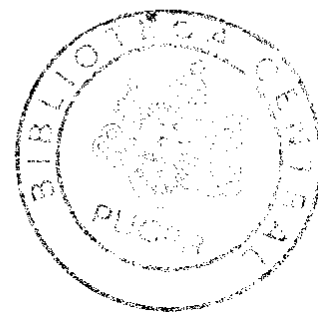


**IRAPURU HARUO FLÓRIDO**



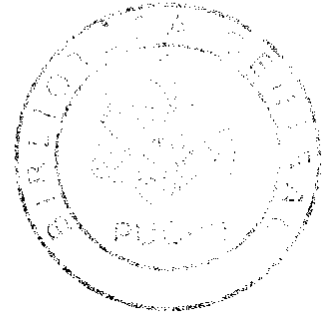
**GESTÃO DO CONHECIMENTO EM SISTEMAS  
DE INFORMAÇÃO LEGADOS –  
“ONTOLEGACY”**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

**CURITIBA**

**2004**

**IRAPURU HARUO FLÓRIDO**



**GESTÃO DO CONHECIMENTO EM SISTEMAS  
DE INFORMAÇÃO LEGADOS – *ONTOLEGACY***

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: Descoberta do Conhecimento e Mineração de Dados

Orientador: Prof. Dr. Bráulio Coelho Ávila

Co-orientador: Prof. Dr. Marcos Augusto Hochuli Shmeil

**CURITIBA**

**2004**

Biblioteca Central  
Gestão do conhecimento em sistemas de informações  
Ac. 196577 - R. 582647 Ex. 1  
Doação - PPGIA  
13/12/2004

Flório, Irapuru Haruo

Gestão do Conhecimento em Sistemas de Informação Legados – “Ontolegacy”  
Curitiba, 2004, 139 p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

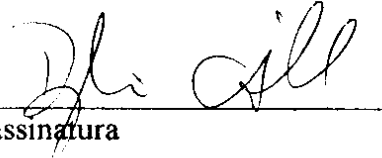

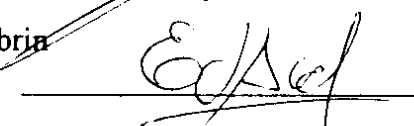
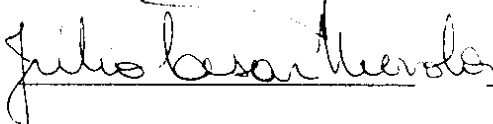

1. Gestão do Conhecimento 2. Sistemas de Informação Legados 3. Ontologias.  
4. Agentes de software I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática Aplicada.



ATA DA SESSÃO PÚBLICA DE DEFESA DE DISSERTAÇÃO DE MESTRADO  
 DO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA  
 DA PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

DEFESA DE DISSERTAÇÃO Nº 102

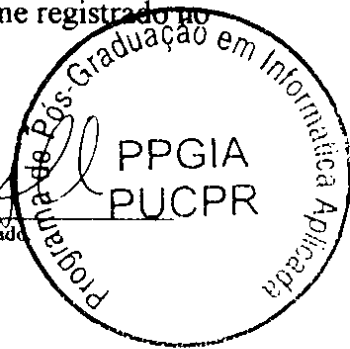
Aos 30 dias do mês de agosto de 2004 realizou-se a sessão pública de defesa da dissertação “**Gestão do Conhecimento em Sistemas de Informação Legados - Ontolegacy**”, apresentado por **Irapuru Haruo Flório** como requisito parcial para a obtenção do título de **Mestre em Informática Aplicada**, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Bráulio Coelho Ávila PUCPR (Orientador)	 assinatura	<u>aprovado</u> parecer(aprov/ reprov.)
Prof. Dr. Marcos Augusto H. Shmeil. PUCPR (co-orientador)		<u>Aprovado</u>
Prof. Dr. Edson Emílio Scalabrin PUCPR		<u>Aprov.</u>
Prof. Dr. Júlio César Nievola PUCPR		<u>APROVADO</u>
Prof. Dr. Roberto Tadeu Raittz UFPR		<u>Aprovado</u>

Conforme as normas regimentais do PPGIA e da PUCPR, o trabalho apresentado foi considerado aprovado (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora, conforme registrado no Livro de Defesas do programa.

Prof. Dr. Bráulio Coelho Ávila  
Diretor do PPGIA PUCPR

07/10/04  
Data e assinatura, após homologação da defesa pelo colegiado



*À minha amada esposa Débora.*

*Aos meus Pais.*

## Agradecimentos

Ao Grande Arquiteto do Universo, Deus, por me consentir trilhar este percurso da vida acadêmica, iluminando-me, guiando-me e permitindo-me superar as barreiras encontradas.

Aos meus pais, Célia e Zito, e ao meu irmão Irapuan por sempre me incentivarem e proporcionarem condições, desde a mais tenra idade aos estudos, sou muito grato.

À minha querida esposa Débora, por estar ao meu lado nas horas difíceis e, com seu amor incondicional, proporcionar força e inspiração para a continuidade desta jornada.

Às minhas queridas filhas Ana Clara e Nara Mayumi, pela ajuda e pela compreensão, diante da minha ausência e da falta de tempo a compartilhar, durante este trabalho.

A dona Jandira pela suas orações inestimáveis e apoio durante minha vida acadêmica.

Ao meu orientador, Professor Marcos Augusto Hochuli Shmeil, por acreditar nos meus ideais e incentivar-me na condução do trabalho, fazendo-me perseverante nos momentos em que havia apenas um fio de chama. Sou agradecido também pela amizade que recebi durante todo o tempo que passamos no Mestrado e outrora no Banco onde trabalhamos.

Ao professor Bráulio, pelo incentivo no início da orientação e que, por força do destino, coloca-se de volta como orientador oficial e também pelos conselhos no desenvolvimento desta dissertação. Aos professores Alcides Calsavara, Jair L. Barboza, Edgard Jamhour, Jamil I. Iskandar, Léa das Graças C. Anastasiou, Carlos A. Maziero, e Wolfgang Arndt por sua competência na condução do curso, com seus ensinamentos.

Pela participação em minha banca, agradeço aos Professores Roberto Tadeu Raittz, (UFPR), Edson Emílio Scalabrin (PUC-PR) e Júlio César Nievola (PUC-PR).

Aos companheiros do laboratório Daniele, Helyane, Jaime, Leonardo, Luiz Gustavo, Rafael, Richard, Silla e Thiago pela paciência, presteza e dicas neste intrincado mundo da Tecnologia de Informação. Meus amigos do Banco que sempre me incentivaram e apoiaram Antonio Carlos, Divonsir, Jaime W., João Luiz, Machado, Marcelo, Marco Antonio, Marlos e Taborda. Agradeço também aos colegas do Mestrado Aline, Claudia, Denise, Filipe, Guarita, Gotardo, Hugo, Irani, Kelly, Kolb, Simone, Sílvio e Pacheco, com quem pude compartilhar idéias, aprender e reaprender. Ao corpo docente da Escola Técnica da UFPR, do qual faço parte, pelos subsídios e condições para conclusão deste trabalho. Enfim a todos que direta ou indiretamente contribuíram para que eu concretizasse este significativo projeto da minha trajetória profissional e acadêmica.

*“Amarás ao teu próximo como a ti mesmo”  
(Jesus Cristo)*

*“A preocupação com o outro é hoje a mais profunda  
e revolucionária idéia”  
(Noam Chomsky)*

# Sumário

<b>Agradecimentos</b>	v
<b>Sumário</b>	vi
<b>Lista de Figuras</b>	x
<b>Lista de Tabelas</b>	xii
<b>Lista de Abreviaturas</b>	xiii
<b>Resumo</b>	xiv
<b>Abstract</b>	xv
<b>Capítulo 1</b>	<b>16</b>
<b>Introdução</b>	<b>16</b>
1.1. Motivação	17
1.2. Cenário	19
1.3. Objetivos	22
1.3.1. Objetivo geral	22
1.3.2. Objetivo específico	23
1.4. Organização	23
<b>Capítulo 2</b>	<b>24</b>
<b>Recorrentes Conceituais e Tecnológicos</b>	<b>24</b>
2.1. Gestão do conhecimento	24
2.1.1. Origem do conhecimento - abordagem filosófica	26
2.1.2. Conhecimento e a cadeia de valores	28
2.1.3. Componentes básicos do conhecimento	29
2.1.4. Conversão e criação do conhecimento	30
2.1.5. Espiral do conhecimento	32
2.1.6. Transferência do conhecimento	32
2.2. Ontologias	34
2.2.1. Ontologia e Filosofia	34
2.2.2. Ontologias e Inteligência Artificial	36
2.2.3. Classificação das Ontologias	38
2.2.4. Metodologias para construção de Ontologias	38



2.2.5.	Linguagens para desenvolvimento de Ontologias	41
2.2.6.	Ferramentas para desenvolvimento de Ontologias	42
2.2.7.	Sistema KISF	43
2.2.8.	Finalidade e uso das Ontologias	45
2.2.9.	Estado da Arte	46
2.3.	Agentes de <i>Software</i>	47
2.3.1.	Conceitos e definições	47
2.3.2.	Competências de agentes de <i>software</i>	49
2.3.3.	Tipologias de agentes de <i>software</i>	50
2.3.4.	Ambiente dos agentes de <i>software</i>	51
2.3.5.	Sistemas multiagentes	52
2.3.6.	Classificações de sistemas multiagentes	53
2.4.	Sistemas de Informação Legados	57
2.4.1.	Definição e evolução	57
2.4.2.	Perspectiva dos Sistemas de Informação Legados	59
2.4.3.	Ambiente dos Sistemas de Informações Legados	60
2.4.4.	Linguagem COBOL	61
2.5.	Considerações finais deste capítulo	63
<b>Capítulo 3</b>		<b>64</b>
<b>Metodologia para gestão do conhecimento em Sistemas de Informação Legados (SIL): <i>Ontolegacy</i></b>		<b>64</b>
3.1.	Cenário da problematização	64
3.2.	A metodologia proposta: <i>Ontolegacy</i>	66
3.2.1.	Construção da Ontologia dos SIL	66
3.2.2.	Definição dos meios para extração dos termos e conceitos	68
3.2.3.	Associação da <i>Ontologia</i> dos SIL aos termos e conceitos extraídos	71
3.2.4.	Compartilhamento do conhecimento dos SIL	72
3.3.	A ferramenta computacional: <i>Ontolegacy</i>	73
3.3.1.	Arquitetura do sistema <i>Ontolegacy</i>	73
3.3.2.	Bases de conhecimento	74
3.3.3.	Sociedade de agentes do <i>Ontolegacy</i>	74
3.3.4.	AgVerificador	75

3.3.5.	AgMantenedor	77
3.3.6.	AgInterface	81
3.4.	Considerações Finais	82
<b>Capítulo 4</b>		<b>83</b>
<b>EXPERIMENTO</b>		<b>83</b>
4.1.	O Sistema de Informação Legado: arrecadação da Previdência Social	83
4.2.	Aplicação da metodologia Ontolegacy	84
4.2.1.	Construção da Ontologia	85
4.2.2.	Definição dos meios para extração dos termos e conceitos	89
4.2.3.	Associação da <i>Ontologia</i> dos SIL com os termos e conceitos extraídos	91
4.3.	Compartilhamento do Conhecimento do SIL	93
4.4.	Considerações Finais	95
4.4.1.	Ambiente do Experimento	95
4.4.2.	Resultados Obtidos	95
<b>Capítulo 5</b>		<b>97</b>
<b>Conclusão</b>		<b>97</b>
5.1.	Contribuição	98
5.2.	Estado da Arte	99
5.3.	Trabalhos Futuros	99
<b>Referências</b>		<b>100</b>
<b>Apêndice A</b>		<b>106</b>
<b>Linguagens para desenvolvimento de Ontologias</b>		<b>106</b>
<b>Apêndice B</b>		<b>109</b>
<b>Informações da GPS</b>		<b>109</b>
<b>Apêndice C</b>		<b>113</b>
<b>CF do Sistema de Informação XYZ</b>		<b>113</b>
C.1 - Entidade e atributos do CF XYZ510		113
C.2 - Código Fonte XYZ510		116
<b>Apêndice D</b>		<b>133</b>
<b>Ontologia do Sistema de Informação Legado XYZ</b>		<b>133</b>

## Lista de Figuras

Figura 1.1: Sistemas e mudanças	20
Figura 1.2: Mundo	22
Figura 2.1: Fator desencadeante [NNT97]	25
Figura 2.2: Os cinco sentidos [AFH04]	27
Figura 2.3: Modos de conversão do conhecimento [NNT97]	31
Figura 2.4: Espiral do Conhecimento [NNT97]	33
Figura 2.5: Definição de Ontologia [GUA98]	37
Figura 2.6: Usos da Ontologia	45
Figura 2.7: XML e <i>Web</i> [W3C04]	46
Figura 2.8: Algumas das dimensões para classificação de agentes [SHM99]	54
Figura 2.9: Evolução computacional	58
Figura 3.1: Ambiente fenomenológico	65
Figura 3.2: Codificação COBOL	69
Figura 3.3: Área de dados ( <i>Data Division</i> COBOL)	70
Figura 3.4: Associação com a Ontologia	71
Figura 3.5: Compartilhamento da Ontologia	72
Figura 3.6: Arquitetura Geral do Sistema Ontolegacy	74
Figura 3.7: Sociedade de Agentes Ontolegacy	75
Figura 3.8: Configuração AgVerificador	76
Figura 3.9: Meta-agente Mantenedor	77
Figura 3.10: Código Fonte Expandido	78
Figura 3.11: Código Fonte selecionado	79
Figura 3.12: Configuração AgInterface	82
Figura 4.1: Guia GPS	84
Figura 4.2: Modelo de dados do Sistema XYZ	85
Figura 4.3: Ontologia com a hierarquia dos conceitos	88
Figura 4.4: Código Fonte VS COBOL II com SQL	89
Figura 4.5: Código Fonte VS COBOL II CICS	90
Figura 4.6: Sociedade de Agentes Ontolegacy	91
Figura 4.7: Associação dos conceitos	92

Figura 4.8: Ontologia exportada para o RDF	93
Figura 4.9: Páginas HTML Gerada pelo <i>Ontolegacy</i>	94

## Lista de Tabelas

Tabela 2.1: Metodologias para Construção de Ontologias	38
Tabela 3.1: Definição Abstrata do AgVerificador	76
Tabela 3.2: Definição Abstrata do agmseleção	78
Tabela 3.3: Definição Abstrata do AgMAnalisador	80
Tabela 3.4: Comandos COBOL selecionados	80
Tabela 3.5: Definição abstrata do AgMConceito	81
Tabela 3.6: Definição Abstrata do AgInterface	81
Tabela 4.1: Termos e Definições	86
Tabela 4.2: Estatística dos CF	91
Tabela 4.3: Entidade e atributos do CF XYZ510	91
Tabela 4.4: Atributos específicos	92

# Lista de Abreviaturas

AMC	Aplicações de Missão Crítica
CF	Códigos Fonte
COBOL	<i>COmmon Business Oriented Language</i>
GPS	Guia da Previdência Social
HTML	<i>Hypertext Markup Language</i>
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
INSS	Instituto Nacional da Seguridade Social
RDF	<i>Resource Description Framework</i>
SI	Sistemas de Informação
SIL	Sistemas de Informação Legados
SQL	<i>Structured Query Language</i>
TI	Tecnologia de Informação
WEB	Rede de computadores da Internet
XML	<i>Extensible Markup Language</i>

# Resumo

A dinâmica atual no mundo dos negócios tem levado as organizações a uma nova atitude, em função da manutenção da competitividade e da sua conseqüente sobrevivência nesse novo ambiente.

Os sistemas de informação destas organizações, imprescindíveis nas tomadas de decisões dos negócios, devem responder prontamente às mudanças e às novas estratégias impostas. Desse modo, a gestão do conhecimento e a inteligência artificial vêm contribuir, nesse cenário, de modo fundamental, ao proporcionarem métodos de trabalho e ferramentas que auxiliem nas mudanças requeridas nesses sistemas de informação.

Esta dissertação propõe o desenvolvimento de uma metodologia e um *software* de apoio, ambos baseados em Ontologias, os quais podem proporcionar ao analista de sistemas a gestão do conhecimento em sistemas de informação legados, e podem auxiliá-lo na manutenção e na preservação do respectivo conhecimento.

O trabalho é dirigido principalmente às organizações financeiras em que 90% dos sistemas de informação que suportam os negócios têm como plataforma principal o *Mainframe*, servidores corporativos e tecnologias de terceira geração como a linguagem COBOL.

Tem-se identificado que tais organizações possuem em seus códigos fontes a totalidade das regras dos negócios, o que conduz a se tornarem, estes códigos fontes, uma das principais fontes do conhecimento organizacional.

**Palavras-Chave:** Gestão do conhecimento, Sistemas de informação legados, Ontologias, Agentes de *software*.

# Abstract

Today's business dynamics has led organizations to a new positioning in order to face such reality, thus maintaining their competitiveness and corresponding survival in such environment.

Information systems of such organizations, which are vital in the process of decision-making, should promptly respond to changes and to the new imposed strategies. Knowledge management and artificial intelligence present a fundamental contribution in such scenario by providing work methods and tools that may be helpful in regard to the required changes of these information systems.

The purpose of this work is the development of a methodology and a supporting software, both based on ontologies, which may provide to the developer the knowledge management of legacy information systems and may help him maintain and preserve the knowledge thereof.

This study is directed mostly to banking companies, where 90% of all information systems that support business have a mainframe as a main platform, corporate servers and third generation technologies such as the COBOL language.

There has been recognition that such organizations hold in their source code the entire set of their business rules. This fact enables the source code to become one of the main sources of the knowledge of the organization.

**Keywords:** Knowledge Management, Legacy Information Systems, Ontology, and Software Agents.



# Capítulo 1

## Introdução

Nos últimos cinquenta anos, os reveses pelos quais a sociedade passou, econômica política e socialmente, fizeram que as organizações existissem num ambiente em que a única certeza era a incerteza [NNT97] [SVB00].

Diante das tendências de globalização da economia, as organizações se obrigaram a constituir nova visão e nova atitude para sua sobrevivência, buscando melhorar suas vantagens competitivas e inovação como forma constante da conquista de novos mercados.

De fato, o caminho para a evolução organizacional exige mudanças de comportamentos, atitudes e quebras de alguns paradigmas que caíram na obsolescência.

Assim, a gestão do conhecimento surge como efeito destas mudanças, na perspectiva de responder a novas demandas do mercado, de modo que adquire gradativamente um papel fundamental na vida das organizações, revestindo-se como o recurso econômico mais importante [TRR00]. Quinn afirma que a maior parte do valor agregado de muitas organizações são consequência da posse de conhecimento específico [QNN90].

Na maioria das vezes, o conhecimento organizacional se apresenta disperso e de forma tácita, necessitando ser explicitado de alguma forma para permitir sua aplicação e manutenção a fim de propiciar qualidade, valor agregado, serviço, inovação, agilidade, flexibilidade e velocidade [TXR00].

Perante os fatos apresentados, lança-se mão das tecnologias computacionais baseadas em Inteligência Artificial (IA), sobretudo quando à representação do conhecimento com o uso de *Ontologias*, para viabilizar as organizações nesse cenário adverso, complexo e competitivo.

## 1.1. Motivação

Conforme Shmeil, numa dada sociedade, as organizações se apresentam como entidades independentes complexas ou distribuídas e se caracterizam pelas suas interações intra e interorganizações. De um modo geral, o resultado desse processo contínuo de interações permite que as organizações envolvidas incrementem e refinem seus conhecimentos, tornando-as mais eficientes e eficazes [SHM99].

Davenport e Prusak ressaltam que “o acúmulo do conhecimento ao longo do tempo resulta num bem intangível para as organizações, não é mensurável como um bem físico, por isso torna-se importante reter e conservar o conhecimento nas organizações” [DVP98].

