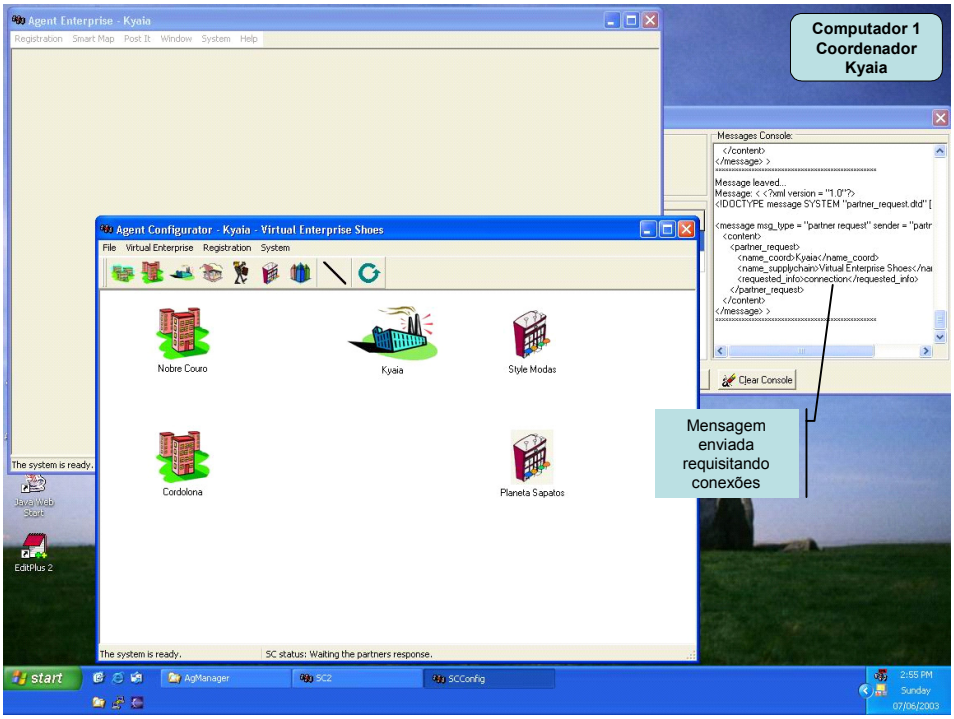


Figura 43 – Configurador enviando mensagens requisitando conexão.

Após receber a resposta dos membros, o agente configurador envia uma mensagem requisitando as conexões (Figura 43). Cada membro responde ao configurador informando suas conexões e seus respectivos produtos intercambiados (Figura 44).

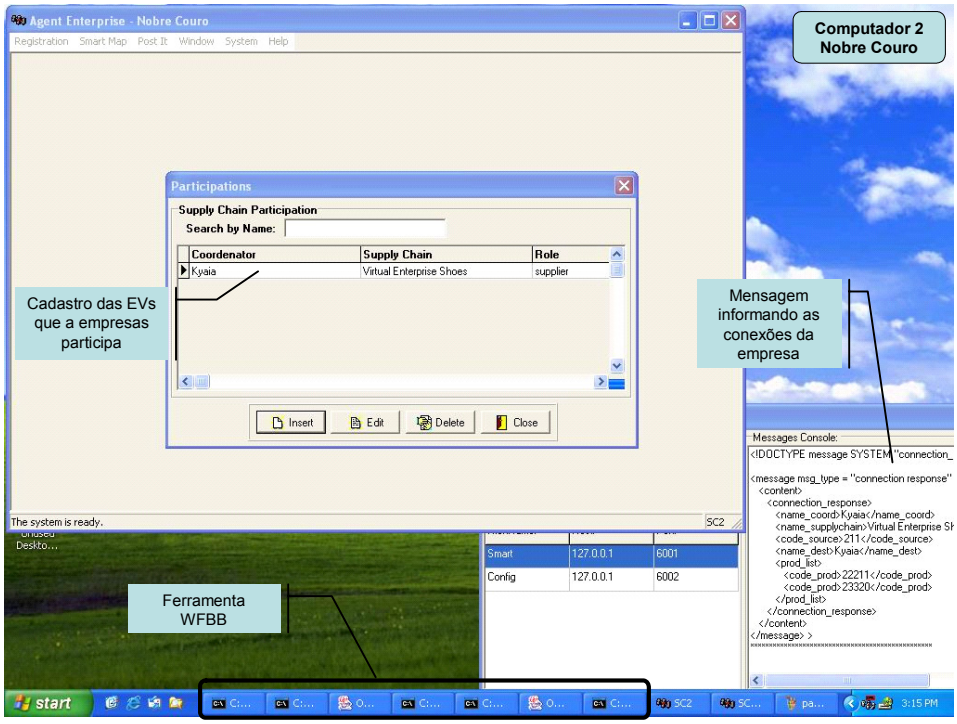


Figura 44 – Empresa informando suas conexões.

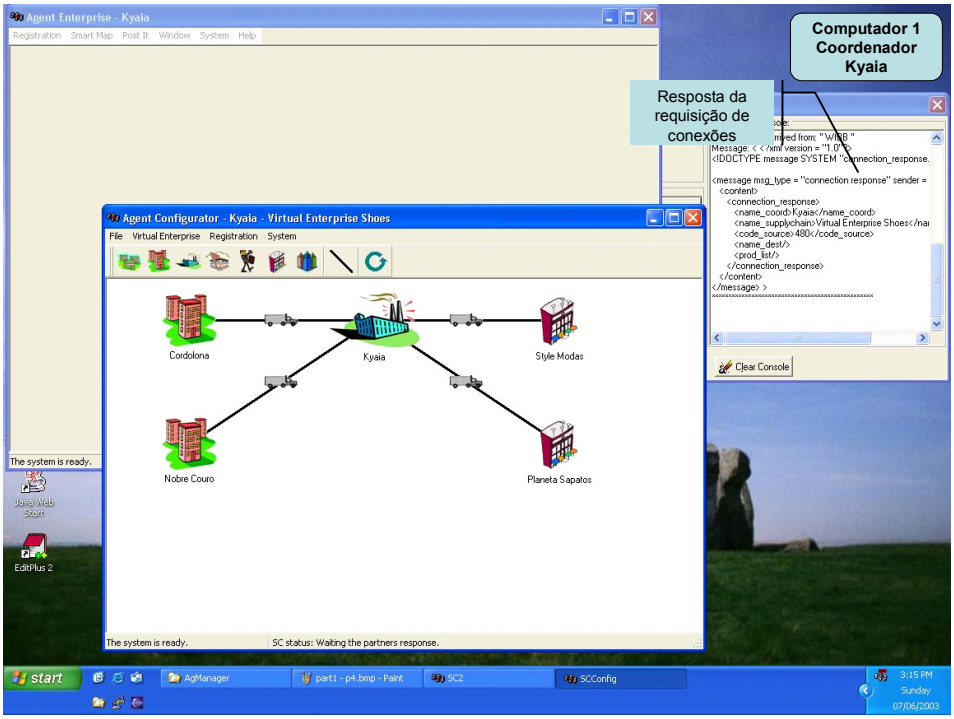


Figura 45 – Configurador incluindo conexões automaticamente na EV.

A cada resposta recebida, o agente configurador inclui as conexões ligando os dois membros informados por ela (Figura 45). A Figura 46 mostra o usuário configurador configurando a visibilidade da topologia para um membro após receber todas as conexões.

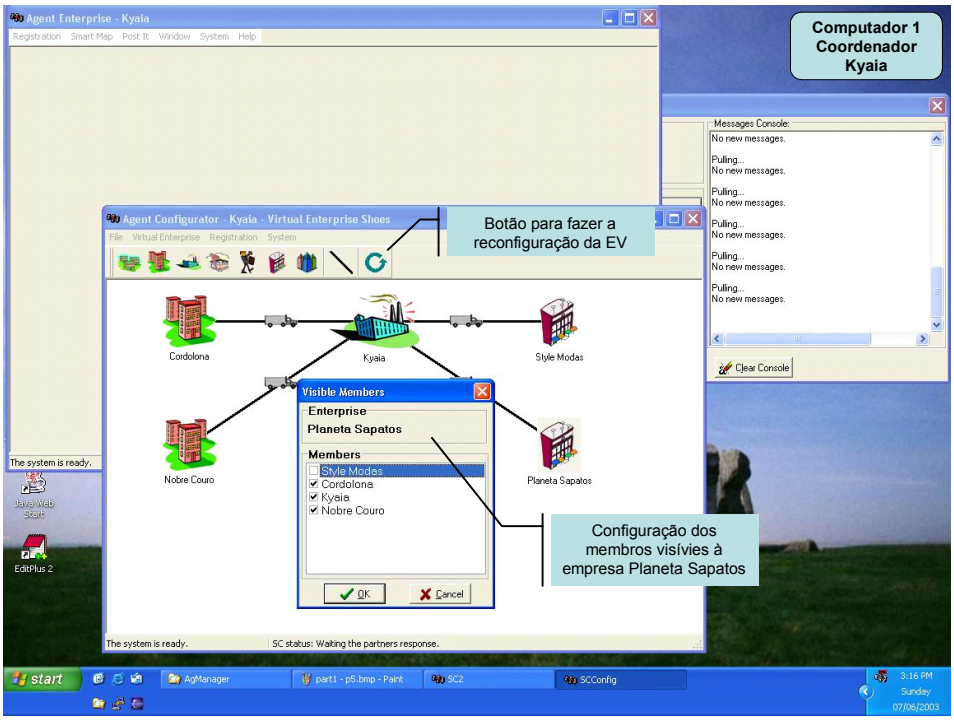


Figura 46 – Coordenador configurando a visibilidade de um membro.

