

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE  
ÁREA: CIÊNCIA DA COMPUTAÇÃO

**NATHÁLIA RIBEIRO SCHALCHER DE OLIVEIRA**

**FORMAÇÃO E CUMPRIMENTO DE CONTRATOS  
ELETRÔNICOS NO SISTEMA DE COMÉRCIO  
INTELIGENTE - ICS**

São Luís  
2004

**NATHÁLIA RIBEIRO SCHALCHER DE OLIVEIRA**

**FORMAÇÃO E CUMPRIMENTO DE CONTRATOS  
ELETRÔNICOS NO SISTEMA DE COMÉRCIO  
INTELIGENTE - ICS**

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Sofiane Labidi

São Luís  
2004

Oliveira, Nathália Ribeiro Schalcher. Formação e cumprimento de contratos eletrônicos no sistema de comércio inteligente – ICS / Nathália Ribeiro Schalcher de Oliveira. – São Luís, 2004.

105f.:il.

Dissertação (Mestrado em Engenharia de Eletricidade) – Universidade Federal do Maranhão, 2004.

1. Inteligência Artificial. 2. Web Semântica. I. Título. CDU 004.8

**NATHÁLIA RIBEIRO SCHALCHER DE OLIVEIRA**

**FORMAÇÃO E CUMPRIMENTO DE CONTRATOS  
ELETRÔNICOS NO SISTEMA DE COMÉRCIO  
INTELIGENTE - ICS**

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Dissertação aprovada em     /     /

**BANCA EXAMINADORA**

---

Prof. Dr. Sofiane Labidi (Orientador)

UFMA

---

Prof. Dr. Rubens Nascimento Melo

PUC-RIO

---

Prof. Dr. Edson Nascimento

UFMA

*À minha família.*

## **AGRADECIMENTOS**

Agradeço a todos que contribuíram direta ou indiretamente para elaboração desta dissertação, em especial:

A Deus, pela criação, pela família e pelas condições de estudo.

A meus pais, pelos estudos, confiança e amor.

A meu marido, Fernando, pelo incentivo, compreensão e apoio.

A meu irmão, Alex.

A meu cunhado, Alexandre, pelos incentivos e explicações.

A meus amigos, Carlos Alfredo, Samyr, Ozéas, João Carlos, Milson, Delano e outros tantos, pela ajuda a realizar este trabalho.

Aos meus sogros, pelo apoio e compreensão.

Ao meu orientador, pela paciência e tranquilidade.

## RESUMO

Este trabalho faz parte do Projeto ICS (Intelligent Commerce System), atualmente em desenvolvimento no Laboratório de Sistemas Inteligentes (LSI), da Universidade Federal do Maranhão (UFMA), sob a coordenação do Prof. Dr. Sofiane Labidi. Possui como objetivo, o desenvolvimento de um ambiente inteligente para lidar com as últimas duas fases do ciclo de vida do ICS: a Formação e o Cumprimento dos Contratos. O ambiente e a arquitetura do ICS são apresentados, inserindo todos os aspectos relativos ao seu desenvolvimento. Um sistema que possibilita automatizar tanto o mecanismo de contratação dos negócios fechados entre os usuários do ICS, quanto o monitoramento das obrigações por eles assumidas, através da utilização de agente inteligente, é implantado. Propomos o uso de padrões e ferramentas da Web Semântica para lidar com o gerenciamento das informações contidas nos contratos. Em relação ao monitoramento, propomos um modelo que faz uso, em conjunto, de Workflow Temporal e de regras ativas fundamentadas no paradigma ECAA de Sistemas de Banco de Dados Ativos.

**Palavras-chave:** Comércio Eletrônico, Contratos Eletrônicos, Agentes Inteligentes, Web Semântica, Workflow Temporal.

## ABSTRACT

This work is part of the ICS Project, which has been developed in LSI at UFMA under the coordinator Sofiane Labidi's assistance. The main objective is to develop an intelligent environment to deal with the last two phases of the ICS lifecycle: the Formation and Fulfillment of Contracts. The ICS architecture and environment are presented as well as its development aspects. An implemented system that uses intelligent agent has the possibility to automatize the mechanism of closed deal contract among ICS users, as well as the monitoring of its assumed obligations. We propose the use of patterns and Semantic Web tools to deal with the information management included in the contracts. Related to the monitoring we propose a model that makes use of the Temporal Workflow and of active rules based on the ECAA's paradigm of Active Database System.

**Keywords:** Electronic Commerce, Electronic Contract, Intelligent Agents, Semantic Web, Temporal Workflow.



## LISTA DE FIGURAS

Figura 1 - Arquitetura da Web Semântica .....	30
Figura 2 - Estrutura da linguagem XML .....	31
Figura 3 - Tipos de ontologia quanto ao nível de dependência .....	37
Figura 4 - Representação esquemática dos conceitos .....	43
Figura 5 - Ambiente do ICS .....	49
Figura 6 - Ciclo de vida do ICS .....	51
Figura 7 - Interface de manipulação dos agentes da plataforma SOMA_.....	52
Figura 8 - Editor de ontologias OiLEd .....	53
Figura 9 - Ambiente de execução da Modelagem do Usuário .....	54
Figura 10 - Exemplo de um Modelo de Usuário_.....	55
Figura 11 - Ontologia do Protocolo de Negociação .....	57
Figura 12 - Ontologia dos Contratos .....	66
Figura 13 - Nova versão do Protocolo de Negociação .....	67
Figura 14 - Diagrama de interação da fase de Formação de Contrato .....	70
Figura 15 - Interface gráfica da plataforma JADE .....	72
Figura 16 - Editor de ontologias Protégé associado ao beangenerator_.....	73
Figura 17 - Exemplo de workflow produzido pelo agente de contrato .....	78

## LISTA DE TABELAS

Tabela 1 - Estrutura de um cluster .....	42
Tabela 2 - Informações sobre obrigações do usuário .....	61

## LISTA DE SIGLAS

ACL – Agent Communication Language

B2B – Business to Business

B2C – Business to Consumer

B2G – Business to Government

C2C – Consumer to Consumer

CSS – Cascade Style Sheet

DAML+OIL – DARPA Agent markup Language + Ontology Interface Layer

DTD – Document Type Definition

ECAA – Event Condition Action Action

EDI – Electronic Data Interchange

FIPA – Foundation for Intelligent Physical Agents

G2C – Government to Consumer

G2G – Government to Government

GUI – Graphical User Interface

HTML – HyperText Markup Language

ICS – Intelligent Commerce System

JADE – Java Agent Development Framework

LGPL – Lesser General Public Licence

LSI – Laboratório de Sistemas Inteligentes

OIL – Ontology Interface Layer

OILEd - Ontology Interface Layer Editor

RDF – Resource Description Framework

SGML – Standard Generalized Markup Language

SOMA – Security and Open Mobile Agents

TI – Tecnologia da Informação

UFMA – Universidade Federal do Maranhão

URI – Uniform Resource Identifier

XSL – Extensible Stylesheet Language

XML – eXtended Markup Language

WfMC – Workflow Management Coalition

WFMS – Workflow Management System

WWW – World Wide Web

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	14
<b>1.1 Objetivo</b> .....	14
<b>1.2 Justificativa e Relevância</b> .....	14
<b>1.3 Estrutura da Dissertação</b> .....	16
<b>2. ESTADO DA ARTE</b> .....	18
<b>2.1 Contrato</b> .....	18
2.1.1 Noção de Contrato .....	18
2.1.2 Formação do Contrato .....	19
2.1.3 Requisitos de validade dos contratos .....	21
2.1.4 Momento em que se reputa celebrado o contrato .....	21
2.1.5 Lugar em que se reputa celebrado o contrato .....	23
2.1.6 Classificação dos Contratos .....	23
<b>2.2 Contrato Eletrônico</b> .....	24
2.2.1 Noção de Contrato Eletrônico .....	24
2.2.2 Formação do contrato eletrônico .....	25
2.2.3 Requisitos de validade .....	25
2.2.4 Momento em que se reputa celebrado o contrato eletrônico .....	26
2.2.5 Lugar em que se reputa celebrado o contrato eletrônico .....	26
2.2.6 Classificação dos contratos eletrônicos .....	27
<b>2.3 Web Semântica</b> .....	28
2.3.1 Contribuições da Web Semântica .....	29
2.3.2 Arquitetura da Web Semântica .....	30
2.3.2.1 Camada Esquema .....	31
2.3.2.1.1 XML .....	31

2.3.2.1.2 RDF .....	33
2.3.2.2 Camada Ontologia .....	34
2.3.2.3 Camada Lógica .....	35
2.3.3 Aplicações .....	35
<b>2.4 Ontologias</b> .....	<b>36</b>
2.4.1 Origem e Conceitos .....	36
2.4.2 Características Fundamentais .....	37
2.4.3 Classificação .....	38
2.4.4 Aplicações e usos .....	39
2.4.5 Comércio eletrônico mediado por agentes inteligentes .....	40
<b>2.5 Workflow</b> .....	<b>41</b>
2.5.1 Conceito .....	41
2.5.2 Vantagens .....	44
2.5.3 Tipos de sistemas de workflow .....	45
2.5.3.1 Workflow de Produção .....	46
2.5.3.2 Workflow Administrativo .....	46
2.5.3.3 Workflow Ad-hoc .....	46
2.5.4 Aplicação de Workflow em Comércio Eletrônico .....	47
<b>3 INTELLIGENT COMMERCE SYSTEM (ICS)</b> .....	<b>48</b>
<b>3.1 Introdução</b> .....	<b>48</b>
<b>3.2 Sistema ICS</b> .....	<b>48</b>
<b>3.3 Ambiente e Arquitetura</b> .....	<b>49</b>
<b>3.4 Ciclo de Vida</b> .....	<b>51</b>
3.4.1 Modelagem do Usuário .....	54
3.4.2 Matchmaking .....	57
3.4.3 Negociação .....	58

3.4.4	Formação do Contrato .....	60
3.4.5	Cumprimento do Contrato .....	60
<b>4</b>	<b>MODELAGEM DOS CONTRATOS ELETRÔNICOS .....</b>	<b>61</b>
<b>4.1</b>	<b><i>Função dos Contratos Eletrônicos no ICS .....</i></b>	<b>61</b>
<b>4.2</b>	<b>Abordagens de modelagem .....</b>	<b>61</b>
4.2.1	Abordagem 1 - Palavras da Linguagem Natural .....	62
4.2.2	Abordagem 2 - Palavras da Linguagem Formal .....	62
4.2.3	Abordagem 3 - Templates de Contratos .....	62
4.2.4	Abordagem 4 - Template de Cláusulas .....	63
<b>4.3</b>	<b><i>Modelagem proposta .....</i></b>	<b>63</b>
<b>4.4</b>	<b><i>Ontologia dos Contratos .....</i></b>	<b>65</b>
<b>4.5</b>	<b><i>Repositório de Contratos .....</i></b>	<b>65</b>
<b>5</b>	<b>FORMAÇÃO DO CONTRATO .....</b>	<b>69</b>
<b>5.1</b>	<b><i>Função .....</i></b>	<b>69</b>
<b>5.2</b>	<b><i>Agente de Contrato .....</i></b>	<b>69</b>
<b>5.3</b>	<b><i>Descrição do cenário de atuação do Agente de Contrato .....</i></b>	<b>70</b>
<b>5.4</b>	<b><i>Diagrama de interação da fase de Formação do Contrato .....</i></b>	<b>71</b>
<b>5.5</b>	<b><i>Implementação .....</i></b>	<b>71</b>
<b>6</b>	<b>CUMPRIMENTO DO CONTRATO .....</b>	<b>75</b>
<b>6.1</b>	<b><i>Objetivo .....</i></b>	<b>75</b>
<b>6.2</b>	<b><i>Modelo de Cumprimento .....</i></b>	<b>75</b>
<b>6.3</b>	<b><i>Exemplo .....</i></b>	<b>77</b>
<b>7</b>	<b>CONCLUSÃO .....</b>	<b>81</b>
	Referências .....	84
	Apêndices .....	86

# 1 INTRODUÇÃO

## 1.1 Objetivo

Este trabalho possui como objetivo contribuir na definição e implantação do ciclo de vida do sistema ICS de Comércio Eletrônico Inteligente, desenvolvido no Laboratório de Sistemas Inteligentes na Universidade Federal do Maranhão, focalizando nas fases de Formação e no Cumprimento do Contrato.

Especificamente, pretende-se:

- a) Desenvolver uma implementação, inicial, de um agente inteligente para lidar com as fases do ICS supracitadas.
- b) Modelar contratos que serão manipulados por este agente.
- c) Propor um modelo de monitoramento e gerenciamento das obrigações contratuais.

## 1.2 Justificativa e Relevância

O crescimento dos negócios entre empresas, através da Internet, mais conhecidos como B2B, vem se destacando no ramo do comércio eletrônico. No Brasil, por exemplo, o total registrado de abril a junho do ano de 2003 chegou a R\$ 34,8 bilhões – valor 2,3% superior ao trimestre anterior. A previsão é de que o volume de negócios ultrapasse 50 bilhões de dólares em 2005. Estimativas do Serasa, serviço especializado em informações financeiras e econômicas, apontam que esse mercado cresce a taxas superiores a 50% no país.



Neste ambiente estão inseridos os chamados e-marketplaces<sup>1</sup>, sistemas de comércio pela Internet que reúnem diversos compradores e fornecedores, formando um verdadeiro mercado eletrônico.

Entretanto, apesar de todo esse aumento, muitas empresas ainda não realizam negócios através da Internet, mesmo que esta prática implique em redução de custos, agilidade nos processos de compra e venda e outras vantagens. As incertezas surgem devido à falta de confiança ainda existente neste tipo de negócio.

A adoção de contratos em transações realizadas nos sistemas de comércio eletrônico consiste em um fator que eleva o nível de confiança na realização de negócios entre empresas. Porém, foi observado que a presença dos mesmos não implica, necessariamente, na alta taxa de transações atualmente registradas; a falta é que pode, significativamente, reduzir o volume de negócios realizados.

A confecção de instrumentos contratuais adequados e detalhistas, capazes de estabelecer os direitos e deveres das partes contratantes e fixar as penalidades pelo descumprimento dessas mesmas regras, deve ser o ponto de partida do estabelecimento de quaisquer relações negociais saudáveis e promissoras, independente do meio em que estas se encontram.

Entendemos então, que a adoção do uso de contratos em sistemas de comércio eletrônicos – denominados de contratos eletrônicos – faz-se relevante e necessária à maior participação de empresas nesse setor. Por este motivo, o ICS (*Intelligent Commerce System*) incluiu, em seu ciclo de vida, as fases de Formação e Cumprimento do Contrato, cujo desenvolvimento consiste no objeto deste trabalho.

---

<sup>1</sup> Electronic Marketplace ou Mercado Eletrônico.

É importante salientarmos que o nosso foco de discussão refere-se à questão da confiança em realizar negócios entre empresas, obtida através de contratos, utilizando o meio virtual. Não faz parte do contexto a segurança das informações utilizadas neste processo, que implica na adoção de métodos adequados, como: Criptografia, Assinaturas Digitais e Autoridades Certificadoras. Obviamente, esta questão pode e deve ser incluída, posteriormente, no sistema ICS, consistindo na realização de trabalhos futuros.

### **1.3 Estrutura da Dissertação**

Esta dissertação está estruturada em sete capítulos. O primeiro trata da introdução ao trabalho, apresentando o objetivo, a justificativa e relevância da pesquisa e a estrutura do trabalho.

O segundo capítulo apresenta o estado da arte, trazendo conceitos relativos aos contratos eletrônicos e às tecnologias e padrões utilizados.

O terceiro capítulo apresenta o sistema ICS, suas principais características e um resumo de cada fase do seu ciclo de vida.

O quarto capítulo aborda a modelagem dos contratos eletrônicos proposta e a justificativa para tal escolha; além da estrutura, das ferramentas e dos padrões utilizados neste processo.

O quinto capítulo apresenta e descreve o agente de contrato, trazendo detalhes como: função, o cenário de atuação e implementação. Também demonstra como ocorre a fase de Formação.

O sexto capítulo trata do modelo proposto para a fase de Cumprimento, incluindo um exemplo para fins de um melhor entendimento do mesmo.

No sétimo e último capítulo é apresentada a conclusão deste trabalho, com destaque para os trabalhos futuros. No final, encontra-se um anexo que contém o código do agente e um template de contrato.

## **2 ESTADO DA ARTE**

### **2.1 Contrato**

Esta seção trata de informações relativas ao importante documento jurídico, conhecido como contrato.

#### **2.1.1 Noção de Contrato**

O novo Código Civil, a exemplo do Código de 1916, deixou de definir o contrato para que a doutrina assim o fizesse. Para tal tarefa, o conceito mais puro de contrato foi definido como:

“o acordo de vontade entre duas ou mais pessoas com a finalidade de adquirir, resguardar, modificar ou extinguir direitos” (BEVILÁQUA, 1950).

Definem-se, também, os contratos como sendo uma das espécies dos negócios jurídicos, os chamados negócios bilaterais, ou seja, aqueles que se aperfeiçoam pela manifestação da vontade de duas ou mais partes, e que são destinados a produzir efeitos jurídicos (RODRIGUES, 2002).

Os contratos possuem uma função social: são o centro de vida dos negócios. Estes colocam, de forma harmoniosa, as posições e os interesses distintos de seus contratantes. Aperfeiçoam-se quando existe a concordância das vontades expressa de um lado pela oferta e, do outro, pela aceitação. Uma vez estabelecido, o contrato liga as partes por meio de um vínculo obrigacional. Estas, por sua vez, devem cumpri-lo, sob pena de responder ao seu inadimplemento.

É tão importante o papel desempenhado pelo contrato que seria irrealizável o desenvolvimento das relações sociais e econômicas sem o mesmo. Por exemplo, o comércio cresceu extraordinariamente, em grande parte, devido ao aperfeiçoamento do contrato.

### **2.1.2 Formação do Contrato**

A formação de um contrato depende dos seguintes elementos (RODRIGUES, 2002):

- a) O acordo de vontades e seu respectivo consenso;
- b) A idoneidade do objeto;
- c) A obediência à forma, quando prescrita em lei.

O contrato apresenta um fator novo em relação aos negócios jurídicos: o acordo das vontades, isto é, a concordância consensual existente entre as partes que o compõem. Este acordo, conforme já fora dito anteriormente, manifesta-se através da oferta e da aceitação.

A oferta, também conhecida como proposta ou policitação, é o passo inicial da formação do contrato. É constituída por uma declaração unilateral de vontade oriunda do proponente<sup>2</sup> e dirigida ao oblato<sup>3</sup>, cujo conteúdo apresenta questões referentes ao objeto do contrato. Possui caráter vinculatório, portanto, deve ser séria, clara e completa. Obriga o proponente a cumpri-la, nas seguintes situações, de acordo com o artigo 427 do Código Civil:

- a) Quando o contrário resulta de seus próprios termos;
- b) Da natureza do negócio;

---

<sup>1</sup> Pessoa que oferece uma proposta de contrato.

<sup>2</sup> Pessoa a quem é dirigida uma proposta de contrato.

c) Das circunstâncias do caso.

A aceitação também é uma declaração unilateral de vontade que apresenta, como conteúdo, a concordância com as questões da proposta. Possui consentimento tácito ou, na maioria das vezes, expresso, situação onde a lei exige uma forma determinada. Somente é considerada válida se for feita sem restrições à proposta e dentro do prazo estabelecido, quando então, o contrato torna-se celebrado.

Se for apresentada fora do prazo ou discordando da proposta, passa a ser considerada uma nova proposta, mais conhecida como contraproposta. Conseqüentemente, retira do proponente a obrigação de contratar.

Embora todo o processo de formação descrito acima venha a ocorrer de forma harmoniosa, existem duas situações onde um contrato deixa de ser aperfeiçoado, a saber:

a) Se a aceitação, embora expedida a tempo, por motivos imprevistos, chegar tarde ao conhecimento do proponente.<sup>4</sup>

Este já se encontrava liberado pelo atraso involuntário, de modo que o contrato não ganha vida, a menos que haja nova declaração da vontade. Todavia, nesse caso, impõe a lei ao proponente o dever de comunicar imediatamente o ocorrido ao oblato, sob pena de responder por perdas e danos.

b) Se antes da aceitação, ou com ela, chegar ao proponente a retratação do oblato.<sup>5</sup>

O contrato, nesse caso, não se completa porque não chegou a se formar, visto não ter ocorrido a coincidência de vontade das partes.

---

<sup>3</sup> Art. 430 do novo Código Civil.

<sup>4</sup> Art. 433 do novo Código Civil.

### **2.1.3 Requisitos de validade dos contratos**

Para que um contrato obtenha validade jurídica, ele deve apresentar os seguintes requisitos (GONÇALVES, 2002):

- a) Seu objeto deve ser lícito, possível, determinado ou determinável;
- b) Os seus agentes ou contratantes devem possuir capacidade jurídica e legitimação para o negócio em questão;
- c) Sua forma deve ser prescrita ou não defesa em lei.

Por capacidade jurídica, entende-se no sentido de ser titular de direito. Inclui-se, também, a de agir, que é a de poder ser parte em processo judicial, isto é, atuar judicialmente.

Quanto à forma, deve-se contratar conforme a lei determinar, ou ao menos não proibir. É o princípio da livre forma. A fim de exemplificação, pode-se citar duas situações:

- a) Não há impedimento legal para compra e venda, mediante contrato verbal, de uma bolsa em uma feira de artesanato. Ou seja, a lei não determina a forma contratual de compra e venda no comércio informal.
- b) Porém, o pacto antenupcial, para ser válido, deve ser lavrado em escritura pública, como determina a lei, de acordo com o artigo 1653 do novo Código civil.

### **2.1.4 Momento em que se reputa celebrado o contrato**

O contrato reputa-se celebrado no momento da aceitação, como se verificou. Quando as partes estão presentes, o conhecimento da aceitação por parte

do contratante é imediato. Entretanto, quando estas se encontram ausentes, existe um intervalo de tempo para que esta situação ocorra, fazendo-se necessário determinar qual o exato momento em que o vínculo contratual se forma.

Para dirimir esta questão, a doutrina elaborou várias teorias, das quais duas são as mais destacadas. Para a teoria da cognição ou informação, o contrato somente se aperfeiçoa no momento em que o proponente toma conhecimento da aceitação do oblato. Baseia sua justificativa no fato de que todo negócio jurídico somente se realiza quando o proponente e o oblato possuem conhecimento da vontade do outro. Entretanto, para a teoria da agnição ou declaração, o contrato se forma através da declaração de vontade do oblato. Divide-se em três subteorias:

- a) Subteoria da declaração: o momento da conclusão coincide com a formulação da aceitação por parte do oblato, em resposta à oferta.
- b) Subteoria da expedição: afirma que não basta somente a formulação da aceitação, sendo necessário e indispensável, também, a sua remessa ao proponente.
- c) Subteoria da recepção: a aceitação deve ser formulada, remetida e recebida pelo proponente. Diferencia-se da teoria da agnição porque esta se baseia no conhecimento do conteúdo da resposta, enquanto que aquela exige apenas a sua recepção, não importando se o proponente tomou conhecimento da mesma.

O novo Código Civil manteve a posição do Código de 1916 que decidiu por acolher a teoria da agnição com subteoria da expedição nos contratos entre ausentes.



### **2.1.5 Lugar em que se reputa o contrato**

O artigo 435 do novo Código Civil dispõe que o contrato reputa-se celebrado no lugar em que foi proposto. A determinação do lugar é importante para escolher o foro competente e as leis contratuais a serem aplicadas. Nas situações onde os contratantes residem em países diferentes, cuja matéria pertence ao direito internacional privado, dispõe-se do art. 9º, § 2º da Lei de Introdução ao Código Civil. Conforme este artigo, a obrigação resultante do contrato reputa-se constituída no lugar em que reside o proponente.

### **2.1.6 Classificação dos Contratos**

A classificação dos contratos não se dá em apenas uma modalidade, compreendendo uma vasta categorização. Esta se encontra fundamentada através de pontos de vistas distintos.

Define-se, por exemplo, a divisão dos contratos quanto à sua natureza em: unilaterais, bilaterais e plurilaterais; gratuitos e onerosos; comutativos e aleatórios e, causais e abstratos.

Quanto à maneira como se aperfeiçoam, pode-se distingui-los em: consensuais e reais, solenes e não solenes.

Levando-se em consideração o fato de a lei lhes atribuir, ou não, um nome e lhes sistematizar as regras, tem-se: nominados e inominados.

Considerando-se o modo pelo qual existem uns em relação os outros: principais e acessórios.

Tendo-se em vista o momento de sua execução: contratos de execução instantânea e contratos de execução diferida no futuro.

Quanto ao seu objeto: definitivo e preliminar.