Como se constata, torna-se importantíssimo desenvolver, manter e reter o conhecimento, tendo em vista facilitar o processo decisório bem como elevar o valor intrínseco das organizações. A *Ontologia*, tratada pela IA, vem ao encontro das questões citadas, porque tem como objetivo tornar formal e explicitar a conceituação de um determinado domínio do conhecimento, ou seja, representá-lo [GRU93].

No Brasil, organizações financeiras – principalmente as bancárias fundadas no século XX – foram precursoras do uso da informática em seus negócios. Herdaram as primeiras tecnologias utilizadas na área, como as linguagens de terceira geração, por exemplo, o COBOL e o uso de computadores de grande porte, ditos *mainframe*.

A constante preocupação com a performance, em função do grande volume de dados gerados pelas características do negócio e em organizações desse porte, levou os grandes fabricantes mundiais de *mainframe*, IBM, Bull-Honeywell, Unisys, Digital e outros, a se concentrarem muito mais no desenvolvimento de tecnologias de *hardware* em detrimento do *software*. Conseqüentemente, as inovações tecnológicas de software evoluíram muito mais lentamente, para proporcionar ferramentas de auxílio ao corpo técnico de sistemas na gestão do conhecimento em Sistemas de Informação (SI).

Ainda hoje, é de grande expressão o volume dos SI nas grandes organizações que utilizam as mencionadas tecnologias legadas que, por sua vez, serão nominados de Sistemas de Informação Legados (SIL).

Conforme levantamento do Gartner Group, empresa internacional de consultoria em TI, 75% dos dados processados nos EUA são feitos em linguagem COBOL; há hoje em torno de 190 bilhões de linhas de código em COBOL, 15% das novas aplicações ainda são escritas em COBOL.

Nesse contexto, muitas organizações tentaram – e alguns o fizeram – migrar seus SIL do ambiente de *mainframe*, *downsizing*, para tecnologias de rede, baixa plataforma, algumas com sucesso, outras não. Muitas se depararam com barreiras tecnológicas e econômicas inviabilizando estas migrações.

Determinada por fatores externos, como medidas de governo e alterações nos rumos da economia, a constante mudança no ambiente dos negócios obriga as organizações financeiras a fazerem contínuas manutenções em seus SIL. Tal necessidade, não raro, sobrecarrega a quantidade de trabalho, *backlog*, do corpo técnico envolvido nos SIL afetados. Então, o uso da tecnologia e de novas ferramentas e métodos devem contribuir de forma a agilizar e tornar mais eficiente o trabalho desse corpo técnico, permitindo não somente facilidades para extrair conhecimentos úteis destes SIL como também viabilizar as manutenções necessárias para o cumprimento das mudanças.

Chega-se então à seguinte questão: como o conhecimento está distribuído nestas organizações? Teixeira afirma que o conhecimento disseminado nas organizações apresenta-se basicamente de duas formas: tácito e explícito. O conhecimento explícito, em geral, apresenta-se registrado de alguma forma e disponível para as pessoas, enquanto que o conhecimento tácito é o que as pessoas possuem e não está registrado de algum modo [TXR00].

A sistematização e a extração do conhecimento tácito se deparam hoje com barreiras tecnologicamente não solucionadas, ou seja, ainda não é possível à extração automática do conhecimento tácito, retirar o conhecimento da mente das pessoas. Mas, por meio do uso de metodologias e ferramentas computacionais, o conhecimento explícito distribuído em diversas mídias – códigos fonte, manuais, textos, banco de dados – permite a representação e o armazenamento dos conhecimentos necessários no auxílio da manutenção e solução dos problemas dos SIL.

Para Davenport e Prusak a única vantagem sustentável que uma empresa tem é aquilo que ela coletivamente sabe, a eficiência com que ela usa o que sabe e a prontidão com que ela adquire e usa novos conhecimentos. [DVP98]

Face ao exposto, busca-se encontrar soluções, desenvolver métodos e ferramentas que auxiliem e permitam a gestão do conhecimento no ambiente organizacional, especificamente em SIL para seu desenvolvimento e manutenção.

## 1.2. Cenário

Toffler [TFL84], em sua obra, *A terceira onda*, é enfático ao afirmar que “a sociedade pós-industrial tem o seu ‘poder’ no conhecimento e que o capital e o trabalho ficaram em segundo plano e não são essenciais na sobrevivência das organizações”.

Compartilhando este pensamento, Teixeira [TXR00] mostra que existem marcos da transformação dessa sociedade pós-industrial, como quando as aplicações comerciais dos *mainframes* em organizações na década de 60 começaram a ser multiplicar ou, quando nos anos 80, houve a popularização do uso dos microcomputadores ou, ainda, quando nos EUA, em 1956, o número de trabalhadores no setor de serviços superou a quantidade de empregados no setor de manufatura.

O ambiente das grandes corporações organizacionais que fazem uso das tecnologias e ambientes de sistemas corporativos centralizados, ditos *mainframe*, é o objeto principal e faz-se presente em todo contexto da presente dissertação.

Para a Tecnologia de Informação, o valor real das organizações está nos SIL. De fato, neles estão acumulados anos de regras de negócio, políticas, *expertise* e *know-how*. Incorporadas nos sistemas, estes SIL são na maioria das vezes a sustentação dos negócios dentro das grandes corporações [HLS99].

No Brasil, convergem para o uso desse ambiente, intensamente, as instituições financeiras bancárias, ou seja, os bancos comerciais e bancos múltiplos de um modo geral. Essas instituições possuem um importantíssimo acervo de SIL, que são críticos e fundamentais no apoio e condução dos negócios, nominados de Aplicações de Missão Crítica (AMC). Estas aplicações não podem parar além de determinado tempo, muitas vezes dezenas de minutos, trazem desgastes e prejuízos intangíveis em função da sua importância dentro do ambiente de negócio.

Como por exemplo, menciona-se a inoperância num determinado dia do SI de conta corrente de uma instituição bancária que, inoperante, “fora do ar”, pode causar sérios prejuízos de ordem econômica e de credibilidade com clientes.

O direcionamento do presente trabalho para este ambiente, especificamente centrado em SIL, deve-se ao fato de ser mais afetado pelas constantes mudanças que ocorrem em suas regras de negócios; por conseqüência, necessita de agilidade e flexibilidade para que isto ocorra, dada também sua importância como AMC.

Aliado a tudo isso, deve-se levar em conta que muitas vezes os SIL das organizações envolvidas manipulam e operam grandes somas monetárias, tornando-se essenciais a eficiência e eficácia nos processos de mudanças e manutenções.

Para melhor compreensão desse cenário, numa determinada instituição bancária, por exemplo, houve o desconto duplicado em conta corrente de pessoas físicas da prestação mensal que vencia em determinado dia, relativa a um produto chamado *Crédito Direto ao Consumidor* (CDC). O fato causou um desgaste de relacionamento junto ao cliente, relevante, como a perda de credibilidade na prestação do serviço e, também, o aumento da carga trabalho para o corpo técnico decorrente das manutenções no SIL. Logicamente, o fato gerou um custo, de certa forma intangível, não só no SIL envolvido como também naqueles relacionados e impactados com o problema. Em grandes corporações, onde há muito compartilhamento de informações entre os SI, estes estão sujeitos ao efeito de recursividade. Isto é, os problemas acontecem gerando outros problemas, em outros SI, e assim por diante.

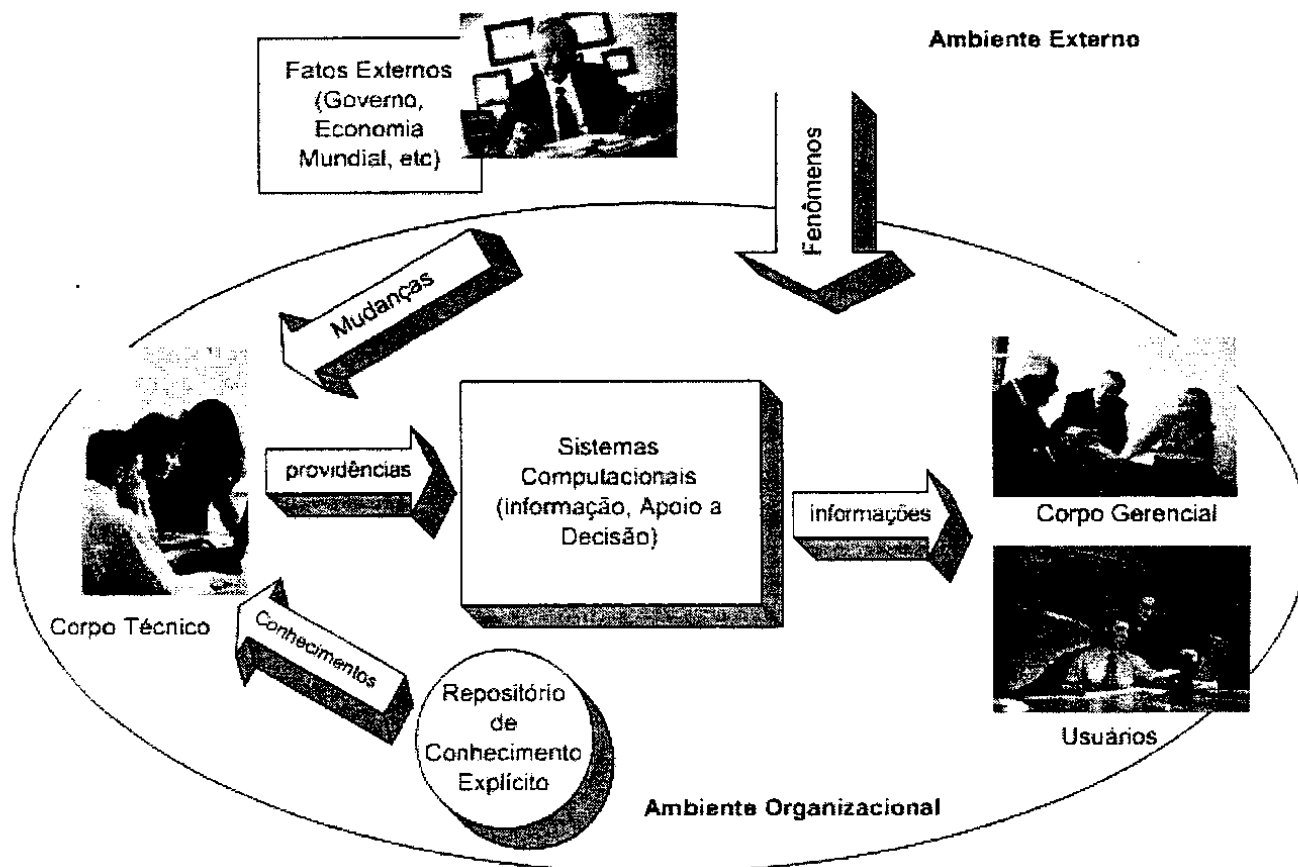


Figura 1.1: Sistemas e mudanças

Conforme exemplo [Figura 1.1], diante de uma mudança, de um fato, um fenômeno no ambiente externo que alterará o ambiente organizacional que possui SI, caso tenham impacto nos mesmos, por efeito, será gerada uma manutenção. Este processo de mudança acarretará providências para o corpo técnico responsável em manter os SI. E quando o conhecimento a respeito dos SI estiver num nível não explícito, ou seja, tácito, o corpo técnico que efetivamente estiver responsável pela manutenção – sobretudo não sendo autores do SIL – terá dificuldades para reverter a ineficácia e a ineficiência na manutenção dos SIL.

Nessas organizações, o corpo técnico responsável pelos SIL, geralmente se depara com dificuldades, principalmente por não conhecê-los, em identificar e localizar os pontos de mudanças e seus impactos nos SIL.

Ao fundamentar-se em sua experiência profissional de mais de dez anos numa organização financeira bancária, na área de desenvolvimento e manutenção de sistemas, o autor desta dissertação viu-se frente à premência de analisar algumas situações críticas no dia-a-dia desse ambiente, especificamente com a questão da manutenção dos SIL. A frequência com que fatos internos – por exemplo, lançamento de novos produtos – e fatos externos, como mudanças de políticas econômicas e medidas provisórias do governo, influenciam a todo o momento os SIL e o corpo técnico responsável.

Desse modo, as equipes de trabalho na área de SI devem ser capazes de ter uma flexibilidade e profundo conhecimento a respeito dos SI envolvidos; todavia, nem sempre isso ocorre. As pessoas que compõem o corpo técnico do SI são suscetíveis a doenças, acidentes, podem estar em férias; enfim, estão sujeitas uma série de variáveis tornando-as indisponíveis em momentos em que há necessidade do respectivo conhecimento para efetivar as mudanças necessárias, num tempo muito restrito. Tal situação configura, em sua maioria, o ambiente dos SI das organizações financeiras bancárias.

Frente a essa conjuntura por vezes adversa, procura-se abordar neste trabalho organizações com algumas especificidades, quais sejam:

- a) organizações financeiras bancárias;
- b) ambiente de grande porte, *mainframe*;
- c) tecnologias de terceira geração, linguagem COBOL; e
- d) área de desenvolvimento e manutenção de SI.

### 1.3. Objetivos

O cenário apresentado está imerso em um mundo com duas perspectivas, o *plano conceitual*, das idéias, do metafísico e o *plano físico* ou sensitivo, conforme [Figura 1.2]. A associação, o relacionamento entre esses dois planos é, sem dúvida, objeto principal deste trabalho. A partir de uma idéia ou especificamente de um conceito, busca-se chegar ao plano físico ou mundo real, natural e ou artificial, considerando também o caminho inverso. Por meio das tecnologias da IA, construção de *ontologias* e aplicação de um ambiente de Sistemas Multiagentes (SMA), pretende-se viabilizar a criação de artefatos que tornem possível um ambiente computacional para solução dos problemas apresentados.

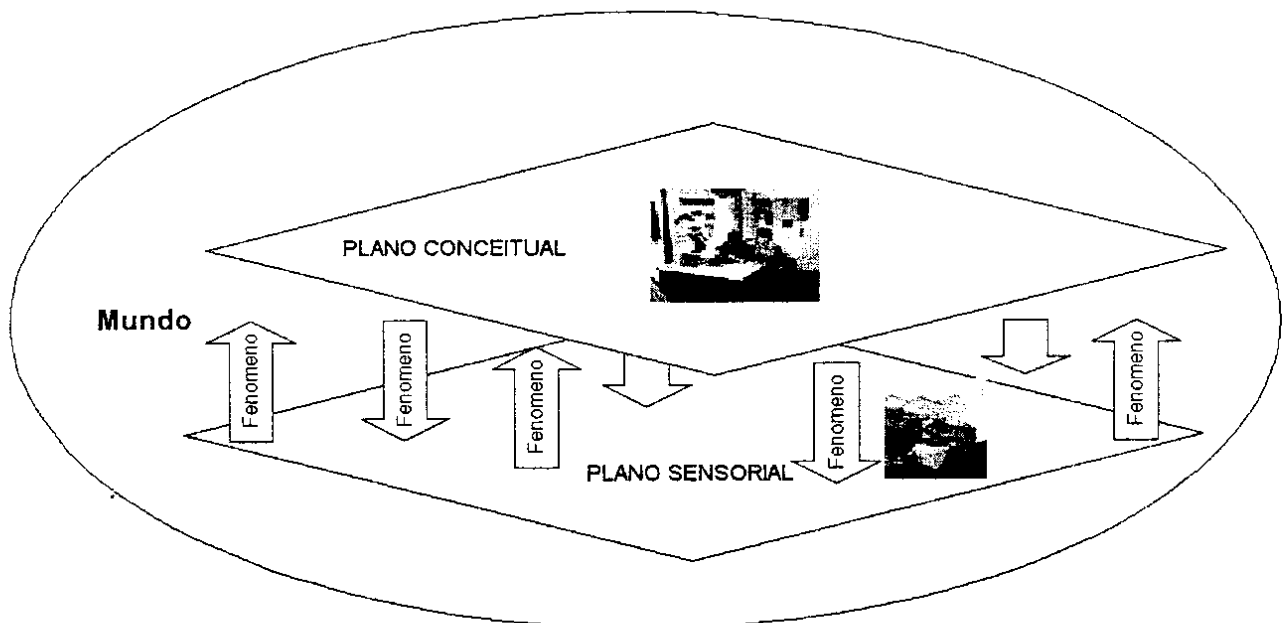


Figura 1.2: Mundo

#### 1.3.1. Objetivo geral

Propõe-se uma metodologia para compartilhar o conhecimento nos sistemas de informação legados SIL, embasados na aplicação de recursos tecnológicos como a Ontologia e Sistemas Multiagentes. A metodologia e a ferramenta computacional, Ontolegacy, fruto deste trabalho, tem o propósito de minerar, extrair termos e conceitos em códigos fonte e realizar o processo de recuperação dos conceitos em SIL, conforme o paradigma da Ontologia.

### 1.3.2. Objetivo específico

A matéria-prima, isto é, o foco da análise da aplicação do sistema Ontolegacy se concentra fundamentalmente nos códigos fonte escritos na linguagem de terceira geração, COBOL.

Têm-se os objetivos específicos de:

- a) recuperar os conhecimentos tácitos dos SIL, que o corpo técnico detém;
- b) criar uma ferramenta, ou seja, um facilitador das manutenções nos SIL;
- c) formalizar a representação do conhecimento por meio da Ontologia;
- d) manter as Ontologias atualizadas por meio de SMA;
- e) compartilhar o conhecimento por meio das Ontologias criadas dos SIL; e
- f) reutilizar as Ontologias para outros SIL.

## 1.4. Organização

O capítulo 2 examina a questão atual da gestão do conhecimento nas organizações e procura apresentar os princípios e teorias do conhecimento, como também o processo para a manutenção dele nas organizações. Na seqüência, são apresentadas as características dos SIL e a importância desses sistemas, no contexto das grandes organizações. São vistos também neste capítulo os fundamentos da Ontologia, propriedades, aplicações, métodos construtivos e sua aplicabilidade no Sistema Ontolegacy. Ainda são apresentados o ambiente de SMA, com seus conceitos, aplicações e contextualização neste trabalho.

No capítulo 3, é exposto o Sistema Ontolegacy, descrevendo sua implementação, plataforma utilizada, definição e funcionamento do SMA incorporado ao Ontolegacy, processo de mineração dos códigos fonte e sua relação com a formalização da Ontologia. Mostra-se também o funcionamento da plataforma *Knowledge Integration & Sharing Framework* (KISF), associado ao Ontolegacy.

No capítulo, 4 apresenta-se o experimento utilizado conforme a metodologia e o sistema *Ontolegacy*. O experimento tem como base um Sistema de Informação legado no domínio de conhecimento *Arrecadação da Previdência Social*. Ao final, são demonstrados os resultados obtidos com o uso do Sistema Ontolegacy.

O capítulo 5 apresenta as conclusões do trabalho, estado da arte e as sugestões para trabalhos futuros.



## Capítulo 2

### Recorrentes Conceituais e Tecnológicos

A tecnologia computacional e determinados conceitos da Inteligência Artificial Distribuída (IAD) e de Sistemas de Informação constituem o suporte da presente dissertação. A Gestão do Conhecimento e os SIL, associados à Ontologia de sistemas e Agentes de *software*, originaram a essência da metodologia e da ferramenta computacional construída, o *Ontolegacy*. Estes recorrentes conceituais e tecnológicos são tratados na continuidade, com o intuito de permitir a compreensão e a visão das relações entre eles, de modo a demonstrar a abordagem adotada.

#### **2.1. Gestão do conhecimento**

A gestão do conhecimento tem adquirido espaço e destaque considerável dentro das organizações como alternativa e nova forma de gestão. Entra em confronto com o modelo tradicional e atual, o qual se mostra ineficiente na exigência da mobilidade, inovação e como diferencial competitivo.

Terra [TRR00] afirma que o conhecimento se tornou o recurso econômico mais importante para a competitividade das empresas e dos países.

Davenport e Prusak [DVP98] consideram que o conhecimento gerado ao longo dos anos é considerado um bem organizacional, um patrimônio da organização; apesar de ser um bem intangível, em oposição a máquinas e equipamentos, reconhece-se como fundamental para sua sobrevivência.