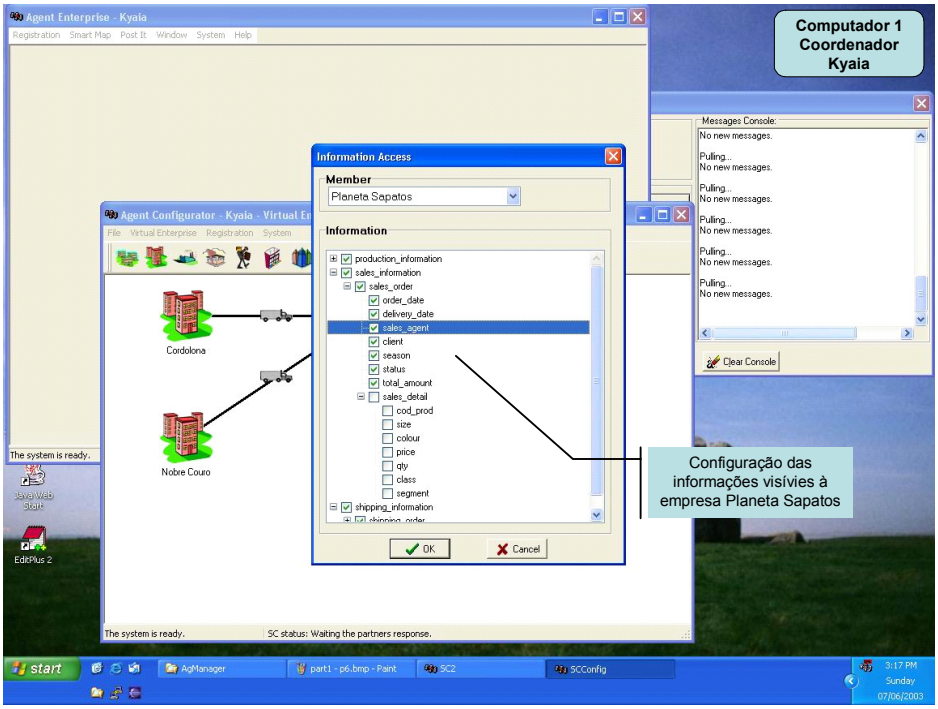


Figura 47 – Coordenador configurando o acesso à informação para um membro.

O último estágio da configuração é a escolha, pelo usuário configurador, das informações de um dado membro acessíveis a cada um dos outros membros da EV – chamada "Cláusulas de Supervisão" – (Figura 47). Após essa configuração uma cópia da EV é enviada a cada parceiro. A Figura 48 apresenta a visão parcial enviada a um membro.

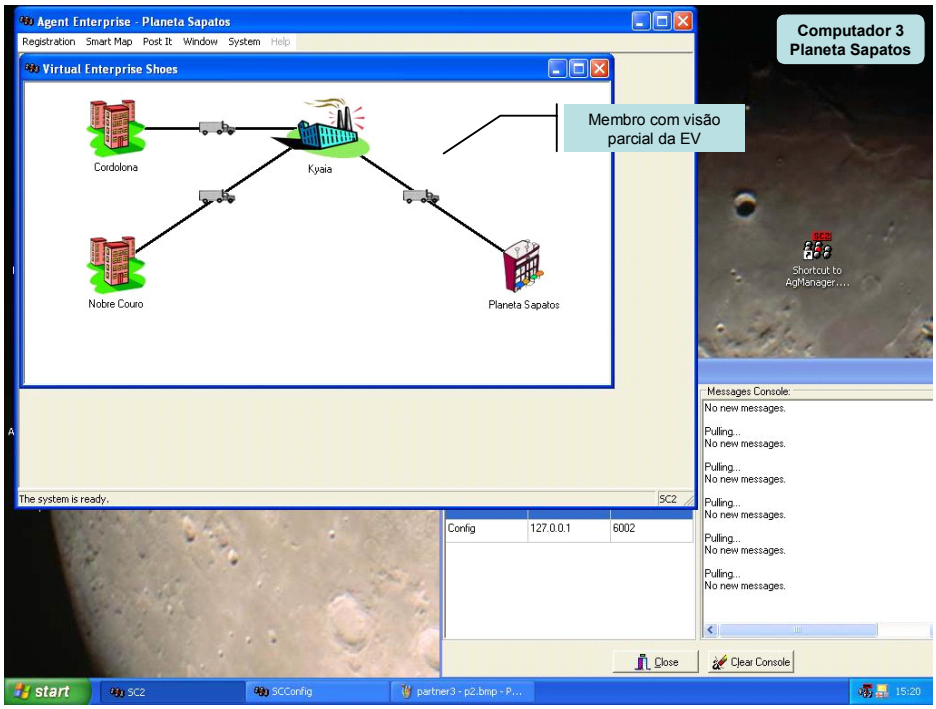


Figura 48 – Membro com visão parcial da EV.

A topologia só é enviada a cada uma das empresas participantes da EV após o usuário configurador se certificar que as informações apresentadas na tela do computador estão corretas e efetuar seu armazenamento no banco de dados do coordenador. Um fato importante abordado no desenvolvimento do protótipo foi a questão da reconfiguração da EV no momento em que a mesma está em operação. Tendo em vista um ambiente dinâmico de funcionamento da empresa virtual, a EV deve ser capaz de automaticamente reconfigurar sua topologia para contemplar as alterações ocorridas com a entrada ou saída de algum parceiro. Além da forma automática de reconfiguração da topologia, estabelecida através de um temporizador que repete o processo de busca das informações nos SMAs dos parceiros a cada período de tempo, o usuário configurador pode, através de um botão na interface do agente configurador, invocar manualmente esse procedimento no momento desejado. Essa característica apresentada pelo protótipo implementado reflete um dos principais objetivos especificados no modelo – tornar o processo de configuração o mais automático e manter as informações o mais atualizadas possível – para EVs dinâmicas. A Figura 46 apresenta a interface do agente configurador destacando o botão de reconfiguração. Após a reconfiguração da topologia, o usuário configurador deverá configurar as restrições de acesso às informações das novas empresas adicionadas à EV.

Na Figura 49 é mostrada uma cópia completa da topologia enviada a um dos membros da EV. Isto se deve ao fato desta empresa não ter nenhuma restrição de visibilidade. Além da topologia, também pode ser vista uma interface mostrando as propriedades de um dos membros da EV. Esta interface mostra as informações básicas da empresa bem como seus produtos intercambiados no escopo da empresa virtual em questão.

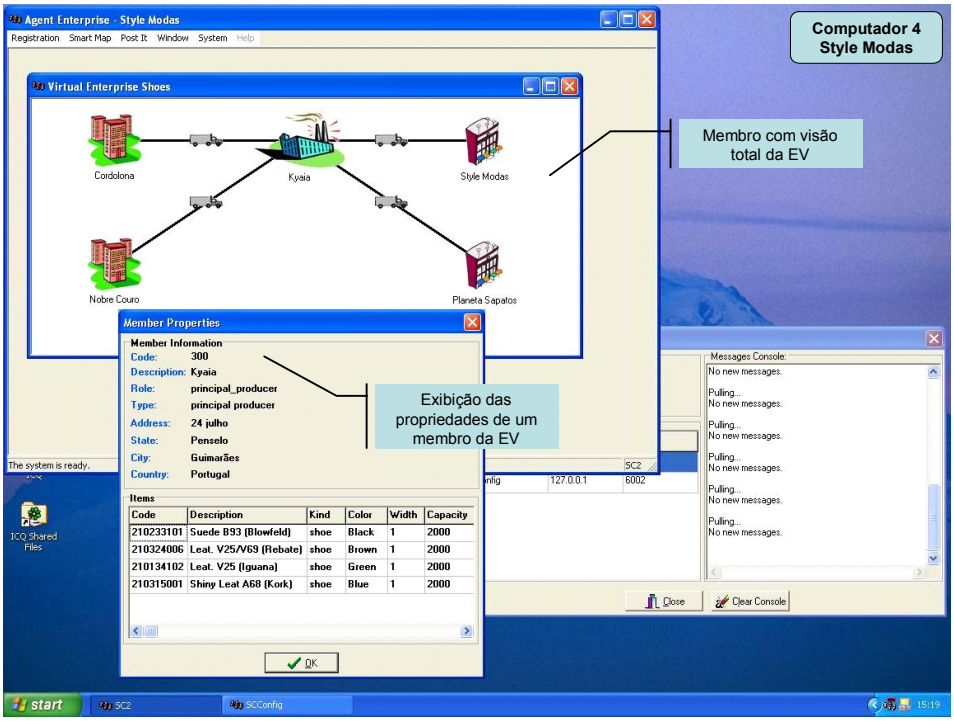


Figura 49 – Membro com visão total da EV.

6.4 Análise dos Resultados

Levando-se em consideração o ambiente de simulação do protótipo, algumas conclusões em relação aos seus resultados podem ser analisadas. Antes de iniciar a discussão dos resultados, faz-se necessária uma breve descrição do cenário de execução utilizado. A Tabela 7 mostra as configurações de capacidade de processamento e quantidade de memória associadas a cada uma das máquinas utilizadas na simulação. Além da configuração das máquinas, também se faz necessário apresentar a infraestrutura de rede sobre a qual essa simulação foi executada. Portanto, as características da *intranet* são: tipo de rede *Ethernet* com taxa de 100 Mbps, tanto do *switch* quanto das placas de rede.

Computador	Configuração	
	Processador	Memória
Computador 1	Intel® Pentium III 550 MHz	128 MB RAM
Computador 2	AMD Athlon™ 1.4 GHz	256 MB RAM
Computador 3	Intel® Pentium III 550 MHz	128 MB RAM
Computador 4	Intel® Pentium III 550 MHz	128 MB RAM
Computador 5	AMD-K6™ 500 MHz	128 MB RAM

Tabela 7 – Recursos dos computadores da simulação.

O principal resultado a ser analisado na simulação é a velocidade do processo de configuração. Aumentar a velocidade e agilidade deste processo foi um dos principais objetivos deste trabalho. O tempo gasto na realização do processo de configuração está diretamente relacionado ao número de membros da EV, e a quantidade de produtos que cada empresa intercambia. Por exemplo, configurar uma pequena empresa virtual do ramo calçadista com cinco ou seis membros trocando em torno de quinze produtos, é um processo mais rápido que a configuração de uma empresa virtual do ramo automobilístico onde a quantidade de membros e produtos é bem maior. É importante salientar que questões relacionadas à velocidade de manipulação do sistema por parte do usuário configurador foram deixadas de lado haja vista que, à priori, não se pode fazer estimativas de tempo relacionadas a esta variável pertencente ao processo. Isto porque a velocidade de manipulação do sistema por parte do usuário envolve o componente humano, e cada pessoa responde ao sistema de maneira diferente dependendo de suas características.