E, finalmente, quanto à sua formação: paritários e de adesão.

## **2.2 Contrato Eletrônico**

Esta seção apresenta e descreve diversos conceitos relativos ao contrato eletrônico, aplicando-lhe a doutrina e a teoria das obrigações contratuais. Visa mostrar como se encontram definidos o valor jurídico, a eficácia e a capacidade, além de identificar as diferenças e semelhanças em relação ao contrato em geral.

### **2.2.1 Noção de Contrato Eletrônico**

A noção de contrato apresentada na seção anterior é totalmente aplicável ao contrato eletrônico. Defende-se que este é uma espécie daquele, diferenciando-se apenas nos seus processos de formação e execução (CUNHA JÚNIOR, 2002). Portanto, também é categorizado como negócio jurídico bilateral, pois se aperfeiçoa através da manifestação da vontade de duas ou mais pessoas; possui a finalidade de adquirir, resguardar, modificar ou extinguir direitos (BEVILÁQUA, 1950) e é destinado a produzir efeitos jurídicos (RODRIGUES, 2002).

Os processos de formação e execução do contrato eletrônico divergem do tradicional porque são implementados através de via eletrônica (CUNHA JÚNIOR, 2002). Conseqüentemente, originam-se os contratos eletrônicos formados, os executados e os formados e executados eletronicamente.

Os primeiros são aqueles cuja manifestação de vontades dos participantes – a oferta e a aceitação – se dá por meio de transmissão eletrônica de dados. Isto é, impõe-se o emprego de recursos tecnológicos na sua formação. Como melhor e mais típico exemplo, tem-se a compra e venda de bens, duráveis ou não, através de páginas eletrônicas na Internet.

Os contratos apenas executados eletronicamente, ao contrário, manifestam-se de forma tradicional e são executados de forma eletrônica. Como exemplo, têm-se os contratos de manutenção de banco de dados e de hospedagem de software, desde que oriundos de uma celebração tradicional.

Finalmente, os formados e executados eletronicamente manifestam-se e são executados utilizando a tecnologia. Os exemplos típicos são: prestação de serviços de informação, que pode ser realizada através de e-mail, pager, celular; contrato de licença de uso de software adquirido por download.

### **2.2.2 Formação do contrato eletrônico**

A formação dos contratos eletrônicos depende dos mesmos elementos aplicados a todos os contratos: o acordo de vontades e seu respectivo consenso, a idoneidade do objeto e a obediência à forma, quando prescrita em lei. Conseqüentemente, as mesmas regras valem para a oferta e a aceitação.

### **2.2.3 Requisitos de validade**

Os requisitos de validade merecem uma análise sobre sua aplicabilidade. As partes contratantes também devem possuir capacidade jurídica e legitimação

para os negócios que são firmados através dos meios eletrônicos, assim como o objeto deve ser lícito, possível, determinável ou determinado, não havendo motivos para contradições. A questão mais importante quanto à validade recai sobre a forma de se expressar um contrato. A forma eletrônica de contratação não está prescrita nem proibida na lei, de acordo com o artigo 129 do Código de Processo Civil, onde prevalece o princípio da ausência de solenidade na celebração dos contratos em geral. Portanto, os contratos eletrônicos possuem validade jurídica alicerçada por este artigo, podendo ser utilizados. Assim, cabe tanto à doutrina quanto à legislação adequá-los às leis existentes, dentro do possível.

#### **2.2.4 Momento em que se reputa celebrado o contrato eletrônico**

O momento da celebração de um contrato eletrônico pode ocorrer de duas maneiras: quando o meio eletrônico utilizado para a formação do mesmo, possuir transmissão instantânea, os contratos devem ser considerados entre presentes. Isto porque a aceitação será imediatamente conhecida pelo proponente, aperfeiçoando o contrato. Contudo, quando a transmissão for prorrogada no tempo, será considerado entre ausentes por haver um intervalo entre a aceitação e o seu conhecimento por parte do proponente. Desta forma, o momento da celebração reger-se-á de acordo com a teoria da agnição com subteoria da expedição.

#### **2.2.5 Lugar em que se reputa o contrato eletrônico**

As leis aplicadas para determinação do lugar em que se reputa um contrato são as mesmas para o eletrônico. Portanto, os contratos eletrônicos são

formados no local onde possui domicílio o proponente, ou seja, no lugar em que foi proposto. O fato destes, em sua maioria, dirigirem-se à empresas ou pessoas indeterminadas, independentemente do país onde se encontram, são irrelevantes para a definição da lei que regerá as obrigações contratuais a serem aplicadas. Esta matéria, conforme afirmado anteriormente, pertence ao direito internacional privado, cujo artigo 9º, § 2º da Lei de Introdução ao Código Civil deve ser aplicado.

### **2.2.6 Classificação dos contratos eletrônicos**

Os contratos eletrônicos podem ser classificados de acordo com as modalidades já mencionadas anteriormente. A classificação pode ser estendida, quando se leva em consideração o aspecto eletrônico em geral, da seguinte maneira (CUNHA JÚNIOR, 2002):

- a) Quanto à natureza jurídica da relação tutelada ou à qualidade das partes envolvidas, estão presentes os contratos:
  - a.1) B2B (*Business to Business*);
  - a.2) B2C (*Business to Consumer*);
  - a.3) C2C (*Consumer to Consumer*);
  - a.4) G2C (*Government to Consumer*);
  - a.5) B2G (*Business to Government*);
  - a.6) G2G (*Government to Government*);
- b) Quanto ao grau de interação homem/máquina, os contratos podem ser interpessoais, interativos ou intersistêmicos.
- c) Quanto à simultaneidade da proposta/aceitação, diferenciam-se em os simultâneos e os não simultâneos.

d) Quanto à subforma eletrônica, que consiste na maneira pela qual a declaração de vontade manifestada eletronicamente pode se expressar, existem as salas de bate-papo, protocolo EDI, videoconferência, e-mail, mensagens interativas, sites, entre outras.

## **2.3 Web Semântica**

Nos anos 90, Tim Berners-Lee trazia ao mundo o conceito da WWW (*World Wide Web*), um potencial sistema de hipertexto baseado na Internet, objetivando compartilhar informação global. Através da linguagem HTML (*HyperText Markup Language*), um grande volume de páginas puderam ser construídas e publicadas na Internet, tornando a Web um enorme repositório de conteúdo.

Entretanto, esse enorme repositório acabou implicando em uma dificuldade no acesso da informação, tornando a busca de dados bastante imprecisa. Outro fator conseqüente do uso de HTML foi criar a maior parte do conteúdo da Internet voltado para a leitura e entendimento das pessoas, e não para que os softwares pudessem manipular significativamente as informações. Estes e outros problemas têm levado a um esforço no sentido de se prover semântica às informações contidas na Web.

Assim, Tim Berners-Lee idealizou outro conceito: a Web Semântica. Este afirma que a Web Semântica é uma nova forma de conteúdo da Web, sendo significativa para os computadores que irão desencadear uma revolução de novas possibilidades (BERNERS-LEE, 2001). Esta não constitui uma Web dissociada, porém uma extensão da existente, na qual é fornecido um sentido mais bem definido

à informação. Isto traz uma possibilidade maior de pessoas e computadores trabalharem juntos.

Um dos desafios da Web Semântica é a evolução da rede atual, composta de documentos, para uma rede de dados na qual toda a informação possui um significado bem definido. Outro consiste na criação de uma linguagem que permita expressar, ao mesmo tempo, o significado dos dados e definir regras para raciocinar sobre os mesmos, de forma a deduzir novos dados e regras e permitir que aquelas existentes em sistemas de conhecimento possam ser exportadas para a Web.

Para desenvolver efetivamente o uso da Web Semântica são necessárias a construção de ontologias de domínio e a adoção de duas novas tecnologias: XML (*eXtended Markup Language*) e RDF (*Resource Description Framework*). Estes conceitos serão posteriormente mais detalhados.

### **2.3.1 Contribuições da Web Semântica**

Uma das tecnologias que já está adquirindo vantagens com a disponibilidade do uso da Web Semântica são os agentes inteligentes. Resumidamente, podemos afirmar que os agentes são programas carregados de informações, que possuem um certo grau de autonomia para utilizá-la no seu funcionamento e que são capazes de realizar tarefas que auxiliem o usuário no desempenho de suas atividades, de acordo com seus interesses (HENDLER, 2001).

Os agentes podem capturar informações para sua base de conhecimentos nos conteúdos de várias fontes na Web e, então, processá-las e fazer intercâmbio desses resultados com outros programas. Através da estruturação

do conteúdo semântico das informações encontradas nas páginas da Internet, possibilita-se a criação de um ambiente onde os agentes de software estão aptos a executar tais tarefas.

Como exemplo de tarefas, podemos citar a procura e a recuperação da informação que serão beneficiadas com mecanismos de busca mais acurados, pois a heterogeneidade da informação dificulta a integração do conteúdo na Web.

Uma área a ser beneficiada será a Representação de Conhecimento. Os sistemas atuais para representação do conhecimento são centralizados, exigindo que todos compartilhem as mesmas definições de conceitos. Para o funcionamento da Web Semântica, os computadores devem possuir acesso a coleções estruturadas de informação e a conjuntos de regras de inferência usados para conduzir o raciocínio automatizado. Desta forma, os sistemas de representação de conhecimento existentes serão beneficiados se suas regras forem exportadas para a Internet.

### **2.3.2 Arquitetura da Web Semântica**

Na proposta de desenvolvimento da Web Semântica (BERNERS-LEE, 2001) foi sugerida uma arquitetura, cuja formação consiste em três camadas, como mostra a Figura 1:



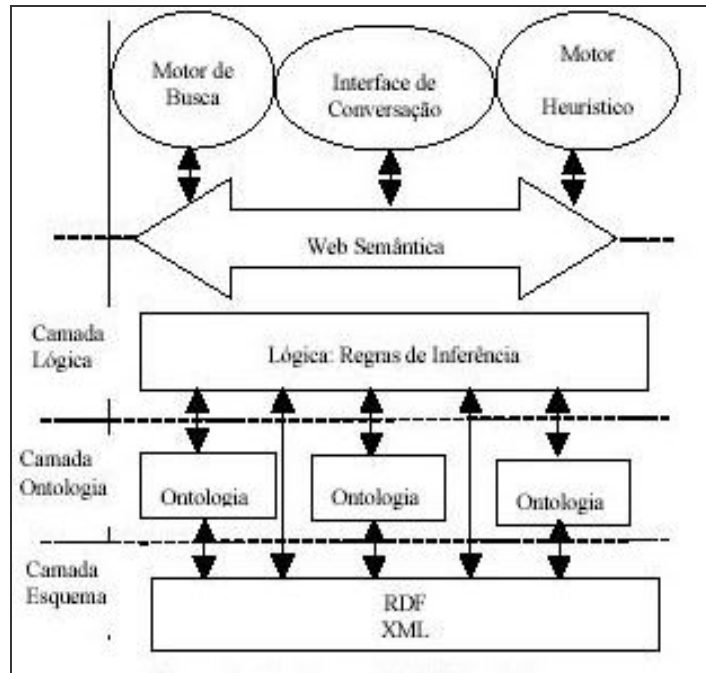


Figura 1: Arquitetura da Web Semântica

### 2.3.2.1 Camada Esquema

A camada base da arquitetura possui a função de estruturar e prover significado aos dados, a fim de possibilitar o raciocínio lógico. É composta pelas tecnologias XML (BRAY, 2000) e RDF (LASSILA, 1999).

#### 2.3.2.1.1 XML

XML é uma linguagem de marcação de dados que provê um formato para descrição de dados estruturados. Isto faz com que as declarações do conteúdo sejam mais precisas, implicando em resultados de buscas mais significativos, realizados através de múltiplas plataformas. Consiste em uma versão do padrão SGML (Standard Generalized Markup Language) voltada para a Web.

XML permite a definição de um número infinito de tags, campos de texto escondidos nas páginas. Porém, com um propósito diferente das tags do HTML. XML usa as tags somente para delimitar trechos de dados e para fornecer a estes uma representação.

A estrutura do XML é composta por três elementos principais (Faria, 2002) conforme a Figura 2:

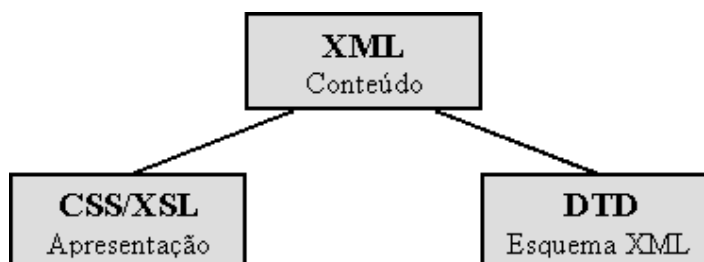


Figura 2: Estrutura da linguagem XML

O DTD (*Document Type Definition*) funciona como uma gramática em forma de um documento do tipo texto, que contém todas as regras para verificação da correção de um documento XML. Também define todos os elementos (tags) que um documento pode conter, determinando sua ordenação e seus atributos, como exemplo, obrigatoriedade ou facultatividade.

O XSL (*Extensible Stylesheet Language*) e o CSS (*Cascade Style Sheet*) são padrões de tecnologia definidos pelo W3C<sup>5</sup> (*World Wide Web Consortium*) que compõem o XML e servem para apresentar os dados. XSL, através da linguagem de transformação, é usado para transformar documentos XML em algo agradável para ser visto, ou seja, em outros formatos que podem ser bem diferentes das estruturas de dados originais do documento XML, assim como para documentos HTML. O CSS pode ser usado para XML simplesmente estruturado, mas não pode apresentar informações em uma ordem diferente de como foi recebida.

---

<sup>5</sup> <http://www.w3.org>.

### 2.3.2.1.2 RDF

O RDF (*Resource Description Framework*) é uma recomendação da W3C (*World Wide Web Consortium*) para definição e uso de metadados na Web. Metadados são dados que possuem informações sobre determinado conjunto de dados. Seu propósito consiste em prover significado às estruturas de dados definidas pelo XML. É formado por:

- a) Modelo de dados do RDF - o modelo do RDF consiste de três objetos: recurso, propriedades e sentenças. Recurso é todo e qualquer objeto unicamente identificado por um URI (Uniform Resource Identifier), um identificador de itens da Web. Propriedade de um recurso descreve características próprias deste e, por fim, uma sentença liga um recurso à sua propriedade adicionando um valor para esta propriedade desse recurso.
- b) Sintaxe RDF - o modelo de dados do RDF consegue representar plenamente os recursos da Web; porém faz-se necessário uma sintaxe representando este modelo para armazenar instâncias do mesmo em arquivos que sejam lidos por máquinas e, também, para comunicá-las entre aplicações. A sintaxe utilizada neste processo é justamente o XML.
- c) Esquema RDF - a questão da semântica do RDF é de responsabilidade do RDF Schema (BRICKLEY, 2003), que define um vocabulário para o modelo de dados em questão. Com o vocabulário, os recursos, as propriedades e os valores possuirão significados

específicos para um domínio particular, além de relacionamentos e combinações entre estes.

### *2.3.2.2 Camada Ontologia*

Após serem estruturados e possuírem significados, os dados geralmente recebem conceituações diferentes dependentes da aplicação que os está utilizando. Para resolver este problema de indefinição ou terminologia, são aplicadas as ontologias.

Na Web Semântica, a ontologia é utilizada no contexto de compartilhamento do conhecimento e tem como objetivo a especificação explícita e formal de uma conceitualização. Uma ontologia requer o uso de um vocabulário específico para descrever os requisitos de um determinado domínio e também um conjunto de axiomas lógicos, necessários para imprimir semântica ao significado pretendido pelas palavras do vocabulário. Assim, a ontologia pode gerar um ambiente com informações documentadas, confiáveis, de fácil manutenção e reutilização, facilitando o trabalho dos agentes de software e viabilizando também o intercâmbio de informações entre eles.

Possibilitam, ainda, um mecanismo de pesquisa mais apurado e restrito às informações realmente relevantes, automação de tarefas que exijam raciocínio, e permitem que os agentes atuem como guias, sugerindo opções e caminhos e auxiliando o usuário no alcance de seus objetivos.

Uma ontologia na Web Semântica possui uma taxonomia e um conjunto de regras de inferência. A taxonomia, que é estabelecida nesta camada, define as classes de objetos e as relações que se estabelecem entre elas. Desta maneira, é

formada uma estrutura onde um conjunto de propriedades é atribuído a determinadas classes e, os objetos que pertencem as estas classes herdam suas características.

### *2.3.2.3 Camada Lógica*

Composta por regras de inferência produzidas pelas estruturas de dados, definidas na camada esquema, e das relações entre estas, definidas na camada ontologia. Estas regras fornecem aos agentes de software o poder de raciocínio sobre os termos e os seus significados.

### **2.3.3 Aplicações**

A Web Semântica propõe, para seu funcionamento, a utilização de várias tecnologias, conforme explanado, como XML, RDF, ontologia, além de outras não citadas; porém em utilização: OIL, DAML+OIL, etc. Toda essa estruturação vem a trazer muitos benefícios quando aplicada a certas áreas da Informática.

Uma área que, constantemente, recebe vantagens é o Comércio Eletrônico, principalmente o mediado por agentes inteligentes. A estruturação e o significado da informação que é provida pelo emprego da Web Semântica permitem aos agentes melhorarem as atividades que costumam realizar, como: entendimento sobre os dados que formam suas bases de conhecimento, raciocínio executado, comunicação, busca, dentre outros.

## 2.4 Ontologias

### 2.4.1 Origem e conceito

A palavra Ontologia possui origem na Filosofia. É vista como um sinônimo de Metafísica e significa a Ciência que estuda a essência do ser. Entretanto, a partir dos anos noventa, o conceito e a utilização de Ontologia vêm sendo aplicados, em um contexto diferente, à Ciência da Computação, mais especificamente na área de Inteligência Artificial.

Atualmente, não existe um consenso sobre a definição do termo Ontologia na comunidade da Ciência da Computação, que passou a ser utilizado como ontologia, com letra minúscula, para ser diferenciado do termo Ontologia proveniente da Filosofia.

Algumas das definições mais citadas são:

- a) “A ontologia é uma especificação explícita de uma conceitualização” (GRUBER, 1995). Neste trabalho, a noção de conceitualização compreende os objetos, conceitos e outras entidades que são assumidos existir em alguma área de interesse, assim como os relacionamentos entre os mesmos. Em outras palavras, uma conceitualização é uma visão simplificada e abstrata do mundo que se deseja representar com algum propósito específico.
- b) Contudo, outro autor defende e justifica a diferença entre os conceitos de ontologia e conceitualização, afirmando que: “a ontologia é uma teoria lógica para relacionar o significado pretendido de um vocabulário formal, isto é, seu comprometimento com uma conceitualização

particular do mundo”. Em outras palavras, ontologia é uma especificação parcial explícita que tenta, da melhor forma possível, aproximar a estrutura de mundo definida por uma conceitualização (GUARINO, 1997).

Inicialmente, ontologias foram empregadas com o objetivo de compartilhar conhecimento entre pessoas e sistemas de aplicação, além de promover a sua reutilização. Para tal situação, existe outra definição mais apropriada, onde o termo ontologia é definido como uma coleção de informações com uma especificação formal e consensual de conceitos, definindo as relações entre estes e que providencia um compartilhamento e um entendimento comum de um domínio que pode ser comunicado entre pessoas e sistemas de aplicação (FARIA, 2002).

#### **2.4.2 Características Fundamentais**

Ao se construir uma ontologia de um determinado domínio, deve-se procurar atender a algumas características fundamentais:

- a) Aberta e dinâmica: a ontologia deve possuir habilidade de adaptar-se às mudanças que possam ocorrer no domínio.
- b) Escalável e interoperável: deve possuir capacidade de ser facilmente estendida ou comprimida, além de possibilitar a integração com outras ontologias.
- c) De fácil manutenção: deve possuir capacidade de ser compreendida por especialistas que não participaram da sua construção.
- d) Semanticamente consistente: deve possuir conceitos e relacionamentos coerentes, não ambíguos.

e) Independente de contexto: deve evitar a utilização de termos muito específicos, pois isto dificulta a integração com outras ontologias.

### 2.4.3 Classificação

As ontologias podem ser classificadas em duas dimensões: ao nível de detalhe e ao nível de dependência (GUARINO, 1997).

Na dimensão de detalhe, encontram-se as ontologias mais detalhadas e as mais simples. As primeiras possuem uma maior aproximação do vocabulário do domínio em questão. A desvantagem desta abordagem refere-se à dificuldade de integração às outras ontologias, justamente por esta proximidade. Contudo, as ontologias mais simples são desenvolvidas com o propósito de compartilhamento e reuso por diversos usuários que concordam com a conceitualização proposta.

Na segunda dimensão, pode-se distinguir entre quatro tipos de ontologias, mostradas na Figura abaixo:

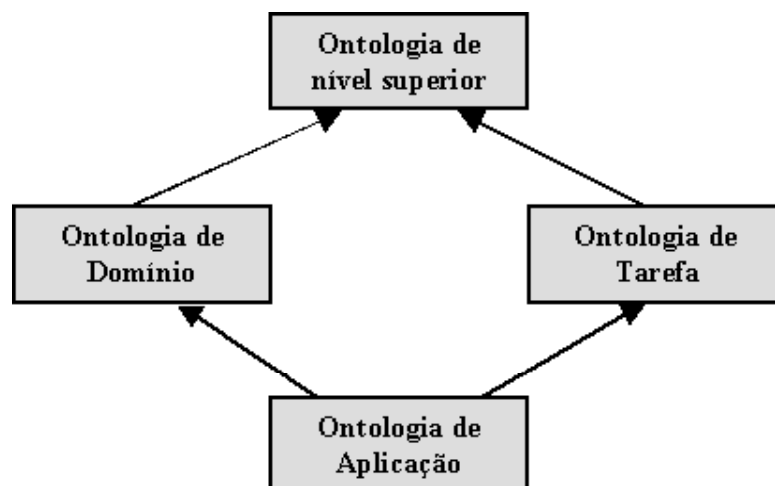


Figura 3: Tipos de ontologia quanto ao nível de dependência



Uma ontologia de nível superior descreve conceitos bastante gerais, como espaço, tempo, matéria, objeto, evento, ação, entre outros, sendo todos independentes de domínio ou problema particular.

Uma ontologia de domínio descreve um vocabulário para um domínio genérico, por exemplo, bebidas ou automóveis, procurando especializar os termos.

Uma ontologia de tarefa também descreve um vocabulário, porém para um domínio de uma tarefa ou atividade genérica, tais como uma compra ou uma venda.

E, finalmente, uma ontologia de aplicação é uma especialização das ontologias de domínio e tarefa, correspondendo a papéis desempenhados por entidades de um domínio enquanto realizam certas atividades.

#### **2.4.4 Aplicações e Usos**

Primeiramente, a Inteligência Artificial empregou ontologias para facilitar o compartilhamento e o reuso de conhecimento. Entretanto, ontologia vêm se tornando um campo de grande interesse em várias áreas de pesquisa, como:

- Engenharia de Conhecimento;
- Comércio Eletrônico;
- Gerenciamento de Conhecimento;
- Integração Inteligente de Informação;
- Banco de Dados;
- Recuperação da Informação;
- Sistemas Multi-Agentes;
- Web Semântica.

Possíveis usos de ontologias dentro das áreas supracitadas encontram-se abaixo:

- a) Funcionar como repositórios para organização de conhecimentos e informação tanto corporativos como científicos;
- b) Como ferramenta para aquisição de informação em situações onde esta funciona como suporte comum para a organização do domínio;
- c) Como ferramenta de referência na construção de sistemas baseados em conhecimento;
- d) Permitir a reutilização de conhecimento existente em novos sistemas;
- e) Servir como base para a construção de linguagens de representação do conhecimento.

Destacamos neste trabalho, o emprego de ontologias na área de Comércio Eletrônico mediado por agentes inteligentes, buscando construir um vocabulário particular de um domínio específico.

#### **2.4.5 Comércio Eletrônico mediado por agentes inteligentes**

O Comércio Eletrônico já é reconhecido como uma área de grande interesse e, conseqüentemente, de grande crescimento, que sempre procura proporcionar alguma evolução em seus métodos de implementação. Atualmente, existe uma tendência de empregar agentes inteligentes para mediar o processo de comercialização na Internet, principalmente na categoria B2B (JENNINGS, 1998).