De acordo com Drucker, “na nova economia, o conhecimento não é apenas mais um recurso, ao lado dos tradicionais fatores de produção, trabalho, capital e terra, mas o único recurso significativo atualmente”.

Para as organizações, de um modo geral, o conhecimento não é novidade, mas a sua sistematização e a utilização como diferencial é acentuada pelos modelos de gestão do conhecimento.

De fato, a gestão do conhecimento envolve processos essenciais como: a identificação, aquisição, desenvolvimento, compartilhamento, distribuição, utilização e retenção do conhecimento, destaca Probst [PRR02]. Com uma visão similar, Nonaka e Takeuchi consideram que a gestão do conhecimento “refere-se à criação sistemática de novos conhecimentos, disseminando-os por toda a organização e incorporando-os rapidamente em novas tecnologias e produtos” [NNT97]. Esta abordagem japonesa da gestão do conhecimento [Figura 2.1] tem como objetivo principal subsidiar a inovação contínua dentro das empresas, fator essencial para a vantagem competitiva.

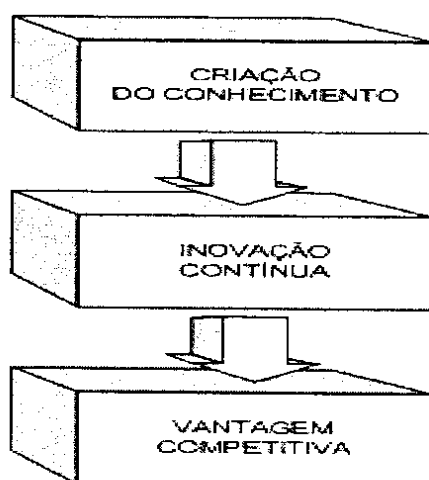


Figura 2.1: Fator desencadeante [NNT97]

Numa visão construtivista, Stano e Leite [STL01] consideram que a gestão do conhecimento possui um papel mediador entre o sujeito e objeto. Assim, o conhecimento não se dá somente na relação entre sujeito e objeto, mas, principalmente, pela mediação da sociedade, por intermédio dos signos e de outras pessoas. Desse modo, torna-se importante predispor meios e pessoas que possam facilitar a compreensão, a assimilação e a própria criação de conhecimentos.

Eis alguns mediadores que facilitam um processo de construção do conhecimento compatível com os objetivos da organização: promover o encontro, confrontar o diferente, estabelecer relações de cortesia e de cooperação entre os funcionários.

Em seu processo criativo, a gestão do conhecimento repercute no ambiente em que ele está inserido. Há necessidade de um novo ambiente propício à gestão do conhecimento, porque tal criação está atrelada a uma nova configuração de organização.

Assim, o desenvolvimento de uma visão compartilhada, relações interpessoais, novas relações de poder, formas de remuneração, reconhecimento e de trabalho, entre outros, são fatores que provêm do que se considera a cultura da organização, pois retratam o conjunto de inter-relações entre crenças e pressupostos organizacionais [STL01].

Conforme Davenport e Prusak [DVP98], mesmo antes da época da “organização que aprende”, bons gerentes já valorizavam a experiência e o *know-how* de seus funcionários, isto é, seu conhecimento. Muitas empresas perceberam logo que necessitam de uma abordagem aleatória – e até mesmo inconsciente – do conhecimento corporativo para vencer na economia atual e futura.

### 2.1.1. Origem do conhecimento - abordagem filosófica

Sabe-se que os fundamentos filosóficos do conhecimento, *epistemologia*, são estudados há séculos. Remontam à época da antiga Grécia, sobretudo com Platão e Aristóteles. Em suas obras, Platão definiu o conhecimento como a “crença verdadeira justificada”.

Na evolução histórica da epistemologia, encontram-se duas vertentes nítidas, mesmo sendo complementares: o *racionalismo* e o *empirismo*. São representantes legítimos destas correntes Platão e Aristóteles e, mais recentemente, Descartes e Locke.

O racionalismo considera ser possível a aquisição do conhecimento por dedução pelo raciocínio, que o verdadeiro conhecimento não é produto da experiência sensorial, mas de um produto de um processo mental ideal, dito conhecimento *a priori*. Por exemplo, a determinação do fenômeno quantidade. Posso determinar *a priori* e construir o grau de sensação da luz solar acrescentando aproximadamente duzentas mil vezes a da Lua.

Por sua vez, o empirismo aponta, em essência, que o conhecimento é adquirido por indução, a partir de experiências sensoriais. Por meio dos cinco sentidos [Figura 2.2], experimentamos sensações e adquirimos o conhecimento sobre determinado objeto.

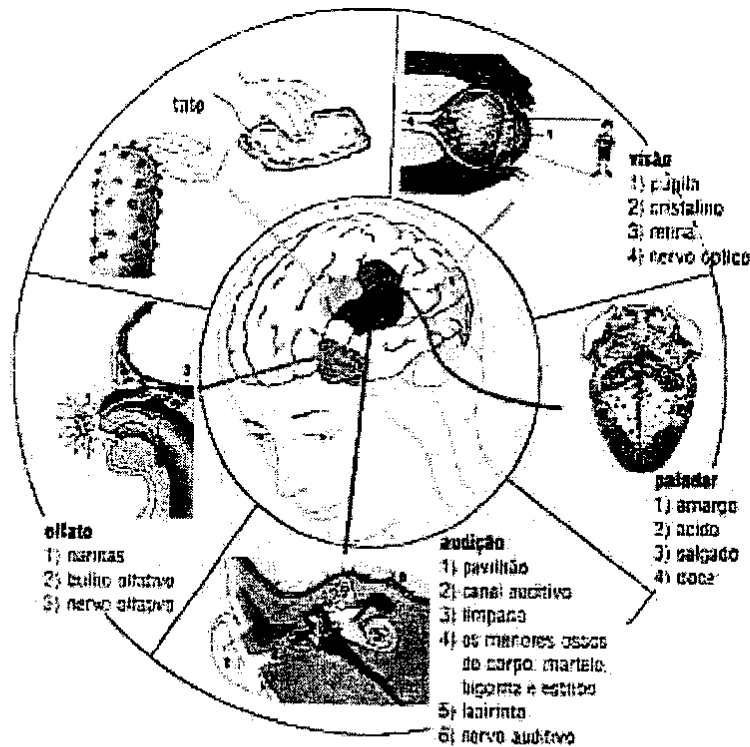


Figura 2.2: Os cinco sentidos [AFH04]

Filosoficamente, não havia um consenso entre o racionalismo e o empirismo. O pensamento racionalista de Descartes era totalmente contraditório a John Locke, defensor do empirismo por natureza. Assim transcorreu durante todo o período histórico da epistemologia até o século XVIII.

Com uma abordagem inovadora e diferenciada, Immanuel Kant, filósofo alemão do século XVIII, faz uma crítica ao racionalismo e ao empirismo. Na época, o empirismo representado por Hume foi criticado por Kant, afirmando que todo conhecimento começa com a experiência, mas não deriva todo da experiência. Kant faz uma síntese das duas correntes filosóficas abordando a questão do racionalismo, concluindo que a mente humana não é uma *tabula rasa*, tampouco o conhecimento não é inato, mas há um conjunto de fatores que determinam o conhecimento “*a coisa em si*”. Não se pode chegar ao conhecimento primeiro

“*a coisa em si*”, ao conhecimento original, a Ontologia, mas apenas aos seus *fenômenos*, o que nos é dado conhecer. Como síntese, vê-se que a origem e o processo da aquisição do conhecimento são complexos e questionáveis no universo de posições e correntes de pensamentos racionalistas, empiristas e a interacionista, que seria um misto das duas correntes. No contexto deste trabalho, considera-se a visão Kantiana como a mais adequada, relativamente ao aspecto da origem do conhecimento.

### 2.1.2. Conhecimento e a cadeia de valores

Na informática, a transformação e aquisição do conhecimento se dão na forma de dados e informações.

Conforme Probst [PRR02], a aplicação de regras de sintaxe sobre símbolos se tornam dados e, quando permitem interpretação num contexto específico, fornecem informações ao receptor. Acrescenta Probst que “quando as informações são interligadas, estas podem ser usadas em um campo de atividade específico, e a isso se pode chamar de conhecimento”.

A informação proporciona um novo ponto de vista para a interpretação de eventos ou objetos, o que torna visíveis significados antes invisíveis ou lança luz sobre conexões inesperadas. Conseqüentemente, torna-se um meio ou material necessário para extrair e construir o conhecimento [NNT97].

A transformação das informações em conhecimento demanda tempo. Está relacionado com dado e informação, o que implicaria então um conjunto formado por experiências, valores, informação de contexto e criatividade aplicado à avaliação de novas experiências e informações. Esta abordagem, útil para aplicações ao trabalho e às organizações, identifica o conhecimento como algo inseparável das pessoas [TXR00].

Pelo motivo do conhecimento não se encontrar somente em documentos, bases de dados e sistemas de informações, leva a consideração dos processos, práticas dos grupos e a experiência acumulada pelas pessoas. Ele é avaliado pelas decisões e ações que desencadeia.

Pela total desconsideração da importância do saber no passado, hoje empresas lutam por compreender melhor o que sabem, o que necessitam descobrir e o que devem fazer com as experiências acumuladas de que dispõem.

Assim, muitas empresas passaram a se preocupar com o seu “capital intelectual”, com a sua “inteligência competitiva”; enfim, com a gestão do conhecimento. Os novos

à construção automatizada de registros do que definimos como conhecimento explícito. São grandes *repositórios de dados* que registram a experiência da organização, além da grande diversidade de dados sobre o seu ambiente interno (processos, rotinas) e externo (clientes, fornecedores, concorrentes).

#### **b) Conhecimento Tácito**

Para Nonaka & Takeuchi [NNT97], o conhecimento tácito está profundamente relacionado e incorporado à experiência pessoal além de envolver fatores intangíveis como, por exemplo, crenças pessoais, perspectivas, valores pessoais. Esses autores afirmam que há uma dificuldade muito grande em verbalizar, escrever, enfim, exteriorizar de modo explícito o que sabemos. Citam também [NNT97] que há presentes dois elementos no conhecimento tácito, elementos cognitivos que consiste de modelos mentais, crenças e perspectivas tão arraigados que são tidos como algo certo, não sujeitos a fácil manifestação. E as habilidades técnicas, de difícil especificação, incorporada ao termo *know-how*.

Como o conhecimento tácito é altamente pessoal e difícil de formalizar, dificulta a transmissão e o compartilhamento com o outro. Incluem-se nessa categoria de conhecimento as conclusões, os – intuições e pressentimentos altamente subjetivos – e palpites subjetivos. Devido à natureza subjetiva e intuitiva do conhecimento tácito, há uma dificuldade imensa no processamento e na transmissão do conhecimento adquirido por qualquer modo sistemático ou lógico. Apesar das dificuldades para que seja gerido, o conhecimento tácito é o principal alvo na maioria dos esforços de gestão do conhecimento, porque representa o conhecimento mais valioso e estratégico da organização.

#### **2.1.4. Conversão e criação do conhecimento**

Nonaka e Takeuchi [NNT97] consideram que na criação do conhecimento organizacional existem processos característicos que permitem alcançar esta criação. Tais processos estão profundamente relacionados às entidades: conhecimentos tácitos e explícitos. São vistos como entidades complementares, pois a partir da interação entre elas chega-se ao que se denomina *conversão do conhecimento*, a qual pressupõe que todo processo tenha elementos de relacionamentos sociais, entre indivíduos, e não num processo isolado, centrado somente em um indivíduo. A conversão é classificada [NNT97] em: socialização, externalização, combinação e internalização [Figura 2.4].

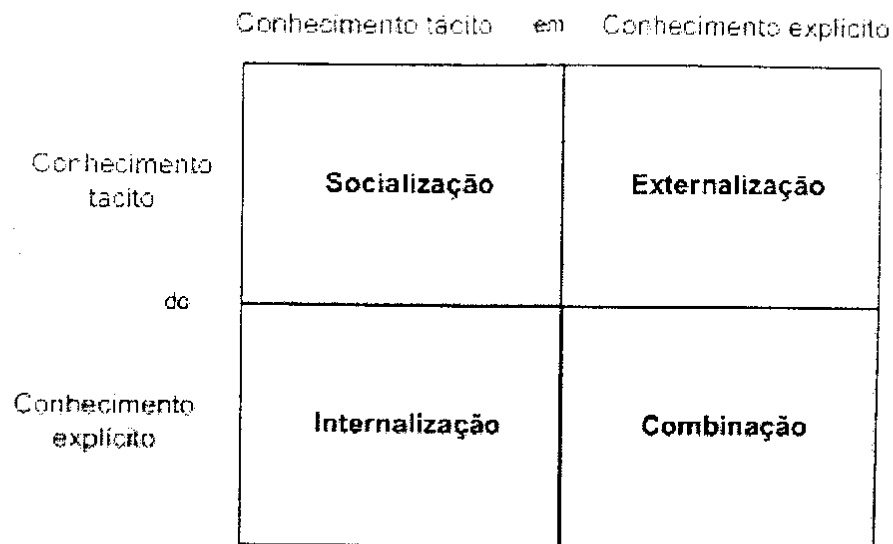


Figura 2.3: Modos de conversão do conhecimento [NNT97]

a) *Socialização* – do conhecimento tácito em conhecimento tácito:

A partir do processo do compartilhamento das experiências, dá-se a criação do conhecimento tácito. Sem alguma experiência compartilhada, é extremamente difícil para uma pessoa projetar-se no processo de raciocínio do outro indivíduo.

b) *Externalização* – do conhecimento tácito em conhecimento explícito:

Processo de articulação do conhecimento tácito em conceitos explícitos, em geral expressos por metáforas, analogias, conceitos, hipóteses e modelos. Após criados os conceitos explícitos, podem ser modelados, expressos em linguagem sistemática e lógica coerente.

c) *Combinação* – do conhecimento explícito em conhecimento explícito:

Processo de sistematização de conceitos em um sistema de conhecimento. As pessoas também são capazes de combinar o componente isolado do conhecimento explícito para a constituição de um novo todo. A reconfiguração das informações por meio da classificação, do acréscimo, da combinação e da categorização do conhecimento explícito pode levar a novos conhecimentos. Usos criativos de redes de computadores de comunicações e bancos de dados em larga escala facilitam esse modo de conversão do conhecimento.

d) *Internalização* – do conhecimento explícito em conhecimento tácito:

Processo de incorporação do conhecimento explícito no conhecimento tácito, para o qual são necessárias a verbalização e a diagramação do conhecimento, sob a forma de documentos, manuais ou histórias orais. Documentos ou manuais facilitam a transferência do conhecimento explícito para outras pessoas.

### 2.1.5. Espiral do conhecimento

Todos os processos citados para conversão do conhecimento possuem uma abordagem epistemológica. Para Nonaka e Takeuchi [NNT97], além da dimensão epistemológica, existe também a dimensão ontológica, em que o nível de conhecimento está distribuído em âmbito individual, grupal, organizacional e interorganizacional. Ou seja, as organizações precisam transpor e movimentar o conhecimento tácito que está nos indivíduos e ampliá-lo organizacionalmente, utilizando-se dos quatro modos de conversão do conhecimento e elevando-os a patamares ontológicos superiores.

A este processo Nonaka e Takeuchi nominaram a *espiral do conhecimento*, que significa a interação do conhecimento tácito e explícito que sai do nível ontológico individual e chega ao nível ontológico interorganizacional. Isto é, sai do individual, amplia comunidades de interação, chega a transpor fronteiras entre seções, departamentos, divisões e organizações [Figura 2.4].

### 2.1.6. Transferência do conhecimento

A espiral do conhecimento é uma das abordagens do processo de transferência do conhecimento. Porém, Davenport e Prusak [DVP98] preconizam que, para transferência do conhecimento, há necessidade de um intenso diálogo, uma “conversa” entre os indivíduos que o possuem. Consideram que a transferência espontânea e não estruturada do conhecimento seja vital para o sucesso da empresa. Para isso, alguns fatores são essenciais, sejam eles formais ou informais. O simples fato de existirem ambientes físicos, salas, para os indivíduos “conversarem” sobre dúvidas e problemas profissionais facilita em muito o processo de compartilhar o conhecimento, principalmente o tácito.



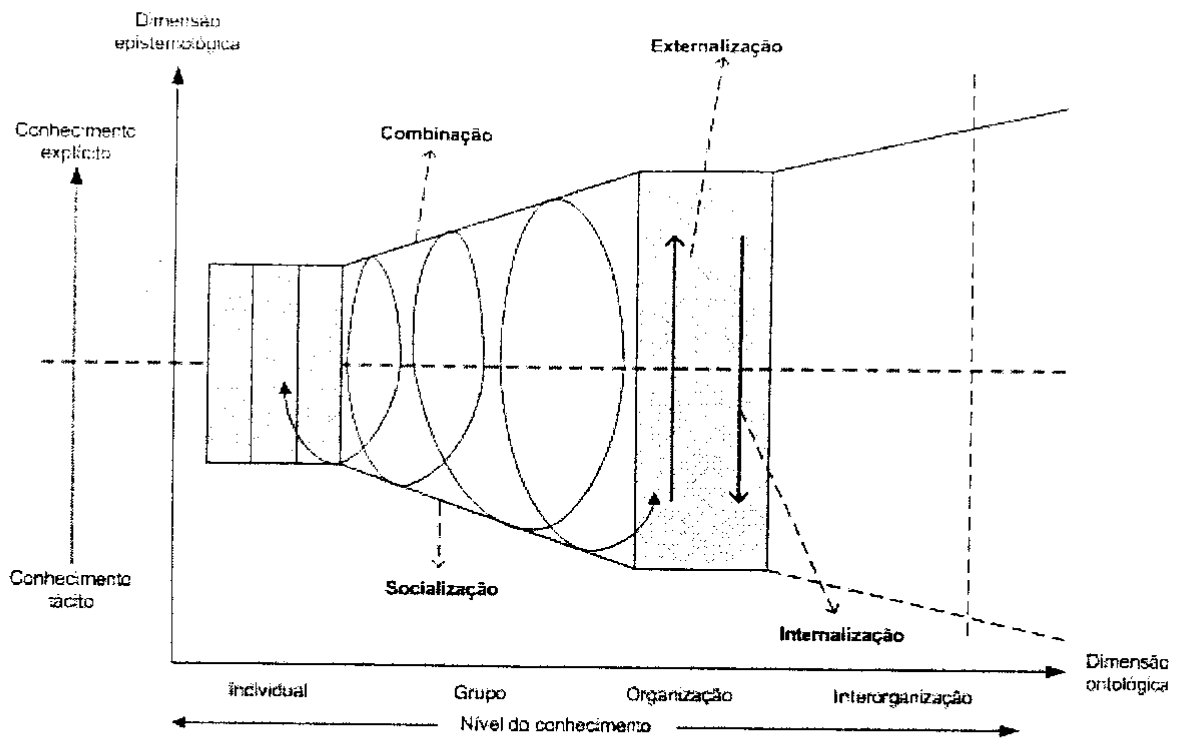


Figura 2.4: Espiral do Conhecimento [NNT97]

Probst [PRR02] ressalta que o compartilhamento e a distribuição de conhecimento ocupam uma posição de destaque na gestão do conhecimento, apoiando fatores competitivos vitais como tempo e qualidade; por isso possuem alavancagem devido a sua importância para outros aspectos da gestão do conhecimento.

Assim, não faz sentido a simples disponibilização do conhecimento, o qual deve ser transmitido e absorvido caracterizando sua transferência. A gestão do conhecimento assume, hoje, um papel fundamental para as organizações que precisam manter o processo de inovação contínua e, conseqüentemente, sua competitividade. A fim de criar conhecimento e permitir sua transferência, lança-se mão de outro recorrente conceitual que é a Ontologia, mencionada neste item e tratada no item 2.2, a seguir.

## 2.2. Ontologias

Esta seção aborda um entendimento, um apontamento, uma direção do que seja *Ontologia*, como ciência e por sua contribuição para a representação do conhecimento. Faz-se inicialmente uma abordagem filosófica demonstrando suas origens e bases, objetivando deixá-la o mais transparente possível. Mostra-se a origem da Ontologia no campo filosófico, estabelecendo uma relação com a IA presente, buscando conceitos, ferramentas e aplicações que auxiliem na contribuição do uso da Ontologia, dentro de um sistema computacional, objetivo maior deste trabalho.