Utilizando o ambiente de simulação apresentado acima, vários testes foram realizados para a formulação de uma estimativa de tempo gasto na configuração em relação à quantidade de empresas envolvidas. Um parâmetro importante a ser destacado nestas simulações é a periodicidade com que o agente XML vai consultar sua fila de mensagens na ferramenta WFBB. Este parâmetro influencia diretamente a velocidade do processo de configuração, visto que o mesmo é especificado pelo usuário do SMA. Para estas simulações foi utilizado o período de 1 segundo para cada consulta ao WFBB para todos os SMA envolvidos na simulação.

Como pode ser visto no gráfico da Figura 50, a relação número de empresas versus tempo de configuração é linear, ou seja, o tempo cresce na proporção em que o número de empresas aumenta. Esta reta pode ser mais ou menos inclinada dependendo da quantidade de informação que cada empresa envia para o configurador. Assim, quanto menor for a quantidade de produtos intercambiados, menor será seu tempo de configuração, e menos inclinada será a reta.

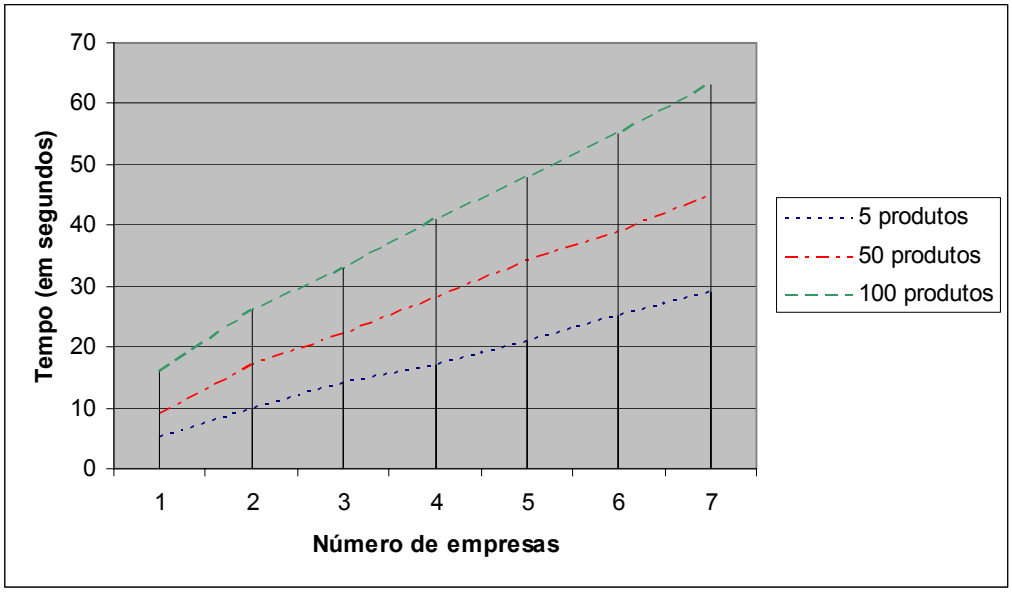


Figura 50 – Gráfico de tempo gasto para configurar uma EV.

Através do gráfico da Figura 50 o coordenador pode ter uma estimativa de quanto tempo irá demorar o processo de configuração dependendo do número de empresas envolvidas. Portanto, se uma empresa não responder dentro de um limite de tempo aceitável, isso pode ser um sinal que a mesma está com problemas de comunicação ou seus sistemas estão desligados. Se isso ocorrer, o SMA do coordenador não conseguirá receber automaticamente as informações requisitadas à empresa em questão, cabendo ao usuário configurador inseri-la manualmente na topologia da EV.

CAPÍTULO 7: Conclusões

Uma empresa virtual dinâmica é aquela que responde positivamente a uma mudança inesperada. Portanto, uma empresa virtual é considerada dinâmica se for formada com a possibilidade de se reconfigurar de forma rápida, barata e eficiente para responder a mudanças na oportunidade de negócio ou a problemas envolvendo algum parceiro (GORANSON, 1999, CAMARINHA-MATOS *et al*., 1999a). Para atender a este tipo de empresas virtuais, a TI deve fornecer ferramentas adequadas para satisfazer as necessidades de agilidade e flexibilidade exigidas pelo mercado globalizado.

Tendo como base o ambiente de empresas virtuais, o objetivo desta dissertação foi o de contribuir para aumentar a agilidade e flexibilidade na fase de criação e evolução de uma empresa virtual tendo como enfoque principal as EVs dinâmicas. Este objetivo foi alcançado através da apresentação de um modelo que auxilia na automatização de certas partes do processo de configuração de EVs, mais especificamente nos processos de configuração da topologia e do acesso à informação dos membros.

O modelo proposto e a implementação elaborada enquadraram-se no âmbito do projeto de cooperação internacional chamado DAMASCOS. Esse projeto visou o desenvolvimento de uma plataforma de suporte ao funcionamento de EVs. Dentro desta plataforma – formada por uma série de módulos de software desenvolvidos pelos vários parceiros do projeto – o grupo GSIGMA / UFSC desenvolveu o sistema SC^2, que é um sistema multiagente para apoiar o gestor de uma EV na coordenação da mesma. Este sistema de gerenciamento é fundamentado em princípios de monitoração / supervisão dos acontecimentos envolvidos no escopo da EV. O sistema SC^2, em uma primeira etapa, era provido de funcionalidades que suportavam apenas a fase de operação da EV. Posteriormente, um módulo para apoiar a busca e seleção de parceiros na fase de criação foi desenvolvido. Este módulo também foi desenvolvido como um sistema multiagente introduzindo como diferencial a utilização de agentes móveis na resolução do problema de seleção (SCHMIDT, 2003). Complementado o sistema SC^2, o modelo de configuração semi-automática proposto nesta dissertação traz uma nova funcionalidade no intuito de aperfeiçoar este sistema e assim torná-lo mais eficiente no gerenciamento efetivo da EV em todas as fases de seu ciclo de vida.

Os resultados do trabalho, tanto em relação ao modelo quanto à implementação do protótipo, contemplaram os objetivos propostos, e de um modo geral atenderam às expectativas de agilidade e flexibilidade necessárias a EVs que operam em ambientes dinâmicos. Os conceitos e tecnologias utilizados foram de primordial importância para se conseguir implementar a proposta, e confirmaram que a utilização de padrões internacionalmente reconhecidos, tais como XML e CORBA, facilitam a concepção e posterior interoperação com aplicações heterogenias. A escolha de sistemas multiagente para a solução do problema trouxe vantagens em dois aspectos. Primeiramente, por ser um problema distribuído, onde os membros estão lógica e fisicamente separados, sabendo apenas sobre suas intenções e seus relacionamentos diretos, não tendo uma visão total da EV, faz com que exista a necessidade de uma interação entre estas partes para se chegar à conclusão de quem faz ou não parte de uma determinada EV. Em segundo lugar, o configurador sendo concebido na forma de um agente, tem uma maior autonomia na realização de sua tarefa, podendo no futuro tomar decisões, sozinho ou com a ajuda do grupo, das quais um sistema monolítico tradicional tem muito menos capacidade.

Apesar do modelo ter alcançado seus objetivos, um ponto importante deve ser levado em consideração. Por ter sido concebido como um processo semi-automático, onde há interações entre o usuário e o processo, certas restrições de desempenho podem ser detectadas. Como a quantidade de interações do usuário é variável, ou seja, dependem do número de restrições impostas aos membros da EV, quanto maior for o número de restrições de acesso às informações, maior será o tempo de iteração ente o sistema e o usuário, e mais demorado será o processo de configuração como um todo. Entretanto, se não existir nenhuma restrição de acesso à informação por parte de nenhum membro, a EV passa do estado de configuração da topologia diretamente para a fase de operação, tornando assim, neste caso em particular, o processo de configuração automático.

Algumas limitações foram detectadas no decorrer no desenvolvimento do projeto. A mais importante destas limitações está relacionada à questão dos tipos pré-definidos de papéis executados pelos membros da EV. No estado atual de desenvolvimento do protótipo, inclusões de novos papéis necessitariam alterações de código e recompilações das aplicações. Uma forma de solucionar este problema seria a adição de procedimentos que permitissem a inclusão de novos papéis por parte do usuário que estivesse operando o sistema. Entretanto, esta não é uma tarefa fácil de ser executada por um usuário tradicional

de sistemas. Isto porque esta tarefa exige a configuração de certos parâmetros, tais como a escolha da imagem correspondente ao novo papel, com que outros papéis o referido poderá se relacionar, etc.; que são difíceis de ser realizadas até mesmo por usuários qualificados.

Outra limitação a ser considerada é a questão da implementação ter sido desenvolvida utilizando com ferramenta o *Borland® C++ Builder™*. Com a utilização desta ferramenta no desenvolvimento da aplicação, a independência de plataforma ficou impossibilitada. Para resolver esse problema, a linguagem de programação *Java* pode ser uma alternativa a ser utilizada no futuro, pois através de sua máquina virtual permite que programas escritos nela sejam executados em praticamente todas as arquiteturas computacionais comercialmente utilizadas.

7.1 Perspectivas para Trabalhos Futuros

Considerando o estágio atual do projeto, algumas sugestões para dar continuidade ao trabalho desenvolvido nesta dissertação são apresentadas no decorrer desta seção.

Como primeira sugestão de melhoramento do estado atual do projeto, tem-se a adoção do ebXML como formato padrão para a informação intercambiada pelo sistema. Esta alteração traria uma maior compatibilidade na troca de informações entre o SMA e outras aplicações envolvidas com comércio eletrônico. O ebXML especifica o formato das mensagens para sistemas de informação que atuam na área de *e-business* e *e-commerce*. No entanto, as iniciativas no âmbito do ebXML estão ainda em desenvolvimento, com poucos resultados disponíveis em termos de modelos de referência para suportar a troca de informações ao longo de todo o ciclo de vida de uma EV.

Outra contribuição importante para o projeto seria a expansão do número de tipos de papéis operacionais que os membros podem desempenhar em uma EV. Levando-se em consideração que o número de papéis contemplados pelo protótipo atual é apenas um subconjunto dos vários tipos possíveis (embora já relativamente grande e contemplando seguramente os papéis mais importantes), com a adição de alguns outros tipos de papéis a utilização do sistema poderia ser expandida para outros segmentos de mercado hoje ainda não contemplados.