Nesses sistemas, os agentes geralmente exercem os papéis de comprador e vendedor, contratante e contratado, dentre outros. Apesar disso, alguns trabalhos existentes utilizam-nos para outros fins, como, por exemplo, intermediários

na comercialização (FONSECA, 2003). Para poder realizar os atos de compra e venda, os agentes devem se comunicar e, para isto, possuir um entendimento comum sobre suas interações. Neste ponto, a adoção de ontologias se torna interessante, pois provê padrões de estruturas e terminologias de informações a fim de descrever itens como as preferências dos clientes, características dos produtos, busca de ofertas/ procuras afins, processos de negócios (negociação, contratação), dentre outros. Isto implica no aumento significativo do grau de automatização na troca de dados e no surgimento de novos modelos de negócios para desenvolvimento e aplicação em mercados eletrônicos.

## **2.5 Workflow**

### **2.5.1 Conceito**

Um grupo de empresas que se uniram com o propósito de estabelecer padrões e conceitos para workflow chamado WfMC (*Workflow Management Coalition*), estabeleceu um glossário (WfMC, 1999) onde se define o termo workflow como:

“A automação de um processo de negócio, em todo ou em parte, durante o qual documentos, informações ou tarefas são passadas entre participantes gerando ações, de acordo com um conjunto de regras de procedimentos”.

Este conceito requer o conhecimento de uma série de outros termos relacionados para alcançar um melhor entendimento, dentre eles:

- a) Processo de Negócio** - conjunto de procedimentos ou atividades relacionadas que, coletivamente, realizam um objetivo de negócios ou

uma meta de políticas, normalmente dentro do contexto de uma estrutura organizacional, definindo papéis e relacionamentos funcionais.

Um processo de negócios pode envolver interações formais ou relativamente informais entre seus participantes, variando amplamente na duração.

Pode consistir de atividades automatizadas, capazes de serem gerenciadas por workflow, e/ou manuais, as quais estão fora de seu escopo.

**b) Definição de Processo** - é a representação de um processo de negócio de maneira tal que suporte um tipo de manipulação automatizada, como uma modelagem, ou um controle através de um sistema de gerenciamento de workflow. A definição de processo consiste em uma rede de atividades e seus relacionamentos, de critérios indicativos do início e do fim de um processo e, informações sobre atividades individuais, tais como: participantes, aplicações de Tecnologia da Informação associadas, dados, etc.

**c) Atividade ou tarefa** - um passo lógico dentro do processo que descreve um fragmento de trabalho. Uma atividade pode ser manual ou automatizada, onde nesta última existe a presença de recursos humanos e/ou computacionais para ser executada. Quando um recurso humano é necessário, uma atividade automatizada é alocada para um agente de workflow.

**d) Instância de Processo** - a representação de um processo particular ou de uma atividade dentro de um processo, incluindo seus dados associados. A diferença entre uma instância de um processo ou de

uma tarefa e seus conceitos relativos está no fato de o primeiro corresponder a um caso específico e, o segundo, a uma idéia abstrata.

- e) **Agente** - um recurso que desenvolve o trabalho representado por uma instância de tarefa automatizada. Geralmente, um agente representa um recurso humano do processo de negócio; porém recursos de software como agentes inteligentes podem ser utilizados.
- f) **Item de trabalho** - a representação do trabalho a ser desenvolvido pelo agente no contexto de uma atividade dentro de uma instância de processo. Ou seja, a atividade gera um conjunto de itens de trabalho a serem alocados aos agentes.
- g) **Lista de trabalho** - a forma de organização dos itens de trabalho quando são apresentados aos agentes.
- h) **WFMS** - um WFMS ou um Sistema Gerenciador de Workflow é uma espécie de sistema que define, cria e gerencia a execução de workflows, através do uso de software, centralizado ou distribuído em uma ou mais máquinas de workflow (workflow engines), que está apto a interpretar a definição do processo, a interagir com os agentes, além de invocar o uso de aplicações e ferramentas de tecnologias de informação (TI).

Consiste em um software responsável por todo suporte e gerenciamento à automatização de um processo de negócios utilizando workflow. A Figura 4 mostra o relacionamento entre alguns dos conceitos vistos:

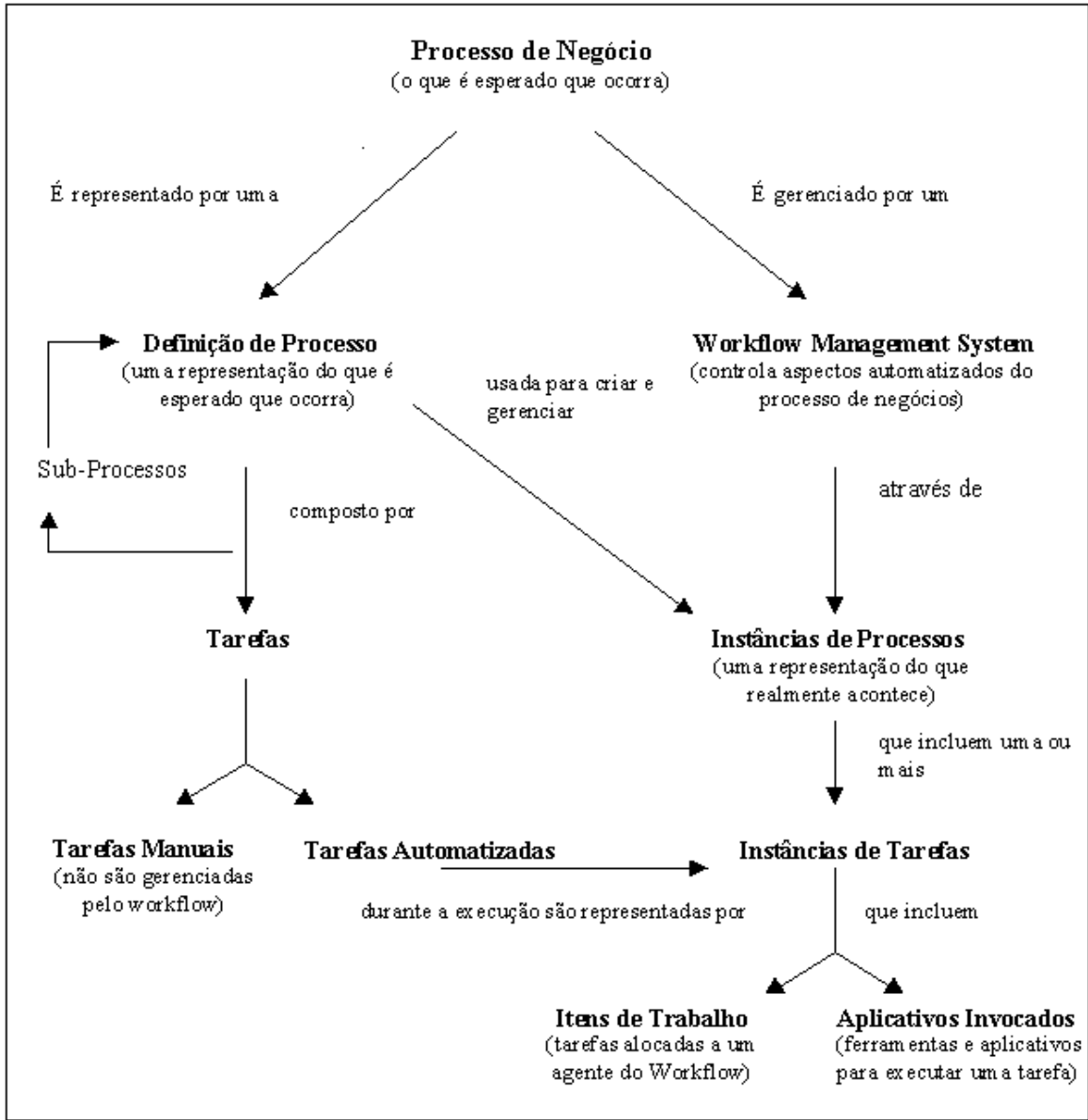


Figura 4: Representação esquemática dos conceitos

## 2.5.2 Vantagens

A utilização de workflow para automatizar qualquer tipo de processo de negócio pode trazer as muitas vantagens, entre elas:

- Garantia de integridade do processo – uma vez definido o processo, o sistema de workflow, WFMS, garante que as regras e a seqüência das

tarefas serão seguidas. Processos compostos por muitas regras são os mais beneficiados.

- Diminuição de custos – a automatização de processos leva a diminuição de custos operacionais, como papéis, além de reduzir a interação humana necessária.
- Integração das atividades da empresa – as atividades que são desenvolvidas por vários segmentos de uma empresa podem ser representadas através do workflow como um único processo.
- Eliminação do tempo de espera entre as atividades – o workflow define todos os passos seqüenciais de um processo e seus respectivos agentes, com isto, não existe um intervalo de tempo entre o término de uma atividade e o início da próxima.
- Manipulação eletrônica de documentos – registros eletrônicos substituindo documentos garantem consistência, utilização de regras e pesquisa rápida.
- Monitoramento da execução do processo – o workflow permite monitorar um determinado processo através do acompanhamento da realização de suas atividades.

### **2.5.3 Tipos de Sistemas de Workflow**

Cada organização possui processos com características próprias que requerem um tipo adequado de workflow para sua automação. Por esta razão, surgiram categorias de sistemas de workflow, onde se pode citar as principais, que são:

### *2.5.3.1 Workflow de Produção*

Seu objetivo é a automação de processos de negócios que incluem tarefas similares a fim de otimizar a produtividade. Estes processos, geralmente são complexos; pois envolvem acesso a múltiplos sistemas de informação, um alto volume de dados, regras de processamento rígidas e restrições de tempo. A ordenação e a coordenação de tarefas podem ser automatizadas. Exemplo: procedimentos de empréstimo.

### *2.5.3.2 Workflow Administrativo*

Automação de processos de negócios repetitivos; porém pouco complexos, objetivando flexibilidade. Geralmente, envolve a utilização de formulários. A ordenação e a coordenação de tarefas também podem ser automatizadas. Exemplo: sistemas que gerenciam roteamento de formulários e relatórios.

### *2.5.3.3 Workflow Ad-hoc*

Envolve coordenação humana, elaboração ou co-decisão. A ordenação e a coordenação de tarefas não são automatizadas, porém controladas por humanos. Exemplos: preparação de uma proposta de negócios, criação de documentos técnicos.



#### **2.5.4 Aplicação de Workflow em Comércio Eletrônico**

Nesta seção, destaca-se a importância e a necessidade de aplicar a tecnologia de workflow na área de comércio eletrônico, particularmente à categoria B2B, visando justificar a adoção da mesma, na fase de Cumprimento dos Contratos.

O Comércio Eletrônico envolve o desenvolvimento e o emprego de processos de negócios, como as atividades de pagamento e de entrega de produtos e serviços, para os quais o workflow torna-se um suporte de tecnologia importante para sua automação. Alguns trabalhos já utilizam workflow no gerenciamento dos processos de negócios que ocorrem na Internet, onde destacamos:

- a) Projeto FlowJet da Hewlett-Packard que objetiva mesclar vários tipos de sistemas de workflow dentro do contexto dos negócios eletrônicos (SHAN, 1999).
- b) Projeto Wise usa tecnologia de gerenciamento de workflows que percorrem, através das organizações, em cenários de comércio eletrônico B2B (ALONSO, 1999).

## **3 INTELLIGENT COMMERCE SYSTEM – ICS**

### **3.1 Introdução**

Neste capítulo, apresentamos uma visão do sistema ICS, fornecendo sua definição e demonstrando seus ambiente e arquitetura. Um resumo das principais características e do funcionamento de cada fase do ciclo de vida é mostrado, deixando para as próximas seções, todo o trabalho realizado nas fases de Formação e Cumprimento do Contrato.

### **3.2 Sistema ICS**

O sistema ICS consiste em uma implementação de comércio eletrônico entre empresas - mais conhecido como B2B - que faz uso de agentes inteligentes móveis e estacionários para sua automação. O desenvolvimento e o funcionamento de seu processo de comércio eletrônico acontece seguindo um ciclo de vida composto por cinco fases.

Os agentes móveis do ICS são responsáveis por representar os usuários deste sistema: empresas que desejam realizar negócios através da Internet, enquanto os estacionários realizam funções internas que permitem implementar seus procedimentos. Uma de suas características interessantes é a utilização de ontologias com a finalidade de prover representação e compartilhamento de conhecimento aos agentes.

A idéia de automatizar o comércio eletrônico através do emprego de agentes inteligentes não é novidade (JENNINGS, 1998). Consiste em colocar sob o comando de agentes algumas tomadas de decisões comerciais.

No ICS, os agentes comunicam-se constantemente e carregam consigo uma grande quantidade de informações para dar suporte ao desenvolvimento de suas funções. Esta característica incentivou o emprego de ontologias, pois proporciona um entendimento comum aos dados através da atribuição de significados aos mesmos e da criação de relacionamentos entre eles. Desta forma, procurou-se fornecer um bom suporte à comunicação entre os agentes e também, um bom entendimento das informações relativas ao processo de comércio por parte destes.

### **3.3 Ambiente e Arquitetura**

O sistema ICS foi concebido para prover um ambiente onde os agentes possam desenvolver suas funções, deslocarem-se (situação dos que são móveis) e se comunicar. Este ambiente é aberto como a Internet, pois deve permitir que os agentes móveis possam entrar e sair dele.

O ICS está organizado em uma arquitetura, mostrada na Figura 5, formada por oito principais componentes: marketplace, região, agente matchmaker, agente mediador, agente de contrato, agente negociadores (comumente, vendedores e compradores), repositório de ontologia e de contratos.

Um **marketplace** é o local onde os agentes realizam suas funções. Nele ocorrem todos os procedimentos internos do ICS que fornecem suporte ao processo de comércio eletrônico por ele provido. Isto é, detalhes sobre protocolos de

comunicação, de negociação, ontologia, identificação de agentes negociadores em vendedores e compradores, armazenamento de anúncios em repositórios, formação de contratos, entre outros. Todos são resolvidos dentro de um marketplace.

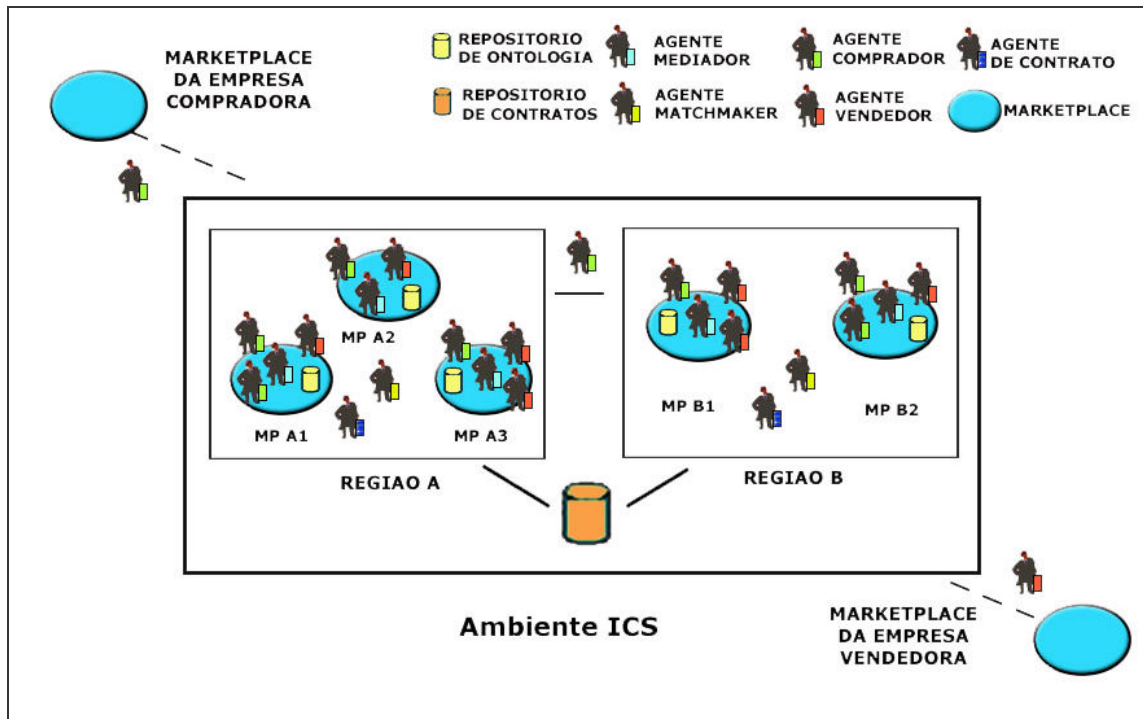


Figura 5: Ambiente do ICS

Uma **região** é o agrupamento de marketplaces que provê um alto nível de abstração para a comunicação entre agentes. Os marketplaces de uma região são reunidos pela mesma área de negócios, modelada através de uma ontologia de domínio. Por isto, a comunicação entre os agentes se dá apenas entre marketplaces de uma mesma região, exceto para os móveis que podem, adicionalmente, trafegar entre diferentes regiões.

Os **agentes negociadores** são os agentes móveis que representam os usuários do ICS. Eles são oriundos do ambiente destas empresas, que trafegam pela rede até chegarem ao ICS carregados de informações sobre como comprar, vender, contratar e ofertar produtos e serviços.

O agente **matchmaker** é o responsável por procurar e juntar agentes negociadores afins, através de suas ofertas, para que estes possam iniciar o processo de negociação. A cada região é alocado um agente matchmaker.

O agente **mediador** atua a partir da fase de Negociação, assumindo um papel de árbitro, aplicando o Protocolo de Negociação, que se encontra armazenado no **repositório de ontologia** – mostrado mais adiante – para garantir integridade a este processo. Assim, pode gerenciar possíveis resoluções de problemas que venham a ocorrer e coordenar a participação de agentes na negociação. Existe um agente mediador em cada marketplace.

Caso ocorra uma negociação bem sucedida, o **agente de contrato** passa a gerenciar o processo de formação do contrato, onde faz uso do **repositório de contratos**, seguido do processo de cumprimento.

### **3.4 Ciclo de vida**

O ciclo de vida adotado no ICS para desenvolvimento e funcionamento de suas transações de negócios eletrônicos foi baseado no modelo proposto por Jennings (JENNINGS et al., 1996). Entretanto, a principal contribuição do sistema ICS foi a extensão do ciclo de vida (LABIDI et al., 2003), incluindo: uma nova fase (Modelagem do Usuário), o conceito de feedback de informações e a definição de ontologias que fornecem um suporte semântico às atividades de comunicação e compartilhamento de informações efetivadas pelos agentes.

A Figura 6 mostra todas as fases e características do ciclo de vida do sistema ICS.

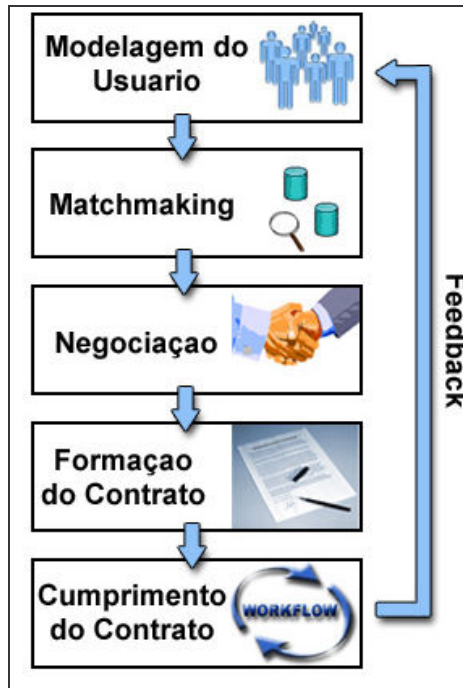


Figura 6: Ciclo de vida do ICS

No ICS, uma transação de negócio bem sucedida desenvolve-se percorrendo as cinco fases do ciclo de vida.

Na primeira fase - Modelagem do Usuário - são capturadas e modeladas informações sobre os usuários do ICS e sobre os produtos e serviços desejados. Todas essas informações são passadas aos agentes negociadores para que estes possam então, se deslocarem de seus marketplaces em busca de possíveis parceiros de negócio.

Quando os agentes negociadores se encontram em um marketplace, tem início a fase de Matchmaking. Eles passam a depositar suas informações, na forma de anúncios, em um repositório de acesso compartilhado. O agente matchmaker é o responsável por procurar possíveis parceiros de negócios através dos anúncios que foram publicados nestes repositórios. Se vier a encontrá-los, ele os reunirá em grupos e os disponibilizará ao mediador.

Na fase de negociação, os agentes negociadores entram em cena novamente, iniciando, junto com seus parceiros, os processos de oferta e procura dos produtos e/ou serviços desejados. Este processo é controlado pela supervisão do agente mediador.

As duas fases seguintes somente ocorrerão, após uma negociação bem sucedida. A fase de Formação do Contrato consiste na determinação explícita, através de um contrato, de todos os termos acordados durante a negociação. Posteriormente, a fase de Cumprimento do Contrato gera um suporte para o acompanhamento e gerenciamento das obrigações resultantes do mesmo. O agente responsável por estas duas funções é o agente de contrato.

Algumas considerações sobre a implementação dos agentes e a modelagem de ontologias que foram produzidas em diversos trabalhos realizados do sistema ICS merecem esclarecimento. Os agentes móveis foram desenvolvidos utilizando a plataforma de agentes SOMA (CORRADI, 1999), mostrada na Figura 7 e as ontologias foram modeladas através da ferramenta OIEd (BECHHOFER, 2001), representada pela Figura 8.

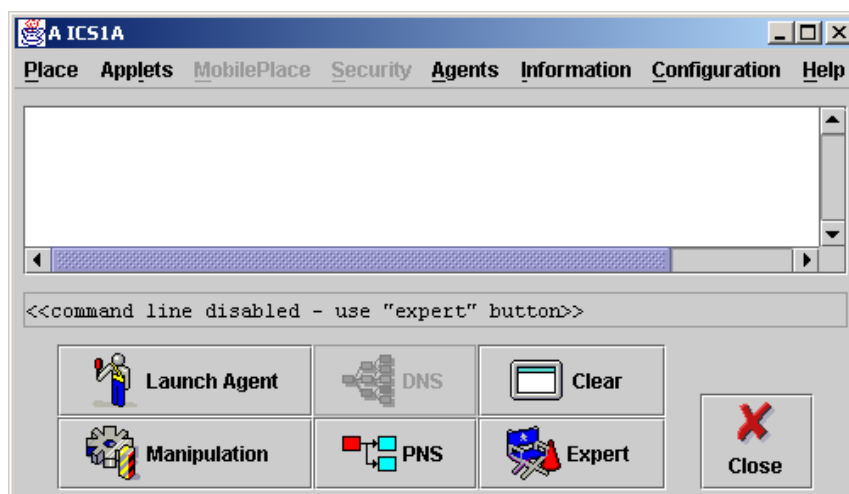


Figura 7: Interface de Manipulação dos Agentes da plataforma SOMA

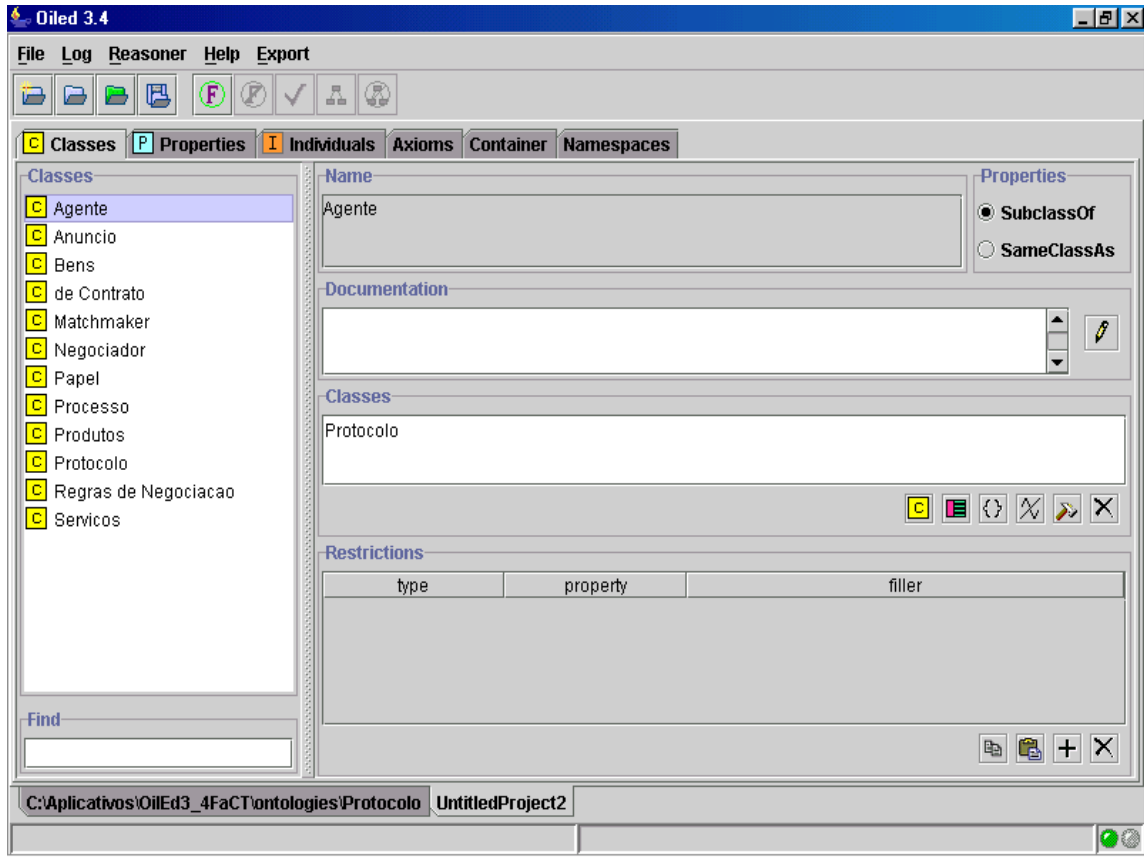


Figura 8: Editor de ontologias OILED

A seguir, detalharemos todas as fases do ICS.