### 2.2.1. Ontologia e Filosofia

A Ontologia é uma área da ciência com conceitos fortemente associados à filosofia, ao mundo das idéias; ou seja, à metafísica. No subcapítulo anterior 2.1, tratou-se da teoria do conhecimento e suas vertentes. Como a Ontologia está intimamente relacionada ao conhecimento, recorre-se aos princípios anteriormente expressos a respeito do mesmo.

A metafísica se apóia na imaginação e na percepção; a imaginação tem o papel de conceber e criar imagens e a percepção permite a vivência e a experiência do mundo real. Com tais elementos, tem-se o mundo das idéias no qual estamos inseridos e que cotidianamente permite nossa sobrevivência. Apesar de algumas correntes filosóficas questionarem esse realismo externo, parte-se do princípio que a percepção deste mundo externo através dos sentidos nos permite ter uma idéia de que realmente existe, está lá, e é do senso comum.

Searle [SRL00] afirma que “o realismo externo, mundo, já existia mesmo antes do aparecimento do ser humano e que são fenômenos independentes da mente, ou seja, independentes de nós, tais como o átomo de hidrogênio, as placas tectônicas, as árvores e as galáxias”. Mas nem todos os fenômenos no mundo são independentes da mente. Por exemplo, a existência de dinheiro, da propriedade, do casamento, das guerras, dos jogos e das festas que dependem de agentes humanos conscientes [SRL00].

Na Grécia antiga, Aristóteles foi o primeiro pensador a apresentar uma definição de Ontologia, ao afirmar que ela é a “ciência que estuda o ser enquanto ser”, a partir do momento

em que uma coisa possui alguma *realidade*, ela *é*. O *ser* é dado pela *forma* e *matéria* das coisas, à *unidade* de forma e a matéria é a *substância*.

Na metafísica aristotélica, a relação de causalidade é a que contribui para a realidade de um ser. Essas causas foram definidas por Aristóteles em:

- a) *causa material*, tudo aquilo de que uma coisa é feita, por exemplo, o mármore de que é feito a estátua;
- b) *causa formal*, o que define o objeto, distinguindo-o dos demais: estátua de homem e não de cavalo;
- c) *causa final*, a idéia da estátua, existente como projeto na mente do escultor, e que o levou a talhar o bloco de mármore para dele fazer uma estátua de homem;
- d) *causa eficiente*, o agente, no caso o escultor, aquele que faz o objeto, atualizando potencialidade de determinada matéria.

A causa formal está intimamente ligada à final, pois seria sempre em vista de um fim que os seres – naturais ou artificiais – são criados e se transformam: a finalidade é que determinaria o que os seres são ou vêm a ser [PNS77].

Para Immanuel Kant, a *Ontologia* está relacionada ao caráter primário da ciência. Esse filósofo alemão contemporâneo afirmava que todo conhecimento começa com a *experiência*, mas isso não quer dizer que tudo derive da experiência, pois há também a contribuição da *faculdade de conhecer*, que não tem um papel apenas passivo [KNT04].

Sob tal perspectiva, o conhecimento não representa as coisas tal como são, mas sim como são para cada um. A realidade em si não é possível ser conhecida, tal como Deus, a imortalidade e a liberdade; acerca de ambas, apenas se pode ter uma certeza moral [KNT04].

Para Kant, pode-se chegar a conceber a *Ontologia* como o “estudo dos conceitos *a priori* que residem no entendimento e têm seu uso na experiência”; isto é, trata-se do “estudo do conjunto de conhecimentos da razão pura”. A *Ontologia* é então “uma parte da metafísica que constitui um sistema de todos os conceitos do entendimento e dos princípios, mas só na medida em que se referem a objetos que podem ser dados aos sentidos e, portanto justificados pela experiência” [KNT04].

Enfim, o que os filósofos gregos e contemporâneos buscam esclarecer sobre a *Ontologia* é que seja uma área das ciências que trata da essência, do caráter primário das coisas, ou seja, o que de fato é a “realidade externa” independentemente da interferência ou visão da mente, conhecer a essência do próprio ser.

## 2.2.2. Ontologias e Inteligência Artificial

Embora a *Ontologia* esteja associada primeira e fundamentalmente à filosofia, na Inteligência Artificial (IA), o termo *Ontologia* tem uma conotação diferente do utilizado nessa área do saber. Empréstase o termo para uso na IA de modo a ser relacionado sobretudo à representação do conhecimento. De acordo com Gruber [GRB93], uma *Ontologia* é “uma especificação formal e explícita de uma conceituação compartilhada”, de maneira que “formal” significa legível para computadores; “especificação explícita” diz respeito a conceitos, propriedades, relações, funções, restrições, axiomas, explicitamente definidos; “compartilhado” quer dizer conhecimento consensual, e “conceituação” implica um modelo abstrato de algum fenômeno do mundo real.

Uschold entende que *Ontologia* é “uma reunião explícita de conhecimento compartilhado em uma área específica” [USC96]. Conseqüentemente, pode resolver problemas de comunicação entre pessoas, organizações e aplicativos.

John Sowa [SWA04], influenciado por Leibniz, propõe a seguinte definição: “O propósito da *Ontologia* é o estudo de categorias de coisas que existem ou devem existir em alguns domínios. O produto de cada estudo é chamado de *Ontologia*, é um catálogo de tipos de coisas que são assumidos para existir em um domínio de interesse D da perspectiva de uma pessoa que utiliza uma linguagem L com a finalidade de falar sobre D [...]”.

Para a IA, o termo se refere a um artefato de engenharia, constituído de um vocabulário de termos organizados em uma taxonomia, suas definições e um conjunto de axiomas formais usados para criar novas relações e para restringir as suas interpretações, conforme um sentido pretendido [GUA98].

Com o intuito de resolver tal impasse terminológico, Guarino [GUA98] propõe que a definição da comunidade da computação seja adotada para o termo *Ontologia* e que para a definição filosófica seja dado o nome de conceituação. Guarino revê e estende a definição de Gruber do seguinte modo: “Uma *Ontologia* é uma teoria lógica que corresponde ao significado intencional de um vocabulário formal, ou seja, um comprometimento ontológico com uma conceituação específica do mundo. Os modelos intencionais de uma linguagem lógica usando este vocabulário são controlados por seu comprometimento ontológico. Este comprometimento e a conceituação subentendida são refletidos na *Ontologia* pela aproximação desses modelos intencionais” [GUA98].

A mesma definição é ilustrada pela figura abaixo:

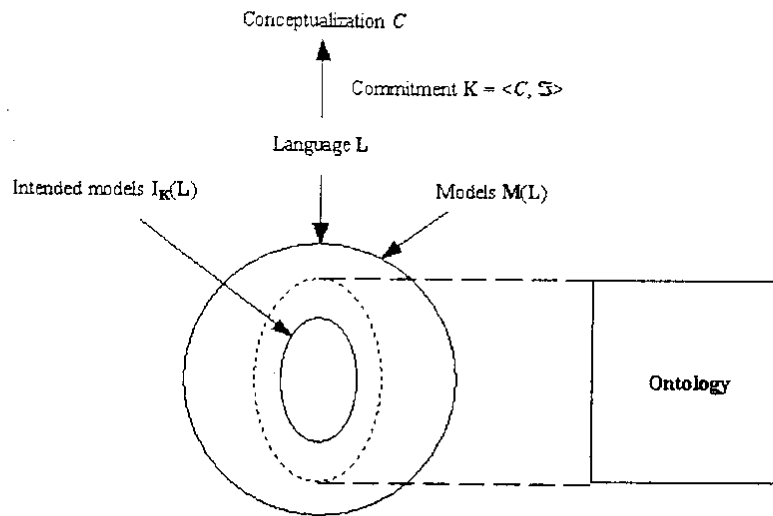


Figura 2.5: Definição de Ontologia [GUA98]

A definição indica claramente as propriedades importantes de uma *Ontologia*. Cada palavra usada para uma relação ou propriedade deve explicitamente declarar seu compromisso ontológico. O comprometimento ontológico dirige o uso e o significado da palavra, e, por consequência, o uso de sua aplicação. Em suma, a *Ontologia* é uma aproximação formal do mundo real. Outra propriedade de suma importância que Guarino inclui é o fato de que a *Ontologia* está para uma dependência da linguagem enquanto a conceituação está para uma independência da linguagem [GUA98]. Isto reflete no seguinte aspecto: duas *Ontologias* diferentes podem compartilhar o mesmo conceito de uma coisa, por exemplo, o conceito do preço, mas podem usar palavras diferentes de seu vocabulário a essa conceituação. Isto significa ser possível conseguir a interoperabilidade entre *Ontologias* diferentes se a conceituação é a mesma em algum sentido. Mas significa também que, se duas aplicações usarem as mesmas palavras, podem não tratar da mesma conceituação [GUA98]. Mesmo sem um consenso sobre sua definição, as *Ontologias* apresentam características comuns. A seção seguinte apresenta classificações das *Ontologias* com respeito ao conteúdo.

### 2.2.3. Classificação das Ontologias

Na medida em que tem crescido o interesse por Ontologias em IA, estas têm sido utilizadas de diferentes maneiras. Com base no seu conteúdo, Ontologias podem ser classificadas em [GUA98]:

- a) *Ontologias genéricas*: descrevem conceitos bastante gerais, tais como, espaço, tempo, matéria, objeto, evento, ação, que são independentes de um problema ou domínio particular;
- b) *Ontologias de domínio*: expressam conceituações de domínios particulares, descrevendo o vocabulário relacionado a um domínio genérico, tal como Medicina;
- c) *Ontologias de tarefas*: expressam conceituações sobre a resolução de problemas, independentemente do domínio em que ocorram; isto é, descrevem o vocabulário relacionado a uma atividade ou tarefa genérica, tal como diagnose ou vendas;
- d) *Ontologias de aplicação*: descrevem conceitos dependentes do domínio e da tarefa particulares. Estes conceitos freqüentemente correspondem a papéis desempenhados por entidades do domínio quando da realização de uma certa atividade; e
- e) *Ontologias de representação*: explicam as conceituações que fundamentam os formalismos de representação de conhecimento.

### 2.2.4. Metodologias para construção de Ontologias

O desenvolvimento de metodologias tem sido feito com o intuito de sistematizar a construção e a manipulação das Ontologias [LPZ99]. A seguir, apresentam-se sinteticamente algumas metodologias conhecidas para a construção de *Ontologias*.

Tabela 2.1: Metodologias para Construção de Ontologias

Metodologia / Autor	Descrição
Enterprise Uschold e King	Identificam o propósito, os conceitos e o relacionamento entre os conceitos, além dos termos utilizados para codificar a Ontologia e, em seguida, documentá-la [USC98].
Methontology	Constrói uma Ontologia por reengenharia sobre outras, utilizando o conhecimento de domínio. As atividades principais são: especificação, conceituação, formalização, implementação e manutenção [LPZ99].
Tove	Representação computacional de estruturas, atividades.

(Toronto Virtual Enterprise) Fox e Gruninger CYC	processos, informações, recursos, pessoas, comportamento, objetivos, regras sobre negócios, sobre governos ou outras informações de outras organizações [FXG98]. Codifica manualmente o conhecimento implícito e explícito das diferentes fontes, e, quando já se tem conhecimento suficiente na Ontologia, um novo consenso pode ser obtido por ferramentas que utilizam linguagem natural [LNG90].
Sensus	Construção taxionômica de terminologias com 70.000 nós, onde o conhecimento adicional pode ser colocado em <i>Framework</i> . É também uma reorganização e extensão do <i>WordNet</i> , projeto da universidade de Princeton. Constrói Ontologias a partir de outras Ontologias, identificando os termos relevantes para o domínio e ligando-os à Ontologia mais abrangente. Um algoritmo monta a estrutura hierárquica do domínio [SWT96].
On-To-Knowledge	Auxilia a administração de conceitos em organizações, identificando metas para as ferramentas da gestão de conhecimento e utilizando cenários e contribuições dos provedores / clientes de informação da organização [STB01].

As metodologias apresentadas possuem abordagens e características diversas, de modo que se torna improvável ocorrer a unificação das propostas em uma única metodologia. Para verificar a utilidade das metodologias e compará-las, é necessário avaliar a Ontologia resultante da aplicação de cada metodologia. A diversidade de abordagens pode ser um fator que dificulta a formulação de metodologias de avaliação formais. Alguns critérios para a avaliação de Ontologias são colocados por Gruber [GRB95], com o objetivo de nortear o processo da construção de uma Ontologia em todas as suas etapas, conforme a seguir:

- a) *clareza*: uma Ontologia deve comunicar efetivamente o significado projetado dos termos definidos e, assim, suas definições devem ser objetivas. Onde for possível, uma definição completa é preferida em relação a uma definição parcial e todas devem ser documentadas em linguagem natural, de modo a reforçar a clareza;
- b) *coerência*: uma Ontologia deve ser coerente, isto é, deve comportar apenas inferências consistentes com as definições. Coerência deve ser observada, também, em relação a conceitos definidos informalmente. Se uma sentença for passível de ser inferida a partir dos axiomas da Ontologia, contradiz uma definição ou exemplo dado informalmente, então a Ontologia é incoerente;
- c) *extensibilidade*: uma Ontologia deve ser projetada para antecipar usos do vocabulário compartilhado e, portanto, sua representação deve poder ser estendida e especializada. Em

outras palavras, deve ser possível definir novos termos para usos especiais, com base no vocabulário existente, sem haver necessidade de rever definições;

- d) *compromissos de codificação mínimos*: a conceituação deve ser especificada no nível de conhecimento sem depender de uma tecnologia particular de representação de conhecimento. Uma tendência de codificação surge quando escolhas de representação são feitas puramente para a conveniência de notação ou implementação. Assim, essa tendência deve ser minimizada, já que agentes compartilhando conhecimento podem ser implementados em diferentes sistemas e paradigmas de representação; e
- e) *compromissos ontológicos mínimos*: o conjunto de compromissos ontológicos de uma Ontologia deve ser o menor possível, capaz de suportar as atividades planejadas de compartilhamento de conhecimento. Uma Ontologia deve fazer tão poucas imposições quanto possíveis sobre o mundo que está sendo modelado, permitindo que as partes comprometidas com a Ontologia fiquem livres para especializar e instanciar a Ontologia na medida do necessário. Uma vez que compromissos ontológicos são baseados no uso consistente de um vocabulário, eles podem ser minimizados pela especificação de uma teoria mais fraca (que admita um maior número de modelos), contendo definições restritas apenas para os termos essenciais à comunicação consistente do conhecimento da teoria.

Na concepção de Noy e McGuinness, a construção de uma *Ontologia* passa por etapas, e nestas etapas criam-se componentes básicos essenciais a sua construção, que são [NYG00]:

- a) *classes* – organizadas em uma taxonomia;
- b) *relações* – representam o tipo de interação entre os conceitos de um domínio;
- c) *axiomas* – utilizados para modelar sentenças sempre verdadeiras; e
- d) *instâncias* – usadas para representar elementos específicos, ou seja, os próprios dados.

Entre as metodologias existentes para a construção de *Ontologias*, destaca-se a de Uschold e Gruninger [UCG96], que propõem uma abordagem sistemática para a sua construção, conforme atividades definidas e organizadas, descritas na seqüência:

- a) *identificação do propósito* – é importante saber claramente porque uma Ontologia está sendo construída, qual é o uso esperado e o seu potencial de usuários;
- b) *construção da Ontologia* – envolve três passos principais: captura, codificação e integração com Ontologias existentes. A captura da Ontologia envolve a identificação dos conceitos e relações relevantes no domínio de interesse, a geração de definições textuais precisas para estes elementos e o estabelecimento de termos para referenciá-los. Na



codificação, a conceituação capturada no estágio anterior é representada em alguma linguagem formal. Durante os passos de captura e codificação, é possível que Ontologias existentes sejam reutilizadas e, portanto, é necessário integrá-las;

- c) *avaliação* – uma Ontologia deve ser avaliada quanto a competência, especificações de requisitos e/ou do mundo real; e
- d) *documentação* – todas as decisões importantes devem ser documentadas, tanto no que tange aos principais conceitos definidos na Ontologia como no que diz respeito às primitivas usadas para expressar definições na Ontologia, isto é, a meta-Ontologia.

### 2.2.5. Linguagens para desenvolvimento de Ontologias

Outro aspecto a ser considerado sobre a construção de Ontologias está relacionado à escolha de uma linguagem para expressá-las. A princípio, qualquer linguagem de representação de conhecimento formal, ou até mesmo informal, pode ser usada. Na prática, têm surgido várias linguagens para este fim, conforme são citadas algumas delas na seqüência. Para verificação mais detalhada, ver *anexo A.1*. Vale apontar que se observa uma classificação do tipo da linguagem, de acordo com o ambiente de utilização.

Linguagens para desenvolvimento de *Ontologias* [ABX03] [SUI02]:

- a) KIF (*Knowledge Interchange Format*);
- b) Ontolíngua;
- c) CycL (*Cyc Language*);
- d) F-logic (*Frame Logic*);
- e) OCML (*Operational Conceptual Modeling Language*);
- f) LOOM; e
- g) TELOS.

Linguagens de desenvolvimento baseadas em *Web* [ABX03] [SUI02]:

- a) OIL (*Ontology Inference Layer*);
- b) DAML + OIL;
- c) XOL (*XML - Based Ontology Exchange Language*);
- d) SHOE (*Simple HTML Ontology Extensions*); e
- e) OWL (*Web Ontology Language*).

Cada uma dessas linguagens possui um número de compromissos ontológicos e a escolha de uma linguagem particular deve ser feita com base na sua adequação ao propósito da representação da Ontologia.

### 2.2.6. Ferramentas para desenvolvimento de Ontologias

Além das metodologias e linguagens, existem também as ferramentas utilizadas no auxílio da construção das Ontologias, as quais são ambientes integrados com linguagens próprias ou associadas àquelas descritas na seção anterior. Possuem também editores de *Ontologias* e opções para exportação em formatos padrões como o XML<sup>1</sup> ou RDF<sup>2</sup>. Dentre as principais ferramentas, de uso sobretudo no meio acadêmico, estão:

*Ontolinguua* – o sistema tem característica *multiusuário*, pois possibilita que várias pessoas naveguem, criem, editem, compartilhem e reusem Ontologias armazenadas em um servidor de Ontologias. O laboratório *Knowledge Systems*, da Universidade de *Stanford*, desenvolveu o sistema no início dos anos 90. O acesso ao editor de Ontologias, baseado em *Frames*, tem uma interface por meio da *Web*. Implementa a troca de conhecimento em redes expresso em *frames* de forma otimizada, ou mandando e recebendo conhecimento das bases através da linguagem KIF. A representação é ambígua e simples, desde que sempre definida por uma série de axiomas KIF. O Frame da Ontologia define o vocabulário da linguagem de apresentação [FFR96].

*WebOnto* – [DMM99] É uma ferramenta baseada em *Web* para visualização, navegação e desenvolvimento de Ontologias e modelos de conhecimento especificados em OCML. Foi desenvolvido pelo Instituto Knowledge Media na Open University como parte de um projeto de pesquisas Européias no final dos anos 90. *WebOnto* é basicamente um cliente Java conectado em um servidor web customizado. Adota o modelo do conhecimento do OCML e diferencia entre domínios, tarefas, métodos de resolução de problemas e aplicações.

---

<sup>1</sup> Sigla de (*eXtensible Markup Language* ou Linguagem de Marcação Estendida), uma especificação desenvolvida pelo W3C. XML é um subconjunto da SGML (Standard Generalized Markup Language, ou Linguagem de Marcação Padrão Generalizada) que permite as desenvolvedores criarem sua própria Tag customizada, permitindo a definição, a transmissão, a validação e a interpretação dos dados entre aplicações e entre organizações.