Para melhorar o desempenho do processo de configuração existem duas outras alternativas: a “configuração automática” e a “configuração inteligente”. Na configuração

automática o processo acontece sem a participação do usuário. Neste tipo de configuração o trabalho do usuário é anterior ao início do processo, ou seja, o mesmo prepara previamente as informações que serão posteriormente repassadas ao coordenador assim que se inicie o processo de configuração. Este processo é considerado um segundo nível de automatização do processo de configuração. A configuração inteligente é o último e mais avançado nível de automatização em relação à configuração. Neste tipo de configuração, além do processo ser automático, não há a necessidade do usuário preparar previamente as informações relevantes ao processo. O sistema de configuração deve ter a capacidade de reconhecer as informações importantes para a configuração dentro do repositório de dados local disponível em cada empresa, e em cima destas informações chegar à conclusão sobre seu papel, relacionamentos e restrições de acesso a suas informações. Estas informações, assim como na alternativa anterior, são posteriormente requisitadas pelo coordenador.

Levando em consideração os tipos de configuração apresentados acima, como último melhoramento surge a pretensão de aprimorar o nível de "inteligência" dos agentes do SMA para tornar a configuração automática ou até mesmo inteligente. Este passo é de grande valor para que se possa, utilizando métodos cognitivos, fazer a configuração da EV. Através desta característica, os SMAs instalados nas empresas consideradas potenciais parceiras teriam a possibilidade de encontrar em seus repositórios de dados locais, informações que identificassem se a mesma faria ou não parte de uma nova EV que estivesse sendo configurada. Esta nova característica é de suma importância em grupos de empresas onde informações deste tipo não estão bem organizadas ou até mesmo incompletas.

Anexo A: Formato das Mensagens Trocadas entre os SMAs

O processo de troca de mensagens entre os sistemas multiagente foi apresentado na seção 5.4.1. Entretanto, seu protocolo especificado em XML não foi descrito naquela ocasião, devido ao fato de algumas mensagens terem um grande conteúdo de informação. Assim, a especificação completa do formato das mensagens, contendo todas as informações pertencentes ao protocolo, foi suprimida do texto da dissertação e introduzido neste anexo.

Antes de entrar em detalhes em cada um dos tipos de mensagens, uma consideração geral sobre o protocolo deve ser feita. Esta consideração diz respeito à característica mais importante presente em todas as mensagens do protocolo, o chamado cabeçalho comum. Este cabeçalho contém dois atributos que identificam respectivamente o "tipo" e o "emissor" da mensagem. O atributo tipo da mensagem (*msg_type*), identifica qual informação a referida mensagem está carregando, e pode possuir um dos seguintes valores: "*partner request*", "*partner response*", "*enterprise registration*", "*connection response*" ou "*supply chain registration*". O atributo emissor da mensagem (*sender*), identifica o criador da mensagem e contém o nome que o referido usuário utiliza para se conectar a ferramenta WFBB. Este último atributo é importante nos casos em que a mensagem necessitar de uma resposta. A seguir, cada um dos tipos de mensagens do protocolo será apresentada e uma breve descrição de seu conteúdo será feita.

Mensagem *PartnerRequest*

Esta mensagem é utiliza pelo coordenador para requisitar informações de possíveis parceiros da EV. Seu conteúdo e composto por um campo que identifica o nome do coordenador (*name_coord*), outro que identifica o nome da empresa virtual sendo formada (*name_virtualEnterprise*), e por último o campo de identifica qual informação está sendo requisitada (*requested_info*). Este último campo pode ter dois valores: se a informações requerida for de empresa, o campo terá como conteúdo "*member*"; entretanto, se a informação requerida for sobre conexões, seu conteúdo será "*connection*". A Figura 51 mostra o DTD que especifica esta mensagem.

```
<!-- DTD Partner Request -->

<!ELEMENT message (content)>
  <!ATTLIST message msg_type CDATA #REQUIRED>
  <!ATTLIST message sender CDATA #REQUIRED>

  <!ELEMENT content (partner_request)>
    <!ELEMENT partner_request (name_coord,
                               name_virtualEnterprise,
                               requested_info)>
      <!ELEMENT name_coord (#PCDATA)>
      <!ELEMENT name_virtualEnterprise (#PCDATA)>
      <!ELEMENT requested_info (#PCDATA)>
```

Figura 51 – DTD da mensagem *PartnerRequest.*

Mensagem *PartnerResponse*

Quando uma empresa recebe uma mensagem *PartnerRequest* do coordenador, a mesma responde se faz parte ou não da empresa virtual através do envio da mensagem *PartnerResponse*. Esta mensagem contém as seguintes informações: o nome do coordenador (*name_coord*), o nome da empresa virtual (*name_virtualEnterprise*), o nome da empresa que está enviando da mensagem (*name_partner*), seu papel na EV (*role*), e por último se faz parte ou não da EV (*is_member*) que pode ter o valor *true* ou *false* respectivamente. É importante ressaltar que se a empresa não faz parte da EV o valor do campo *is_member* será *false* e o conteúdo do campo *role* estará vazio. Na Figura 52 é apresentado o DTD que especifica esta mensagem.

```
<!-- DTD Partner Response -->

<!ELEMENT message (content)>
  <!ATTLIST message msg_type CDATA #REQUIRED>
  <!ATTLIST message sender CDATA #REQUIRED>

  <!ELEMENT content (partner_response)>
    <!ELEMENT partner_response (name_coord,
                                name_virtualEnterprise,
                                code_partner,
                                name_partner,
                                role,
                                is_member)>
      <!ELEMENT name_coord (#PCDATA)>
      <!ELEMENT name_virtualEnterprise (#PCDATA)>
      <!ELEMENT code_partner (#PCDATA)>
      <!ELEMENT name_partner (#PCDATA)>
      <!ELEMENT role (#PCDATA)>
      <!ELEMENT is_member (#PCDATA)>
```

Figura 52 – DTD da mensagem *PartnerResponse.*

Mensagem *EnterpriseRegistration*

A mensagem *EnterpriseRegistration* contém as informações básicas de uma empresa em específico. Esta mensagem é enviada por uma empresa ao coordenador da EV somente se sua resposta (*PartnerResponse*) a procura por parceiros for afirmativa. Ou seja, o coordenador só receberá estas informações de uma empresa se efetivamente a mesma participar da EV. Dentre as informações importantes compreendidas por esta mensagem estão: o endereço da empresa, os possíveis contatos através de seus representantes, e os produtos produzidos pela mesma que serão intercambiados com os outros membros do consórcio. A especificação completa do DTD desta mensagem pode ser visto na Figura 53.

```
<!-- DTD Enterprise Information -->

<!ELEMENT message (content)>
 <!ATTLIST message msg_type CDATA #REQUIRED>
 <!ATTLIST message sender CDATA #REQUIRED>

 <!ELEMENT content (enterprise_info+)>
  <!ELEMENT enterprise_info (generic_info, product_info*)>
   <!ELEMENT generic_info (code_ent, name_ent, type, contact*, address*)>
    <!ELEMENT code_ent (#PCDATA)>
    <!ELEMENT name_ent (#PCDATA)>
    <!ELEMENT type (#PCDATA)>
    <!ELEMENT contact (contact_name, contact_phone, contact_fax?, email?, function?, department?)>
     <!ELEMENT contact_name (#PCDATA)>
     <!ELEMENT contact_phone (#PCDATA)>
     <!ELEMENT contact_fax (#PCDATA)>
     <!ELEMENT email (#PCDATA)>
     <!ELEMENT function (#PCDATA)>
     <!ELEMENT department (#PCDATA)>
    <!ELEMENT address (street, number, aditional, zipcode, POBox, city, region?, country, web)>
     <!ELEMENT street (#PCDATA)>
     <!ELEMENT number (#PCDATA)>
     <!ELEMENT aditional (#PCDATA)>
     <!ELEMENT zipcode (#PCDATA)>
     <!ELEMENT POBox (#PCDATA)>
     <!ELEMENT city (#PCDATA)>
     <!ELEMENT region (#PCDATA)>
     <!ELEMENT country (#PCDATA)>
     <!ELEMENT web (#PCDATA)>
   <!ELEMENT product_info (code_prod, name_prod, max_capacity, allocated_capacity,
                production_price, color?, volume?, height?, width?, weight?, material?, productType?)>
    <!ELEMENT code_prod (#PCDATA)>
    <!ELEMENT name_prod (#PCDATA)>
    <!ELEMENT max_capacity (#PCDATA)>
    <!ELEMENT allocated_capacity (#PCDATA)>
    <!ELEMENT production_price (#PCDATA)>
    <!ELEMENT color (#PCDATA)>
    <!ELEMENT volume (#PCDATA)>
    <!ATTLIST volume unit CDATA #REQUIRED>
    <!ELEMENT height (#PCDATA)>
    <!ATTLIST height unit CDATA #REQUIRED>
    <!ELEMENT width (#PCDATA)>
    <!ATTLIST width unit CDATA #REQUIRED>
    <!ELEMENT weight (#PCDATA)>
    <!ATTLIST weight unit CDATA #REQUIRED>
    <!ELEMENT material (#PCDATA)>
    <!ELEMENT productType (#PCDATA)>
```

Figura 53 – DTD da mensagem *EnterpriseRegistration.*

Mensagem *ConnectionResponse*

Esta mensagem é enviada por uma empresa ao coordenador em resposta a um pedido de conexões. Ela informa que empresas estão relacionadas e quais os produtos uma intercambia com a outra. Seu conteúdo é formado pelas seguintes informações: nome do coordenador (*name_coord*), nome da EV (*name_virtualEnterprise*) e a lista de conexões (*connection_list*). A lista de conexões é formada pelo código da empresa que origina a conexão (*code_source*), o nome da empresa destino da conexão (*name_dest*) e uma lista com os códigos dos produtos da empresa origem que fazem parte deste relacionamento (*prod_list*). A Figura 54 mostra o DTD que especifica esta mensagem.

```
<!-- DTD Connection Response -->

<!ELEMENT message (content)>
  <!ATTLIST message msg_type CDATA #REQUIRED>
  <!ATTLIST message sender CDATA #REQUIRED>

  <!ELEMENT content (connection_response)>
    <!ELEMENT connection_response (name_coord,
                                   name_vitualEnterprise,
                                   connection_list*)>
      <!ELEMENT name_coord (#PCDATA)>
      <!ELEMENT name_vitualEnterprise (#PCDATA)>
      <!ELEMENT connection_list (code_source,
                                   name_dest,
                                   prod_list)>
        <!ELEMENT code_source (#PCDATA)>
        <!ELEMENT name_dest (#PCDATA)>
        <!ELEMENT prod_list (code_prod*)>
          <!ELEMENT code_prod (#PCDATA)>
```