### 3.4.1 Modelagem do Usuário

Esta fase do ICS objetiva adquirir todas as características, comportamentos, preferências e restrições dos usuários, assim como informações específicas sobre os produtos e serviços que eles pretendem obter ou contratar. Todas essas informações servem tanto para compor estratégias de negociação quanto para formar os anúncios (BASTOS FILHO, 2003).

Abaixo, na Figura 9, é representado o ambiente de execução da fase de Modelagem do Usuário.



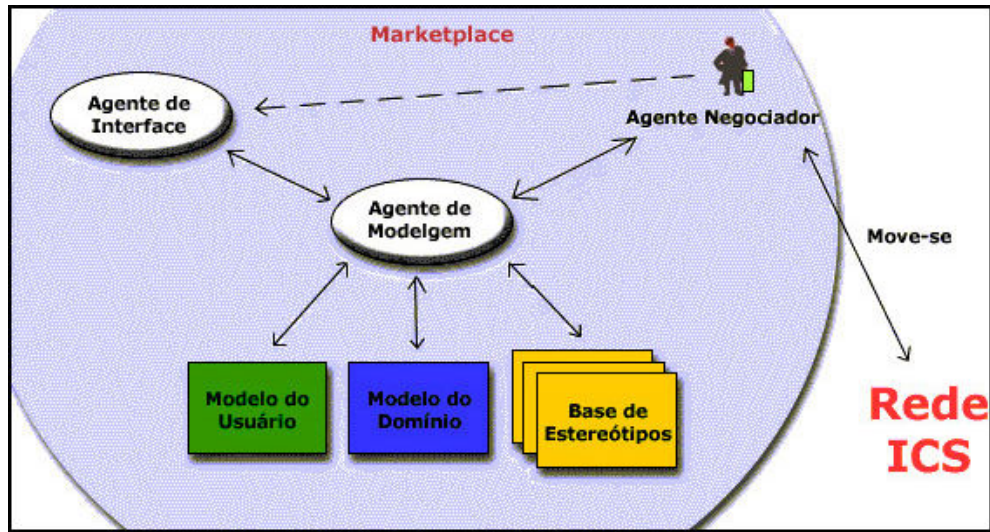


Figura 9: Ambiente de execução da Modelagem do Usuário

Todo o processo de modelagem foi desenvolvido utilizando-se agentes de software. A execução inicia-se com o **agente de interface**, que funciona como um elo entre o usuário e o sistema que dá suporte à modelagem. De um lado, ele captura informações diretamente do usuário e as repassa ao agente de modelagem. Do outro, ele recebe os resultados da negociação fornecidos pelo agente negociador que, neste momento, já retornou de sua jornada, com informações sobre sua atuação.

O **agente de modelagem** gerencia como as informações provenientes do agente de interface, do Modelo do Usuário, do Modelo do Domínio e da Base de Estereótipos serão apresentadas ao negociador:

**Modelo do Usuário** – repositório que armazena os modelos de usuários cadastrados no sistema. Esses modelos estão organizados em duas categorias: características do usuário e características de busca. As características do usuário contêm dados que compõem o perfil do agente negociador. Entretanto, as de busca registram aspectos relacionados à formação de anúncios. A Figura 10 mostra um

exemplo de modelo do usuário, cuja estrutura é representada através de uma ontologia.

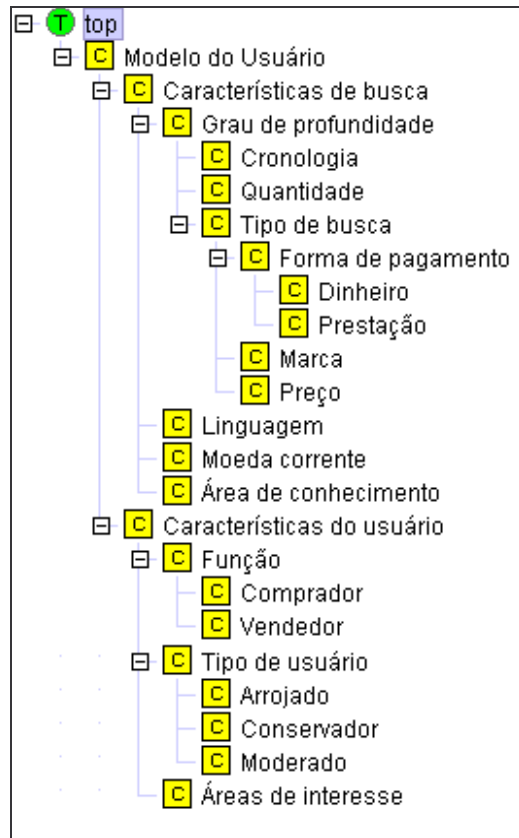


Figura 10: Exemplo de um Modelo de Usuário

**Modelo do domínio** – define atributos sobre produtos e serviços específicos, modelados através de ontologias. Exemplo: o produto livro possui os atributos título, autor, editora, ano de publicação, edição, assunto, etc.

**Base de estereótipos** – descreve características básicas dos usuários que possam integrá-los em grupos de perfis similares.

O agente negociador constitui a mobilidade do ICS. No primeiro momento, ele recebe as informações do usuário que representará nos processos de busca e negociação, partindo em direção ao ICS. Posteriormente, retorna ao seu ambiente com os resultados que obteve.

### 3.4.2 Matchmaking

Após a saída dos agentes negociadores dos marketplaces dos usuários, eles são direcionados ao ambiente ICS para tentar alcançar seus objetivos. Ao chegarem a um determinado marketplace, como dissemos anteriormente, depositam os anúncios que carregam em repositórios compartilhados. Neste momento, o matchmaker, um agente local do ICS, inicia seu procedimento. Sua função é descobrir oportunidades de negócios para os agentes que representam as empresas e então, colocá-los em um mesmo grupo para iniciarem a negociação.

O processo de união destes agentes se desenvolve através de um emparelhamento das características de um produto ou serviço específico. Estas informações estão presentes nos anúncios armazenados nos repositórios.

Em cada região do ICS existe um matchmaker que inicia o processo de match em intervalos de tempo pré-determinados (TOMAZ, 2003). Se houver negociadores afins, o resultado deste match resulta em uma base de clusters. Clusters são unidades de informação de estrutura bem simples, geradas pelo matchmaker e que representam uma possível dupla de parceiros de negócios. A Tabela 1 mostra um exemplo desta estrutura.

Cluster
ID_AGENTE_COMPRA
ID_AGENTE_VENDA
ESTADO

Tabela 1. Estrutura de um cluster

Após a geração de clusters, o matchmaker passa a agrupar aqueles que possuem o mesmo produto ou serviço, compondo o que foi denominado de grupos de negociação. Esses grupos contêm um conjunto de agentes que estão interessados em, por exemplo, vender/comprar um mesmo produto ou serviço.

O próximo passo do matchmaker é instanciar quantos agentes mediadores forem o número de grupos de negociação, finalizando a fase de Matchmaking e iniciando a de Negociação.

### 3.4.3 Negociação

A negociação de produtos ou serviços no ICS desenvolve-se envolvendo agentes negociadores e mediadores. Os agentes mediadores cumprem o papel de árbitros, gerenciando todo o processo através do Protocolo de Negociação. Este protocolo impõe regras na maneira como tudo deve ocorrer e controla o fluxo de mensagens entre negociadores. É importante que as regras sejam definidas com o propósito de garantir “justiça” aos participantes do processo (FONSECA, 2003).

Todo o protocolo foi modelado em uma Ontologia, pois sua estrutura de dados e seus respectivos significados são armazenados nos marketplaces para serem compartilhados entre todos os agentes participantes da negociação (cf Figura 11). Cada marketplace utiliza o seu próprio protocolo, que pode ser dos seguintes tipos: leilão, barganha, votação, etc.

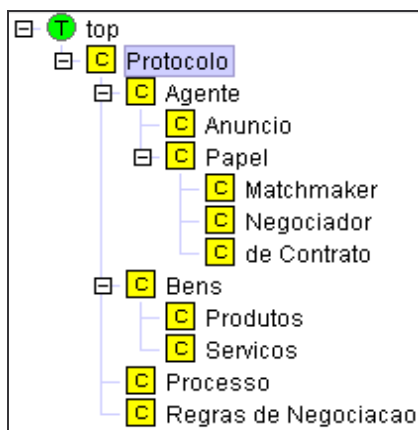


Figura 11: Ontologia do Protocolo de Negociação

Uma descrição em seqüência de passos é necessária para demonstrar como ocorre o processo de negociação:

- a) Primeiramente, o agente mediador invoca todos os negociadores do grupo.
- b) Solicita as ofertas de negócio de cada um.
- c) Valida as ofertas conforme as restrições impostas no Protocolo de Negociação.
- d) Distribui somente aquelas que foram validadas aos negociadores que também obtiveram suas ofertas válidas, excluindo os demais.
- e) Avisa aos negociadores restantes a quantidade de agentes envolvidos, para que estes possam elaborar suas estratégias de negociação.
- f) Negociadores analisam as ofertas recebidas e elaboram suas estratégias.

A partir deste momento, a negociação propriamente dita inicia-se, acontecendo em série de rodadas. Um acordo somente é alcançado se, no mínimo, dois agentes fazem ofertas, um para o outro, tão boas ou até melhores do que aquela feita por eles mesmos. Matematicamente representando:

$O_1$  – oferta do negociador 1.

$O_2$  – oferta do negociador 2.

Acordo:  $utilidade_1(O_2) \geq utilidade_1(O_1)$  ou  $utilidade_2(O_1) \geq utilidade_2(O_2)$ .

Se nenhum acordo for alcançado, o mediador inicia uma nova rodada, com a condição de que nenhum negociador pode propor oferta menos interessante aos outros do que a sua na rodada anterior. Prosseguindo a mesma situação, o agente mediador finaliza a negociação e informa ao grupo que esta chegou ao fim.

Porém, se um acordo for fechado, os negociadores envolvidos devem avisar o fato ao mediador. Isto faz com que ele possa avisar o ocorrido ao grupo, liberando-os desta transação e finalizando a negociação.

Esta ação conclui a fase de negociação. A vontade coincidente de fechar um acordo é que determina a existência das próximas etapas: Formação e Cumprimento do Contrato.

#### **3.4.4 Formação do Contrato**

Esta fase objetiva assegurar a transação de negócio realizada através do ICS. A decisão de utilizar contratos neste sistema visa tornar os negócios mais atrativos (OLIVEIRA, 2004). Todos os aspectos referentes à modelagem e implementação utilizados nesta etapa do ICS serão mostrados nos capítulos seguintes.

#### **3.4.5 Cumprimento do Contrato**

Embora toda transação bem sucedida leve à adoção de um contrato, proporcionando-a uma certa garantia, a fase de Cumprimento possui justificativas para sua existência. De um lado, o ICS oferece mais um serviço aos seus clientes, quando acompanha e verifica se as obrigações de cada parte estão sendo executadas. Do outro, enriquece o Modelo do Usuário através da reutilização dessas informações.

## **4 MODELAGEM DOS CONTRATOS ELETRÔNICOS**

Neste capítulo, procuramos apresentar e justificar o modelo proposto de contratos eletrônicos para ser utilizado no sistema ICS. A modelagem envolveu a adoção de uma estrutura de armazenagem e a construção de uma ontologia utilizada e compartilhada pelos agentes envolvidos nesta fase.

### **4.1 Função dos contratos eletrônicos no ICS**

A função dos contratos eletrônicos, no sistema ICS, consiste em registrar as transações de negócios efetuadas entre os seus clientes. Todos os parâmetros que foram acordados na fase de Negociação serão transportados para um determinado contrato, de acordo com o tipo de negócio que foi realizado. O ICS, inicialmente, provê quatro tipos de transações que podem ser realizadas entre empresas: compra e venda de produtos, troca de produtos ou serviços, locação de imóveis e prestação de serviços.

Outra característica importante desses contratos encontra-se no fato de que as obrigações de suas partes integrantes (empresas) servem como base para a fase de Cumprimento, mostrada mais adiante.

### **4.2 Abordagens de modelagem**

Destaca-se na literatura (LUDWIG, 2001), quatro abordagens para modelagem de contratos eletrônicos e representação de seus elementos negociáveis.

#### *4.2.1 Abordagem 1 – Palavras da Linguagem Natural*

Nesta abordagem, os elementos negociáveis são palavras da linguagem natural, da mesma maneira que todo o contrato. A desvantagem da utilização da linguagem natural é a preocupação com os problemas de más interpretações, ambigüidades e imprecisões, além da análise sintática. Isto tudo dificulta a automação.

#### *4.2.2 Abordagem 2 – Palavras da Linguagem Formal*

Esta abordagem requer a disponibilidade de uma linguagem específica para lidar com o domínio dos contratos. Sendo assim, a formação do contrato consistirá na utilização do vocabulário e na aplicação da adequada sintaxe, ambos pertencentes à referida linguagem.

#### *4.2.3 Abordagem 3 – Templates de Contratos*

Esta abordagem faz uso de templates, que nada mais são do que modelos padrões de contratos. Esses templates são compostos de texto juntamente com campos destacados que receberão, posteriormente, seus respectivos valores. Esta abordagem permite automação, pois os templates podem ser armazenados em repositórios e acessados, enquanto os campos, preenchidos através de algum processo automatizado.



#### *4.2.4 Abordagem 4 – Template de Cláusulas*

Nesta abordagem, templates são as cláusulas formadoras de um contrato. Então, um contrato é formado pela composição de vários templates de cláusulas. A dificuldade fica por conta da correta composição de um contrato, que implica no conhecimento da posição das cláusulas no corpo do mesmo. Esta abordagem também é passível de automação.

### **4.3 Modelagem proposta**

Como objetivo maior desta dissertação, nosso trabalho consiste em prover as duas últimas fases do ciclo de vida do sistema ICS, como afirmamos anteriormente. O ICS é um sistema que faz uso de agentes inteligentes e móveis para atingir seus objetivos, de forma que durante todo o projeto procurou-se prover uma considerável estrutura de comunicação, entendimento e compartilhamento das informações entre estes agentes. Os exemplos são as ontologias que foram utilizadas para modelar a maioria das informações internas ao sistema. Maiores detalhes sobre as fases anteriores podem ser encontrados nas referências citadas nesta dissertação, localizadas no capítulo 3.

Procuramos seguir este caminho, de forma que nosso objetivo consistiu em modelar os contratos eletrônicos através de uma representação voltada para leitura e entendimento da máquina ou “machine-understandable”, que são os agentes inteligentes. Desta forma, propomos o uso de padrões, conceitos e ferramentas da Web Semântica como uma opção à modelagem.

Das abordagens acima demonstradas, optamos por aquela que trata dos templates de contratos. A idéia de formar modelos de contratos padrões, possuindo estrutura fixa e campos negociáveis a serem preenchidos, foi observada em nossa pesquisa como uma prática comum adotada por profissionais da área jurídica. Isto se justifica pela presença de um grande número de elementos que são comuns ao mesmo tipo de contrato, implicando numa redução de custos na atividade de configuração dos mesmos.

Em nossa modelagem, os templates são formas de contratos padrões completos, que incluem campos a serem preenchidos. Estas formas possuem estrutura fixa e pré-definida, de acordo com o tipo de transação a ser efetuada, que leva a um tipo de contrato a ser utilizado. Entretanto, os campos representam os parâmetros que são negociados entre as partes. Por enquanto, os contratos providos pelo ICS são: Compra e Venda, Troca, Locação e Prestação de Serviço.

Para lidar com a automação desta modelagem e aplicar a representação pretendida, fizemos uso do padrão XML e construímos uma ontologia de domínio, denominada de ontologia dos Contratos.

Os templates foram codificados em XML, onde as tags provêm estruturação e representação à informação textual (escrita em linguagem natural), além de representarem os campos a serem preenchidos. No Apêndice deste trabalho, encontra-se um template de contrato de Compra e Venda, no formato XML.

A ontologia dos Contratos levou em consideração todos os parâmetros pertencentes a cada tipo de contrato provido. Desta forma, os agentes que necessitam de informações sobre contratos e seus respectivos parâmetros – caso do agente de contrato e dos negociadores – podem encontrar nesta estrutura, que está armazenada num repositório de acesso compartilhado (cf Figura 5).

## 4.4 Ontologia dos Contratos

Durante a fase de Negociação, os agentes negociadores desenvolvem suas tarefas seguindo a imposição do Protocolo de Negociação, que se encontra modelado como uma ontologia (cf Figura 11). Aqui, nós o estendemos, adicionando o conceito de Contratos, sua estrutura e seus parâmetros. Desta forma, criamos um vocabulário particular deste domínio, que possui como objetivos: a melhora da negociação e a validação dos templates.

Quanto à melhora da negociação, podemos afirmar que esta tende a ficar mais coesa, uma vez que o agente negociador passa a possuir conhecimento sobre os parâmetros relativos ao tipo de transação que venha a efetuar.

Quanto à validação, o agente de contrato deve realizá-la, usando os parâmetros que foram a ele enviados, para poder se certificar de que tudo ocorreu conforme o protocolo. E isto somente se torna possível, através do acesso à ontologia compartilhada.

A Figura 12 a seguir, mostra a ontologia dos Contratos, representada através do editor de ontologias Protegé (NOY, 2001). Consiste de classes, subclasses e atributos que representam os componentes dos contratos. Encontra-se inserida na ontologia do Protocolo de Negociação que, por sua vez, está armazenada em um repositório de acesso compartilhado a todos os agentes que fazem uso da mesma.

A classe principal é Contrato, dividida em três subclasses que representam sua estrutura como um todo: Identificacao\_Partes, Especies e Disposicao\_Geral. Cada uma destas possui seus próprios atributos e possíveis subclasses.

A subclasse Identificação\_Parties trata das informações relativas às partes contratantes (nome, endereço, código, etc). Especies é a subclasse que contém os tipos de contratos e, Disposição\_Geral compreende nas informações complementares (local, data, foro, etc).

## **4.5 Repositório de Contratos**

Para facilitar o desenvolvimento do processo de automação dos contratos eletrônicos, propusemos, também, o conceito de instâncias de contratos e de um repositório para armazenar os templates e as instâncias.

As instâncias são contratos que já foram fechados pelas partes, ou seja, templates cujos campos já tenham sido preenchidos depois de uma negociação bem sucedida. Estas devem ser armazenadas no repositório com a finalidade de registrar os negócios que foram realizados no ICS entre duas empresas.

O repositório somente é acessado pelo agente de contrato. Deve fornecer os mecanismos básicos de armazenagem e recuperação de contratos, além de prover operações relativas a gerenciamento. Por exemplo: adição de novos templates, especialização dos existentes, estabelecimento de relacionamentos, entre outras.

A Figura 5 mostra como é disposto, no ambiente ICS, o repositório de contratos.

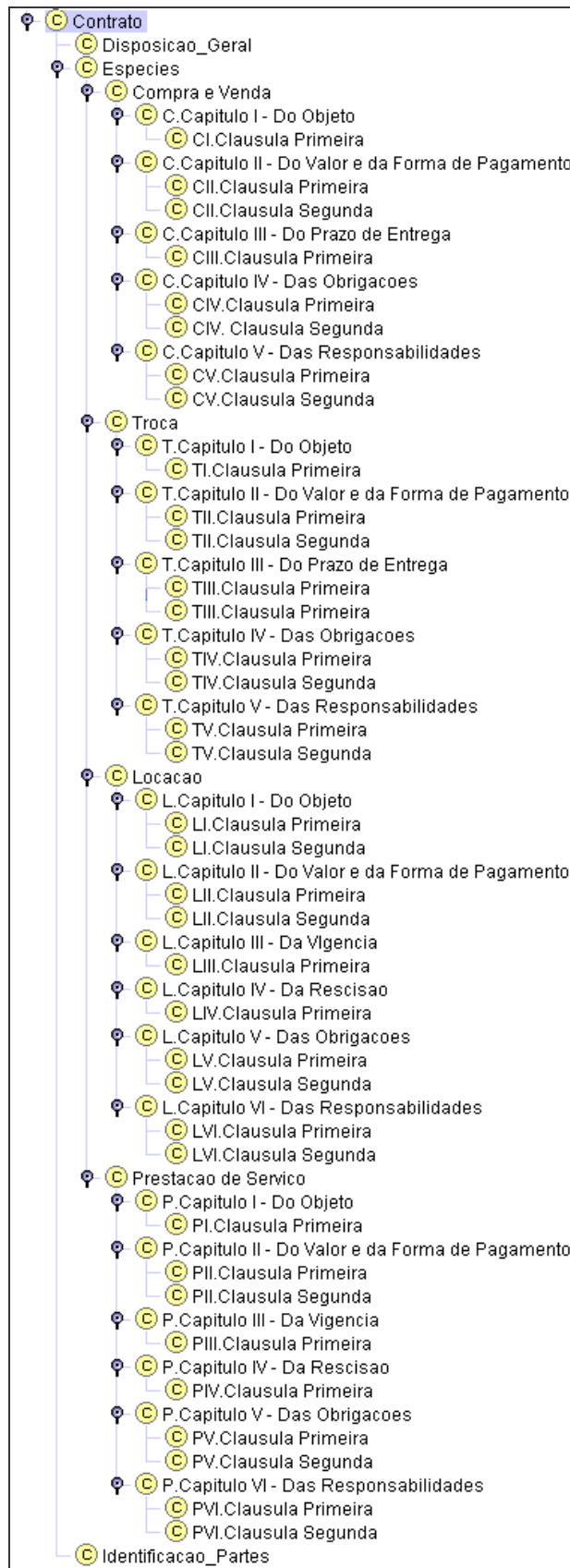


Figura 12: Ontologia dos Contratos

Abaixo, a Figura 13 mostra a nova versão do Protocolo de Negociação, desta vez, com as características do contrato.

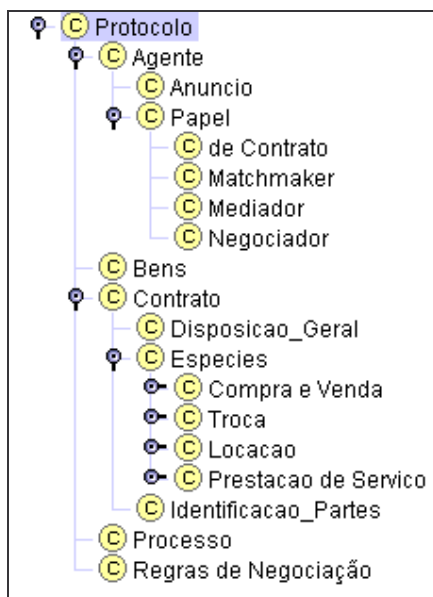


Figura 13: Nova versão do Protocolo de Negociação

## **5 FORMAÇÃO DO CONTRATO**

Este capítulo descreve a fase de Formação do Contrato, apresentando o responsável por gerenciá-la: o agente de contrato. Descrevemos sua função, mostrando suas interações com outras entidades e apresentamos uma pequena implementação, utilizando a ferramenta JADE, no apêndice deste trabalho.

### **5.1 Função**

Esta fase incrementa o sistema ICS, através da configuração de um contrato, incluindo todos os parâmetros negociados na fase anterior. Esta medida, conforme destacamos anteriormente, visa tornar as transações de negócios do ICS mais atrativas; pois o contrato é uma forma de registro do que foi fechado entre as empresas.

### **5.2 Agente de contrato**

Neste trabalho, um dos nossos objetivos consistiu no desenvolvimento da função exercida pelo agente de contrato: gerenciamento da constituição dos contratos eletrônicos do ICS. Para realizar tal função, este agente deve realizar uma série de tarefas, como: comunicar-se com os agentes negociadores que fecharam um acordo, acessar o repositório de contratos e recuperar um template, validá-lo utilizando o Protocolo de Negociação e, por fim, armazená-lo de volta no repositório.