<sup>2</sup> RDF (*Resource Description Framework*): recomendação do *World Wide Web Consortium* - W3C, constitui uma arquitetura genérica de metadados que permite descrever semanticamente recursos no contexto *Web*.

*WebODE* – [GPZ01] é um *workbench* escalável para a engenharia ontológica com uso no ambiente *Web*. O grupo *Ontology* e o *Knowledge Reuse Group*, da Universidade Técnica de Madrid, desenvolveram o *WebODE workbench*. Possui uma arquitetura de três camadas comumente encontrada em aplicações *Web*, camada de dados, camada da lógica do negócio e a camada de interface com o usuário. As principais construções do modelo do conhecimento de *WebODE* são: conceitos, grupos dos conceitos, relações, constantes e instanciações.

*Protégé-2000* – [NFM00] é um sistema integrado com plataforma independente para o desenvolvimento e manutenção de sistemas baseados em conhecimento. Foi desenvolvido pela *Stanford Medical Informatics*. O *Protégé-2000* tem um modelo baseado em *frames* para representação do conhecimento, completamente compatível com o protocolo *OKBC (Open Knowledge-Base Connectivity)*, permitindo a interoperabilidade com outros sistemas de representação de conhecimento. Além da modelagem do conhecimento pela definição de classes organizadas hierarquicamente e de suas relações entre elas, o *Protégé-2000* oferece uma interface de introdução de dados, instâncias, específicos para a criação de uma base de dados. Além de modelar aspectos genéricos como tipos, também podem ser registradas instâncias específicas das classes finais, permitindo a criação de um sistema especialista.

*OntoEdit* – [MSS00] É um ambiente de desenvolvimento para projeto e manutenção de Ontologias, criado pelo *Knowledge Management Group* da Universidade de Karlsruhe. Suporta o desenvolvimento em várias línguas e o modelo do conhecimento é relacionado com as linguagens baseadas em *frames*.

*OilEd* - [BHG01] é uma ferramenta de desenvolvimento para a Ontologia baseada na linguagem *OIL*. Integra o sistema raciocinador *FaCT (Fast Classification of Terminologies)* e estende o poder expressivo de outras ferramentas baseadas em *Frames*. Seu elemento principal é usar raciocínios para verificar a consistência dos conceitos na *Ontologia* e inferir sobre associações implícitas. As funcionalidades mais interessantes do *FaCT* são sua lógica expressiva, sua implementação otimizada do *tableaux* e sua arquitetura cliente-servidor.

### 2.2.7. Sistema KISF

Além das ferramentas citadas, existem várias outras. Convém destacar nesta seção a ferramenta *KISF (Knowledge Integration&Sharing Framework)*, desenvolvida pelo Msc. professor Edson José Pacheco, da Pontifícia Universidade Católica do Paraná, em 2003,

como trabalho de conclusão de dissertação do seu curso Mestrado. A mesma será utilizada nos experimentos deste trabalho, sobretudo relativamente à construção da *Ontologia* e na composição do ambiente multiagente necessário no trabalho.

O sistema KISF (*Knowledge Integration & Sharing Framework*) é um sistema computacional baseado em metodologias da Inteligência Artificial Distribuída, que objetiva, de forma interativa e incremental, construir e manter um ambiente para a criação, manutenção e evolução de Ontologias de domínio [PAC03].

No sistema, a Ontologia é representada pela explícita especificação conceitual, sendo caracterizada por planos de representação e realizada mediante a configuração dos seguintes componentes [PAC03]:

- a) *camada conceitual*: diz respeito à explicitação do plano simbólico, com a caracterização de conceitos e relações;
- b) *propriedades*: integrada a camada conceitual, propiciam os meios para a personalização das associações, de forma a enquadrar um valor sintático superior para a leitura da Ontologia;
- c) *camada sensível*: nela, representam-se as instâncias observadas, pela atuação dos sentidos, no circundante, e
- d) *perfil*: permite a personalização individual de cada ente representado na Ontologia, propiciando a adequação representacional entre discursos diferenciados em um domínio comum.

Conforme Pacheco [PAC03], “a criação e evolução conceitual são realizadas mediante a atuação e percepção fenomenológica, seja pela direta intervenção do Engenheiro de Conhecimento (agente humano) ou pela atuação da sociedade de agentes configurada”.

O autor destaca que a adição de novos agentes, pela explicitação de novas competências, é realizada de maneira direta, não exigindo reescrita do projeto (ou parte dele), provendo os meios para que novas estratégias de atuação sejam adicionadas, mediante a inserção de novas competências na sociedade e, seqüencialmente, da personalização da atuação dos agentes pelo Engenheiro de Conhecimento.

## 2.2.8. Finalidade e uso das Ontologias

Na concepção de pesquisadores [NYG00] [GRB93] [AGP99], o desenvolvimento das *Ontologias*, em geral, tem uma visão comum principalmente no se refere às questões de:

- a) *compartilhamento* e entendimento das estruturas de informação entre pessoas ou agentes de *software*, possibilitando que na construção das aplicações se utilizem os mesmos recursos. Este foi um fator preponderante e decisivo para a explosão das pesquisas em Ontologias;
- b) *reusabilidade* de um domínio de conhecimento, significando que, quando se constrói uma nova aplicação, reutilizam-se os componentes já prontos, levando-se a algumas vantagens como: menor custo financeiro, menor tempo na construção e utilização de menos recursos.

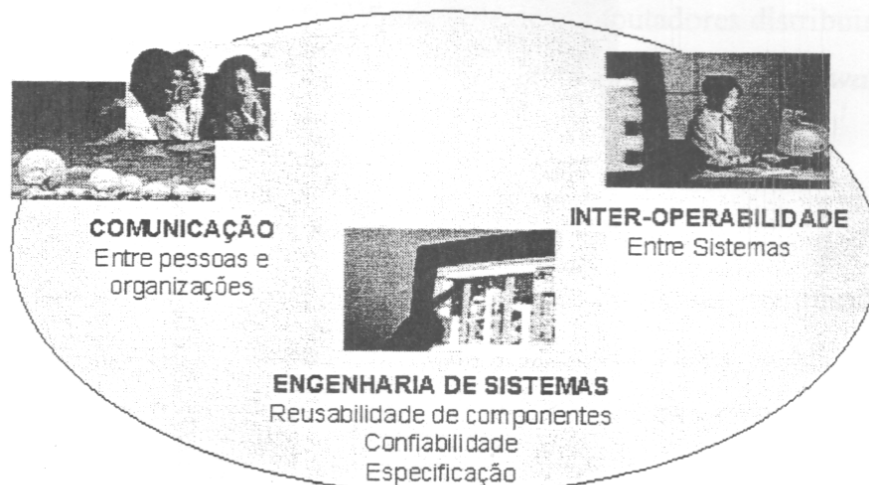


Figura 2.6: Usos da Ontologia

Outros aspectos que envolvem o uso das Ontologias [UCG96] [NFY00], relevantes no contexto do domínio do conhecimento:

- a) tornar explícita a compreensão do domínio do conhecimento;
- b) separar o domínio do conhecimento do conhecimento operacional; e
- c) analisar o domínio do conhecimento.

As Ontologias são projetadas também para compartilhar o conhecimento com e entre os agentes computacionais. Este compartilhamento entende-se como uma descrição de conceitos e relacionamentos usados por um agente ou uma comunidade de agentes [GRB93].

### 2.2.9. Estado da Arte

Muito se tem pesquisado sobre Semântica WEB, em função do grandioso repositório de dados e informações que é a *Web*. Nos primórdios da Internet, a preocupação era somente com a apresentação dos conteúdos, HTML. Hoje, tem-se dificuldade imensa em buscar, consultar informações úteis, em decorrência dessa estrutura voltada à apresentação. Diante desse cenário, deu-se início a várias pesquisas com a preocupação de estabelecer um padrão, modelo de como estruturar as informações na Web. Como frutos dessas pesquisas surgiram linguagens Semânticas na Web [W3C04], como o XML, RDF e outras derivadas, como o OWL. A partir de então, buscou-se criar páginas projetadas e construídas com essas linguagens para propiciar facilidades na busca de conteúdos e informações, sistematicamente.

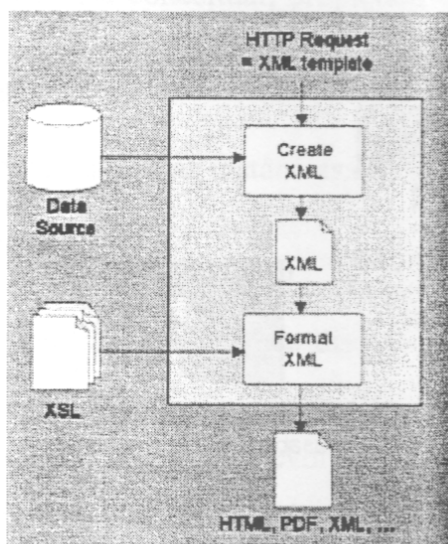


Figura 2.7: XML e *Web* [W3C04]

Ao aprofundar-se um pouco mais neste ambiente semântico, pesquisam-se também formas de como disponibilizar e compartilhar conhecimentos, possibilitar a extração do conhecimento da Internet. Para tanto, as páginas na Web precisam disponibilizar Ontologias; ou seja, páginas construídas com linguagens semânticas, por exemplo, o OWL, propiciando a compreensão e a extração do conhecimento.

### 2.3. Agentes de Software

É uma área da qual participam vários pesquisadores de diferentes disciplinas como a *Inteligência Artificial*, *Sistemas Distribuídos*, *Interface Humana com Computadores* e *Robótica*. Juntos, possuem como objetivos principais: atender aos novos requisitos exigidos por determinadas aplicações e facilitar a interação usuário máquina e a construção de sistemas inteligentes distribuídos.

Vale mencionar que o nome *Agente de Software* (AS) é considerado resultante de pesquisas desenvolvidas pelas comunidades de agentes inteligentes e de agentes de *software*. A primeira está envolvida com a construção de sistemas inteligentes capazes de reagir a eventos tanto em ambientes do mundo real, robôs, quanto do mundo virtual, programas. E a segunda se ocupa do desenvolvimento de programas de *software* mais confiáveis e menores, com a propriedade de se mover dentro de uma rede de computadores distribuindo informações para outros agentes que podem ser tanto humanos quanto aplicações de *software* [HEN96].

#### 2.3.1. Conceitos e definições

A partir das definições observadas nos trabalhos de alguns pesquisadores, é possível perceber que vários autores possuem diferentes idéias sobre a área de AS. Dessa forma, não existe uma definição unânime sobre o que seja exatamente um agente de *software*; porém, muitas dessas definições estudadas são complementares. Na seqüência, é apresentado um resumo das definições encontradas.

Russel e Norvig [RSN95] definem que "um agente é tudo que pode ser visto como percebendo seu ambiente através de sensores e agindo sobre ele através de atuadores". Realizando uma analogia simples, um humano possui diversos sensores como olhos, ouvido, nariz e atuadores como braços, pernas e boca. Um robô possui, por exemplo, câmeras de vídeo, antenas de rádio e fontes de raios infravermelhos como sensores e motores efetadores, enquanto um programa tem suas percepções e ações codificadas em uma seqüência de bits.

Maes [MAE94] e Hendler [HEN96] mencionam sobre agentes inteligentes como programas de IA, cuja finalidade é agir em diversos ambientes de importância para os humanos, sendo divididos em duas categorias: agentes físicos e agentes de informação. Os agentes físicos trabalham em um ambiente onde seja difícil colocar um ser humano, por

exemplo, espaço, ou que seja perigoso, por exemplo, o núcleo de um reator nuclear. Os agentes de informação, ou *softbots (software robots)*, atuam em um mundo virtual onde exista grande quantidade de informações espalhadas por diversos computadores, por exemplo, a Internet.

Wooldridge e Jennings [WDJ94] distinguem um conjunto de propriedades do uso do termo agentes, noção "fraca" de agência: o termo agente é usado, geralmente, para denotar tanto *hardwares* quanto *softwares* que possuam as seguintes propriedades:

- a) *autonomia*: agentes operam sem a intervenção direta de humanos ou outros agentes e possuem algum tipo de controle sobre suas ações e estados internos;
- b) *sociabilidade*: agentes interagem com outros agentes (ou humanos) por meio de algum tipo de linguagem de comunicação compreensível por ambas as partes;
- c) *reatividade*: agentes percebem o ambiente (físico ou virtual) e reagem em tempo às mudanças nele ocorridas; e
- d) *pró-atividade*: agentes não respondem apenas às mudanças no seu ambiente; podem ainda exibir um comportamento orientado a objetivo, ou seja, ele também pode tomar a iniciativa.

E noção "forte" de agência, uma visão mais especializada atribuída, sobretudo por pesquisadores envolvidos em IA, definem agentes como sistemas de computador que além de possuírem as propriedades anteriormente citadas são conceituados e implementadas, utilizando conceitos que normalmente são aplicados aos humanos. Por exemplo, é muito comum em IA caracterizar um agente usando noções mentais, como conhecimento, crença, intenção e obrigação.

Bradshaw [BRS97] acrescenta outras propriedades que os agentes devem possuir para se diferenciar de um simples programa de computador, a saber:

- a) *capacidade de inferência* – agentes podem agir sobre especificações abstratas de tarefas utilizando conhecimentos anteriores, conseguindo ir além das informações fornecidas e devem possuir algum modelo próprio, de usuários, de situações e de outros agentes;
- b) *continuidade* – agentes conseguem fazer persistir a sua identidade e estados durante longos períodos de tempo;
- c) *adaptabilidade* – agentes são capazes de aprender e melhorar com a experiência; e
- d) *mobilidade*: agentes podem migrar de forma intencional de um determinado local para outro.



Uma outra abordagem para se definir os agentes de *software* é diferenciar um agente de um simples programa de computador. Dessa forma, apesar de existirem diferentes definições, torna-se possível estabelecer alguns critérios das propriedades as quais os AS devam possuir. Deve ser ressaltado que todos os AS são programas, porém, nem todos os programas podem ser considerados agentes. Os AS podem possuir algumas das "habilidades" citadas, como mobilidade, reatividade, autonomia. Logo, eles podem ser classificados em função de suas características peculiares. Ainda, os agentes podem estar contidos em vários ambientes distintos, onde cada qual possui um comportamento particular. Dessa forma, os tipos de agentes e os possíveis ambientes são discutidos na seqüência.

### 2.3.2. Competências de agentes de *software*

Antes de classificar-se os AS, faz-se necessário estabelecer alguns critérios e características particulares que permitam sua realização. Nwana [NWA96] apresentou um estudo selecionando algumas propriedades que facilitam tal tarefa. Na concepção apresentada, os agentes podem ser classificados de acordo com suas habilidades, tais como:

- a) *mobilidade* – se um agente possuir a capacidade de transportar-se por meio de uma rede de computadores armazenando seu estado anterior, então ele é considerado um agente móvel ou, caso contrário, estático;
- b) *comportamento* – a existência de um modelo simbólico interno de raciocínio que permita ao agente planejar e negociar com outros agentes na execução de tarefas de forma coordenada classifica o agente como deliberativo. Por outro lado, os agentes ditos reativos não possuem nenhum modelo simbólico interno de seu ambiente e, ao invés disto, reagem a estímulos vindos do seu ambiente;
- c) *exibição de atributos primários* – as propriedades de autonomia, cooperação e aprendizagem classificam os agentes como colaborativos, de interface ou inteligentes;
- d) *papel* – os agentes podem ser classificados por seus papéis. Exemplo: os agentes de Internet, cujo papel principal é buscar informações na rede mundial de computadores; e
- e) *filosofia híbrida* – os agentes podem ter duas ou mais filosofias de agentes simultaneamente.

As características apresentadas cobrem grande parte das habilidades que os agentes conseguem apresentar.

### 2.3.3. Tipologias de agentes de *software*

Faz-se necessário separar os agentes pertencentes a classes distintas para compreender melhor o que são essas entidades. Algumas classes de agentes encontradas na literatura são classificadas numa determinada tipologia desses agentes, a qual se refere ao estudo dos tipos de entidades. Os agentes de *software* podem existir em diversos tipos de ambientes; por isso, alguns dos possíveis ambientes de agentes também são citados na seqüência. As propriedades citadas anteriormente possibilitam a definição de uma tipologia de agentes de *software*. Dessa forma, algumas das categorias mais comuns de agentes existentes são:

- a) *agentes colaborativos* – enfatizam a autonomia e cooperação com outros agentes para realizar tarefas. Podem aprender, porém este aspecto não é muito relevante para este tipo de agente. Para que haja uma coordenação entre si, devem ser capazes de negociar [NWA96];
- b) *agentes de interface* – enfatizam a autonomia e o aprendizado para realizar as suas tarefas. Metaforicamente, agem como se fossem assistentes pessoais que colaboram com o usuário no mesmo ambiente de trabalho. Tal colaboração difere da apresentada pelos agentes colaborativos, uma vez não ser necessário existir uma linguagem formal explícita para a comunicação, como é o caso da comunicação intra-agentes [MAE94];
- c) *agentes móveis* – são programas autônomos capazes de migrar de uma máquina para outra em uma rede heterogênea de computadores como a Internet. Em outras palavras, o agente que esteja executando em um computador *A* consegue suspender sua execução em qualquer ponto, transportar seu código e estado para um computador *B* e terminar sua execução podendo voltar em seguida para a máquina de origem ou seguir para uma outra. Esta migração para uma fonte eletrônica de recursos permite ao agente acessar localmente os recursos disponíveis, eliminando, assim, a transferência dos dados pela rede [GRA97];
- d) *agentes de informação / Internet* – são agentes capazes de "navegar" por redes heterogêneas de computadores (e.g Internet) para localizar, combinar, processar, organizar e trazer informações requisitadas por usuários ou programas;
- e) *agentes reativos* – respondem a estímulos enviados pelo ambiente operando de forma autônoma em tarefas específicas (e.g. sensoriamento). Porém, nenhuma especificação de comportamento é previamente definida; isto é, não têm um sistema de planejamento,

sistema de aprendizagem e máquinas de inferência como em alguns sistemas clássicos de IA. Em geral, a comunicação entre eles não ocorre via *broadcasting* de mensagens e sim por transmissões realizadas "agente a agente", que tende a ser minimizada, permitindo que cada agente tome decisões por si e tornando suas respostas a estímulos mais eficientes [MAE94] e [NWA96];

- f) *agentes inteligentes* – baseiam-se nos sistemas de planejamento e de raciocínio lógico de Inteligência Artificial. Possuem representações do seu ambiente e inferem a respeito dos eventos que ocorrem dentro deles. Seu componente principal é a sua base de conhecimento, utilizada para formar representações do mundo em que vive e que são utilizadas por processos de inferência para derivar novas representações, permitindo que sejam realizadas deduções sobre o que deve ser feito. Estes agentes se adaptam a mudanças ocorridas em seu meio, e assim, conseguem aceitar novas tarefas [RSN95]; e
- g) *agentes híbridos* – possuem características existentes em diversos tipos de agentes.

#### **2.3.4. Ambiente dos agentes de software**

Os agentes devem conviver em algum tipo de ambiente, os quais também podem possuir diversas características. Tais propriedades estão apresentadas na seqüência. A relação entre os agentes e o ambiente onde residem, comumente, é vista da seguinte forma: as ações são executadas pelos agentes em respostas aos eventos que estes conseguem perceber do ambiente, cujos tipos podem ser descritos nos seguintes termos [RSN95]:

- a) *acessibilidade* – se o aparato sensorial de um agente fornece a ele acesso ao estado completo do ambiente, então o ambiente é dito acessível. Um ambiente é eficientemente acessível se os sensores do agente detectam todos os aspectos relevantes para a escolha de uma ação. Um ambiente inacessível às vezes é conveniente porque o agente não precisa manter todos os possíveis estados internos do seu mundo;
- b) *previsibilidade* – se o próximo estado do ambiente é totalmente determinado pelo estado atual e ações selecionados pelos agentes, então o ambiente é totalmente considerado determinístico. Caso contrário, não o é;
- c) *episódios* – num ambiente episódico, a experiência de um agente é dividida em episódios, sendo que cada um se consiste do agente o perceber e, então, agir. A qualidade das ações

tomadas pelo agente depende apenas do episódio atual, uma vez que os episódios subsequentes não dependem das ações tomadas em episódios anteriores;

- d) *variabilidade* – se um ambiente pode se modificar enquanto um agente está deliberando, então o ambiente é tido como dinâmico para o agente. De outra forma, é considerado estático. Ambientes estáticos são menos complexos para os agentes, pois não precisam ficar “olhando” para o seu mundo enquanto decidem qual ação tomar; e
- e) *tamanho* – se existe um número limitado de ações e percepções claramente definidas, então o ambiente é chamado de discreto. Em contrapartida, num ambiente contínuo, esse número é alterado a todo o momento e, assim, o agente deve conseguir tratar essas mudanças tomando decisões corretas para tal fim.