Figura 54 – DTD da mensagem *ConnectionResponse.*

Mensagem *VERegistration*

Após a configuração da empresa virtual, o coordenador envia uma mensagem para cada um dos membros da EV com a topologia da mesma. Esta mensagem é chamada de *VERegistration* e contém todas as informações necessárias para sua exibição na interface gráfica dos membros. Dentre as informações que fazem parte desta mensagem podemos destacar as seguintes: o código (*code*) e o nome (*name*) da EV, a lista de membros (*members_list*), a lista de conexões (*connections_list*) e a lista de empresas (*enterprises_list*). A lista de membros contém informações tais como: nome, papel e posição do mesmo da EV. A lista de conexões traz informações dos códigos das empresas origem e destino da conexão, bem como uma lista com os códigos dos produtos que fazem

parte desta conexão. A lista de empresas contém as informações básicas das empresas que fazem parte da empresa virtual configurada. As informações contidas nesta lista são as mesmas enviadas por cada empresa para o coordenador nas mensagens do tipo *EnterpriseRegistration*. Na Figura 55 é apresentada a especificação completa do DTD desta mensagem.

```
<!-- DTD Virtual Enterprise -->

<!ELEMENT message (content)>
  <!ATTLIST message msg_type CDATA #REQUIRED>
  <!ATTLIST message sender CDATA #REQUIRED>

  <!ELEMENT content (virtual_enterprise)>
    <!ELEMENT virtual_enterprise (code, name, members_list, connections_list, enterprises_list)>
      <!ELEMENT code (#PCDATA)>
      <!ELEMENT name (#PCDATA)>
      <!ELEMENT members_list (member*)>
        <!ELEMENT member (cd_member, cd_enterprise, label, pos_x, pos_y, custer)>
          <!ATTLIST member role (pre_supplier |supplier |principal_producer |distributor |broker |
                                 retailer |cluster ) #REQUIRED>
          <!ELEMENT cd_member (#PCDATA)>
          <!ELEMENT cd_enterprise (#PCDATA)>
          <!ELEMENT label (#PCDATA)>
          <!ELEMENT pos_x (#PCDATA)>
          <!ELEMENT pos_y (#PCDATA)>
          <!ELEMENT custer (#PCDATA)>
      <!ELEMENT connections_list (connection*)>
        <!ELEMENT connection (code_conn, member_src, member_dest, items_list)>
          <!ELEMENT code_conn (#PCDATA)>
          <!ELEMENT member_src (#PCDATA)>
          <!ELEMENT member_dest (#PCDATA)>
          <!ELEMENT items_list (item+)>
            <!ELEMENT item (enterprise_src, cd_item)>
              <!ELEMENT enterprise_src (#PCDATA)>
              <!ELEMENT cd_item (#PCDATA)>
      <!ELEMENT enterprises_list (enterprise*)>
        <!ELEMENT enterprise (cd_enterprise, nm_name, nm_identifier, nm_street, nm_city,
                              nm_state, nm_country, products_list)>
          <!ELEMENT nm_name (#PCDATA)>
          <!ELEMENT nm_identifier (#PCDATA)>
          <!ELEMENT nm_street (#PCDATA)>
          <!ELEMENT nm_city (#PCDATA)>
          <!ELEMENT nm_state (#PCDATA)>
          <!ELEMENT nm_country (#PCDATA)>
          <!ELEMENT products_list (product*)>
            <!ELEMENT product (cd_product, nm_description, nm_producttype, nm_color)>
              <!ELEMENT cd_product (#PCDATA)>
              <!ELEMENT nm_description (#PCDATA)>
              <!ELEMENT nm_producttype (#PCDATA)>
              <!ELEMENT nm_color (#PCDATA)>
```

Figura 55 – DTD da mensagem *VERegistration.*

Anexo B: Detalhes de Implementação do SMA

O capítulo 4 apresentou, de forma geral, todos os aspectos envolvidos na implementação do protótipo que validou o modelo. Entretanto, este capítulo não tinha como propósito apresentar de forma detalhada a implementação dos diversos algoritmos implementados para a resolução do problema. Portanto, este anexo tem por objetivo apresentar uma pequena parte da implementação dos algoritmos desenvolvidos no protótipo de configuração de EVs.

A título de exemplo, o trecho de código apresentado neste anexo está relacionado à inclusão de um novo membro na topologia da EV. A descrição do processo inicia-se com a chegada de uma mensagem do SMA de uma empresa ao agente XML do SMA coordenador, indicando a inserção desta empresa como um novo membro da EV sendo configurada. A classe *TxmlProcessorPartnerResponse* do agente XML e a função *DOMToPartnerRespose* desta classe, que interpreta a mensagem de inclusão de um novo membro, são apresentadas respectivamente na Figura 56 e na Figura 57. O diagrama de classes completo do agente XML pode ser visto na Figura 33.

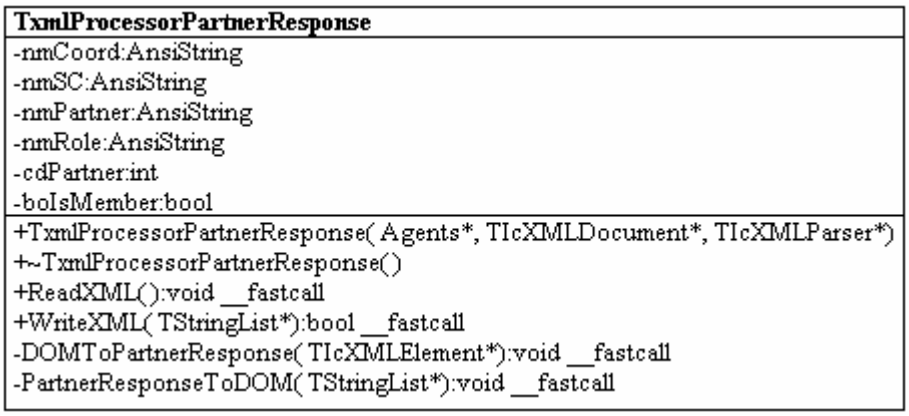


Figura 56 – Classe *TxmlProcessorPartnerResponse* do agente XML.

Após a mensagem ser decodificada pela função *DOMToPartnerRespose* do agente XML do coordenador, o referido agente envia uma mensagem com os parâmetros recebidos do SMA da empresa para o agente Configurador do coordenador solicitando a inclusão de um novo membro na EV.

```
//---------------------------------------------------------------------------
//  Function: DOMToPartnerResponse
//   Purpose:  This function sends a message to inform the agent
//             Configurator that a new member should be put in the EV.

void __fastcall TxmlProcessorPartnerResponse::DOMToPartnerResponse
      (TIcXMLElement *el)
{
   TIcXMLElement *tmpEle = 0;

   tmpEle = el->GetFirstChild(); // name_coord
   nmCoord = TestString(tmpEle);
   tmpEle = tmpEle->NextSibling(); // name_supplychain
   nmSC = TestString(tmpEle);
   tmpEle = tmpEle->NextSibling(); // code_partner
   cdPartner = ConvertToInt(tmpEle);
   tmpEle = tmpEle->NextSibling(); // name_partner
   nmPartner = TestString(tmpEle);
   tmpEle = tmpEle->NextSibling(); // role
   nmRole = TestString(tmpEle);
   tmpEle = tmpEle->NextSibling(); // is_member
   boIsMember = StrToBool(TestString(tmpEle));
   if(boIsMember) {
     agent->hsSend( "Config", "m, "+IntToStr(cdPartner)+", "+nmPartner+",
     "+nmRole);
   }
}
//---------------------------------------------------------------------------
```

Figura 57 – Função *DOMToPartnerRespose* da classe *TxmlProcessorPartnerResponse.*

Assim que a mensagem chega ao agente Configurador, a mesma é analisada e a ação correspondente ao seu conteúdo é tomada. Como a mensagem em questão tem por objetivo adicionar uma nova empresa como membro da EV, esta será a ação a ser tomada pelo agente Configurador. A classe *TAgents* do agente Configurador e a função *checkMessage* desta classe, que interpreta a mensagem recebida do agente XML e invoca um objeto da classe *TSmartMap* para inserir um novo membro na EV, são apresentadas respectivamente na Figura 58 e na Figura 59. O diagrama de classes completo do agente Configurador pode ser visto na Figura 28.