### 5.3 Descrição do cenário de atuação do agente de contrato

Conforme afirmamos, a fase de Formação de Contrato somente inicia, caso a fase de Negociação tiver sido bem sucedida, resultando em um acordo entre duas empresas.

O cenário básico de atuação do agente de contrato será descrito nos parágrafos seguintes:

- a) Os agentes negociadores que fecharam um acordo emitem um aviso ao agente de contrato;
- b) Este, então, passa a acessar o repositório de contratos e recuperar um template correspondente ao tipo de transação (Compra e Venda, Troca, Locação ou Prestação de Serviços).
- c) Em seguida, requisita os parâmetros de negociação aos agentes negociadores;
- d) Estes enviam os parâmetros de volta;
- e) Agente de contrato passa a validá-los utilizando o Protocolo de Negociação;
- f) Após a validação, o agente preenche o template com os parâmetros recebidos e o armazena no repositório, fazendo com que este passe a se tornar uma instância de um contrato que foi fechado entre dois clientes do ICS.

Por fim, os negociadores podem retornar aos seus marketplaces de origem.



## 5.4 Diagrama de interação da fase de Formação do Contrato

Na figura abaixo, um diagrama demonstra as interações entre todas as entidades participantes da fase de Formação do Contrato, conforme descrito na seção anterior. Destacamos, em particular, o agente de contrato.

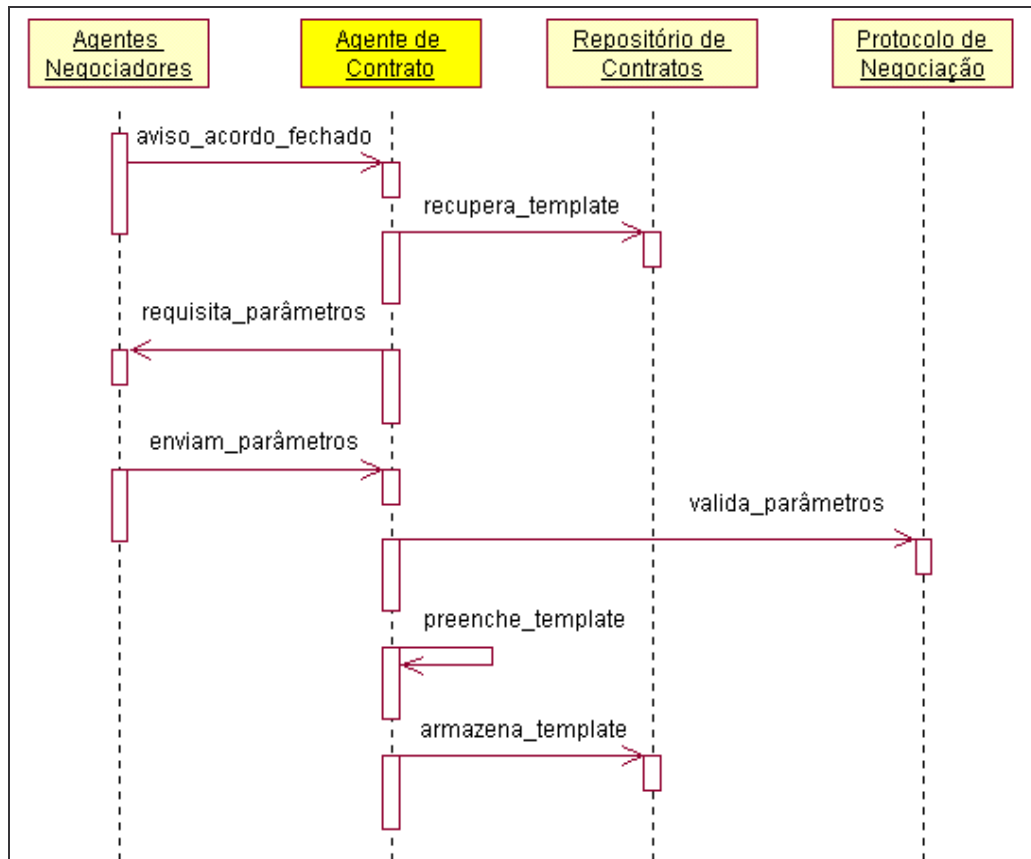


Figura 14: Diagrama de interação da fase de Formação de Contrato

## 5.5 Implementação

Este trabalho não focaliza a implementação dos agentes negociadores. Contudo, para demonstrarmos o funcionamento desta fase, foi necessário que os implantássemos; porém apenas os aspectos relativos à comunicação que estes

mantêm com o agente de contrato. Tanto os negociadores quanto o agente de contrato foram implementados através da ferramenta JADE, pois esta é a ferramenta padrão utilizada neste projeto.

JADE<sup>7</sup> (*Java Agent Development Framework*) é um framework para desenvolvimento de aplicações multiagentes, completamente escrito na linguagem Java e que se encontra em conformidade com os padrões FIPA<sup>8</sup> (*Foundation for Intelligent Physical Agents*).

JADE é um software livre, concebido e desenvolvido pela TILab<sup>9</sup>. Está disponível a toda comunidade sob os termos da LGPL (*Lesser General Public License Version 2*).

Este framework inclui dois produtos principais: uma plataforma de agentes genéricos e um pacote para desenvolvimento de agentes específicos em linguagem Java.

Possui uma interface gráfica GUI, mostrada na Figura 17, que fornece suporte às fases de debug e desenvolvimento dos agentes. A plataforma de agentes pode ser distribuída em várias máquinas e a configuração dos agentes controlada via esta interface.

A estrutura de comunicação é bastante flexível e eficiente. Desenvolvida através de passagem de mensagens, onde a linguagem FIPA ACL é utilizada para representação das mesmas. Cada agente possui uma fila privada de mensagens ACL, que pode ser acessada por uma combinação de modos diferentes. Inclui, ainda, suporte para ontologias e linguagens de conteúdo.

---

<sup>1</sup> <http://jade.cselt.it/>.

<sup>2</sup> [www.fipa.org](http://www.fipa.org).

<sup>3</sup> <http://www.telecomitalialab.com/s>

Com relação às ontologias, JADE e o editor de ontologias Protégé proporcionam uma considerável facilidade na construção e acesso das mesmas, graças a um certo plug-in conhecido como **beangenerator**. É possível definir ontologias utilizando Protégé e, então, passar a responsabilidade para o beangenerator. Este gera, automaticamente, em código Java, tanto a classe de definição da ontologia, quanto às dos conceitos, predicados e ações dos agentes.

Sendo assim, criamos novamente a ontologia do Protocolo de Negociação, desta vez utilizando Protégé e beangenerator (cf Figura 18), incluindo a classe Contratos, que representa a ontologia de Contratos. Esta ontologia é declarada como um simples objeto e disponibilizada aos agentes através de um método de acesso. Isto permite o compartilhamento da mesma entre todos os agentes (negociadores e de contrato).

No apêndice B deste trabalho, encontra-se o código-fonte (em Java) dos agentes negociadores e de contrato e, no C, o código da ontologia gerada através do beangenerator.

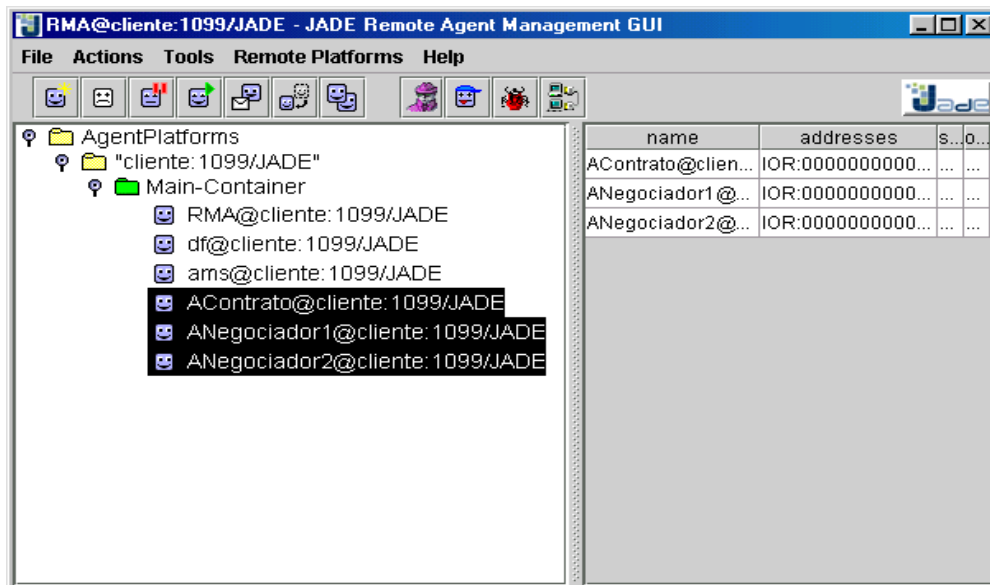


Figura 15: Interface gráfica da plataforma JADE

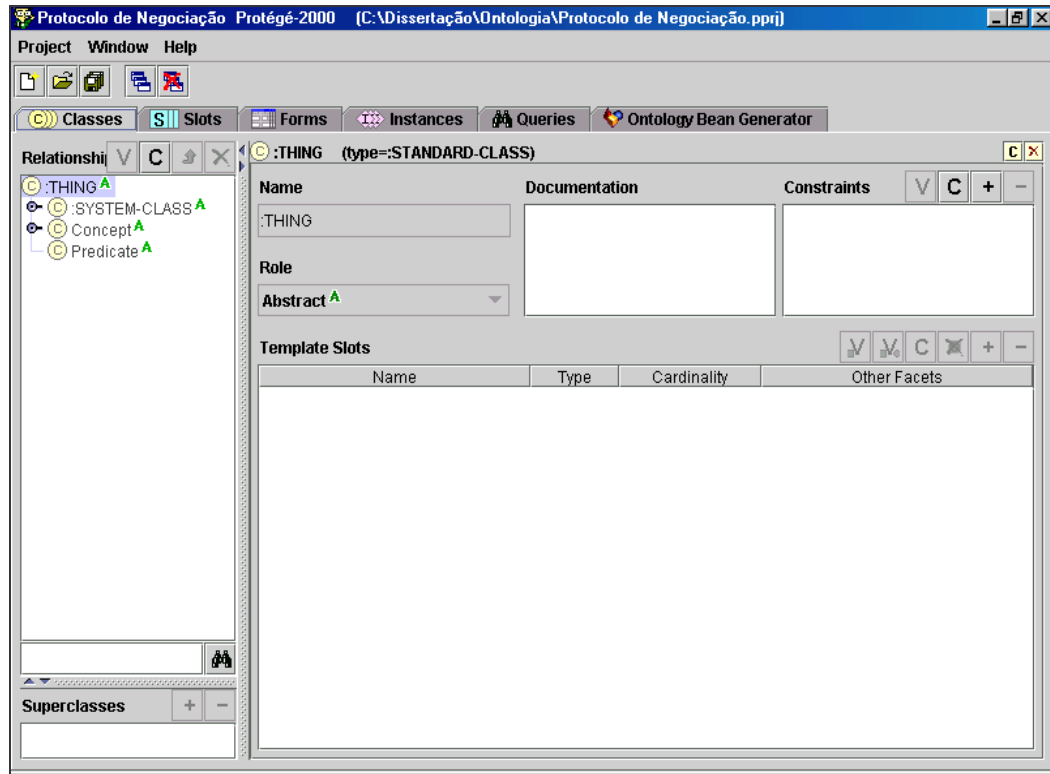


Figura 16: Editor de ontologias Protégé associado ao beangenerator

## **6 CUMPRIMENTO DO CONTRATO**

Na última etapa deste trabalho, apresentamos a fase de Cumprimento do Contrato. Esta encerra o ciclo de vida do ICS, fornecendo um serviço de acompanhamento das ações dos usuários deste sistema que efetivaram uma transação.

### **6.1 Função**

O objetivo desta fase consiste em monitorar e gerenciar o cumprimento das obrigações que foram assumidas por empresas, no momento de uma transação efetivada através do ICS. Esta medida acarreta vantagens para o ICS e para os seus clientes.

Os clientes envolvidos em um negócio possuirão informações, em forma de aviso, sobre o comportamento de seus parceiros. Isto pode, por exemplo, levá-los a tomar decisões sobre medidas corretivas, nos casos de quebra do contrato.

Em contrapartida, estas mesmas informações servirão para enriquecer o repositório Modelo do Usuário da fase de Modelagem, através de uma atualização sobre o comportamento de um usuário que acabara de participar de uma transação. Isto justifica a presença do conceito de feedback no ciclo de vida do ICS.

### **6.2 Modelo proposto de cumprimento**

Neste trabalho, propomos um modelo de monitoramento e gerenciamento das obrigações de um contrato, utilizando Workflow Temporal e regras ativas

fundamentadas no paradigma ECAA de Sistemas de Banco de Dados Ativos. A justificativa para a escolha do Workflow Temporal resultou da observação de que um contrato pode ser considerado como um processo de negócio, onde as obrigações funcionam como atividades a serem cumpridas em datas limites (prazos). O paradigma ECAA consiste em um conjunto de regras constituídas de eventos, condições e ações. Estes elementos foram considerados muito adequados ao modelo de monitoramento proposto.

Desta forma, as regras ativas são usadas pelo agente de contrato para monitorar, no momento exato, o estado das obrigações de cada parte (empresa).

Na etapa de Cumprimento, aplicamos este modelo em associação com o agente de contrato. Em nosso método, o agente desempenha as seguintes ações:

- a) extrai as informações do contrato relativas às obrigações;
- b) produz um workflow para acompanhar a seqüência de obrigações a serem cumpridas e seus respectivos prazos, além das atividades relacionadas;
- c) cria uma tabela para cada usuário, contendo informações sobre as obrigações e seus respectivos estados e prazos;
- d) gera um conjunto de regras que levam em consideração o tempo limite de cada obrigação, os estados das mesmas e as ações, por ele, a serem tomadas.

Considerações:

No método de funcionamento do ICS, os usuários devem notificar o sistema quando completam suas obrigações. Essas informações são mantidas em tabelas, uma para cada usuário. As tabelas contêm os campos Obrigações, Estado e tempo final (tf), mostrados abaixo:

Obrigações	Estado	tf

Tabela 2: Informações sobre obrigações do usuário

Uma obrigação cumprida dentro do prazo estabelecido possui um estado com valor TRUE. Caso contrário, será FALSE.

O modelo de Workflow Temporal escolhido leva em consideração o fator tempo de cada atividade do processo de negócio (ALVES, 2000). Desta forma, o workflow produzido pelo agente consistirá numa seqüência de atividades (obrigações) associadas aos seus tempos inicial, de duração e final. A unidade de tempo adequada para este tipo de processo é o dia.

As regras ativas possuem o seguinte formato:

**ON EVENTO  
IF CONDIÇÃO VERDADEIRA  
DO AÇÃO1  
ELSE DO AÇÃO2**

Os eventos ocorrem quando a variável T alcança os prazos finais de cada atividade do workflow, definidos no campo **tf** da tabela de cada usuário, acrescentados em mais um dia. As condições avaliam se os estados foram cumpridos, ou seja, se os seus valores são TRUE. As ações são executadas pelo agente em duas situações: quando a condição for satisfeita ou quando não for. Para os casos de condições satisfeitas, ou seja, situação onde uma parte cumpriu com sua obrigação dentro do prazo, o agente deve emitir um aviso para a outra parte, informando que a situação está sob controle. Para os casos contrários, ele também avisa o ocorrido e, se necessário, desativa as regras restantes.

Utilizamos uma variável T, cuja atualização deve ser diária, para controlar o momento em que o evento venha a ocorrer. Por conseguinte, no início do processo T recebe valor 0, sendo incrementada em uma unidade, a cada dia que passa.

### 6.3 Exemplo

Demonstraremos a aplicação de nosso modelo de cumprimento, supondo uma transação de compra e venda que foi efetuada entre duas empresas, através do ICS. O contrato, contendo todas as informações referentes à esta transação, segue o mesmo modelo do que está anexo a este trabalho. As obrigações relativas são:

- a) A empresa compradora deve efetuar o pagamento correto e enviar o seu comprovante à empresa vendedora, no prazo acertado.
- b) A empresa vendedora deve entregar o produto no prazo combinado.

Em ambos os casos, caso não haja o cumprimento, o contrato será cancelado.

De posse deste contrato, o agente inicia sua jornada. No primeiro momento, ele desempenha a extração de informações relativas às obrigações e seus prazos. Posteriormente, produz o workflow, cria a tabela e gera as regras ativas.

Todo esse procedimento e as informações associadas são mostrados abaixo. A Figura 15 mostra o modelo de Workflow Temporal produzido pelo agente, diante das obrigações geradas pela adoção do contrato de Compra e Venda.

Tabela da empresa compradora:

Obrigação	Estado	tf
Enviar comprovante de pagamento		5

Tabela da empresa vendedora:

Obrigação	Estado	tf
Verificar recebimento de comprovante		6



Assim que as partes cumprirem suas obrigações, elas devem notificar este fato ao ICS, através de uma interface que está disponível às mesmas. Desta forma, as tabelas serão atualizadas com o valor TRUE no campo Estado da obrigação relacionada.

No momento correto, a variável T atingirá o tempo final de uma obrigação, disparando a regra 1 para o agente. Este testará a condição, que se for satisfeita, ou seja, a presença do valor TRUE no campo Estado, provocará a ativação da ação correspondente – avisar a empresa vendedora que o contrato está sendo cumprido.

No caso de não cumprimento, o valor do campo Estado será FALSE, o que levará a condição a não ser satisfeita e, conseqüentemente, ao agente realizar a ação correspondente ao ELSE – desativar a regra 2 e notificar a empresa vendedora da quebra do contrato.

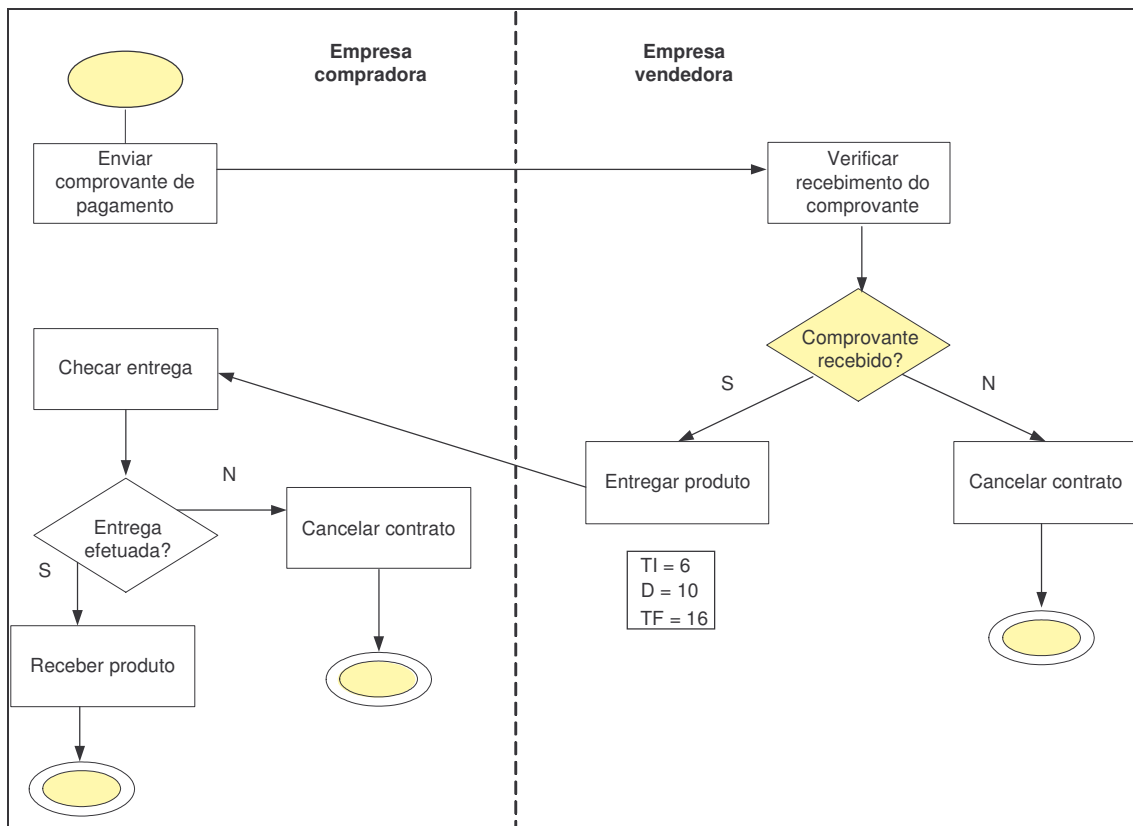


Figura 15: Workflow produzido pelo agente de contrato

**Regra1:**

ON T = tf1 + 1

IF Enviar comprovante de pagamento.Estado = TRUE

DO Aviso "OK" para empresa vendedora

ELSE \* Aviso "Não cumprimento" para empresa vendedora;

\* Desativar as regras restantes.

**Regra2:**

ON T = tf2 + 1

If Entregar produto.Estado = TRUE

DO Aviso "OK" para empresa compradora

SENÃO \* Aviso "Não cumprimento" para empresa compradora;

FIM

## 7 CONCLUSÃO

Os contratos são instrumentos jurídicos legais que, por serem fontes da obrigação, proporcionam mais confiança às relações por eles estabelecidas. Portanto, a adoção dos mesmos em um sistema de comércio eletrônico como o ICS, visa tornar os negócios mais atrativos.

Com a finalidade de obter um maior conhecimento dos usuários, conseqüentemente, melhorando a aquisição de informações, a fase de Cumprimento torna-se de grande importância.

O Projeto ICS foi concebido com o propósito de aplicar a tecnologia de agentes em sua arquitetura a fim de automatizar todo o processo de comércio eletrônico. Desta forma, também fizemos uso desta mesma tecnologia, propondo assim, o agente de contrato. Este agente é o responsável por gerenciar as duas últimas fases do ciclo de vida que tratam das questões que envolvem os contratos eletrônicos utilizados neste sistema.

O gerenciamento destas fases envolve duas atividades principais: a interação do agente de contrato com outros agentes, os negociadores; além da manipulação de informações bastante específicas do domínio dos contratos.

Para lidar com a complexidade e os problemas encontrados, propomos a modelagem dos contratos eletrônicos através de uma representação voltada para leitura e entendimento dos agentes que fazem uso dos mesmos. Desta maneira, utilizamos os padrões, conceitos e ferramentas da Web Semântica.

Por ser um padrão de fato para representação de documentos e por permitir uma completa estruturação à informação contida nos mesmos, adotamos XML na confecção dos templates de contratos atualmente disponíveis.

O motivo para a construção da ontologia encontra-se nos fatos de que se faz necessário tanto prover um entendimento comum a todos os agentes que necessitam de informações sobre um determinado contrato, quanto melhorar a comunicação entre eles.

O modelo proposto para lidar com o cumprimento das obrigações de um contrato, composto por Workflow Temporal e regras ativas fundamentadas no paradigma ECAA de Sistemas de Banco de Dados Ativos, possibilita as seguintes vantagens:

- Workflow Temporal produz um conjunto de atividades relacionadas que realizam um objetivo de negócios, levando em consideração o fator tempo, de relevante importância no contexto das obrigações de um contrato. Isto permite um acompanhamento das ações dos usuários do ICS envolvidos em uma transação de negócio.
- As regras ativas foram visualizadas como um mecanismo de disparo para a tomada de ações pelo agente de contrato, no momento exato e de acordo com as condições associadas a estas ações.

As contribuições deste trabalho são realizadas dentro do contexto do Projeto ICS, pois levamos em consideração o seu próprio funcionamento. Não tivemos a pretensão de fazer deste modelo um padrão para as questões referentes aos contratos eletrônicos que venham a ser aplicadas aos diversos sistemas de comércio eletrônico existentes ou a serem desenvolvidos.

Concluimos assim, que estas fases são necessárias e importantes para o funcionamento completo a que o sistema ICS se propõe e, que todos os aspectos levantados e considerados nesta pesquisa foram fundamentados nas áreas que

envolvem questões de contratos eletrônicos, agentes inteligentes e naquelas que fornecem um suporte para realização das atividades complementares.

Como sugestões para trabalhos futuros, propomos:

- Maior flexibilização dos templates, configurando-os com mais especificações de elementos contratuais;
- Melhorar a implementação do agente de contrato;
- Implementar o comportamento do agente na fase de cumprimento, ou seja, desenvolvimento da extração das informações do contrato, produção do workflow e geração das regras;
- Prover um feedback do contrato preenchido às empresas que fecharam negócio, para que estas possam se certificar de que este corresponde à tudo aquilo que elas concordaram com suas parceiras de negócio. Este ponto implica na adoção de uma estrutura de segurança das informações trafegadas, durante a transmissão do contrato, ou seja, lidar com questões do tipo: assinatura digital e criptografia, por exemplo.