A classificação de agentes e ambientes é necessária para se entender estas entidades computacionais. Um outro aspecto importante que deve ser levado em consideração diz respeito a uma abordagem mais ampla com relação a AS, a ser tratado na próxima seção.

### 2.3.5. Sistemas multiagentes

Nos anos mais recentes, a Inteligência Artificial Distribuída (IAD) tem se tornado um setor de pesquisa em oposição a IA clássica. Comparativamente, a IA clássica se baseia no comportamento humano individual, cuja ênfase é posta na representação do conhecimento e no método de inferência. Por sua vez, a IAD é baseada em comportamento social, cuja ênfase é colocar ações e interações. Uma das áreas que estudam estes modelos da IAD é a disciplina de Sistemas Multiagentes (SMA).

Denomina-se Sistema Multiagente aquele projetado e implementado com vários agentes interativos situados em um ambiente comum, objetivando a resolução de um problema. De acordo com Ferber [FRB99], o termo SMA é aplicado a um sistema que compreende os seguintes elementos:

- a) um ambiente, **E**, representado por um espaço que possui uma métrica;
- b) um conjunto de objetos, **O**. É possível, em um dado momento associar qualquer objeto com uma posição em **E**. Esses objetos são passivos; ou seja, podem ser percebidos, criados, destruídos e modificados pelos agentes;
- c) uma comunidade de agentes, **A**, que são objetos específicos ( $A \square O$ ), representando as entidades ativas do sistema;

- d) uma comunidade de relações, **R**, que liga objetos (portanto, agentes) entre si;
- e) uma comunidade de operações, **Op**, que permite aos agentes de **A** perceberem, produzirem, consumirem, transformarem e manipularem objetos do **O**; e
- f) operadores com a tarefa de representar a aplicação dessas operações e a reação do mundo para suas tentativas de modificação.

Um SMA pode ser definido como uma rede, fracamente ligada, de agentes que trabalham em conjunto para resolver problemas que ultrapassam a capacidade e o conhecimento de cada um em particular. São especialmente adaptados para representar problemas com múltiplas possibilidades de solução e diversas perspectivas e entidades. Esses sistemas são capazes de resolver problemas tradicionalmente distribuídos e permitem sofisticados padrões de interação, tais como:

- a) *cooperação* – trabalhar em conjunto em prol de um objetivo comum;
- b) *coordenação* – organizar a resolução de problemas mediante a exploração de interações benéficas e evitando interações prejudiciais; e
- c) *negociação* – tentar atingir um acordo que seja aceitável para todas as partes envolvidas.

A base desse paradigma e seu diferencial sobre as demais abordagens são a flexibilidade e o alto nível das interações [JEN98].

Durante a concepção de um agente computacional, espera-se que a ação ocorra em benefício de algo ou alguém, no intuito da realização de uma tarefa específica que lhe foi designada. Entretanto, a função de configurar o agente, com todos os detalhes de seu comportamento, pode ser uma tarefa complexa. Apresenta-se, então, a necessidade de desenvolver uma capacidade, intrínseca ao agente, de compreensão do contexto do ambiente para a execução eficiente da tarefa designada. Portanto, a formação de um agente computacional envolve não apenas o exercício de uma forma particular de *expertise*, mas também a análise das peculiaridades do ambiente e das situações envolvidas.

### 2.3.6. Classificações de sistemas multiagentes

Os agentes, que constituem um SMA, apresentam propriedades as quais, além de permitirem a distinção e caracterização entre agentes e entidades, possibilitam a formação de categorias. Essas são baseadas em dimensões (concepção, constituição, granularidade, etc.), e



como agentes. É super quando o agente é constituído por partes menores as quais são agentes elementares ou superagentes.

Quanto às capacidades e comportamentos dos agentes, podem ser:

- a) *cognitivos (ou deliberativos)* – a hipótese de símbolos físicos de Simon [SIM68] é base para a maioria dos modelos de agentes usados na IA e tem como núcleo a representação do ambiente e dos Estados Mentais. Esses estão sujeitos às alterações pelas inúmeras formas do raciocínio simbólico. As crenças de um agente revelam o que ele espera do estado atual do ambiente e mostram suas expectativas sobre o curso das ações que deverão ser executadas para que ele possa atingir um determinado objetivo. As crenças são modeladas tendo como base a concepção de ambientes possíveis, associadas a cada tipo de representação, percebido por um agente, de um determinado ambiente. De maneira abstrata, os desejos especificam preferências sobre estados futuros ou do curso dos acontecimentos do ambiente. Eles não são necessariamente consistentes, de modo que um agente pode crer que um desejo venha a acontecer ou não. As formações de desejos por intermédio dos processos de seleção de objetivos são representadas pela intenção. Os cognitivos são chamados de intencionais na medida em que a estrutura de conhecimento para a aplicação do raciocínio se caracteriza pela existência de crenças, desejos e intenções (*BDI-architecture, Beliefs, Desires and Intentions*). Essa estrutura do conhecimento expressa uma atitude mental do agente [WDJ94], representando respectivamente a informação, a motivação e o raciocínio, os quais também são referidos na literatura como agentes deliberativos;
- b) *reativos* – fortemente baseados na psicologia comportamental, estes agentes formam suas decisões em tempo de execução. Geralmente, baseiam-se em pouca informação e mediante a utilização de regras de ação simples têm como filosofia de trabalho a hipótese de Simon, que diz que “a complexidade do comportamento de um agente pode ser uma reflexão da complexidade do ambiente no qual este agente está operando a mais do que seu modelo interno” [MOU96]. São reativos na medida em que percebem o ambiente no qual estão inseridos e respondem rapidamente [MOU96], sendo que as trocas ocorrem durante o processo. O termo *rapidamente* está associado à idéia de uma baixa atividade de raciocínio – permitindo reações rápidas – o que leva a caracterizá-los como incapazes de manipular os seus objetivos. Por exemplo, suas ações são executadas como resultado de disparos de regras simples, dados pelo par *percepção e reação*. Quanto à inteligência dos

agentes reativos, essa emerge não do comportamento individual, mas do comportamento global da comunidade.

- c) *sociabilidade* – enquadra os agentes em termos de possuírem explicitamente, ou não, modelos dos outros agentes da comunidade, de maneira que, por meio do raciocínio, consideram esse conhecimento para a tomada de decisão; e
- d) *especialidade* – enquadra os agentes em termos do domínio das tarefas que os mesmos executam.



## 2.4. Sistemas de Informação Legados

O maior patrimônio que as organizações detêm na área de informática são suas aplicações corporativas, apropriadamente chamadas de Sistemas de Informação Legados (SIL). Desenvolvidas e aperfeiçoadas ao longo de vários anos, os SIL são o alicerce de muitas organizações e, na maioria das vezes, de missão crítica, essenciais para a sua operação diária.

A grande preocupação das administrações ligadas às áreas de tecnologia de informação com relação aos SIL é a sua manutenção em função de que a atualização destes sistemas para novas tecnologias é um processo problemático, sobretudo por causar um desperdício de tempo, recursos humanos e investimentos realizados ao longo dos anos no desenvolvimento e manutenção dos SIL. Os SIL representam para as organizações um imenso investimento em termos de tecnologia de informação, processos de negócios, procedimentos e estruturas organizacionais [HLS99].

### 2.4.1. Definição e evolução

Para Jacobson [JCB99], “Sistema Legado é um sistema existente herdado por um projeto. Usualmente, é um sistema velho que foi criado usando tecnologias de implementação mais ou menos obsoletas, mas que precisa ser incorporado ou reusado, quer na sua totalidade, quer parcialmente, quando um novo sistema é construído pelo projeto”.

Boland [HLS99] define os SIL como tecnologia de informação (e.g máquinas, aplicativos e rede) e modelo de negócios implícito na aplicação da tecnologia (e.g estrutura organizacional, fluxos de trabalhos, procedimentos e processos) com a organização. Em geral, considera-se que os SIL possuam uma tecnologia ultrapassada, pois se hoje analisarmos esses aspectos nos ambientes computacionais, são considerados SIL principalmente os sistemas cooperativos de informação bancários. Isso porque, em sua grande maioria, funcionam ainda em ambientes centralizados em *mainframe*, escritos em linguagens de terceira geração, acessando em muitos casos banco de dados hierárquicos e arquivos seqüenciais.

As organizações dos anos 50 eram dependentes dos computadores para um número limitado de funções críticas; por exemplo, salários e pesquisa de mercado. As máquinas centralizadas, *mainframes*, eram usadas em um grande número de organizações durante os

anos 60, como, por exemplo, em bancos e companhias de seguro. No fim dos anos 70 e início dos 80, a arquitetura de informação se tornou mais complexa, com a informação distribuída pela telecomunicação e com a introdução de computador pessoal. Nos anos 90, viu-se emergir a arquitetura cliente-servidor, com as empresas conectadas aos fornecedores e os clientes com a *Web*.

Os sistemas da primeira geração que datam dos anos 60 e 70 estavam na linguagem de máquina, mas a maioria foi desenvolvida com versões adiantadas das linguagens de programação de terceira geração tais como COBOL ou Fortran. Esses sistemas eram geralmente do tipo *batch* (em lote) que funcionavam em *mainframes*. Os sistemas de segunda geração, do final dos anos 70 e início dos anos 80, possuíam algum grau de modularidade e muitos eram usados em processamento *on-line* (em linha). Tais sistemas operavam sobre *mainframes* e em uma variedade de computadores médios tais como o AS400. Muitos desses sistemas foram desenvolvidos na linguagem COBOL, mas, alguns foram escritos nas linguagens de quarta geração, baseadas em sistemas gerenciadores de banco de dados. No final dos anos 80, houve o desenvolvimento de sistemas de terceira geração, usando-se ferramentas gráficas com o desenvolvimento rápido das aplicações.

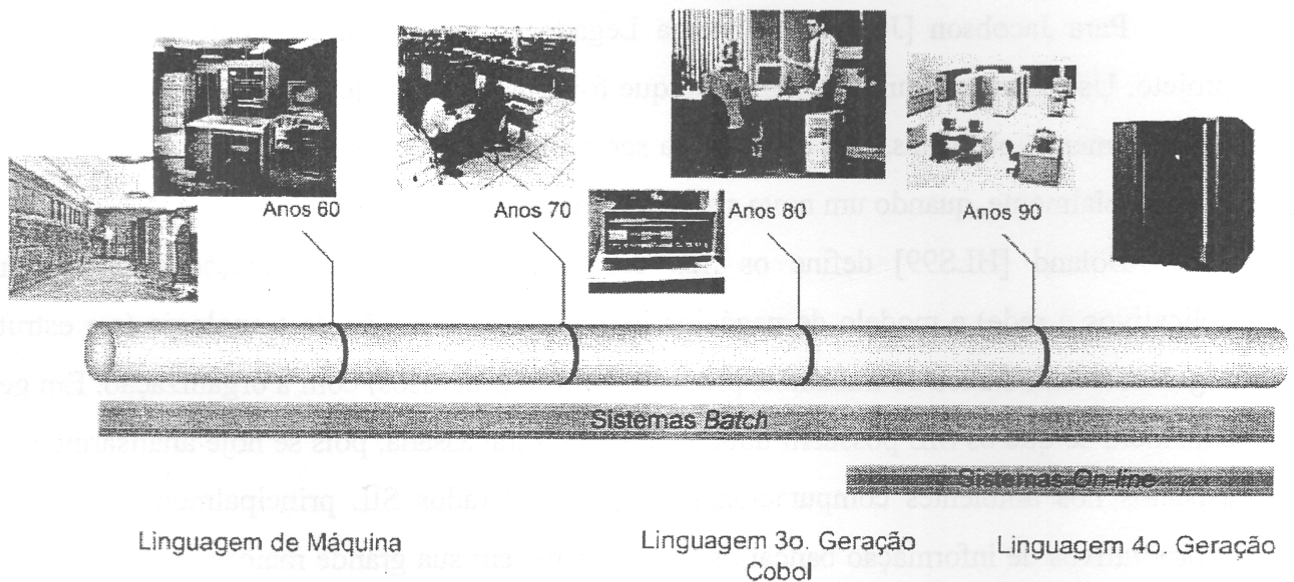


Figura 2.9: Evolução computacional

Mais recentemente, as organizações adotaram pacotes de planejamento de recurso de empresa (ERP), que são produtos comerciais disponíveis para o mercado, pré-escritos, pré-codificados, conjunto de programas integrados que automatizam as atividades principais das

empresas, assim como a gerência do processo produtivo, de recurso humano, de finanças e de matéria-prima, operando em ambiente cliente servidor [HLS99].

Não é incomum para as organizações manterem sistemas de informação criados no início dos anos 80. De fato, nos Estados Unidos, sistemas implantados nos anos 60 e 70 estão ainda na ativa, embora isto não signifique um problema. No Brasil, muitos SIL na área bancária datam da década de 80 e 90, mas não deixam de ser um grande recurso à organização. O desenvolvimento cuidadoso e o ajuste fino de vários anos significam que são relativamente estáveis e podem continuar a suportar a visão estratégica de uma organização. Além disso, têm o potencial de manter milhares de clientes e são executados rapidamente e de forma segura. “O valor real da organização encontra-se no acúmulo de anos de regras de negócio, de políticas, de perícia e de *know how* embutidos nos SIL” [KIM97].

Por sua vez, problemas podem surgir quando os sistemas não permitem que a organização se adapte rapidamente a um ambiente de negócio em mudança e/ou é incapaz de suportar a nova visão estratégica da organização. A vantagem competitiva das organizações nesse novo cenário social e econômico, em que estão imersas, relaciona-se diretamente com a maneira de como os SIL respondem a essa nova situação de negócio.

#### **2.4.2. Perspectiva dos Sistemas de Informação Legados**

Nos anos 90, ao analisarem os SIL sob uma perspectiva técnica, as organizações concluíram que teriam substanciais economias nos custos do *downsizing* do *hardware*, de modo que migraram seus SI para uma arquitetura de aplicações cliente/servidor com sustentação na vantagem da GUI, em relação aos dos sistemas em *mainframe*.

Nesse processo, embora a troca de *hardware* fosse simples, organizações se depararam com uma barreira mais complexa, que seria o *software*. Por isso, nem todas as empresas foram unânimes em migrar para plataformas cliente/servidor. Recentemente, as organizações realizaram nos SIL ajustes para o ano 2000, quando os sistemas não conseguiam distinguir anos entre séculos, de maneira que, conseqüentemente, algo deveria ser feito para impedir um possível descontrole no processamento dos dados e, por efeito, nas informações geradas. Outro problema técnico, não publicado extensamente, foi a troca de moeda (euro) em alguns países da Europa, a qual afetou, em muitos casos, organizações que tratavam dos clientes ou

dos fornecedores europeus. A complexidade para a manutenção dos SIL aumenta a escala desse problema, quando não devidamente identificado.

### **2.4.3. Ambiente dos Sistemas de Informações Legados**

No Brasil, as maiores organizações detentoras de SIL são as instituições financeiras bancárias. O ambiente com relação a novas estratégias de negócios não poderia ser diferente, as várias mudanças na área econômica as obrigaram por várias vezes a tomar novos caminhos para sua sobrevivência. Algumas organizações, de fato, não sobreviveram a esses reveses. Na visão de Holland [HLS99], a perspectiva atual de negócios no ambiente dos SIL tem sido alterada por três poderosos fatores:

- a) primeiro, a tendência crescente de globalização da economia;
- b) segundo, a transformação das economias da sociedade industrial em economias lideradas por conhecimento e bases de informação, por exemplo, a competição baseada em tempo e produtos de curto ciclo de vida, e;
- c) terceiro, a transformação dos negócios das empresas, pela qual a organização se afasta de uma hierarquia estruturada centralizadamente, para tornar-se achatada, menos hierarquizada e descentralizada.

Diante deste cenário, as organizações em geral, incluindo as financeiras, necessitam de uma agilidade para reagir a estas mudanças impostas pela nova economia mundial. E os SIL, mesmo consumindo grande parte dos recursos destas organizações, não conseguem responder as determinadas necessidades, requeridas em função desse novo paradigma.

Um estudo efetuado em quinze grandes organizações nos Estados Unidos mostra a importância e o envolvimento dos SIL, na estrutura, no ambiente de negócios e na visão estratégica da empresa. Em um ambiente estável de negócios, é possível responder e envolver a infra-estrutura de tecnologia de informação e o modelo de negócios para taxas de mudanças relativamente pequenas. Porém, num ambiente oposto, onde as taxas de mudanças são muito mais altas, torna-se difícil adaptar a tecnologia de informação de sistemas legados a um modelo de negócio, rápida e suficientemente para suportar a visão estratégica [HLS99].

Assim, conforme o estudo, se as organizações financeiras bancárias se situam num ambiente instável de negócios com altas taxas de mudanças, reforça-se a demanda de uma metodologia que possibilite atender rapidamente a visão estratégica da empresa.

#### 2.4.4. Linguagem COBOL

A grande maioria dos SIL que estão ativos em ambientes de *mainframe* ainda utilizam a linguagem de programação COBOL. Mesmo sendo uma linguagem que nasceu na década de 60, está adaptada para ser utilizada com banco de dados como o “DB2”, “Datacom”, “Adabas” e outros. Em 1959, um grupo de profissionais, sob o patrocínio do governo dos Estados Unidos da América, com representantes do governo, dos fabricantes de computadores, instituições de ensino superior e usuários, reuniu-se na CoDaSyL (*Conference on Data Systems Language*). Um dos objetivos dessa conferência era estabelecer os padrões para o desenvolvimento de uma linguagem voltada à resolução de problemas comerciais. Concordou-se que ela deveria ter os seguintes aspectos:

- a) ter uma forma natural de expressar programas;
- b) ser de fácil aprendizado;
- c) possuir facilidade de autodocumentação;
- d) ser completamente independente do equipamento;
- e) possuir uma estrutura que permita uma evolução sem que a linguagem mude na essência.

A primeira versão desta linguagem, que recebeu o nome de COBOL, acrônimo de “*COmmon Business Oriented Language*” (Linguagem Orientada para Negócio Comum), surgiu em dezembro de 1959. Pode-se dizer que os objetivos dessa conferência foram atingidos com pleno sucesso, pois o COBOL apresenta todas estas características.

Durante vários anos, o COBOL sofreu contínuos refinamentos e padronizações; contudo, a linguagem permanece, em essência, a mesma. Atualmente, existe um padrão aprovado pelo ANSI (American National Standard Institute), instituição voltada a normas e padrões válidos para o território dos EUA, conhecido como COBOL ANS. Esta versão implementa a maioria dos computadores, em âmbito mundial. Abaixo, são listados os vários dialetos do COBOL e seus compiladores [BRD97]:

- a) 68, 74, 85. A primeira, segunda e terceira versão do COBOL. Estas linguagens foram liberadas em 1968, 1974 e 1985, respectivamente;
- b) UNISYS COBOL, OS/VS COBOL, COBOL 370, Microfocus COBOL. São produtos de *software*: compiladores COBOL, seus correspondentes COBOL/ dialetos, e ferramentas para suporte a programação no seu particular dialeto;

Nota-se que muitos dialetos têm incorporado, *embedded*, como SQL<sup>3</sup>, JCL<sup>4</sup>, CICS<sup>5</sup> etc. Como seu próprio nome indica, o COBOL é uma linguagem voltada sobretudo para problemas comerciais. Diferentemente dos problemas científicos, os problemas comerciais não requerem fórmulas matemáticas complexas, mas necessitam manipular eficientemente arquivos extensos e de constante atualização. Por este motivo, o COBOL possui muitos recursos para a manipulação de dados de entrada e saída.