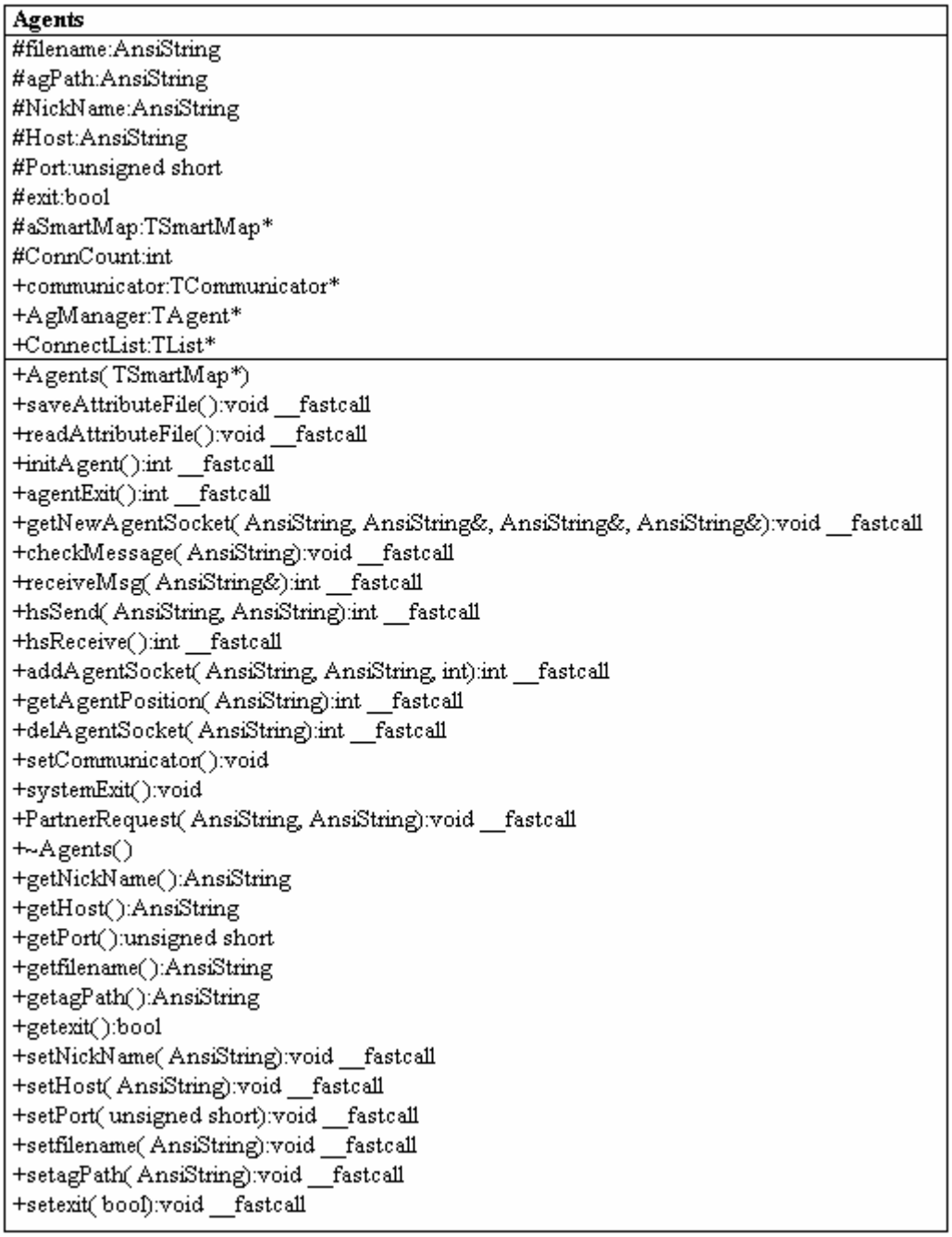


Figura 58 – Classe *TAgents* do agente Configurador.

```
//---------------------------------------------------------------------------
//  Function: checkMessage
//  Purpose: This Function analyzes all messages what arrived to the agent.

void __fastcall Agents::checkMessage( AnsiString mesg)
{
  ...
  // Message to add a new member in the EV.
  // <m, nmPartner, cdPartner, nmRole>
  case 'm' :
    {
        AnsiString nmPartner, nmRole;
        int cdPartner;

        ConnCount = 0;
        // Reject the first token 'm'
        nmPartner = Trim(strtok(mesg.c_str(), separador.c_str()));
        cdPartner = StrToInt(Trim(strtok(NULL, separador.c_str())));
        nmPartner = Trim(strtok(NULL, separador.c_str()));
        nmRole = Trim(strtok(NULL, separador.c_str()));

        TMember_DB *aMember = new TMember_DBInterbase();
        aMember->setcdSupplyChain(100);  //Defalt value
        aMember->setcdMember(0); //Defalt value
        aMember->setnmRole(nmRole);
        aMember->setcdDP(cdPartner);
        aMember->setnmLabel(nmPartner);
        aMember->setcdCluster(0);
        TPoint p;
        p.x = 1;
        p.y = 1;
        aSmartMap->setPosXY(StrToRole(aMember->getnmRole()), p);
        aMember->setnuPosX(p.x);
        aMember->setnuPosY(p.y);
        aSmartMap->createImage(StrToRole(aMember->getnmRole()),
       p, aMember);
    }
    break;
    ...
}
//---------------------------------------------------------------------------
```

Figura 59 – Função *checkMessage* da classe *TAgents*.

A partir do momento que o agente Configurador interpreta a mensagem vinda do agente XML, o mesmo invoca a função *createImage* da classe *TSmartMap*. Esta função é responsável por criar e adicionar a imagem / ícone correspondente ao papel desempenhado pela empresa que está sendo incluída como novo membro da EV. A classe *TSmartMap* do agente Configurador e a função *createImage* desta classe são apresentadas respectivamente na Figura 60 e na Figura 61.

A partir do momento que a imagem / ícone da empresa que enviou a mensagem for incluída na topologia, o processamento da mesma é encerrado e uma nova mensagem vinda de outras empresas pode ser processada.

Figura 60 – Classe *TSmartMap* do agente Configurador.

```
//---------------------------------------------------------------------------
//  Function: createImage
//  Purpose: This Function create a new TCustomImage instance and put it
//           in the VE topology.

void __fastcall TSmartMap::createImage(TRole role, TPoint p,
                                       TMember_DB *aMember)
{
  TCustomImage *image = new TCustomImage(this,role,p, aMember);

  setmodified(true);

  // To unselect the connection before selected.
  connectionList->connectionExists(-1, -1);

  if (image->getRole() == cluster)
    image->PopupMenu = clusterImageMenu;
  else
    image->PopupMenu = imageMenu;
  image->OnMouseDown = ImageMouseDown;
  image->OnMouseUp =  ImageMouseUp;
  image->OnMouseMove =  ImageMouseMove;
  supplyChain->addMember(aMember);
  imageList->Add(image);
  imageList->selectImage(image);
  if (aMember->getcdSupplyChain() == 0)
    if (image->getRole() != cluster)
      showFormSelectEnterprise();
    else
      showFormClusterEnterprise();
  paintBox->Invalidate();
}
//---------------------------------------------------------------------------
```

Figura 61 – Função *createImage* da classe *TSmartMap.*

Glossário

A
ACTS – *Advanced Communication Technologies and Services*.
ANSI – *American National Standard Institute*.
API – *Application Programming Interfaces*.

B
BMBF – *Federal Ministry of Education and Resource*.

C
CD – *Cluster Directory*.
CORBA – *Common Object Request Broker Architecture*.
CSCW – *Computer Supported Cooperative Work*.

D
DAMASCOS – *Dynamic Forecast for Master Production Planning with Stock and Capacity Constraints*.
DII – *Dynamic Invocation Interface*.
DOM – *Document Object Model*.
DTD – *Document Type Definition*.

E
ebXML – *Electronic Business Extensible Markup Language*.
EDIFACT – *Electronic Data Interchange for Administration, Commerce and Transport*.
ERP – *Enterprise Recourse Planning*.
ESD – *External Suppliers Directory*.
ESIOP – *Environment-Specific Inter-ORB Protocol*.
ETTN – *European Technology Transfer Network*.
EV – Empresa Virtual.

G
GIOP – *General Inter-ORB Protocol*.

H
HTML – *Hypertext Markup Language*.

I
IA – Inteligência Artificial.
IAD – Inteligência Artificial Distribuída.
IDL – *Interface Definition Language*.
IFIP – *International Federation for Information Processing*.
IIOP – *Internet Inter-ORB Protocol*.
IMS – *Intelligent Manufacturing Systems*.
ISD – *Internal Suppliers Directory*.
ISO – *International Organization for Standardization*.

IST – *Information Society Technologies*.

M
Massyve-Kit – *Multiagent Agile Manufacturing Scheduling Systems for Virtual Enterprise*.
Middleware – Camada de software que provê serviços para aplicações distribuídas.
MR – Modelo de Referência.

O
OMA – *Object Management Architecture*.
OMG – *Object Management Group*.
ORB – *Object Request Broker*.

P
POA – *Portable Object Adaptor*.
PRODNET – *Production Planning and Management in an Extended Enterprise*.

R
RPN – Reengenharia de Processo de Negócio.

S
SC^2 – *Supply Chain Smart Coordination*.
SDP – Solução Distribuída de Problemas.
SGBD – Sistema de Gerenciamento de Banco de Dados.
SGML – *Standard Generalized Markup Language*.
SGWF – Sistema de Gerenciamento de *Workflow*.
SMA – Sistema Multiagente.
SME – *Small and Medium Enterprises*.
SOCKET – É um objeto do sistema operacional que representa o ponto de acesso a um canal de comunicação entre processos. Teve origem no UNIX BSD 4.1 (1982), mas hoje é adotado por quase todos os sistemas operacionais.
SPD – Solução de Problemas Distribuídos.
SQL – *Structured Query Language*.
STEP – *Standard for the Exchange of Product Model Data.*

T
TCP/IP – *Transmission Control Protocol / Internet Protocol*.
TI – Tecnologia de Informação.

V
VERA – *Virtual Enterprise Reference Architecture*.

X
XML – *Extensible Markup Language*.

W
W3C – *World Wide Web Consortium*.
WAN – *Wide Area Network*.
WAPI – *Workflow API and Interchange*.

WFBB – *Workflow Backbone*.
WfMC – *Workflow Management Coalition*.
WPDL – *Workflow Process Definition Language*.
WWW – *World Wide Web*.

Referências Bibliográficas

ALONSO, G.; FIEDLER, U.; LAZCANO, A.; *et al.*; 1999. WISE: An Infrastructure for E-Commerce. In: WORKSHOP INFORMATIK '99 ENTERPRISE-WIDE AND CROSS-ENTERPRISE WORKFLOW MANAGEMENT: CONCEPTS, SYSTEMS, APPLICATIONS (Oct. 1999 : Paderborn, Germany). *Proceedings*. p. 3-11.

ARMSTRONG, E.; 2001. *Working with XML. The Java API for XML Parsing Tutorial*. www.java.sun.com.

ATLURI, V.; 2001. Security for Workflow Systems. Information Security Technical Report, Elsevier Science, v. 6, n.2, p. 59-68.

ÁVILA, P.; PUTNIK, G. D.; CUNHA, M. M.; 2002. Brokerage Function in Agile Virtual Enterprise Integration – A Literature Review. In: THIRD WORKING CONFERENCE ON INFRASTRUCTURES FOR VIRTUAL ENTERPRISES (PRO-VE'02 : May 2002). *Proceedings*. Sesimbra, Portugal. p. 65-72.