## REFERÊNCIAS

ALONSO, Gustavo et al. *WISE: Business to Business E-Commerce*. Proceedings of Int. Workshop on Research Issues in Data Engineering, Sydney, Australia, 1999.

ALVES, Ana Clenir Freitas da Costa. *Definição de um Modelo de Workflow Integrando a Cooperação e a Organização Temporal*. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís-Ma, 2000.

BASTOS FILHO, Othon de Carvalho. *Modelagem do Usuário para o Sistema ICS de Comércio Eletrônico*. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís-Ma, 2003.

BECHHOFFER, Sean et al. *OILEd: A Reason-able Ontology Editor for the Semantic Web*. Proceedings of KI2001, joint German/Austrian Conference on Artificial Intelligence, September, Vienna. Springer-Verlag LNAI Vol. 2174, pp 396-408. 2001.

BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora. *The Semantic Web*. Scientific American, Maio 2001.

BEVILÁQUA, Clóvis. *Código Civil dos Estados Unidos do Brasil*. São Paulo, 1950.

BRAY, Tim et al. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation. 6 October 2000. Capturado em 25 de Outubro de 2003. On-Line. Disponível em: <http://www.w3c.org/tr/rec-xml>

BRICKLEY, Dan; Guha, R.V.; MCBRIDE, Brian. *RDF Vocabulary Description Language 1.0: RDF Schema W3C Working Draft 10 October 2003*. Capturado em 25 de outubro de 2003. On-Line. Disponível na Internet em: <http://www.w3c.org/tr/rdf-schema/2003>

CORRADI, A. et al. *Mobile Agent Integrity for Electronic Commerce Applications*. Information Systems, v. 24, n. 6, 1999.

CUNHA JÚNIOR, Eurípedes Brito. *Os Contratos Eletrônicos e o Novo Código Civil*. Conferência proferida no “Congresso Internacional de Direito e Tecnologias da Informação”, realizado pelo Centro de Estudos Judiciários, nos dias 03 e 04 de Outubro de 2002, no Auditório do Superior Tribunal de Justiça. Brasília-DF.

FARIA, Carla Gomes de. *Uma Análise da Web Semântica e suas implicações no acesso à Informação*. Capturado em 15 de junho de 2003. On-Line. Disponível na Internet em: <http://maae.deinf.ufma.br/Ensino/IA/Artigos/UmaAnalisedaWebSemanticaesuasimplificacoesnoacessoainformacao>.

FONSECA, Luís Carlos Costa. *Sistemas Multiagentes para Negociação no Ambiente ICS de Comércio Eletrônico*. Dissertação (Mestrado em Ciência da Computação) –

Curso de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís-Ma, 2003.

GONÇALVES, Carlos Alberto. *Direito das Obrigações: parte especial, volume 6, tomo I: Contratos*. Saraiva, 2002.

GRUBER, Thomas R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. International Journal of Human and Computer Studies, 907-928.

GUARINO, Nichola. *Formal Ontology and Information Systems*. In N. Guarino, Editor, Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy, pages 3--15. IOS Press, June 1998.

HENDLER, James. *Agents and the Semantic Web*. Disponível em: <http://www.cs.umd.edu/users/hendler/agentweb.html>, acesso em maio de 2003.

JENNINGS, Nicholas R.; WOOLDRIDGE, Michael. J. *Applications of Intelligent Agents*. In Agent Technologies: Foundations, Applications and Markets. N. R. Jennings and M. J. Wooldridge (Eds.) (To appear) 1998.

LABIDI, Sofiane et al. *Intelligent B2B Commerce System*. In: Innovatex/Formatex. (Org.). Techno-Legal Aspects of Information Society and New Economy: An Overview. Badajoz, 2003, V. I.

LASSILA, Ora; SWICK, Ralph R. *Resource Description Framework (RDF) Model and Syntax Specification*. Capturado em 26 de Janeiro de 2002. On-Line. Disponível na Internet em: <http://www.w3c.org/tr/1999/rec-rdf-syntax-19990222>.

LUDWIG, Heiko; GISLER, Michael. *Electronic Contracts and Contract-Based Computing*. Invited Tutorial at the IFIP I3E Conference, Zurich, October 2001.

OLIVEIRA, Nathália et al. *Formation and Fulfillment of Electronic Contracts in ICS*. To appear in the Proceedings of 6<sup>th</sup> International Conference on Enterprise Information Systems, Porto – Portugal, April, 2004.

NOY, Natalya F et al. *Creating Semantic Web Contents with Protégé-2000*. IEEE Intelligent Systems 48(2):60-71, 2001.

RODRIGUES, Silvio. *Direito Civil*. Saraiva, 2002.

SHAN, Ming-Chien. *FlowJet: Internet-Based E-Service Process Management*. Proceedings of Introduction Process Technology Workshop, Villard de Lans, France, 1999.

Tomaz, Ricardo Ferraz. *Uma Arquitetura Baseada em Web Services Semânticos para Agrupamento dos Agentes Negociantes no Ambiente ICS de Comércio Eletrônico*. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís-Ma, 2003.

*Workflow Management Coalition – Terminology and Glossary*. The Workflow Management Coalition Specification, Fevereiro, 1999.

## APÊNDICES



## APÊNDICE A – Template de um contrato de Compra e Venda codificado em XML

```
<?xml version="1.0" encoding="UTF-8"?>
<contrato>
  <titulo>Contrato de Compra e Venda</titulo>
  <identificacao_partes>
    <p1>Pelo presente instrumento particular de Promessa de Compra e Venda que fazem entre
si, de um lado, doravante denominado PROMITENTE VENDEDOR,</p1>
    <nome_p1/>
    <codigo_p1/>
    <p2>estabelecido(a) à </p2>
    <endereco_p1/>
    <p3>, neste ato representado conforme seu Estatuto Social e, de outro lado, doravante
denominado PROMISSARIO COMPRADOR </p3>
    <nome_p2/>
    <codigo_p2/>
    <p4>, estabelecido(a) à , </p4>
    <endereco_p2/>
    <p5>, neste ato representado conforme seu Estatuto Social, tem justo e convencionado o que
se segue, conforme cláusulas e condições abaixo descritas:</p5>
  </identificacao_partes>
  <capitulos>
    <capitulol>
      <nome_capl>Capítulo I - Do Objeto</nome_capl>
      <clausulal.cl>
        <pcl.I1>Cláusula Primeira: O PROMITENTE VENDEDOR é senhor e legítimo possuidor
do objeto deste Instrumento, a seguir descrito e caracterizado:
          Objeto:</pcl.I1>
        <objeto/>
      </clausulal.cl>
    </capitulol>
    <capituloll>
      <nome_capll>Capítulo II - Do Valor e da Forma de Pagamento</nome_capll>
      <clausulal.cll>
        <pcl.II1>Cláusula Primeira: Pelo presente instrumento e na melhor forma de direito, o
PROMITENTE VENDEDOR, se compromete a vender ao PROMISSÁRIO COMPRADOR, e este a
adquirir do PROMITENTE VENDEDOR, o objeto acima descrito, pelo valor de</pcl.II1>
        <preco/>
        <preco_escrito/>
        <pcl.II2>, a ser pago nas seguintes condições:</pcl.II2>
        <cond_pagamento/>
      </clausulal.cll>
      <clausulall.cll>
        <pcl.II1>Cláusula Segunda: Fica ajustada a multa equivalente a</pcl.II1>
        <valor_multa/>
        <multa_escrito/>
        <pcl.II2>a ser aplicada sobre o atraso injustificado no pagamento da(s) parcela(s) do
saldo devedor, a ser paga pelo PROMISSÁRIO COMPRADOR ao PROMITENTE VENDEDOR, mais
juros de </pcl.II2>
        <valor_juros/>
        <juros_escrito/>
        <pcl.II3>ao mês e atualização monetária na forma legal.</pcl.II3>
      </clausulall.cll>
    </capituloll>
    <capitulolll>
      <nome_caplll>Capítulo III - Do Prazo de Entrega</nome_caplll>
      <clausulal.clll>
        <pcl.III1>Cláusula Primeira: O prazo de entrega será efetuado dentro de </pcl.III1>
```

<n\_dias\_entrega/>  
<pcl.III2>dias, contados a partir da data de assinatura do presente Contrato, salvo por motivo de caso fortuito ou força maior.</pcl.III2>  
</clausulal.cIII>  
</capituloIII>  
<capituloIV>  
<nome\_capIV>Capítulo IV - Das Obrigações</nome\_capIV>  
<clausulal.cIV>  
<pcl.IV1>Cláusula Primeira: São obrigações do PROMITENTE VENDEDOR:</pcl.IV1>  
<obrigacoes\_p1/>  
</clausulal.cIV>  
<clausulall.cIV>  
<pcl.IV1>Cláusula Segunda: São obrigações do PROMISSÁRIO  
COMPRADOR:</pcl.IV1>  
<obrigacoes\_p2/>  
</clausulall.cIV>  
</capituloIV>  
<capituloV>  
<nome\_capV>Capítulo V - Das Responsabilidades</nome\_capV>  
<clausulal.cV>  
<pcl.V1>Cláusula Primeira: São responsabilidades do PROMITENTE  
VENDEDOR:</pcl.V1>  
<responsabilidades\_p1/>  
</clausulal.cV>  
<clausulall.cV>  
<pcl.V1>Cláusula Segunda: São responsabilidades do PROMISSÁRIO  
COMPRADOR:</pcl.V1>  
<responsabilidades\_p2/>  
</clausulall.cV>  
</capituloV>  
</capitulos>  
<disposicoes\_gerais>  
<pd1>Fica eleito o Foro da Comarca de </pd1>  
<foro/>  
<pd2>, com renúncia a qualquer outro por mais privilegiado que seja, para nele serem dirimidas dúvidas ou questões oriundas do presente Contrato.  
E por estarem de acordo, assinam as partes o presente Instrumento em 02 (duas) vias de igual teor e forma, na presença das testemunhas abaixo:</pd2>  
<local/>  
<data/>  
<nome\_p1/>  
<nome\_p2/>  
<pd3>Testemunhas:</pd3>  
<test\_p1/>  
<cod\_test\_p1/>  
<test\_p2/>  
<cod\_test\_p2/>  
</disposicoes\_gerais>  
</contrato>

## APÊNDICE B – Código dos agentes negociadores e de contrato

### \* Agente de contrato:

```
package examples.Contrato;
import javax.swing.JOptionPane;
import jade.core.*;
import jade.core.AID;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.DFService;
import jade.domain.FIPAException;

public class Contrato extends Agent {
    class EnviaMensagemParâmetros extends SimpleBehaviour {
        private boolean finished = false;
        String nome_agente = "ANegociador";
        AID id = new AID(nome_agente, AID.ISLOCALNAME);
        public EnviaMensagemParâmetros (Agent a) {
            super(a);
        }

        public void action() {
            ACLMessage msg = myAgent.receive();
            if(msg != null){
                if(msg.getPerformative() == ACLMessage.NOT_UNDERSTOOD)
                    JOptionPane.showMessageDialog(null,"Recebeu uma performativa NOT-
UNDERSTOOD");
                else{
                    ACLMessage reply = msg.createReply();
                    // Recebeu mensagem de acordo fechado
                    if(msg.getPerformative()== ACLMessage.INFORM){
                        String content = msg.getContent();
                        if ((content != null) && (content.indexOf("acordo fechado") != -1)){
                            // Interage com usuário
                            JOptionPane.showMessageDialog(null,"Confirmação de acordo fechado");
                            reply.setPerformative(ACLMessage.REQUEST);
                            reply.setContent("Enviar parametros");
                            myAgent.send(reply);
                            String p1 = JOptionPane.showInputDialog("Digite o parâmetro objeto");
                            String p2 = JOptionPane.showInputDialog("Digite o parâmetro preço");
                            String p3 = JOptionPane.showInputDialog("Digite o parâmetro preço por escrito");
                            String p4 = JOptionPane.showInputDialog("Digite o parâmetro descrição do
preço");
                            String p5 = JOptionPane.showInputDialog("Digite o parâmetro condições de
pagamento");
                            String p6 = JOptionPane.showInputDialog("Digite o parâmetro multa");
                            String p7 = JOptionPane.showInputDialog("Digite o parâmetro valor da multa por
escrito");
                            String p8 = JOptionPane.showInputDialog("Digite o parâmetro juros");
                            String p9 = JOptionPane.showInputDialog("Digite o parâmetro valor dos juros por
escrito");
                            String p10 = JOptionPane.showInputDialog("Digite o parâmetro dias de entrega");
                            String p11 = JOptionPane.showInputDialog("Digite o parâmetro obrigações");
                            String p12 = JOptionPane.showInputDialog("Digite o parâmetro
responsabilidades ");
                            String p13 = JOptionPane.showInputDialog("Digite o parâmetro foro");
```

```

    }
  }
  else{
    // Recebeu uma performativa incorreta
    reply.setPerformative(ACLMessage.NOT_UNDERSTOOD);
    reply.setContent("(Unexpected-act
"+ACLMessage.getPerformative(msg.getPerformative()+")
    JOptionPane.showMessageDialog(null,"Recebeu uma Performativa incorreta!");
  }
}
else{
  block();
}
}

public boolean done() {
  return finished;
}
} //fim da classe EnviamensagemParâmetros

```

```

protected void setup() {

  /** Registro no DF */
  DFAgentDescription dfd = new DFAgentDescription();
  ServiceDescription sd = new ServiceDescription();
  sd.setType("Agente de Contrato");
  sd.setName(getName());
  sd.setOwnership("ddf");
  sd.addOntologies("ContratosOntology");
  dfd.setName(getAID());
  dfd.addServices(sd);
  try {
    DFService.register(this,dfd);
  } catch (FIPAException e) {
    System.err.println(getLocalName()+" Registro no DF sem sucesso. Razão:
"+e.getMessage());
    doDelete();
  }

  EnviamensagemParâmetros PingBehaviour = new EnviamensagemParâmetros(this);
  addBehaviour(PingBehaviour);
}

```

```

} //fim da classe Contrato

```

### \* Agente negociador

```

package examples.Contrato;
import jade.core.Agent;
import jade.core.AID;
import jade.core.behaviours.*;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import javax.swing.JOptionPane;
public class Negociador extends Agent {

```

```

class EnviaMensagemParâmetros extends SimpleBehaviour {

    private boolean finished = false;
    String nome_agente = "AContrato";
    AID id = new AID(nome_agente, AID.ISLOCALNAME);

    public EnviaMensagemParâmetros(Agent a) {
        super(a);
    }

    public void action() {
        ACLMessage msg = blockingReceive();
        if(msg != null){
            if(msg.getPerformative() == ACLMessage.NOT_UNDERSTOOD)
                JOptionPane.showMessageDialog(null,"Recebeu uma performativa NOT-UNDERSTOOD");
            else{
                ACLMessage reply = msg.createReply();
                if(msg.getPerformative() == ACLMessage.REQUEST){
                    String content = msg.getContent();
                    if ((content != null) && (content.indexOf("Enviar parâmetros") != -1)){
                        reply.setPerformative(ACLMessage.INFORM);
                        reply.setContent("Parâmetros enviados com sucesso");
                        myAgent.send(reply);
                    }
                }
            }
        }
    }

    public boolean done() {
        return finished;
    }
} //fim da classe EnviaMensagemParâmetros

protected void setup() {

    // Registro no DF
    DFAgentDescription dfd = new DFAgentDescription();
    ServiceDescription sd = new ServiceDescription();
    sd.setType("AgenteNegociador");
    sd.setName(getName());
    sd.setOwnership("ddf");
    sd.addOntologies("ContratosOntology");
    dfd.setName(getAID());
    dfd.addServices(sd);
    try {
        DFService.register(this,dfd);
    } catch (FIPAException e) {
        System.err.println(getLocalName()+" Registro no DF sem sucesso. Razão: "+e.getMessage());
        doDelete();
    }
    EnviaMensagemParâmetros PingBehaviour = new EnviaMensagemParâmetros(this);
    addBehaviour(PingBehaviour);
}
} // fim da classe Negociador

```

## APÊNDICE C – Código da ontologia dos contratos

// file: ContratosOntology.java generated by ontology bean generator.

```
package contratos.ontology;
import jade.content.onto.*;
import jade.content.schema.*;
import jade.util.leap.HashMap;
import jade.content.lang.Codec;
import jade.core.CaseInsensitiveString;
import nl.uva.psy.swi.beangenerator.ProtegeIntrospector;
import nl.uva.psy.swi.beangenerator.SlotHolder;
import nl.uva.psy.swi.beangenerator.ProtegeTools;

/** file: ContratosOntology.java @author ontology bean generator @version 2004/01/22, 00:25:16 */

public class ContratosOntology extends jade.content.onto.Ontology implements
ProtegeTools.ProtegeOntology {
    /** * These hashmap store a mapping from jade names to either protege names of SlotHolder
containing the protege names. And vice versa */

private HashMap jadeToProtege;
    //NAME
    public static final String ONTOLOGY_NAME = "Contratos";
    // The singleton instance of this ontology
    private static ProtegeIntrospector introspect = new ProtegeIntrospector();
    private static Ontology theInstance = new ContratosOntology();
    public static Ontology getInstance() {
        return theInstance;
    }
    // ProtegeOntology methods
    public SlotHolder getSlotNameFromJADENAME(SlotHolder jadeSlot) {
        return (SlotHolder) jadeToProtege.get(jadeSlot);
    }
    // storing the information
    private void storeSlotName(String jadeName, String javaClassName, String slotName){
        jadeToProtege.put(new SlotHolder(javaClassName, jadeName), new SlotHolder(javaClassName,
slotName));
    }
    // VOCABULARY
    public static final String
LVI.CLAUSULA_SEGUNDA_RESPONSABILIDADES_P2="responsabilidades_p2";
    public static final String LVI.CLAUSULA_SEGUNDA="LVI.Clausula Segunda";
    public static final String P.CAPITULO_V__DAS_OBRIGACOES="P.Capitulo V - Das Obrigacoes";
    public static final String TIII.CLAUSULA_PRIMEIRA_N_DIAS_ENTREGA="n_dias_entrega";
    public static final String TIII.CLAUSULA_PRIMEIRA="TIII.Clausula Primeira";
    public static final String P.CAPITULO_III__DA_VIGENCIA="P.Capitulo III - Da Vigencia";
    public static final String LII.CLAUSULA_PRIMEIRA_TIPO_LOCACAO="tipo_locacao";
    public static final String LII.CLAUSULA_PRIMEIRA_DESCRICAO_PRECO="descricao_preco";
    public static final String LII.CLAUSULA_PRIMEIRA_PRECO="preco";
    public static final String LII.CLAUSULA_PRIMEIRA_TIPO_REAJUSTE="tipo_reajuste";
    public static final String LII.CLAUSULA_PRIMEIRA_LOCAL_PAGAMENTO="local_pagamento";
    public static final String LII.CLAUSULA_PRIMEIRA_DIA_PAGAMENTO="dia_pagamento";
    public static final String LII.CLAUSULA_PRIMEIRA_ORGAO_REAJUSTE="orgao_reajuste";
    public static final String LII.CLAUSULA_PRIMEIRA="LII.Clausula Primeira";
    public static final String P.CAPITULO_VI__DAS_RESPONSABILIDADES="P.Capitulo VI - Das
Responsabilidades";
    public static final String T.CAPITULO_III__DO_PRAZO_DE_ENTREGA="T.Capitulo III - Do Prazo
de Entrega";
    public static final String PIII.CLAUSULA_PRIMEIRA_TIPO_VIGENCIA="tipo_vigencia";
```

```

public static final String PIII.CLAUSULA_PRIMEIRA="PIII.Clausula Primeira";
public static final String L.CAPITULO_IV__DA_RESCISAO="L.Capitulo IV - Da Rescisao";
public static final String C.CAPITULO_V__DAS_RESPONSABILIDADES="C.Capitulo V - Das
Responsabilidades";
public static final String LOCACAO="Locacao";
public static final String PIV.CLAUSULA_PRIMEIRA_N_DIAS_RESCISAO="n_dias_rescisao";
public static final String PIV.CLAUSULA_PRIMEIRA="PIV.Clausula Primeira";
public static final String T.CAPITULO_I__DO_OBJETO="T.Capitulo I - Do Objeto";
public static final String TII.CLAUSULA_PRIMEIRA_COND_TROCA="cond_troca";
public static final String TII.CLAUSULA_PRIMEIRA="TII.Clausula Primeira";
public static final String CONTRATO="Contrato";
public static final String TIV.CLAUSULA_PRIMEIRA_OBRIGACOES_P1="obrigacoes_p1";
public static final String TIV.CLAUSULA_PRIMEIRA="TIV.Clausula Primeira";
public static final String
P.CAPITULO_II__DO_VALOR_E_DA_FORMA_DE_PAGAMENTO="P.Capitulo II - Do Valor e da
Forma de Pagamento";
public static final String COMPRA_E_VENDA="Compra e Venda";
public static final String PII.CLAUSULA_SEGUNDA_JUROS="juros";
public static final String PII.CLAUSULA_SEGUNDA_DESCRICAO_MULTA="descricao_multa";
public static final String PII.CLAUSULA_SEGUNDA_MULTA="multa";
public static final String PII.CLAUSULA_SEGUNDA_ORGAO_REAJUSTE="orgao_reajuste";
public static final String PII.CLAUSULA_SEGUNDA="PII.Clausula Segunda";
public static final String CI.CLAUSULA_PRIMEIRA_OBJETO="objeto";
public static final String CI.CLAUSULA_PRIMEIRA="CI.Clausula Primeira";
public static final String P.CAPITULO_I__DO_OBJETO="P.Capitulo I - Do Objeto";
public static final String LV.CLAUSULA_PRIMEIRA_OBRIGACOES_P1="obrigacoes_p1";
public static final String LV.CLAUSULA_PRIMEIRA="LV.Clausula Primeira";
public static final String C.CAPITULO_I__DO_OBJETO="C.Capitulo I - Do Objeto";
public static final String T.CAPITULO_V__DAS_RESPONSABILIDADES="T.Capitulo V - Das
Responsabilidades";
public static final String T.CAPITULO_IV__DAS_OBRIGACOES="T.Capitulo IV - Das Obrigacoes";
public static final String TROCA="Troca";
public static final String IDENTIFICACAO_PARTES_NOME_P2="nome_p2";
public static final String IDENTIFICACAO_PARTES_NOME_P1="nome_p1";
public static final String IDENTIFICACAO_PARTES_ENDERECO_P1="endereco_p1";
public static final String IDENTIFICACAO_PARTES_ENDERECO_P2="endereco_p2";
public static final String IDENTIFICACAO_PARTES_CODIGO_P2="codigo_p2";
public static final String IDENTIFICACAO_PARTES_CODIGO_P1="codigo_p1";
public static final String IDENTIFICACAO_PARTES="Identificacao_Partес";
public static final String PII.CLAUSULA_PRIMEIRA_DESCRICAO_PRECO="descricao_preco";
public static final String PII.CLAUSULA_PRIMEIRA_PRECO="preco";
public static final String PII.CLAUSULA_PRIMEIRA_TIPO_REAJUSTE="tipo_reajuste";
public static final String PII.CLAUSULA_PRIMEIRA_LOCAL_PAGAMENTO="local_pagamento";
public static final String PII.CLAUSULA_PRIMEIRA_DIA_PAGAMENTO="dia_pagamento";
public static final String PII.CLAUSULA_PRIMEIRA_ORGAO_REAJUSTE="orgao_reajuste";
public static final String PII.CLAUSULA_PRIMEIRA_COND_PAGAMENTO="cond_pagamento";
public static final String PII.CLAUSULA_PRIMEIRA="PII.Clausula Primeira";
public static final String CII.CLAUSULA_PRIMEIRA_DESCRICAO_PRECO="descricao_preco";
public static final String CII.CLAUSULA_PRIMEIRA_PRECO="preco";
public static final String CII.CLAUSULA_PRIMEIRA_COND_PAGAMENTO="cond_pagamento";
public static final String CII.CLAUSULA_PRIMEIRA="CII.Clausula Primeira";
public static final String PV.CLAUSULA_SEGUNDA_OBRIGACOES_P2="obrigacoes_p2";
public static final String PV.CLAUSULA_SEGUNDA="PV.Clausula Segunda";
public static final String CIV._CLAUSULA_SEGUNDA_OBRIGACOES_P2="obrigacoes_p2";
public static final String CIV._CLAUSULA_SEGUNDA="CIV. Clausula Segunda";
public static final String
TV.CLAUSULA_SEGUNDA_RESPONSABILIDADES_P2="responsabilidades_p2";
public static final String TV.CLAUSULA_SEGUNDA="TV.Clausula Segunda";
public static final String
CV.CLAUSULA_SEGUNDA_RESPONSABILIDADES_P2="responsabilidades_p2";