Para tornar-se autodocumentado, o programa pode conter comentários que o expliquem. Além disso, o COBOL foi estruturado para ter quatro grandes divisões inter-relacionadas que facilitam identificar setores do programa e a sua leitura, quais sejam:

- a) *Identification Division*: divisão de identificação;
- b) *Environment Division*: divisão de equipamento;
- c) *Data Division*: divisão de dados;
- d) *Procedure Division*: divisão de procedimentos, programa propriamente dito.

Eis um exemplo de um código fonte em COBOL com as principais divisões:

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. ALOMUNDO.
000300
000400*
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000700 SOURCE-COMPUTER. IBM3090.
000800 OBJECT-COMPUTER. IBM3090.
000900
001000 DATA DIVISION.
001100 FILE SECTION.
001200
100000 PROCEDURE DIVISION.
100100
100200 LOGICA-PRINCIPAL SECTION.
100300 INICIO.
100400     DISPLAY "ALO MUNDO !!!".
100500     STOP RUN.
100600 LOGICA-PRINCIPAL-SAIDA.
100700     EXIT.
```

<sup>3</sup> SQL significa *Structured Query Language*, Linguagem Estruturada de Pesquisa. De grande utilização com Banco de Dados, teve seus fundamentos no modelo relacional de Codd (1970).

<sup>4</sup> JCL Acrônimo de *Job Control Language*. Linguagem de comandos usada em sistemas *mainframe* IBM OS/360. A JCL é usada para executar aplicações e especifica informações sobre o tempo de execução, o tamanho do programa e os arquivos de programa usados para cada aplicação.

<sup>5</sup> CICS, abreviação de *Customer Information Control System*, é um monitor de processamento de transações da IBM, desenvolvido originalmente para fornecer e processar transações em *mainframes* da IBM. Controla a interação entre aplicações e usuários e deixa programadores desenvolverem aplicações *on-line*, sem conhecimento detalhado dos terminais em uso.

## 2.5. Considerações finais deste capítulo

Quando se pesquisa sobre SIL, pouco se encontra sobre soluções que permitam gerenciar de um modo mais simples e eficiente esse vasto conhecimento internalizado nas organizações e de grande valor intrínseco. A maioria das referências diz respeito a mudanças da tecnologia de informação, metodologias centradas em processos de migração dessa plataforma para ambientes orientados, sobretudo, a objetos.

No entanto, pelo aspecto dimensional do volume e longevidade dos SIL, não se deve ignorar a importância do ambiente, que muitas vezes armazena regras de negócio no qual o autor não mais pertence ao quadro de recursos humano da organização.

Eis por que o emprego das Ontologias tem se tornado tecnologicamente viável para a gestão do conhecimento dos SIL, juntamente com o uso de agentes computacionais. Contudo, a maior dificuldade nesse processo está em transformar e representar o conhecimento na forma tácita para a explícita, porque “o conhecimento tácito é altamente pessoal e difícil de formalizar, o que dificulta a transmissão e o compartilhamento com os outros” [NNT97].

Além disso, vale destacar que inexistente ferramenta para aquisição do conhecimento automático e será muito improvável que haja, pelo menos em curto prazo, algo a respeito, sendo encontradas algumas que o fazem de maneira semi-automática e é neste sentido que se propõe esta dissertação.

## Capítulo 3

# Metodologia para gestão do conhecimento em Sistemas de Informação Legados (SIL): *Ontolegacy*

Na Informática, a transposição de um conhecimento do plano metafísico para um objeto real, é extensa fonte de pesquisa na disciplina de *Inteligência Artificial Distribuída*.

De fato, metodologias e tecnologias são intensamente pesquisadas para aplicação na solução de problemas. Eis por que se busca, com este trabalho, desenvolver uma metodologia capaz de melhorar a gestão do conhecimento em *Sistemas de Informação Legados*. Além da metodologia proposta, é objeto desta dissertação a construção de uma ferramenta para o suporte da adoção dessa metodologia. Assim, antes de se discorrer sobre ela, as características e o funcionamento da ferramenta *Ontolegacy* são descritos, conforme a seção seguinte.

### 3.1. Cenário da problematização

O atraso tecnológico dos SIL, em ambientes de *mainframes*, recai sobretudo no uso de ferramentas e linguagens de desenvolvimento ultrapassadas. Além de sua obsolescência, constata-se outros fatores, como, por exemplo, a ausência da explicitação do conhecimento respectivo. Isto é, os SIL estão em constante transformação e, para tal, o corpo técnico responsável requer o conhecimento acerca do SIL para levar a cabo suas mudanças.

Uma organização pode ser vista como uma *interface* entre um ambiente interior (*inner*) e um ambiente exterior (*outer*) [SHM99], sendo estimulada continuamente por fenômenos a ela relacionados. Conforme o Dicionário Aurélio [ARL99], fenômeno é “um fato, aspecto ou ocorrência passível de observação”. Esquemáticamente, [Figura 3.1] há uma constante transformação do *Inner e Outer*, relativa a fenômenos ocorridos no mundo a afetar as organizações, que precisam ser flexíveis para manter sua competitividade e inovação.



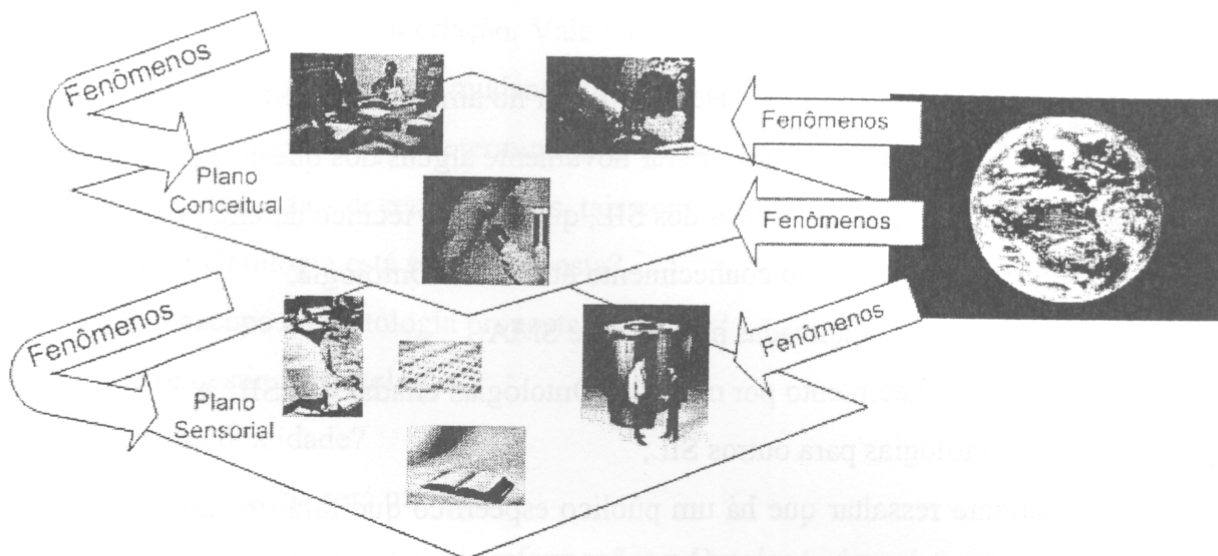


Figura 3.1: Ambiente fenomenológico

No ambiente dos SI, sujeito com ligações ao *outer*, o conceito dos sistemas e suas Ontologias são afetados por fenômenos que ocorrem constantemente. Tais fenômenos incidem nas questões dos negócios das organizações, *inner*. Por exemplo, a alteração da CPMF (Contribuição Provisória sobre Movimentação Financeira) do valor de 0,28% para 0,33%. Pergunta-se: onde este fenômeno gerará mudanças no ambiente organizacional e, conseqüentemente, em quais dos SIL?

A resposta será compatível com determinados graus de esforços, dependentes antes do conhecimento tácito do corpo técnico e do conhecimento técnico que estiver explicitado. Quanto maior o conhecimento tácito a respeito do SIL envolvido, menor o esforço e menor o custo para a mudança. Para alcançar um nível satisfatório do conhecimento tácito, é preciso que o conteúdo do SIL esteja explícito, formalizado em documentos para a sua *internalização*.

De acordo com os modos de conversão do conhecimento [NNT97], (*Capítulo 2*), atinge-se algum grau de auto-suficiência quando o corpo técnico houver realizado a *socialização* e a *internalização* do conhecimento. Mas, quando essas formas de conversão do conhecimento não forem possíveis, não há conhecimento explicitado para sua *internalização* nem pessoas para a *socialização*. A metodologia proposta tem o objetivo de resolver esta questão, possibilitar formas de representação do conhecimento para se fazer a *internalização*.

### 3.2. A metodologia proposta: *Ontolegacy*

A metodologia *Ontolegacy* busca contribuir no ambiente dos SIL, sobretudo quanto às manutenções desses sistemas. Cabe lembrar novamente alguns dos quesitos a serem atingidos:

- a) recuperar os conhecimentos tácitos dos SIL, que o corpo técnico detém;
- b) formalizar a representação do conhecimento através da Ontologia;
- c) manter as Ontologias atualizadas por meio de SMA;
- d) compartilhar o conhecimento por meio das Ontologias criadas dos SIL; e
- e) reutilizar as Ontologias para outros SIL;

É importante ressaltar que há um público específico que fará uso dessa metodologia. Assim, elege-se como o maior interessado o corpo técnico envolvido com os SIL, de modo específico profissionais que de fato precisam desse conhecimento: os técnicos que em geral iniciam o aprendizado dos SIL como novos integrantes das equipes e/ou situações em que são transferidas responsabilidades pela manutenção dos SIL para outras equipes.

Com o intuito de atingir os objetivos propostos, em termos formais, é preciso estabelecer algumas etapas à construção do ambiente, cuja metodologia assim se estabelece:

- a) construção da *Ontologia* dos SIL;
- b) definição dos meios para extração dos termos e conceitos;
- c) associação da *Ontologia* dos SIL com os termos e conceitos extraídos; e
- d) compartilhamento do conhecimento dos SIL.

A seguir, estão as etapas e suas especificidades, conforme a metodologia proposta.

#### 3.2.1. Construção da *Ontologia* dos SIL

Cabe ressaltar que esta etapa tem papel fundamental para a metodologia como também deve ser a inicial; isto é, as outras etapas são dependentes desta para a condução do trabalho.

Há uma diversidade de métodos para construção das *Ontologias*. Aqui, devido às características da ferramenta KISF [PAC03], procura-se seguir principalmente as recomendações adotadas pelo SMI (Stanford Medical Institute) no relatório de Noy e McGuinness [NYG00].

Para esses autores, não há um processo definido para construir *Ontologias*, mas alguns passos podem facilitar a sua criação. Vale lembrar que as etapas da criação da Ontologia são um processo interativo, no qual mudanças podem ser sugeridas com revisões e refinamentos.

Numa primeira etapa, determina-se o escopo da *Ontologia* proposta, quando algumas perguntas fundamentais devem ser feitas, tais como [NGY00]:

- a) Por que a Ontologia está sendo proposta?
- b) O que o escopo da Ontologia proposta irá cobrir?
- c) A que se destina a Ontologia?
- d) Qual a sua finalidade?
- e) Como a Ontologia será utilizada?
- f) Para quais tipos de questões de informações a *Ontologia* deverá prover respostas?
- g) Quem serão os usuários da Ontologia?
- h) Quem terá a responsabilidade de manter a Ontologia?

As respostas a estas questões são de suma importância, base do escopo da Ontologia. Embora as respostas respectivas possam sofrer alterações no processo de desenvolvimento da Ontologia, busca-se que o escopo do modelo esteja o próximo possível das demandas existentes e que esteja bem definido para que as próximas etapas do processo. [NGY00].

Como segunda etapa, pode-se considerar o reuso das Ontologias existentes. Essa medida pode se tornar um requisito à Ontologia proposta, pela necessidade de interagir com outras aplicações que já tenham vocabulários controlados por *ontologias* particulares.

Numa terceira etapa, devem-se enumerar os termos que comporão a Ontologia; ou seja, deve-se criar uma terminologia definida para o domínio de conhecimento em questão. Desse modo, pode-se criar uma lista de seus termos gerais, descrever a que se destinam e quais suas propriedades. Esses termos serão muito úteis para as etapas seguintes.

Na próxima etapa, são estabelecidas as classes e a sua hierarquia, a partir da terminologia definida, ou seja, as classes serão criadas com base nos termos gerais definidos.

De acordo com Uschold e Gruninger [UCG96], existem muitas abordagens possíveis para o desenvolvimento da hierarquia de classes, conforme se pode verificar:

- a) *Top Down* – este processo se inicia com a definição dos conceitos mais gerais de um domínio para, posteriormente, chegar-se a um nível de especialização em torno deles;
- b) *Bottom-up* – inicia-se com a definição das classes mais específicas do domínio para então agrupá-las aos conceitos mais gerais;

c) *Combination* – é a combinação dos processos de desenvolvimento *top-down* e *bottom-up*. Inicia-se definindo os conceitos fundamentais para depois fazer a generalização e a especialização, quando apropriado.

Nenhum desses métodos pode ser considerado melhor ou pior um em relação ao outro. O processo a ser escolhido depende muito da visão pessoal e do domínio de conhecimento a ser mapeado. Por exemplo: se um desenvolvedor tem uma visão *top-down* sistemática de um domínio, então pode ser fácil para ele utilizar a abordagem *top-down*. Por sua vez, o processo de desenvolvimento *combination* tem sido freqüentemente utilizado por desenvolvedores de *Ontologias* por ser considerado o mais fácil e adaptativo dos três.

Noy e McGuinness [NGY00] definem uma *Ontologia* como “uma descrição explícita formal de conceitos num domínio do discurso, classes, propriedades de cada conceito descrevendo características, atributos dos conceitos e restrições sobre as propriedades”. Uma única classe não provê informação suficiente para responder questões relativas ao escopo da *Ontologia*. Portanto, uma vez definida a hierarquia de classes, precisa-se descrever a estrutura interna de seus conceitos.

Esta é a quinta etapa, na qual se descrevem os *slots* que integram as classes, suas propriedades e valores permitidos. Então, os termos gerais definidos na terceira etapa serão transformados em *slots* das classes. Todas as subclasses de uma classe herdam seus *slots* e também as restrições de valores para estes *slots*.

Na etapa seguinte, definem-se quais tipos de valores (*string*, *integer*, *boolean* ou outro) os *slots* das classes podem assumir, quais valores serão estabelecidos, qual o número de valores permitido (cardinalidade) e outras características que os valores dos *slots* possam ter.

Para finalizar a criação da *Ontologia*, podem-se criar instâncias individuais das classes da hierarquia. Para isso, definir uma instância individual de uma classe requer escolhê-la, criar-lhe uma instância individual e preencher os valores dos *slots*. Ao se formalizar as etapas anteriores, adota-se um editor de *Ontologias* que permita sua implementação. Na metodologia proposta, utilizar-se-á o editor KISF [PAC03] para a criação da camada ontológica.

### 3.2.2. Definição dos meios para extração dos termos e conceitos

Com a ferramenta construída, a metodologia enfoca primeiro um aspecto específico dos SIL. Há uma concentração de esforços num meio definido, uma fonte de informação

única para a extração dos termos e conceitos. O foco da análise da aplicação do sistema *Ontolegacy* está nos Códigos Fonte (CF), escritos na linguagem de terceira geração, COBOL.

O COBOL tem como característica ser uma linguagem procedural, com divisões de estrutura (como dados e procedimentos), sendo também posicional. Isto é, existem posições (colunas) nas quais devem ser colocados os comandos para seu funcionamento [Figura 3.2].

```
000100 IDENTIFICATION DIVISION.  
000200 PROGRAM-ID. ALOMUNDO.  
000300  
000400  
000500 ENVIRONMENT DIVISION.  
000600 CONFIGURATION SECTION.  
000700 SOURCE-COMPUTER. IBM3090.  
000800 OBJECT-COMPUTER. IBM3090.  
000900  
001000 DATA DIVISION.  
001100 FILE SECTION.  
001200  
100000 PROCEDURE DIVISION.  
100100  
100200 LOGICA-PRINCIPAL SECTION.  
100300 INICIO.  
100400     DISPLAY "ALO MUNDO !!!".  
100500     STOP RUN.  
100600 LOGICA-PRINCIPAL-SAIDA.  
100700     EXIT.
```

Margem A (coluna 8)

Margem B (coluna 12)

Figura 3.2: Codificação COBOL

Cabe citar que a extração dos termos nos CF ocorre de forma semi-automática, com o auxílio da ferramenta *Ontolegacy* e um agente específico, para este fim.

Nas divisões de um programa COBOL, a *DATA DIVISION* concentra informações sobre os seus dados, num nível macro pelos SIL. A esta divisão dá-se especial atenção, pois aí está a estrutura de metadado das *entidades e atributos*, usados pelos SIL [Figura 3.3].

```

FILE SECTION.
FD XYZS502
   BLOCK                0
   RECORDING            F
   RECORD               200
   LABEL RECORD IS     STANDARD.
   COPY 'XYZD502B'.

01 D502B-REG.
   05 D502B-TIPO-REG      PIC 9(001).
   05 D502B-DT-PROCES    PIC 9(008).
   05 D502B-NUM-AG       PIC 9(005)          COMP-3.
   05 D502B-NUM-POSTO   PIC 9(003)          COMP-3.
   05 D502B-V-INSS      PIC 9(013)V99        COMP-3.
   05 D502B-TP-PGTO     PIC 9(002).
   05 D502B-TP-DOC      PIC 9(001).

WORKING-STORAGE SECTION.
*****EXEC SQL
*****  INCLUDE XYZBCPA
*****END-EXEC
*****
* COBOL DECLARATION FOR TABLE DBAPF.CTL_PRO_XYZ *
*****
01 DCLCTL-PRO-XYZ.
   10 CPA-NUM-SIST       PIC S9(3)V USAGE COMP-3.
   10 CPA-DT-PROCES     PIC S9(8)V USAGE COMP-3.
   10 CPA-DT-PROCES-ANT PIC S9(8)V USAGE COMP-3.
   10 CPA-NUM-REMES-GPS PIC S9(7)V USAGE COMP-3.

01 AREAS-DE-SALVAMENTO.
   05 AS-RETURN-CODE-4000 PIC S9(004) COMP VALUE +4000.
   05 AS-DV-AG          PIC X(001)          VALUE SPACES.
   05 AS-NUM-SIST       PIC 9(003)          VALUE 370.
   05 AS-NUM-POSTO     PIC 9(003)          VALUE ZEROES.
   05 AS-NUM-AG        PIC 9(005)          VALUE ZEROES.