BARATA, J.; CAMARINHA-MATOS, L. M.; 2002. Contract Management in Agile Manufacturing Systems. In: THIRD WORKING CONFERENCE ON INFRASTRUCTURES FOR VIRTUAL ENTERPRISES (PRO-VE'02 : May 2002). *Proceedings*. Sesimbra, Portugal. p. 109-122.

BERG, R.; HANNUS, M.; TOLLE, M.; 2000. GLOBEMEN.EU: Evaluation of State of the Art Technologies. Deliverable 411.

BIGUS, J.; BIGUS, J.; 1998. *Constructing Intelligent Agents with Java*. Willey Computer Publishing.

BROOS, R.; BELL, A.; DILLENSEGER, B.; 2000. MIAMI: Mobile Intelligent Agents for Managing the Information Infrastructure.

BULTJE, R.; WIJK J.; (1998). *Taxonomy of Virtual Organisations, Based on Definitions, Characteristics and Typology*. v. 2, n. 3, p. 7-20.

BYRNE, J. A.; 1993. The Virtual Corporation. In: INTERNATIONAL BUSINESS WEEK (Fev. 1993). *Proceedings*. p. 36-41.

CAMARINHA-MATOS, L. M.; AFSARMANESH, H.; 1997. Virtual Enterprises: Life Cycle Supporting Tools and Technologies. In: Handbook of Life Cycle Engineering: Concepts, Tools and Techniques. A. Kusiak. Chapman and Hall.

CAMARINHA-MATOS, L. M.; AFSARMANESH, H.; 1999a. The Virtual Enterprise Concept. In: WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ENTERPRISES (PRO-VE'99 : Oct. 1999 : Porto, Portugal). *Proceedings*. Porto, Portugal. p. 3-14.

CAMARINHA-MATOS, L. M.; CARDOSO, T.; 1999b. Selection of Partners for a Virtual Enterprise. In: WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ENTERPRISES (PRO-VE'99 : Oct. 1999 : Porto, Portugal). *Proceedings*. Porto, Portugal. p. 259-278.

CAMARINHA-MATOS, L. M.; AFSARMANESH, H.; 1999c. The PRODNET Architecture. In: WORKING CONFERENCE ON INFRASTRUCTURE FOR

VIRTUAL ENTERPRISES (PRO-VE'99 : Oct. 1999 : Porto, Portugal). *Proceedings*. Porto, Portugal. p. 109-126.

CAMARINHA-MATOS, L. M.; AFSARMANESH, H.; 1999d. The PRODNET Goals and Approach. In: WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ENTERPRISES (PRO-VE'99 : Oct. 1999 : Porto, Portugal). *Proceedings*. Porto, Portugal. p. 97-108.

CAMARINHA-MATOS, L. M.; AFSARMANESH, H.; RABELO, R. J.; 2000. Support Agility in Virtual Enterprise. In: SECOND WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ORGANIZATIONS (PRO-VE'00 : Dec. 2000). *Proceedings*. Florianópolis, Brasil. p. 89-104.

CAMARINHA-MATOS, L. M.; AFSARMANESH, H.; RABELO, R. J.; 2001. Infrastructure Developments for Agile Virtual Enterprises. *International Journal on Computer Integrated Manufacturing*.

CARVALHO, M.; MACHADO, C.; 2000. Environment for Design and Analysis of System Integration. In: SECOND WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ORGANIZATIONS (PRO-VE'00 : Dec. 2000). *Proceedings*. Florianópolis, Brasil. p. 127-134.

COSTA, A. C. P. L.; 1997. Expert-Coop: *Um ambiente para desenvolvimento de Sistemas Multi-Agentes Cognitivos*. Florianópolis. Dissertação (Mestrado em Engenharia Elétrica) – Centro Tecnológico, Universidade Federal de Santa Catarina.

DAS, S.; KOCHUT, K.; MILLER, J.; *et al.*; 1997. ORBWork: A Reliable Distributed CORBA-Based Workflow Enactment System for METEOR_2, Technical Report #UGA-CS-TR-97-001. Department of Computer Science, University of Georgia.

DELOACH, S. A.; 1999. Multiagent Systems Engineering: A Methodology and Language for Designing Agent Systems. In: AGENT-ORIENTED INFORMATION SYSTEM (AOIS'99 : May, 1999 : Seattle). Proceedings. p. 45-57.

DOZ, Y. L.; HAMEL, G.; 1998. *Alliance Advantage. The Art of Creating Value through Partinering.* Boston: Harvard Business School Press.

DU, W.; ELMAGARMID, A.; 1997. Workflow Management: State of the Art vs. State of the Products. Hewlett Packard: Software Technology Laboratory.

EBXML; 2001. Requirements Specification. v. 1.06.

FERBER, J.; 1999. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Reading : Addison-Wesley.

FERREIRA, D.; FERREIRA, H.; ANTUNES, N.; 2000a. Architecture of DAMASCOS Suite and of Distributed Workflow Backbone.

FERREIRA, D.; FERREIRA, H.; FERREIRA, J.; 2000b. Distributed Workflow Backbone.

FILOS, E.; DEVINE, M.; 2000. Virtual Teams and the Organizational Grapevine. In: SECOND WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ORGANIZATIONS (PRO-VE'00 : Dec. 2000). *Proceedings*. Florianópolis, Brasil. p. 413-424.

FRANKLIN, S.; GRAESSER, A.; 1996. Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In: INTERNATIONAL WORKSHOP ON AGENT

THEORIES, ARCHITECTURES, AND LANGUAGES (3rd : Aug. 1996 : Budapest, Hungary). *Proceedings*. p. 121-131.

FRAYRET, J. M.; CLOUTIER, L.; MONTREUIL, B.; *et al.*; 2000. A Distributed Framework for Collaborative Supply Network Integration. In: SECOND WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ORGANIZATIONS (PRO-VE'00 : Dec. 2000). *Proceedings*. Florianópolis, Brasil. p. 233-243.

FRENKEL, A.; AFSARMANESH, H.; GARITA, C.; *et al.*; Supporting Information Access Rights and Visibility Levels in Virtual Enterprises. In: SECOND WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ORGANIZATIONS (PRO-VE'00 : Dec. 2000). *Proceedings*. Florianópolis, Brasil. p. 177-192.

GLEIZES, M. P.; GLIZE, P.; 2002. ABROSE: Multi Agent Systems for Adaptive Brokerage. In: INTERNATIONAL BI-CONFERENCE WORKSHOP ON AGENT-ORIENTED INFORMATION SYSTEMS (4th : May 2002 : Toronto, Canada). *Proceedings*. p. 30-34.

GORANSON, H. T.; 1999. *The Agile Virtual Enterprise*. Quorum Books.

GUIMARÃES, N.; PEREIRA, A. P.; ANTUNES, P.; 1997. Bridging WorkFlow and Collaboration Tools. In: 8th EUROGDSS WORKSHOP / 7th MINI EURO CONFERECE (Mar., 1997 : Bruges). *Proceedings*.

HALARIS, C.; BAFOUTSOU, G.; PAPAVASSILIOU, G.; *et al.*; 2001. A System for Virtual Tendering and Bidding. In: PANHELLINIC CONFERENCE IN INFORMATICS (8th : Nov. 2001: Nicosia, Cyprus). *Proceedings*. p. 173-183;

HANDS, J.; BESSONOV, M.; BLINOV, M.; *et al.*; 2000. An Inclusive and Extensible Architecture for Electronic Brokerage. In: DECISION SUPPORT SYSTEMS. *Proceedings*. p. 305-321.

HAROLD, E. R.; MEANS, W. S.; 2002. *XML in a Nutshell*. 2 ed. O'Reilly.

HOFFNER, Y.; FIELD, S.; GREFEN, P.; LUDWIG, H.; 2001. *Contract-driven creation and operation of virtual enterprises. Computer Network*, v. 37, p. 111-136.

HOFFNER, Y.; SCHADE, A.; FIELD, S.; 2002. Negotiation Protocol Characterization and Mechanisms for Virtual Markets and Enterprises. In: THIRD WORKING CONFERENCE ON INFRASTRUCTURES FOR VIRTUAL ENTERPRISES (PRO-VE'02 : May 2002). *Proceedings*. Sesimbra, Portugal. p. 133-140.

HORN, E.; KUPRIES, M.; GLÖDE, D.; 1998. Properties and Models of Software Agents and Agent Systems. In: AAAI WORKSHOP TECHNICAL REPORT. *Proceedings*. Madison Wisconsin. p. 124- 140.

IBM; 2001. Introduction to ebXML. www.ibm.com/developerworks.

IDRIS, N.; 1999. Should I use SAX or DOM? developerlife.com.

JACOMINO, A.; GESZYCHTER, M. B.; 1999. *Massyve Kit: Um Ambiente de Suporte ao Desenvolvimento de Aplicações Multiagente*. Florianópolis. Trabalho de conclusão de curso (Bacharel em Ciências da Computação) – Centro Tecnológico, USFC.

KANET, J. J.; FAISST, W.; MERTENS, P.; 1999. Application of Information Technology to a Virtual Enterprise Broker: The Case of Bill Epstein. *International Journal of Production Economics*, n. 62, p. 23-32.

KATZY, B. R.; SUNG, G.; 2003. State of the Art of Virtual Organization Modeling. IN: THE E-BUSINESS E-WORK CONFERENCE (Venice, Italy). *Proceedings*.

KENDALL, E. A.; MALKOUN, M. T.; JIANG, C. H.; 1997. Multiagent System Design Based on Object Patterns. The Journal of Object Oriented Programming. June.

KLEN, A.; RABELO, R. J.; KLEN, E.; 2002. *Deliverable D 8.1 – Requirements Identification*. System Requirements and User Needs.

KÜRÜMLÜOGLU, M.; 2003. VOSTER: Virtual Organisations Cluster – Virtial Organization Concepts. IST-2001-32031.

LANGE, D. B.; 1998. Mobile Objects and Mobile Agents: The Future of Distributed Computing. In: THE EUROPEAN CONFERENCE ON OBJECT-ORIENTED PROGRAMMING (July, 1998 : Brussels, Belgium). *Proceedings*. p. 63-75.

LESSER, V. R.; LAASRI, B.; LAASRI, H.; *et al.*; 1992. A Generic Model for Intelligent Negociation Agents. *International Journal on Intelligent Cooperative Information Systems*, v. 1, n. 2 (Jan.), p. 291-317.