```



```

    public static final String CV.CLAUSULA_SEGUNDA="CV.Clausula Segunda";
    public static final String
PVI.CLAUSULA_PRIMEIRA_RESPONSABILIDADES_P1="responsabilidades_p1";
    public static final String PVI.CLAUSULA_PRIMEIRA="PVI.Clausula Primeira";
    public static final String LIII.CLAUSULA_PRIMEIRA_TIPO_VIGENCIA="tipo_vigencia";
    public static final String LIII.CLAUSULA_PRIMEIRA="LIII.Clausula Primeira";
    public static final String LI.CLAUSULA_SEGUNDA="LI.Clausula Segunda";
    public static final String
C.CAPITULO_II_DO_VALOR_E_DA_FORMA_DE_PAGAMENTO="C.Capitulo II - Do Valor e da
Forma de Pagamento";
    public static final String
TV.CLAUSULA_PRIMEIRA_RESPONSABILIDADES_P1="responsabilidades_p1";
    public static final String TV.CLAUSULA_PRIMEIRA="TV.Clausula Primeira";
    public static final String DISPOSICAO_GERAL_FORO="foro";
    public static final String DISPOSICAO_GERAL_TEST_P1="test_p1";
    public static final String DISPOSICAO_GERAL_NOME_P2="nome_p2";
    public static final String DISPOSICAO_GERAL_TEST_P2="test_p2";
    public static final String DISPOSICAO_GERAL_DATA="data";
    public static final String DISPOSICAO_GERAL_NOME_P1="nome_p1";
    public static final String DISPOSICAO_GERAL_COD_TEST_P1="cod_test_p1";
    public static final String DISPOSICAO_GERAL_COD_TEST_P2="cod_test_p2";
    public static final String DISPOSICAO_GERAL_LOCAL="local";
    public static final String DISPOSICAO_GERAL="Disposicao_Geral";
    public static final String L.CAPITULO_I_DO_OBJETO="L.Capitulo I - Do Objeto";
    public static final String
LVI.CLAUSULA_PRIMEIRA_RESPONSABILIDADES_P1="responsabilidades_p1";
    public static final String LVI.CLAUSULA_PRIMEIRA="LVI.Clausula Primeira";
    public static final String
T.CAPITULO_II_DO_VALOR_E_DA_FORMA_DE_PAGAMENTO="T.Capitulo II - Do Valor e da
Forma de Pagamento";
    public static final String P.CAPITULO_IV_DA_RESCISAO="P.Capitulo IV - Da Rescisao";
    public static final String
L.CAPITULO_II_DO_VALOR_E_DA_FORMA_DE_PAGAMENTO="L.Capitulo II - Do Valor e da
Forma de Pagamento";
    public static final String LI.CLAUSULA_PRIMEIRA_OBJETO="objeto";
    public static final String LI.CLAUSULA_PRIMEIRA="LI.Clausula Primeira";
    public static final String
CV.CLAUSULA_PRIMEIRA_RESPONSABILIDADES_P1="responsabilidades_p1";
    public static final String CV.CLAUSULA_PRIMEIRA="CV.Clausula Primeira";
    public static final String CIII.CLAUSULA_PRIMEIRA_N_DIAS_ENTREGA="n_dias_entrega";
    public static final String CIII.CLAUSULA_PRIMEIRA="CIII.Clausula Primeira";
    public static final String LIV.CLAUSULA_PRIMEIRA_N_DIAS_RESCISAO="n_dias_rescisao";
    public static final String LIV.CLAUSULA_PRIMEIRA="LIV.Clausula Primeira";
    public static final String TIV.CLAUSULA_SEGUNDA_OBRIGACOES_P2="obrigacoes_p2";
    public static final String TIV.CLAUSULA_SEGUNDA="TIV.Clausula Segunda";
    public static final String TII.CLAUSULA_SEGUNDA_JUROS="juros";
    public static final String TII.CLAUSULA_SEGUNDA_DESCRICAO_MULTA="descricao_multa";
    public static final String TII.CLAUSULA_SEGUNDA_MULTA="multa";
    public static final String TII.CLAUSULA_SEGUNDA_DESCRICAO_JUROS="descricao_juros";
    public static final String TII.CLAUSULA_SEGUNDA="TII.Clausula Segunda";
    public static final String PV.CLAUSULA_PRIMEIRA_OBRIGACOES_P1="obrigacoes_p1";
    public static final String PV.CLAUSULA_PRIMEIRA="PV.Clausula Primeira";
    public static final String L.CAPITULO_III_DA_VIGENCIA="L.Capitulo III - Da Vigencia";
    public static final String C.CAPITULO_IV_DAS_OBRIGACOES="C.Capitulo IV - Das Obrigacoes";
    public static final String TI.CLAUSULA_PRIMEIRA_OBJETO1="objeto1";
    public static final String TI.CLAUSULA_PRIMEIRA_OBJETO2="objeto2";
    public static final String TI.CLAUSULA_PRIMEIRA="TI.Clausula Primeira";
    public static final String L.CAPITULO_V_DAS_OBRIGACOES="L.Capitulo V - Das Obrigacoes";
    public static final String LII.CLAUSULA_SEGUNDA_JUROS="juros";
    public static final String LII.CLAUSULA_SEGUNDA_DESCRICAO_MULTA="descricao_multa";

```



```

public static final String LII.CLAUSULA_SEGUNDA_MULTA="multa";
public static final String LII.CLAUSULA_SEGUNDA_ORGAO_REAJUSTE="orgao_reajuste";
public static final String LII.CLAUSULA_SEGUNDA="LII.Clausula Segunda";
public static final String CII.CLAUSULA_SEGUNDA_JUROS="juros";
public static final String CII.CLAUSULA_SEGUNDA_DESCRICAO_MULTA="descricao_multa";
public static final String CII.CLAUSULA_SEGUNDA_MULTA="multa";
public static final String CII.CLAUSULA_SEGUNDA_DESCRICAO_JUROS="descricao_juros";
public static final String CII.CLAUSULA_SEGUNDA="CII.Clausula Segunda";
public static final String PI.CLAUSULA_PRIMEIRA_OBJETO="objeto";
public static final String PI.CLAUSULA_PRIMEIRA="PI.Clausula Primeira";
public static final String LV.CLAUSULA_SEGUNDA_OBRIGACOES_P2="obrigacoes_p2";
public static final String LV.CLAUSULA_SEGUNDA="LV.Clausula Segunda";
public static final String ESPECIES="Especies";
public static final String CIV.CLAUSULA_PRIMEIRA_OBRIGACOES_P1="obrigacoes_p1";
public static final String CIV.CLAUSULA_PRIMEIRA="CIV.Clausula Primeira";
public static final String PRESTACAO_DE_SERVICO="Prestacao de Servico";
public static final String L.CAPITULO_VI_DAS_RESPONSABILIDADES="L.Capitulo VI - Das
Responsabilidades";
public static final String
PVI.CLAUSULA_SEGUNDA_RESPONSABILIDADES_P2="responsabilidades_p2";
public static final String PVI.CLAUSULA_SEGUNDA="PVI.Clausula Segunda";
public static final String C.CAPITULO_III_DO_PRAZO_DE_ENTREGA="C.Capitulo III - Do Prazo
de Entrega";
/** * Constructor */
private ContratosOntology(){
super(ONTOLOGY_NAME, BasicOntology.getInstance());
introspect.setOntology(this);
jadeToProtege = new HashMap();
try {
// adding Concept(s)
ConceptSchema c.Capitulo_III_Do_Prazo_de_EntregaSchema = new
ConceptSchema(C.CAPITULO_III_DO_PRAZO_DE_ENTREGA);
add(c.Capitulo_III_Do_Prazo_de_EntregaSchema,
contratos.ontology.C.Capitulo_III_Do_Prazo_de_Entrega.class);
ConceptSchema pvl.Clausula_SegundaSchema = new
ConceptSchema(PVI.CLAUSULA_SEGUNDA);
add(pvl.Clausula_SegundaSchema, contratos.ontology.PVI.Clausula_Segunda.class);
ConceptSchema l.Capitulo_VI_Das_ResponsabilidadesSchema = new
ConceptSchema(L.CAPITULO_VI_DAS_RESPONSABILIDADES);
add(l.Capitulo_VI_Das_ResponsabilidadesSchema,
contratos.ontology.L.Capitulo_VI_Das_Responsabilidades.class);
ConceptSchema prestacao_de_ServicoSchema = new
ConceptSchema(PRESTACAO_DE_SERVICO);
add(prestacao_de_ServicoSchema, contratos.ontology.Prestacao_de_Servico.class);
ConceptSchema civ.Clausula_PrimeiraSchema = new
ConceptSchema(CIV.CLAUSULA_PRIMEIRA);
add(civ.Clausula_PrimeiraSchema, contratos.ontology.CIV.Clausula_Primeira.class);
ConceptSchema especiesSchema = new ConceptSchema(ESPECIES);
add(especiesSchema, contratos.ontology.Especies.class);
ConceptSchema iv.Clausula_SegundaSchema = new
ConceptSchema(LV.CLAUSULA_SEGUNDA);
add(iv.Clausula_SegundaSchema, contratos.ontology.LV.Clausula_Segunda.class);
ConceptSchema pl.Clausula_PrimeiraSchema = new ConceptSchema(PI.CLAUSULA_PRIMEIRA);
add(pl.Clausula_PrimeiraSchema, contratos.ontology.PI.Clausula_Primeira.class);
ConceptSchema cil.Clausula_SegundaSchema = new
ConceptSchema(CII.CLAUSULA_SEGUNDA);
add(cil.Clausula_SegundaSchema, contratos.ontology.CII.Clausula_Segunda.class);
ConceptSchema lil.Clausula_SegundaSchema = new
ConceptSchema(LII.CLAUSULA_SEGUNDA);
add(lil.Clausula_SegundaSchema, contratos.ontology.LII.Clausula_Segunda.class);

```

```

ConceptSchema l.Capitulo_V__Das_ObrigacoesSchema = new
ConceptSchema(L.CAPITULO_V__DAS_OBRIGACOES);
add(l.Capitulo_V__Das_ObrigacoesSchema,
contratos.ontology.L.Capitulo_V__Das_Obrigacoes.class);
ConceptSchema tl.Clausula_PrimeiraSchema = new ConceptSchema(TI.CLAUSULA_PRIMEIRA);
add(tl.Clausula_PrimeiraSchema, contratos.ontology.TI.Clausula_Primeira.class);
ConceptSchema c.Capitulo_IV__Das_ObrigacoesSchema = new
ConceptSchema(C.CAPITULO_IV__DAS_OBRIGACOES);
add(c.Capitulo_IV__Das_ObrigacoesSchema,
contratos.ontology.C.Capitulo_IV__Das_Obrigacoes.class);
ConceptSchema l.Capitulo_III__Da_VigenciaSchema = new
ConceptSchema(L.CAPITULO_III__DA_VIGENCIA);
add(l.Capitulo_III__Da_VigenciaSchema, contratos.ontology.L.Capitulo_III__Da_Vigencia.class);
ConceptSchema pV.Clausula_PrimeiraSchema = new
ConceptSchema(PV.CLAUSULA_PRIMEIRA);
add(pV.Clausula_PrimeiraSchema, contratos.ontology.PV.Clausula_Primeira.class);
ConceptSchema til.Clausula_SegundaSchema = new
ConceptSchema(TII.CLAUSULA_SEGUNDA);
add(til.Clausula_SegundaSchema, contratos.ontology.TII.Clausula_Segunda.class);
ConceptSchema tiV.Clausula_SegundaSchema = new
ConceptSchema(TIV.CLAUSULA_SEGUNDA);
add(tiV.Clausula_SegundaSchema, contratos.ontology.TIV.Clausula_Segunda.class);
ConceptSchema liV.Clausula_PrimeiraSchema = new
ConceptSchema(LIV.CLAUSULA_PRIMEIRA);
add(liV.Clausula_PrimeiraSchema, contratos.ontology.LIV.Clausula_Primeira.class);
ConceptSchema ciil.Clausula_PrimeiraSchema = new
ConceptSchema(CIII.CLAUSULA_PRIMEIRA);
add(ciil.Clausula_PrimeiraSchema, contratos.ontology.CIII.Clausula_Primeira.class);
ConceptSchema cV.Clausula_PrimeiraSchema = new
ConceptSchema(CV.CLAUSULA_PRIMEIRA);
add(cV.Clausula_PrimeiraSchema, contratos.ontology.CV.Clausula_Primeira.class);
ConceptSchema ll.Clausula_PrimeiraSchema = new ConceptSchema(LI.CLAUSULA_PRIMEIRA);
add(ll.Clausula_PrimeiraSchema, contratos.ontology.LI.Clausula_Primeira.class);
ConceptSchema l.Capitulo_II__Do_Valor_e_da_Forma_de_PagamentoSchema = new
ConceptSchema(L.CAPITULO_II__DO_VALOR_E_DA_FORMA_DE_PAGAMENTO);
add(l.Capitulo_II__Do_Valor_e_da_Forma_de_PagamentoSchema,
contratos.ontology.L.Capitulo_II__Do_Valor_e_da_Forma_de_Pagamento.class);
ConceptSchema p.Capitulo_IV__Da_RescisaoSchema = new
ConceptSchema(P.CAPITULO_IV__DA_RESCISAO);
add(p.Capitulo_IV__Da_RescisaoSchema, contratos.ontology.P.Capitulo_IV__Da_Rescisao.class);
ConceptSchema t.Capitulo_II__Do_Valor_e_da_Forma_de_PagamentoSchema = new
ConceptSchema(T.CAPITULO_II__DO_VALOR_E_DA_FORMA_DE_PAGAMENTO);
add(t.Capitulo_II__Do_Valor_e_da_Forma_de_PagamentoSchema,
contratos.ontology.T.Capitulo_II__Do_Valor_e_da_Forma_de_Pagamento.class);
ConceptSchema lVI.Clausula_PrimeiraSchema = new
ConceptSchema(LVI.CLAUSULA_PRIMEIRA);
add(lVI.Clausula_PrimeiraSchema, contratos.ontology.LVI.Clausula_Primeira.class);
ConceptSchema l.Capitulo_I__Do_ObjetoSchema = new
ConceptSchema(L.CAPITULO_I__DO_OBJETO);
add(l.Capitulo_I__Do_ObjetoSchema, contratos.ontology.L.Capitulo_I__Do_Objeto.class);
ConceptSchema disposicao_GeralSchema = new ConceptSchema(DISPOSICAO_GERAL);
add(disposicao_GeralSchema, contratos.ontology.Disposicao_Geral.class);
ConceptSchema tV.Clausula_PrimeiraSchema = new
ConceptSchema(TV.CLAUSULA_PRIMEIRA);
add(tV.Clausula_PrimeiraSchema, contratos.ontology.TV.Clausula_Primeira.class);
ConceptSchema c.Capitulo_II__Do_Valor_e_da_Forma_de_PagamentoSchema = new
ConceptSchema(C.CAPITULO_II__DO_VALOR_E_DA_FORMA_DE_PAGAMENTO);
add(c.Capitulo_II__Do_Valor_e_da_Forma_de_PagamentoSchema,
contratos.ontology.C.Capitulo_II__Do_Valor_e_da_Forma_de_Pagamento.class);
ConceptSchema ll.Clausula_SegundaSchema = new ConceptSchema(LI.CLAUSULA_SEGUNDA);

```

```

add(II.Clausula_SegundaSchema, contratos.ontology.II.Clausula_Segunda.class);
ConceptSchema liil.Clausula_PrimeiraSchema = new
ConceptSchema(LIII.CLAUSULA_PRIMEIRA);
add(liil.Clausula_PrimeiraSchema, contratos.ontology.LIII.Clausula_Primeira.class);
ConceptSchema pvl.Clausula_PrimeiraSchema = new
ConceptSchema(PVI.CLAUSULA_PRIMEIRA);
add(pvl.Clausula_PrimeiraSchema, contratos.ontology.PVI.Clausula_Primeira.class);
ConceptSchema cV.Clausula_SegundaSchema = new
ConceptSchema(CV.CLAUSULA_SEGUNDA);
add(cV.Clausula_SegundaSchema, contratos.ontology.CV.Clausula_Segunda.class);
ConceptSchema tV.Clausula_SegundaSchema = new
ConceptSchema(TV.CLAUSULA_SEGUNDA);
add(tV.Clausula_SegundaSchema, contratos.ontology.TV.Clausula_Segunda.class);
ConceptSchema ciV.Clausula_SegundaSchema = new
ConceptSchema(CIV.CLAUSULA_SEGUNDA);
add(ciV.Clausula_SegundaSchema, contratos.ontology.CIV.Clausula_Segunda.class);
ConceptSchema pV.Clausula_SegundaSchema = new
ConceptSchema(PV.CLAUSULA_SEGUNDA);
add(pV.Clausula_SegundaSchema, contratos.ontology.PV.Clausula_Segunda.class);
ConceptSchema cil.Clausula_PrimeiraSchema = new
ConceptSchema(CII.CLAUSULA_PRIMEIRA);
add(cil.Clausula_PrimeiraSchema, contratos.ontology.CII.Clausula_Primeira.class);
ConceptSchema pil.Clausula_PrimeiraSchema = new
ConceptSchema(PII.CLAUSULA_PRIMEIRA);
add(pil.Clausula_PrimeiraSchema, contratos.ontology.PII.Clausula_Primeira.class);
ConceptSchema identificacao_PartesModulea = new ConceptSchema(IDENTIFICACAO_PARTES);
add(identificacao_PartesModulea, contratos.ontology.Identificacao_PartesModulea.class);
ConceptSchema trocaSchema = new ConceptSchema(TROCA);
add(trocaSchema, contratos.ontology.Troca.class);
ConceptSchema t.Capitulo_IV_Das_ObrigacoesSchema = new
ConceptSchema(T.CAPITULO_IV_DAS_OBRIGACOES);
add(t.Capitulo_IV_Das_ObrigacoesSchema,
contratos.ontology.T.Capitulo_IV_Das_Obrigacoes.class);
ConceptSchema t.Capitulo_V_Das_ResponsabilidadesSchema = new
ConceptSchema(T.CAPITULO_V_DAS_RESPONSABILIDADES);
add(t.Capitulo_V_Das_ResponsabilidadesSchema,
contratos.ontology.T.Capitulo_V_Das_Responsabilidades.class);
ConceptSchema c.Capitulo_I_Do_ObjetoSchema = new
ConceptSchema(C.CAPITULO_I_DO_OBJETO);
add(c.Capitulo_I_Do_ObjetoSchema, contratos.ontology.C.Capitulo_I_Do_Objeto.class);
ConceptSchema IV.Clausula_PrimeiraSchema = new
ConceptSchema(LV.CLAUSULA_PRIMEIRA);
add(IV.Clausula_PrimeiraSchema, contratos.ontology.LV.Clausula_Primeira.class);
ConceptSchema p.Capitulo_I_Do_ObjetoSchema = new
ConceptSchema(P.CAPITULO_I_DO_OBJETO);
add(p.Capitulo_I_Do_ObjetoSchema, contratos.ontology.P.Capitulo_I_Do_Objeto.class);
ConceptSchema cl.Clausula_PrimeiraSchema = new ConceptSchema(CI.CLAUSULA_PRIMEIRA);
add(cl.Clausula_PrimeiraSchema, contratos.ontology.CI.Clausula_Primeira.class);
ConceptSchema pil.Clausula_SegundaSchema = new
ConceptSchema(PII.CLAUSULA_SEGUNDA);
add(pil.Clausula_SegundaSchema, contratos.ontology.PII.Clausula_Segunda.class);
ConceptSchema compra_e_VendaSchema = new ConceptSchema(COMPRAS_E_VENDAS);
add(compra_e_VendaSchema, contratos.ontology.Compras_e_Vendas.class);
ConceptSchema p.Capitulo_II_Do_valor_e_da_Forma_de_PagamentoSchema = new
ConceptSchema(P.CAPITULO_II_DO_VALOR_E_DA_FORMA_DE_PAGAMENTO);
add(p.Capitulo_II_Do_valor_e_da_Forma_de_PagamentoSchema,
contratos.ontology.P.Capitulo_II_Do_valor_e_da_Forma_de_Pagamento.class);
ConceptSchema tiV.Clausula_PrimeiraSchema = new
ConceptSchema(TIV.CLAUSULA_PRIMEIRA);
add(tiV.Clausula_PrimeiraSchema, contratos.ontology.TIV.Clausula_Primeira.class);

```

```

ConceptSchema contratoSchema = new ConceptSchema(CONTRATO);
add(contratoSchema, contratos.ontology.Contrato.class);
ConceptSchema til.Clausula_PrimeiraSchema = new
ConceptSchema(TII.CLAUSULA_PRIMEIRA);
add(til.Clausula_PrimeiraSchema, contratos.ontology.TII.Clausula_Primeira.class);
ConceptSchema t.Capitulo_I_Do_ObjetoSchema = new
ConceptSchema(T.CAPITULO_I_DO_OBJETO);
add(t.Capitulo_I_Do_ObjetoSchema, contratos.ontology.T.Capitulo_I_Do_Objeto.class);
ConceptSchema piV.Clausula_PrimeiraSchema = new
ConceptSchema(PIV.CLAUSULA_PRIMEIRA);
add(piV.Clausula_PrimeiraSchema, contratos.ontology.PIV.Clausula_Primeira.class);
ConceptSchema locacaoSchema = new ConceptSchema(LOCACAO);
add(locacaoSchema, contratos.ontology.Locacao.class);
ConceptSchema c.Capitulo_V_Das_ResponsabilidadesSchema = new
ConceptSchema(C.CAPITULO_V_DAS_RESPONSABILIDADES);
add(c.Capitulo_V_Das_ResponsabilidadesSchema,
contratos.ontology.C.Capitulo_V_Das_Responsabilidades.class);
ConceptSchema l.Capitulo_IV_Da_RescisaoSchema = new
ConceptSchema(L.CAPITULO_IV_DA_RESCISAO);
add(l.Capitulo_IV_Da_RescisaoSchema, contratos.ontology.L.Capitulo_IV_Da_Rescisao.class);
ConceptSchema piil.Clausula_PrimeiraSchema = new
ConceptSchema(PIII.CLAUSULA_PRIMEIRA);
add(piil.Clausula_PrimeiraSchema, contratos.ontology.PIII.Clausula_Primeira.class);
ConceptSchema t.Capitulo_III_Do_Prazo_de_EntregaSchema = new
ConceptSchema(T.CAPITULO_III_DO_PRAZO_DE_ENTREGA);
add(t.Capitulo_III_Do_Prazo_de_EntregaSchema,
contratos.ontology.T.Capitulo_III_Do_Prazo_de_Entrega.class);
ConceptSchema p.Capitulo_VI_Das_ResponsabilidadesSchema = new
ConceptSchema(P.CAPITULO_VI_DAS_RESPONSABILIDADES);
add(p.Capitulo_VI_Das_ResponsabilidadesSchema,
contratos.ontology.P.Capitulo_VI_Das_Responsabilidades.class);
ConceptSchema lii.Clausula_PrimeiraSchema = new ConceptSchema(LII.CLAUSULA_PRIMEIRA);
add(lii.Clausula_PrimeiraSchema, contratos.ontology.LII.Clausula_Primeira.class);
ConceptSchema p.Capitulo_III_Da_VigenciaSchema = new
ConceptSchema(P.CAPITULO_III_DA_VIGENCIA);
add(p.Capitulo_III_Da_VigenciaSchema, contratos.ontology.P.Capitulo_III_Da_Vigencia.class);
ConceptSchema tiil.Clausula_PrimeiraSchema = new
ConceptSchema(TIII.CLAUSULA_PRIMEIRA);
add(tiil.Clausula_PrimeiraSchema, contratos.ontology.TIII.Clausula_Primeira.class);
ConceptSchema p.Capitulo_V_Das_ObrigacoesSchema = new
ConceptSchema(P.CAPITULO_V_DAS_OBRIGACOES);
add(p.Capitulo_V_Das_ObrigacoesSchema,
contratos.ontology.P.Capitulo_V_Das_Obrigacoes.class);
ConceptSchema lvi.Clausula_SegundaSchema = new
ConceptSchema(LVI.CLAUSULA_SEGUNDA);
add(lvi.Clausula_SegundaSchema, contratos.ontology.LVI.Clausula_Segunda.class);
// adding AgentAction(s)
// adding AID(s)
// adding Predicate(s)
// adding fields
pvi.Clausula_SegundaSchema.add(PVI.CLAUSULA_SEGUNDA_RESPONSABILIDADES_P2,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
ciV.Clausula_PrimeiraSchema.add(CIV.CLAUSULA_PRIMEIRA_OBRIGACOES_P1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
IV.Clausula_SegundaSchema.add(LV.CLAUSULA_SEGUNDA_OBRIGACOES_P2,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
pl.Clausula_PrimeiraSchema.add(PI.CLAUSULA_PRIMEIRA_OBJETO,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
cil.Clausula_SegundaSchema.add(CII.CLAUSULA_SEGUNDA_DESCRICAO_JUROS,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);