LESSER, V. R.; 1999. Cooperative Multiagent Systems: A Personal View of the State of the Art. *IEEE Transaction on Knowledge and Data Engineering*, v. 11, n. 1 (Jan.), p. 133-142.

LIMA, C. P.; 2001. *Um Modelo Multinível de Coordenação em Ambiente de Empresa Virtual. Lisboa*, Portual. Tese (Doutorado em Engenharia Eletronica) – Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

LOGAN, B.; 1998. Classifying Agent Systems. In: CONFERENCE WORKSHOP ON SOFTWARE TOOLS FOR DEVELOPING AGENTS (AAAI-98 : California). Proceedings. p-53-63.

LUNG, L. C.; 2001. *Experiências com Tolerância a Faltas no CORBA e Extensões ao FT-CORBA para Sistemas Distribuídos de Larga Escala*. Florianópolis. Tese (Doutorado em Engenharia Elétrica) – Centro Tecnológico, Universidade Federal de Santa Catarina.

MAES, P.; 1991. *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. London : The MIT Press.

MARTIN, J.; 1999. COWORK: IT Tools to Support Concurrent Project Development in Networks of SMEs. (ICE' 99 : Mar. 1999). *Proceedings*. The Hague, Netherlands.

MEJÍA, R.; ACA, J.; GARCÍA, E.; MOLINA, A.; 2002. Knowledge and Technology Integration in Production and Service. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY FOR BALANCED AUTOMATION SYSTEMS (BASYS' 02 : Sep. 2002). *Proceedings*. Cancun, México. p. 141-148.

MENG, J.; SU, S.; LAM, H.; *et al.*; 2002. Achieving Dynamic Inter-organizacional Workflow Management by Integrating Business Processes, Events and Rules. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HISS-35). *Proceedings*. Hawaii. p. 10-35.

MIN, H.; ZHOU, G.; 2002. Supply Chain Modeling: past, present and future. *Computers & Industrial Engineering*, v. 43, p. 231-249.

MORRISON, M.; *et al.*; 2000. XML Unleashed. SAMS.

NWANA, H. S.; 1996. Software Agents: An Overview. *Knowledge Engineering Review*. Cambridge University Press, v. 11, n. 3 (nov. 1996), p. 205-244.

OLLUS, M.; KARVONEN, I.; JANSSON, K.; *et al.*; 2003. *Deliverable D1 – Interim Report on Consolidated Baseline*. Roadmap Design for Collaborative Virtual Organizations in Dynamic Business Ecosystems.

OMG; 1996. Object Management Architecture. Madrid, Aug.

OMG; 1998. Workflow Management Facility. OMG BODTF RFP #2 Submission. July, 1998.

OMG; 2002. Common Object Request Broker Architecture: Core Specification. version 3.0.2. Dec.

OMG; 2003. XML Metadata Interchange (XMI) Specification. version 2.0. May.

OUZOUNIS, E. K.; TSCHAMMER, V.; 1999. A Framework for Virtual Enterprise Support Services. In: 32nd INTERNATIONAL CONFERENCE ON SYSTEMS AND SCIENCES (HICSS32) (Jan. : Maoui, Hawaii). *Proceedings*.

OUZOUNIS, E. K.; 2001. *An Agent-Based Platform for the Management of Dynamic Virtual Enterprises*. Berlin, Germany. Thesis Dr. Ing. Electronic and Informatic Department, University of Berlin.

PANG, G.; 2000. *Implementation of an Agent-Based Business Process*. Sichuan, China. thesis. Institute of Informatics, Zuric University.

PARK, K. H.; FAVREL, J.; 1999. Virtual Enterprise – Information System and Networking Solution. *Computers & Industrial Engineering*, n. 37, p. 441-444.

RABELO, R. J.; SPINOSA, L. M.; 1997. Mobile-Agent-Based Supervision in Supply Chain Management in the Food Industry. In: WORKSHOP ON SUPPLY CHAIN MANAGEMENT (AGROSOFT'97). *Proceedings*.

RABELO, R. J.; AFSARMANESH, H.; CAMARINHA-MATOS, L. M.; 1999a. Applying Federated Databases to Inter-Organizational Multiagent Scheduling. In: FIRST INTERNATIONAL IFAC WORKSHOP ON MULTIAGENT SYSTEMS IN PRODUCTION (MAS'99 : Vienna, Austria). *Proceedings*.

RABELO, R. J.; KLEN, A. P.; SPINOSA, L. M.; 1999b. Agile Supply-Chain Coordination in the Virtual Enterprise. In: FORTH BRAZILIAN SIMPOSIUM OF INTELLIGENT AUTOMATION (4th SBAI : São Paulo, Brasil). *Proceedings*.

RABELO, R. J.; CAMARINHA-MATOS, L. M.; VALLEJOS, R. V.; 2000. Agent-Based Brokerage for Virtual Enterprise Creation in the Moulds Industry. In: SECOND WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ORGANIZATIONS (PRO-VE'00 : Dec. 2000). *Proceedings*. Florianópolis, Brasil. p. 281-290.

RABELO, R. J.; KLEN, A. P.; KLEN, E. R.; 2002. A Multiagent System for Smart Coordination of Dynamic Supply Chains. In: THIRD WORKING CONFERENCE ON INFRASTRUCTURES FOR VIRTUAL ENTERPRISES (PRO-VE'02 : May 2002). *Proceedings*. Sesimbra, Portugal. p. 379-386.

REID, R. L.; ROGERS, K. J.; JOHNSON, M. *et al.*; 1996. Engineering the Virtual Enterprise. In: 5th IERC (May : Minneapolis, MN). *Proceedings*. Minneapolis, 1996.

RESNICK, P.; AVERY, R.; 1994. Roles for Electronic Brokers. In: TELECOMMUNI CATIONS POLICY RESEARCH CONFERENCE. *Proceedings*. p. 289-304.

REZGUI, Y.; ZARLI, A.; WILSON, I.; 2000. OSMOS: Project Presentation. Deliverable 5.3.

SCHMELZER, R.; 2001. The Pros and Cons of XML. ZapThink Foundation Report. Sep., 2001. www.zapthink.com.

SCHMIDT, R.; 2003. *Busca e Seleção de Parceiros para Empresas Virtuais: Uma Abordagem Baseada em Agentes Móveis*. Florianópolis. Dissertação (Mestrado em Engenharia Elétrica) – Centro Tecnológico, Universidade Federal de Santa Catarina.

SEVERWRIGHT, J.; SINGH, R.; 1998. VE Technological Requirements: The VENICE Project. In: OBJECTS, COMPONENTS AND THE VIRTUAL ENTERPRISE (OCVE'98). Proceedings. p. 53-65.

SOARES, A. L.; SOUSA, J. P.; 2002. Multiple Perspective Configuration of Virtual Enterprises Using Social Actors Networks. In: THIRD WORKING CONFERENCE ON INFRASTRUCTURES FOR VIRTUAL ENTERPRISES (PRO-VE'02 : May 2002). *Proceedings*. Sesimbra, Portugal. p. 273-280.

SPINOSA, L. M.; RABELO, R. J.; KLEN, A. P.; 1998. High-Level Coordination of Business Processes in a Virtual Enterprise. In: THE TENTH INTERNATIONAL IFIP TC5 WG-5.2 WG-5.3 CONFERENCE (PROLAMAT'98). *Proceedings*. p. 55-67.

STONE, P.; VELOSO, M.; 2000. Multiagent Systems: A Survey from Machine Learning Perspective. *Autonomous Robotics*, v. 8, n. 3, p. 72-129.

SYCARA, K. P.; 1998. Multiagent Systems. *AI Magazine*, v. 19, n. 2, p. 79-92.

TOLLE, M.; ZWEGERS, A.; VESTERAGER, J.; 2003. GLOBEMEN.EU: Virtual Enterprise Reference Architecture and Methodology (VERAM). Joint D41 / D43 deliverable.

UN/CEFACT; 2002. ebXML Core Components Technical Specification, Part 1.

VINOSKI, S.; 1998. New Features for CORBA 3.0. *Communication of the ACM*, v. 41, n. 10 (Aug.), p. 44-52.

XIANG, Y.; 2002. Probabilistic Reasoning in Multiagent Systems. 1. ed. Cambridge, UK : Cambridge University Press.

WALTON, J.; WHICKER, L.; 1996. Virtual Enterprise: Myth and Reality. (Oct. 1996).

WFMC; 1995. Workflow Management Coalition: The Workflow Reference Model, WFMC-TC-1003. Hampshire, United Kingdown.

WFMC; 1996. Workflow Management Coalition: Terminology and Glossary, WFMC-TC-1011. Brussels, Belgium.

WFMC; 1999. Workflow Management Coalition: Process Definition Meta-Model and WPDL, WFMC-TC-1016. Brussels, Belgium.

WHITAKER, B.; 1999. *Intelligent Agents for Electronic Commerce*.

WOGNUM, P. M.; FABER, E. C.; 1999. A Framework for Improving the Quality of Operation in a Virtual Enterprise. In: WORKING CONFERENCE ON

INFRASTRUCTURE FOR VIRTUAL ENTERPRISES (PRO-VE'99 : Oct. 1999 : Porto, Portugal). *Proceedings*. Porto, Portugal. p. 365-376.

WOOLDRIDGE, M.; JENNINGS, N. R.; 1995. Intelligent Agents: Theory and Practice. Knowledge Engineering Review. v. 10, n. 2 (Jan.), p. 115-152.

WOOLDRIDGE, M.; JENNINGS, N. R.; 1998. A Pitfalls of Agent-Oriented Development. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES (5th : July 1998 : Paris, France). *Proceedings*. p.385-391.

ZARLI, A.; POYET, P.; 1999. A Framework for Distributed Information Management in the Virtual Enterprise: The VEGA Project. In: WORKING CONFERENCE ON INFRASTRUCTURE FOR VIRTUAL ENTERPRISES (PRO-VE'99 : Oct. 1999 : Porto, Portugal). *Proceedings*. Porto, Portugal. p. 293-306.

ZARLI, A.; REZGUI, Y.; 2001. OSMOS: Inter-company Interaction Process Model. Deliverable 1.2.

ZHONGQIAO, L; 2000. *Using Collaboration Task in Orbwork Enactment System for Meteor Workflow Management System*. Athens, Georgia. Thesis Dr. Sc. University of Georgia.