```



```
cil.Clausula_SegundaSchema.add(CII.CLAUSULA_SEGUNDA_MULTA,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
cil.Clausula_SegundaSchema.add(CII.CLAUSULA_SEGUNDA_DESCRICAO_MULTA,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
cil.Clausula_SegundaSchema.add(CII.CLAUSULA_SEGUNDA_JUROS,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
lil.Clausula_SegundaSchema.add(LII.CLAUSULA_SEGUNDA_ORGAO_REAJUSTE,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
lil.Clausula_SegundaSchema.add(LII.CLAUSULA_SEGUNDA_MULTA,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
lil.Clausula_SegundaSchema.add(LII.CLAUSULA_SEGUNDA_DESCRICAO_MULTA,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
lil.Clausula_SegundaSchema.add(LII.CLAUSULA_SEGUNDA_JUROS,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
tl.Clausula_PrimeiraSchema.add(TI.CLAUSULA_PRIMEIRA_OBJETO2,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
tl.Clausula_PrimeiraSchema.add(TI.CLAUSULA_PRIMEIRA_OBJETO1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
pV.Clausula_PrimeiraSchema.add(PV.CLAUSULA_PRIMEIRA_OBRIGACOES_P1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
til.Clausula_SegundaSchema.add(TII.CLAUSULA_SEGUNDA_DESCRICAO_JUROS,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
til.Clausula_SegundaSchema.add(TII.CLAUSULA_SEGUNDA_MULTA,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
til.Clausula_SegundaSchema.add(TII.CLAUSULA_SEGUNDA_DESCRICAO_MULTA,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
til.Clausula_SegundaSchema.add(TII.CLAUSULA_SEGUNDA_JUROS,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
tiV.Clausula_SegundaSchema.add(TIV.CLAUSULA_SEGUNDA_OBRIGACOES_P2,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
liV.Clausula_PrimeiraSchema.add(LIV.CLAUSULA_PRIMEIRA_N DIAS_RESCISAO,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
ciiI.Clausula_PrimeiraSchema.add(CIII.CLAUSULA_PRIMEIRA_N DIAS_ENTREGA,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
cV.Clausula_PrimeiraSchema.add(CV.CLAUSULA_PRIMEIRA_RESPONSABILIDADES_P1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
II.Clausula_PrimeiraSchema.add(LI.CLAUSULA_PRIMEIRA_OBJETO,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
IvI.Clausula_PrimeiraSchema.add(LVI.CLAUSULA_PRIMEIRA_RESPONSABILIDADES_P1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_LOCAL,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_COD_TEST_P2,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_COD_TEST_P1,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_NOME_P1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_DATA,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_TEST_P2,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_NOME_P2,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_TEST_P1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
disposicao_GeralSchema.add(DISPOSICAO_GERAL_FORO,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
tV.Clausula_PrimeiraSchema.add(TV.CLAUSULA_PRIMEIRA_RESPONSABILIDADES_P1,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
```

liil.Clausula\_PrimeiraSchema.add(LIIL.CLAUSULA\_PRIMEIRA\_TIPO\_VIGENCIA,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pvl.Clausula\_PrimeiraSchema.add(PVI.CLAUSULA\_PRIMEIRA\_RESPONSABILIDADES\_P1,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
cV.Clausula\_SegundaSchema.add(CV.CLAUSULA\_SEGUNDA\_RESPONSABILIDADES\_P2,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
tV.Clausula\_SegundaSchema.add(TV.CLAUSULA\_SEGUNDA\_RESPONSABILIDADES\_P2,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
ciV.\_Clausula\_SegundaSchema.add(CIV.\_CLAUSULA\_SEGUNDA\_OBRIGACOES\_P2,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pV.Clausula\_SegundaSchema.add(PV.CLAUSULA\_SEGUNDA\_OBRIGACOES\_P2,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
cil.Clausula\_PrimeiraSchema.add(CII.CLAUSULA\_PRIMEIRA\_COND\_PAGAMENTO,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
cil.Clausula\_PrimeiraSchema.add(CII.CLAUSULA\_PRIMEIRA\_PRECO,  
(TermSchema)getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);  
cil.Clausula\_PrimeiraSchema.add(CII.CLAUSULA\_PRIMEIRA\_DESCRICAO\_PRECO,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_PrimeiraSchema.add(PII.CLAUSULA\_PRIMEIRA\_COND\_PAGAMENTO,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_PrimeiraSchema.add(PII.CLAUSULA\_PRIMEIRA\_ORGAO\_REAJUSTE,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_PrimeiraSchema.add(PII.CLAUSULA\_PRIMEIRA\_DIA\_PAGAMENTO,  
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);  
pil.Clausula\_PrimeiraSchema.add(PII.CLAUSULA\_PRIMEIRA\_LOCAL\_PAGAMENTO,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_PrimeiraSchema.add(PII.CLAUSULA\_PRIMEIRA\_TIPO\_REAJUSTE,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_PrimeiraSchema.add(PII.CLAUSULA\_PRIMEIRA\_PRECO,  
(TermSchema)getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);  
pil.Clausula\_PrimeiraSchema.add(PII.CLAUSULA\_PRIMEIRA\_DESCRICAO\_PRECO,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
identificacao\_PartesModule.add(IDENTIFICACAO\_PARTES\_CODIGO\_P1,  
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);  
identificacao\_PartesModule.add(IDENTIFICACAO\_PARTES\_CODIGO\_P2,  
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);  
identificacao\_PartesModule.add(IDENTIFICACAO\_PARTES\_ENDERECO\_P2,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
identificacao\_PartesModule.add(IDENTIFICACAO\_PARTES\_ENDERECO\_P1,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
identificacao\_PartesModule.add(IDENTIFICACAO\_PARTES\_NOME\_P1,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
identificacao\_PartesModule.add(IDENTIFICACAO\_PARTES\_NOME\_P2,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
IV.Clausula\_PrimeiraSchema.add(LV.CLAUSULA\_PRIMEIRA\_OBRIGACOES\_P1,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
cl.Clausula\_PrimeiraSchema.add(CI.CLAUSULA\_PRIMEIRA\_OBJETO,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_SegundaSchema.add(PII.CLAUSULA\_SEGUNDA\_ORGAO\_REAJUSTE,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_SegundaSchema.add(PII.CLAUSULA\_SEGUNDA\_MULTA,  
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);  
pil.Clausula\_SegundaSchema.add(PII.CLAUSULA\_SEGUNDA\_DESCRICAO\_MULTA,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
pil.Clausula\_SegundaSchema.add(PII.CLAUSULA\_SEGUNDA\_JUROS,  
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);  
tiV.Clausula\_PrimeiraSchema.add(TIV.CLAUSULA\_PRIMEIRA\_OBRIGACOES\_P1,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);  
til.Clausula\_PrimeiraSchema.add(TII.CLAUSULA\_PRIMEIRA\_COND\_TROCA,  
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);

```

    piV.Clausula_PrimeiraSchema.add(PIV.CLAUSULA_PRIMEIRA_N_DIAS_RESCISAO,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
    piil.Clausula_PrimeiraSchema.add(PIII.CLAUSULA_PRIMEIRA_TIPO_VIGENCIA,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
    lil.Clausula_PrimeiraSchema.add(LII.CLAUSULA_PRIMEIRA_ORGAO_REAJUSTE,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
    lil.Clausula_PrimeiraSchema.add(LII.CLAUSULA_PRIMEIRA_DIA_PAGAMENTO,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
    lil.Clausula_PrimeiraSchema.add(LII.CLAUSULA_PRIMEIRA_LOCAL_PAGAMENTO,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
    lil.Clausula_PrimeiraSchema.add(LII.CLAUSULA_PRIMEIRA_TIPO_REAJUSTE,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
    lil.Clausula_PrimeiraSchema.add(LII.CLAUSULA_PRIMEIRA_PRECO,
(TermSchema)getSchema(BasicOntology.FLOAT), ObjectSchema.OPTIONAL);
    lil.Clausula_PrimeiraSchema.add(LII.CLAUSULA_PRIMEIRA_DESCRICAO_PRECO,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
    lil.Clausula_PrimeiraSchema.add(LII.CLAUSULA_PRIMEIRA_TIPO_LOCACAO,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);
    tiil.Clausula_PrimeiraSchema.add(TIII.CLAUSULA_PRIMEIRA_N_DIAS_ENTREGA,
(TermSchema)getSchema(BasicOntology.INTEGER), ObjectSchema.OPTIONAL);
    lvi.Clausula_SegundaSchema.add(LVI.CLAUSULA_SEGUNDA_RESPONSABILIDADES_P2,
(TermSchema)getSchema(BasicOntology.STRING), ObjectSchema.OPTIONAL);

    // adding name mappings
    storeSlotName("responsabilidades_p2", "contratos.ontology.PVI.Clausula_Segunda",
"responsabilidades_p2");
    storeSlotName("obrigacoes_p1", "contratos.ontology.CIV.Clausula_Primeira", "obrigacoes_p1");
    storeSlotName("obrigacoes_p2", "contratos.ontology.LV.Clausula_Segunda", "obrigacoes_p2");
    storeSlotName("objeto", "contratos.ontology.PI.Clausula_Primeira", "objeto");
    storeSlotName("descricao_juros", "contratos.ontology.CII.Clausula_Segunda", "descricao_juros");
    storeSlotName("multa", "contratos.ontology.CII.Clausula_Segunda", "multa");
    storeSlotName("descricao_multa", "contratos.ontology.CII.Clausula_Segunda", "descricao_multa");
    storeSlotName("juros", "contratos.ontology.CII.Clausula_Segunda", "juros");
    storeSlotName("orgao_reajuste", "contratos.ontology.LII.Clausula_Segunda", "orgao_reajuste");
    storeSlotName("multa", "contratos.ontology.LII.Clausula_Segunda", "multa");
    storeSlotName("descricao_multa", "contratos.ontology.LII.Clausula_Segunda", "descricao_multa");
    storeSlotName("juros", "contratos.ontology.LII.Clausula_Segunda", "juros");
    storeSlotName("objeto2", "contratos.ontology.TI.Clausula_Primeira", "objeto2");
    storeSlotName("objeto1", "contratos.ontology.TI.Clausula_Primeira", "objeto1");
    storeSlotName("obrigacoes_p1", "contratos.ontology.PV.Clausula_Primeira", "obrigacoes_p1");
    storeSlotName("descricao_juros", "contratos.ontology.TII.Clausula_Segunda", "descricao_juros");
    storeSlotName("multa", "contratos.ontology.TII.Clausula_Segunda", "multa");
    storeSlotName("descricao_multa", "contratos.ontology.TII.Clausula_Segunda", "descricao_multa");
    storeSlotName("juros", "contratos.ontology.TII.Clausula_Segunda", "juros");
    storeSlotName("obrigacoes_p2", "contratos.ontology.TIV.Clausula_Segunda", "obrigacoes_p2");
    storeSlotName("n_dias_rescisao", "contratos.ontology.LIV.Clausula_Primeira", "n_dias_rescisao");
    storeSlotName("n_dias_entrega", "contratos.ontology.CIII.Clausula_Primeira", "n_dias_entrega");
    storeSlotName("responsabilidades_p1", "contratos.ontology.CV.Clausula_Primeira",
"responsabilidades_p1");
    storeSlotName("objeto", "contratos.ontology.LI.Clausula_Primeira", "objeto");
    storeSlotName("responsabilidades_p1", "contratos.ontology.LVI.Clausula_Primeira",
"responsabilidades_p1");
    storeSlotName("local", "contratos.ontology.Disposicao_Geral", "local");
    storeSlotName("cod_test_p2", "contratos.ontology.Disposicao_Geral", "cod_test_p2");
    storeSlotName("cod_test_p1", "contratos.ontology.Disposicao_Geral", "cod_test_p1");
    storeSlotName("nome_p1", "contratos.ontology.Disposicao_Geral", "nome_p1");
    storeSlotName("data", "contratos.ontology.Disposicao_Geral", "data");
    storeSlotName("test_p2", "contratos.ontology.Disposicao_Geral", "test_p2");
    storeSlotName("nome_p2", "contratos.ontology.Disposicao_Geral", "nome_p2");
    storeSlotName("test_p1", "contratos.ontology.Disposicao_Geral", "test_p1");

```

```

storeSlotName("foro", "contratos.ontology.Disposicao_Geral", "foro");
storeSlotName("responsabilidades_p1", "contratos.ontology.TV.Clausula_Primeira",
"responsabilidades_p1");
storeSlotName("tipo_vigencia", "contratos.ontology.LIII.Clausula_Primeira", "tipo_vigencia");
storeSlotName("responsabilidades_p1", "contratos.ontology.PVI.Clausula_Primeira",
"responsabilidades_p1");
storeSlotName("responsabilidades_p2", "contratos.ontology.CV.Clausula_Segunda",
"responsabilidades_p2");
storeSlotName("responsabilidades_p2", "contratos.ontology.TV.Clausula_Segunda",
"responsabilidades_p2");
storeSlotName("obrigacoes_p2", "contratos.ontology.CIV.Clausula_Segunda", "obrigacoes_p2");
storeSlotName("obrigacoes_p2", "contratos.ontology.PV.Clausula_Segunda", "obrigacoes_p2");
storeSlotName("cond_pagamento", "contratos.ontology.CII.Clausula_Primeira",
"cond_pagamento");
storeSlotName("preco", "contratos.ontology.CII.Clausula_Primeira", "preco");
storeSlotName("descricao_preco", "contratos.ontology.CII.Clausula_Primeira", "descricao_preco");
storeSlotName("cond_pagamento", "contratos.ontology.PII.Clausula_Primeira",
"cond_pagamento");
storeSlotName("orgao_reajuste", "contratos.ontology.PII.Clausula_Primeira", "orgao_reajuste");
storeSlotName("dia_pagamento", "contratos.ontology.PII.Clausula_Primeira", "dia_pagamento");
storeSlotName("local_pagamento", "contratos.ontology.PII.Clausula_Primeira", "local_pagamento");
storeSlotName("tipo_reajuste", "contratos.ontology.PII.Clausula_Primeira", "tipo_reajuste");
storeSlotName("preco", "contratos.ontology.PII.Clausula_Primeira", "preco");
storeSlotName("descricao_preco", "contratos.ontology.PII.Clausula_Primeira", "descricao_preco");
storeSlotName("codigo_p1", "contratos.ontology.Identificacao_Part", "codigo_p1");
storeSlotName("codigo_p2", "contratos.ontology.Identificacao_Part", "codigo_p2");
storeSlotName("endereco_p2", "contratos.ontology.Identificacao_Part", "endereco_p2");
storeSlotName("endereco_p1", "contratos.ontology.Identificacao_Part", "endereco_p1");
storeSlotName("nome_p1", "contratos.ontology.Identificacao_Part", "nome_p1");
storeSlotName("nome_p2", "contratos.ontology.Identificacao_Part", "nome_p2");
storeSlotName("obrigacoes_p1", "contratos.ontology.LV.Clausula_Primeira", "obrigacoes_p1");
storeSlotName("objeto", "contratos.ontology.CI.Clausula_Primeira", "objeto");
storeSlotName("orgao_reajuste", "contratos.ontology.PII.Clausula_Segunda", "orgao_reajuste");
storeSlotName("multa", "contratos.ontology.PII.Clausula_Segunda", "multa");
storeSlotName("descricao_multa", "contratos.ontology.PII.Clausula_Segunda", "descricao_multa");
storeSlotName("juros", "contratos.ontology.PII.Clausula_Segunda", "juros");
storeSlotName("obrigacoes_p1", "contratos.ontology.TIV.Clausula_Primeira", "obrigacoes_p1");
storeSlotName("cond_troca", "contratos.ontology.TII.Clausula_Primeira", "cond_troca");
storeSlotName("n_dias_rescisao", "contratos.ontology.PIV.Clausula_Primeira", "n_dias_rescisao");
storeSlotName("tipo_vigencia", "contratos.ontology.PIII.Clausula_Primeira", "tipo_vigencia");
storeSlotName("orgao_reajuste", "contratos.ontology.LII.Clausula_Primeira", "orgao_reajuste");
storeSlotName("dia_pagamento", "contratos.ontology.LII.Clausula_Primeira", "dia_pagamento");
storeSlotName("local_pagamento", "contratos.ontology.LII.Clausula_Primeira", "local_pagamento");
storeSlotName("tipo_reajuste", "contratos.ontology.LII.Clausula_Primeira", "tipo_reajuste");
storeSlotName("preco", "contratos.ontology.LII.Clausula_Primeira", "preco");
storeSlotName("descricao_preco", "contratos.ontology.LII.Clausula_Primeira", "descricao_preco");
storeSlotName("tipo_locacao", "contratos.ontology.LII.Clausula_Primeira", "tipo_locacao");
storeSlotName("n_dias_entrega", "contratos.ontology.TIII.Clausula_Primeira", "n_dias_entrega");
storeSlotName("responsabilidades_p2", "contratos.ontology.LVI.Clausula_Segunda",
"responsabilidades_p2");
// adding inheritance
c.Capitulo_III_Do_Prazo_de_EntregaSchema.addSuperSchema(compra_e_VendaSchema);
pvl.Clausula_SegundaSchema.addSuperSchema(p.Capitulo_VI_Das_ResponsabilidadesSchema);
l.Capitulo_VI_Das_ResponsabilidadesSchema.addSuperSchema(locacaoSchema);
prestacao_de_ServicoSchema.addSuperSchema(especiesSchema);
ciV.Clausula_PrimeiraSchema.addSuperSchema(c.Capitulo_IV_Das_ObrigacoesSchema);
especiesSchema.addSuperSchema(contratoSchema);
IV.Clausula_SegundaSchema.addSuperSchema(l.Capitulo_V_Das_ObrigacoesSchema);
pl.Clausula_PrimeiraSchema.addSuperSchema(p.Capitulo_I_Do_ObjetoSchema);

```



cil.Clausula\_SegundaSchema.addSuperSchema(c.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema);  
lii.Clausula\_SegundaSchema.addSuperSchema(i.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema);  
    i.Capitulo\_V\_Das\_ObrigacoesSchema.addSuperSchema(locacaoSchema);  
    ti.Clausula\_PrimeiraSchema.addSuperSchema(t.Capitulo\_I\_Do\_ObjetoSchema);  
    c.Capitulo\_IV\_Das\_ObrigacoesSchema.addSuperSchema(compra\_e\_VendaSchema);  
    l.Capitulo\_III\_Da\_VigenciaSchema.addSuperSchema(locacaoSchema);  
    pV.Clausula\_PrimeiraSchema.addSuperSchema(p.Capitulo\_V\_Das\_ObrigacoesSchema);  
til.Clausula\_SegundaSchema.addSuperSchema(t.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema);  
    tiV.Clausula\_SegundaSchema.addSuperSchema(t.Capitulo\_IV\_Das\_ObrigacoesSchema);  
    liV.Clausula\_PrimeiraSchema.addSuperSchema(i.Capitulo\_IV\_Da\_RescisaoSchema);  
    ciil.Clausula\_PrimeiraSchema.addSuperSchema(c.Capitulo\_III\_Do\_Prazo\_de\_EntregaSchema);  
    cV.Clausula\_PrimeiraSchema.addSuperSchema(c.Capitulo\_V\_Das\_ResponsabilidadesSchema);  
    II.Clausula\_PrimeiraSchema.addSuperSchema(i.Capitulo\_I\_Do\_ObjetoSchema);  
I.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema.addSuperSchema(locacaoSchema);  
p.Capitulo\_IV\_Da\_RescisaoSchema.addSuperSchema(prestacao\_de\_ServicoSchema);  
t.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema.addSuperSchema(trocaSchema);  
IvI.Clausula\_PrimeiraSchema.addSuperSchema(i.Capitulo\_VI\_Das\_ResponsabilidadesSchema);  
I.Capitulo\_I\_Do\_ObjetoSchema.addSuperSchema(locacaoSchema);  
disposicao\_GeralSchema.addSuperSchema(contratoSchema);  
tV.Clausula\_PrimeiraSchema.addSuperSchema(t.Capitulo\_V\_Das\_ResponsabilidadesSchema);  
c.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema.addSuperSchema(compra\_e\_VendaSchema);  
    II.Clausula\_SegundaSchema.addSuperSchema(i.Capitulo\_I\_Do\_ObjetoSchema);  
    liil.Clausula\_PrimeiraSchema.addSuperSchema(i.Capitulo\_III\_Da\_VigenciaSchema);  
    pVI.Clausula\_PrimeiraSchema.addSuperSchema(p.Capitulo\_VI\_Das\_ResponsabilidadesSchema);  
cV.Clausula\_SegundaSchema.addSuperSchema(c.Capitulo\_V\_Das\_ResponsabilidadesSchema);  
    tV.Clausula\_SegundaSchema.addSuperSchema(t.Capitulo\_V\_Das\_ResponsabilidadesSchema);  
ciV.Clausula\_SegundaSchema.addSuperSchema(c.Capitulo\_IV\_Das\_ObrigacoesSchema);  
    pV.Clausula\_SegundaSchema.addSuperSchema(p.Capitulo\_V\_Das\_ObrigacoesSchema);  
cil.Clausula\_PrimeiraSchema.addSuperSchema(c.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema);  
piI.Clausula\_PrimeiraSchema.addSuperSchema(p.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema);  
    identificacao\_PartesSchema.addSuperSchema(contratoSchema);  
    trocaSchema.addSuperSchema(especiesSchema);  
    t.Capitulo\_IV\_Das\_ObrigacoesSchema.addSuperSchema(trocaSchema);  
    t.Capitulo\_V\_Das\_ResponsabilidadesSchema.addSuperSchema(trocaSchema);  
    c.Capitulo\_I\_Do\_ObjetoSchema.addSuperSchema(compra\_e\_VendaSchema);  
    IV.Clausula\_PrimeiraSchema.addSuperSchema(i.Capitulo\_V\_Das\_ObrigacoesSchema);  
    p.Capitulo\_I\_Do\_ObjetoSchema.addSuperSchema(prestacao\_de\_ServicoSchema);  
    cl.Clausula\_PrimeiraSchema.addSuperSchema(c.Capitulo\_I\_Do\_ObjetoSchema);  
piI.Clausula\_SegundaSchema.addSuperSchema(p.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema);  
compra\_e\_VendaSchema.addSuperSchema(especiesSchema);  
p.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema.addSuperSchema(prestacao\_de\_ServicoSchema);  
    tiV.Clausula\_PrimeiraSchema.addSuperSchema(t.Capitulo\_IV\_Das\_ObrigacoesSchema);  
til.Clausula\_PrimeiraSchema.addSuperSchema(t.Capitulo\_II\_Do\_Valor\_e\_da\_Forma\_de\_PagamentoSchema);  
    til.Capitulo\_I\_Do\_ObjetoSchema.addSuperSchema(trocaSchema);  
piV.Clausula\_PrimeiraSchema.addSuperSchema(p.Capitulo\_IV\_Da\_RescisaoSchema);  
locacaoSchema.addSuperSchema(especiesSchema);  
c.Capitulo\_V\_Das\_ResponsabilidadesSchema.addSuperSchema(compra\_e\_VendaSchema);  
I.Capitulo\_IV\_Da\_RescisaoSchema.addSuperSchema(locacaoSchema);  
piil.Clausula\_PrimeiraSchema.addSuperSchema(p.Capitulo\_III\_Da\_VigenciaSchema);  
t.Capitulo\_III\_Do\_Prazo\_de\_EntregaSchema.addSuperSchema(trocaSchema);  
p.Capitulo\_VI\_Das\_ResponsabilidadesSchema.addSuperSchema(prestacao\_de\_ServicoSchema);

```
lii.Clausula_PrimeiraSchema.addSuperSchema(l.Capitulo_II__Do_Valor_e_da_Forma_de_Pagament  
oSchema);  
    p.Capitulo_III__Da_VigenciaSchema.addSuperSchema(prestacao_de_ServicoSchema);  
tiii.Clausula_PrimeiraSchema.addSuperSchema(t.Capitulo_III__Do_Prazo_de_EntregaSchema);  
    p.Capitulo_V__Das_ObrigacoesSchema.addSuperSchema(prestacao_de_ServicoSchema);  
lvi.Clausula_SegundaSchema.addSuperSchema(l.Capitulo_VI__Das_ResponsabilidadesSchema);  
    }catch (java.lang.Exception e) {e.printStackTrace();}  
    }  
}
```