PROCEDURE DIVISION.

```

Figura 3.3: Área de dados (*Data Division* COBOL)

Ao acompanhar a evolução da TI, o COBOL se adaptou ao uso do banco de dados em conjunto com a linguagem. Nota-se no, exemplo acima, que além dos campos (*em negrito*) do arquivo seqüencial definido na *FD*, são mostrados campos de bancos de dados, definidos na *Working-Storage Section*, utilizados na aplicação. Convém ressaltar que além da identificação das entidades e atributos contidos nos CF, faz-se necessário um processo anterior que filtre os CF porque é utilizado um meio no qual estão incorporados os metadados dos arquivos seqüenciais (*copybook*) e os *layouts* das tabelas do banco de dados.

### 3.2.3. Associação da *Ontologia* dos SIL aos termos e conceitos extraídos

A maior dificuldade encontrada nos processos de manutenção dos SIL se refere ao seu desconhecimento. O passo primeiro para a manutenção dos SIL é a *socialização* [NNT97], tentativa de adquirir o conhecimento diretamente do corpo técnico responsável pelo SIL, para que seja assimilado. Outra opção seria a *internalização* [NNT97], isto é, a aquisição de conhecimento por meio de documentos e manuais dos SIL, o que implica compreender a natureza dos SIL pelos conhecimentos que estão no formato explícito.

Quando esses meios não estão disponíveis, busca-se o conhecimento diretamente nos objetos (CF) ativos que refletem a situação real dos SIL. Nesta etapa, pretende-se resolver partes dos problemas expostos com a complementação da *Ontologia* do SIL, com os termos e conceitos extraídos dos CF, por meio da ferramenta Ontolegacy.

Os termos gerados a partir do CF, definidos conforme seção anterior, são selecionados pelo analista responsável ou engenheiro do conhecimento, de forma a complementar a *Ontologia* do SIL. De acordo com a Figura 3.4, aparece na *Ontologia* o conceito *Número-Remessa*, de modo que na extração dos termos no CF exemplificados o usuário seleciona ou não o termo CPA-NUM-REMESS-GPS que fica associado ao conceito *Número-Remessa*.

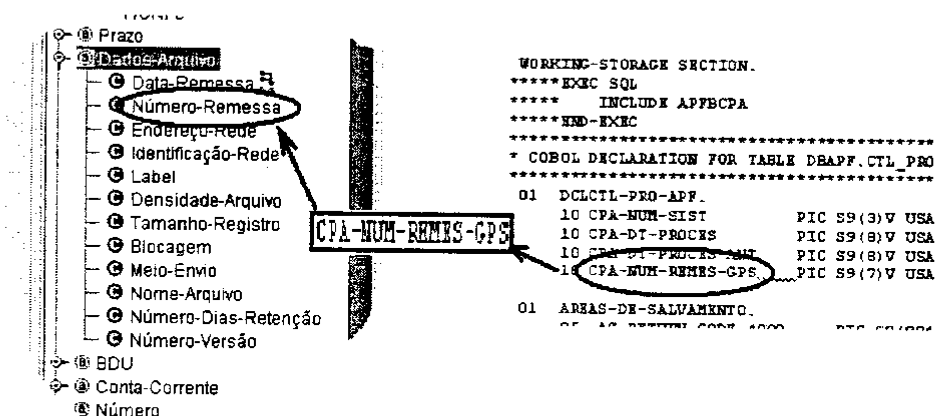


Figura 3.4: Associação com a *Ontologia*

### 3.2.4. Compartilhamento do conhecimento dos SIL

O compartilhamento das *Ontologias* dos SIL se faz importante para disseminar o conhecimento entre as pessoas que trabalham num mesmo ambiente, ou outro, para que possam aproveitar e reutilizar a *Ontologia* criada. A *Web* é um meio para tal, dentro e fora das organizações.

A emergente pesquisa e criação de linguagens para a Semântica *Web* tem sido desenvolvida paralelamente às *Ontologias* [FNM01]. O entendimento, a compreensão e a troca de informações na *Web* devem ocorrer com base numa linguagem formal e comum em *Ontologias*.

Por efeito, é essencial que as ferramentas para a criação das *Ontologias* permitam exportá-las em formatos inteligíveis na *Web* como o XML e o RDF e, mais recentemente [W3C04], o OWL como padrão de referência para Semântica *Web* pela *World Wide Web Consortium* (W3C).

Conforme a seguir [Figura 3.5], visualiza-se a transformação da *Ontologia* em dados padrão RDF para disponibilização na *Web*.

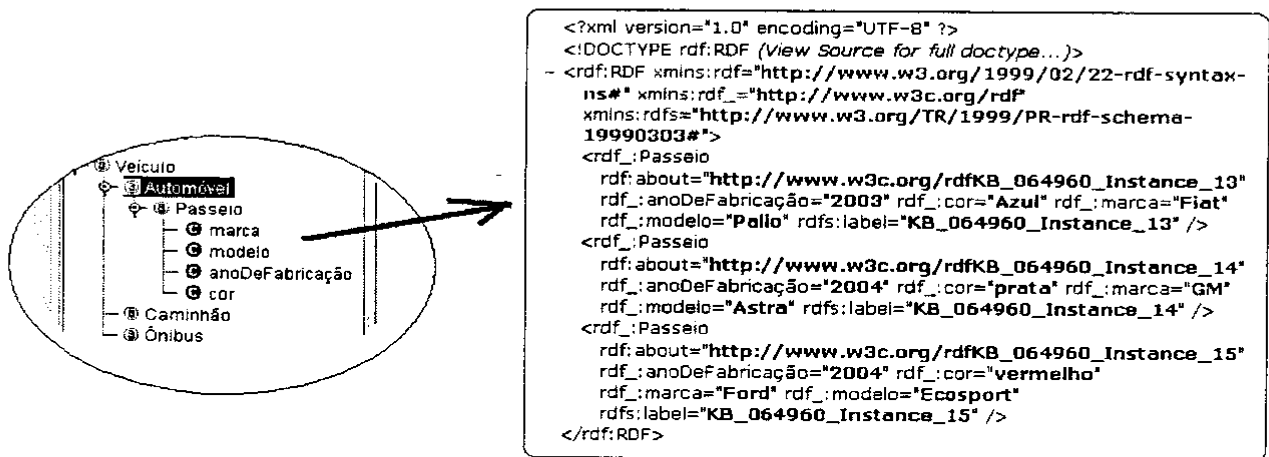


Figura 3.5: Compartilhamento da Ontologia

A seguir, são abordadas as características da ferramenta *Ontolegacy*, essenciais no contexto desta metodologia, viabilizando, sobretudo, a etapa da associação da *Ontologia* dos SIL aos termos e conceitos extraídos.



### 3.3. A ferramenta computacional: *Ontolegacy*

A ferramenta *Ontolegacy* é um facilitador na associação da *Ontologia* do SIL dos objetos reais – códigos fontes – aos objetos conceituais – *Ontologia* propriamente dita dos SIL. Considera-se que uma das características mais importantes do *Ontolegacy* seja a capacidade de extrair e gerar termos e conceitos para inclusão na *Ontologia* dos SIL e também o mapeamento e localização dos termos nos códigos fontes relacionados ao SIL em questão.

Apesar dos artefatos computacionais em IA em geral usarem linguagens de programação como LISP ou PROLOG, optou-se como plataforma de implementação do *Ontolegacy* a linguagem JAVA. Seu uso é de grande difusão na comunidade acadêmica da computação e de fácil comunicação entre os profissionais da área programação, existindo, assim, uma boa maturidade e estabilidade nas soluções adotadas. Outra razão para escolher a plataforma foi a portabilidade e interoperabilidade entre diversos sistemas operacionais, já que pelas características e aplicações do sistema, deverá estar apto para se integrar a diversos ambientes como, por exemplo, ser acessada via Web.

O sistema KISF [PAC03], citado na seção 2.2.6, *framework*, baseado em agentes computacionais, será usado em associação com a ferramenta ou sistema *Ontolegacy* a fim de:

- a) disponibilizar o uso do editor de *Ontologias*;
- b) armazenar as *Ontologias*;
- c) compartilhamento do ambiente de SMA; e
- d) interface para o ambiente de pesquisa e consultas das *Ontologias*.

#### 3.3.1. Arquitetura do sistema *Ontolegacy*

Na arquitetura do *Sistema Ontolegacy*, têm-se basicamente os seguintes elementos: *bases de conhecimento e sociedade de agentes* [Figura 3.6].

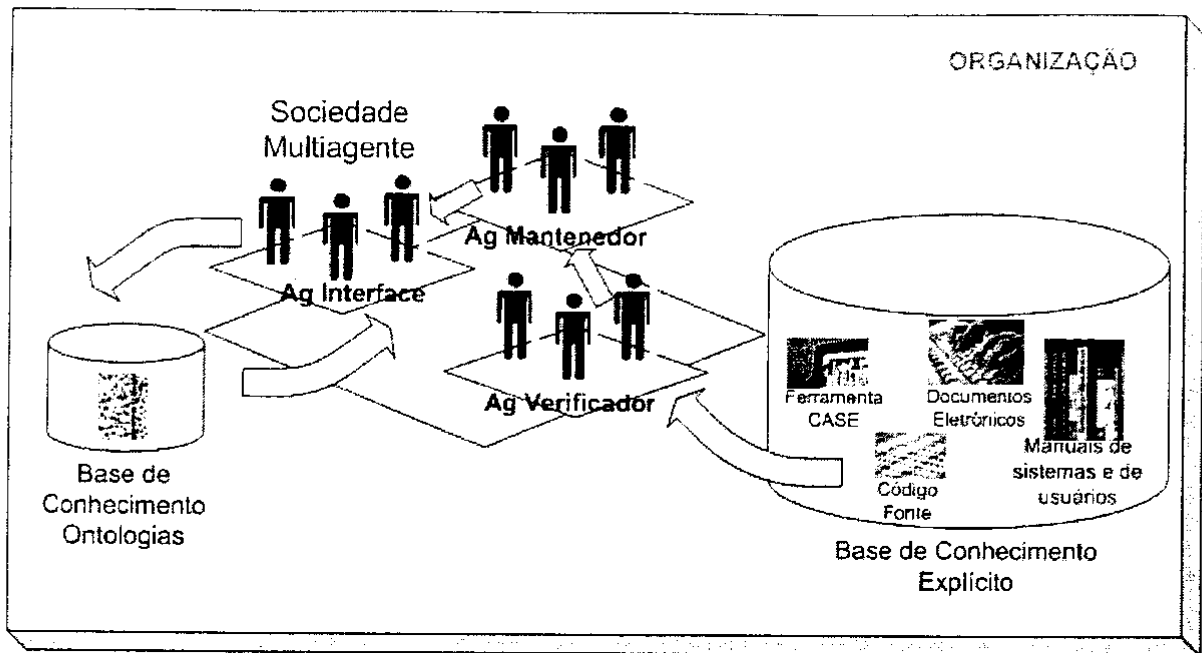


Figura 3.6: Arquitetura Geral do Sistema Ontolegacy

### 3.3.2. Bases de conhecimento

São providas de conhecimentos compartilhados pela sociedade de agentes para o alcance dos seus objetivos. Divididas em dois grupos principais são constituídas de:

- a) *base de conhecimento ontológico* – onde estão armazenadas as *Ontologias* dos SIL. Local onde o engenheiro do conhecimento ou o analista de sistemas do corpo técnico responsável pelo sistema armazena e mantém a *Ontologia* do SIL.
- b) *base de conhecimento explícito* – onde está armazenado o conhecimento explícito do SIL, submetido ao processo de *externalização* do conhecimento [NNT97]. Nas organizações, há uma diversidade de mídias nas quais está formalizado o conhecimento tácito dos SIL. No contexto deste trabalho, é considerada apenas a mídia Código Fonte COBOL, por possuir a maior parte das informações do negócio a respeito dos SIL; ainda, por deter, em termos quantitativos, a maioria das fontes, e por refletir a situação real dos SIL, ou seja, o sistema que está ativo funciona exclusivamente em função do conteúdo do código fonte.

### 3.3.3. Sociedade de agentes do *Ontolegacy*

A sociedade de agentes do *Ontolegacy* é composta de três grupos, formados de um ou mais subagentes, no sistema multiagente proposto, tendo uma *interface* de comunicação com

a base de dados do sistema em que são acessados os dados gerados na investigação dos códigos fontes. Tais dados são usados no processo de execução dos agentes, conforme segue.

- a) *agente verificador* (AgVerificador) – é responsável por verificar a ocorrência de novos fenômenos relevantes ao ambiente dos SIL da organização. Ou seja, em locais pré-determinados, são armazenados os códigos fonte do SIL e toda alteração nos CF. O agente perceberá a alteração e repassará a informação ao AgMantenedor.
- b) *agente mantenedor* (AgMantenedor) – é um meta-agente, responsável pelo gerenciamento de um conjunto de agentes, os quais executam as operações de extração dos conceitos. Esse conjunto (AgMSeleção, AgMAnalisador, AgMConceitos) apresenta características específicas, relacionadas à extração de novas informações dos SIL.
- c) *agente interface* (AgInterface) – é responsável pela exibição dos resultados gerados pelos agentes mantenedores. Apresenta a arquitetura da *sociedade de agentes* [Figura 3.7].

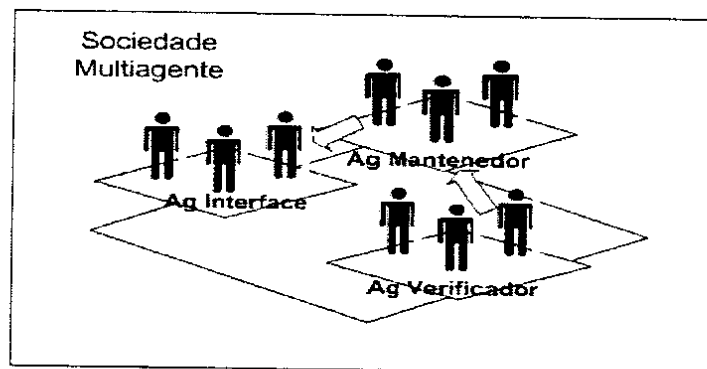


Figura 3.7: Sociedade de Agentes Ontolegacy

#### 3.3.4. AgVerificador

A fim de obter-se uma *Ontologia* consistente a respeito dos SIL de uma organização, é essencial que a *Ontologia* seja atualizada no instante em que qualquer novo fenômeno ocorra. A tarefa desse agente é verificar nas fontes de informação a incidência de novos fenômenos. Entende-se por fonte de informação um espaço lógico que contém um conjunto de fenômenos. Conforme citado anteriormente, a fonte de informação básica, eleita para uso com o sistema *Ontolegacy*, refere-se aos códigos fontes (programas) em COBOL.

Tabela 3.1: Definição Abstrata do AgVerificador

Parâmetros	Descrição
Competência	Verificador de alteração do ambiente: percepção dos fenômenos a partir de um conjunto de meio configurado.
Informação de entrada	Local da ocorrência dos fenômenos, diretórios ou pastas.
Informação de saída	Cada fenômeno percebido a partir do meio selecionado

Os fenômenos são, constantemente, observados pelo agente em suas respectivas fontes de informação. Dessa forma, o agente *AgVerificador* possui também um procedimento de atualização na busca por novos fenômenos. Assim, a cada intervalo de tempo pré-definido, verifica-se sua ocorrência. Todos os fenômenos obtidos a partir da fonte de informação selecionada são encaminhados para o agente *AgMantenedor*.

O *AgVerificador* possui cinco componentes de configuração, como mostra a Figura 3.8. O componente de coordenação do agente utiliza a coordenação por padronização, uma vez que os agentes sabem quais são suas tarefas e como elas devem ser executadas.

Assim, os agentes criados detêm conhecimento necessário para obter os fenômenos nas fontes de informação. O planejamento do agente indica a ordem de análise da fonte de informação configurada.

Com relação à aprendizagem, o agente adiciona à sua estrutura as fontes de informação já cadastradas, não sendo necessário recadastrar tais informações para a criação de um novo agente, de maneira que a *Expertise* desse agente é representada pela percepção dos fenômenos ocorridos na fonte de informação. Finalmente, o componente de comunicação gerencia a troca de mensagens e de informações entre os agentes.

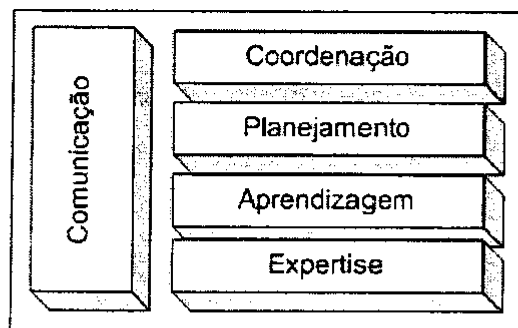


Figura 3.8: Configuração AgVerificador

### 3.3.5. AgMantenedor

O agente denominado *AgMantenedor* é um meta-agente responsável por coordenar um conjunto de agentes cujas competências são bem definidas, quais sejam: a seleção dos CF, a transformação dos CF para análise e a extração dos conceitos e termos respectivos. Conforme ilustração, [Figura 3.9], demonstra-se a estrutura do *AgMantenedor* e seus subagentes.

Os agentes mantenedores utilizam a sintaxe *AgM<Competência>*; conseqüentemente, o meta-agente fica composto dos subagentes, conforme a seguinte seqüência: *AgMSeleção*; *AgMAnalisador* e *AgMConceito*.

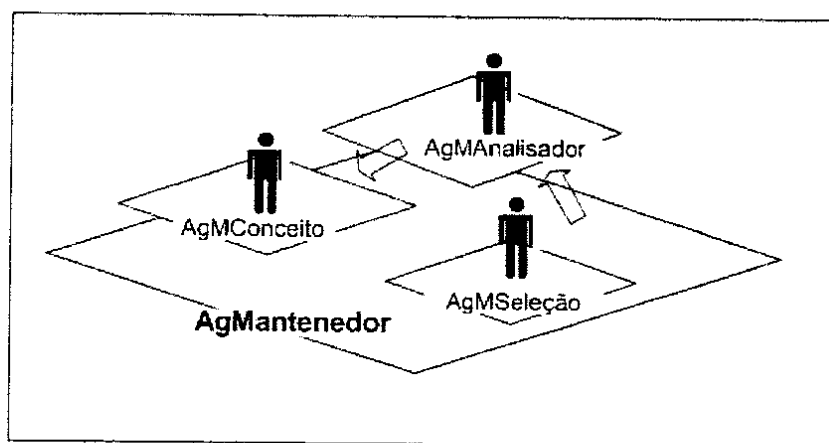


Figura 3.9: Meta-agente Mantenedor

#### a) AgMSeleção

O meio configurado para análise posterior são os CF e este se apresenta em dois formatos; ou seja, com o arquivo de texto do código fonte “puro” ou o arquivo de texto no qual se tem todo o código submetido à compilação; isto é, a listagem expandida em que o código foi submetido anteriormente por um pré-compilador.

Tabela 3.2: Definição Abstrata do agmseleção

Parâmetros	Descrição
Competência	Seleção do meio configurado: a partir dos fenômenos nele ocorridos, prepara-se o meio para análise
Informação de entrada	Cada fenômeno percebido a partir do meio selecionado
Informação de saída	Meio preparado para análise dos conceitos

Nesta dissertação, optou-se pelo segundo meio, por ter embutidos os *layouts* dos arquivos e tabelas de banco de dados do CF [Figura 3.10], embora este meio esteja de alguma forma “poluído”, com linhas que não dizem respeito ao código fonte em si.

```

IPP 5668-958 IBM VS COBOL II Release 4.0 09/15/92                               Date 03/07/02 Time 15:44:46 Page 1
Invocation parameters:
NOADV, FLAG(I, I), LIB, MAP, NONAME,, OBJECT, OFF, APOST, RENT, RES, TRUNC (BIN), VBREF, X,
Options in effect:
NOADV
APOST
NOAWO
BUFSIZE(4096)
NOWORD
XREF(FULL)
ZWB
IPP 5668-958 IBM VS COBOL II Release 4.0 09/15/92                               XYZ510 Date 03/07/02 Time 15:44:46 Page 2
LineID  PL SL  -----A-1-B-----2-----3-----4-----5-----6-----7-|-----8 Map and Cross Reference
0 000001          IDENTIFICATION DIVISION.
000003          PROGRAM-ID.                                XYZ510.
000007          000700* APLICACAO..... XYZ - ARREC PREVIDENCIA *
000009          000900* ANALISTA..... FULANO DE TAL *
000013          001300* CODIFICADO EM..... MARCO/1999 *
000024          002400* FUNCAO..... CONSISTENCIA DO MOVIMENTO DIARIO GPS *
000035          ENVIRONMENT DIVISION.
000037          CONFIGURATION SECTION.
000039          SPECIAL-NAMES.
000041          DECIMAL-POINT                                IS COMMA.
000043          INPUT-OUTPUT SECTION.
000045          FILE-CONTROL.
000047          SELECT XYZE501                                66
000048          ASSIGN                                       XYZE501
000049          STATUS                                        AS-STATUS.                                686
PP 5668-958 IBM VS COBOL II Release 4.0 09/15/92                               XYZ510 Date 03/07/02 Time 15:44:46 Page 3
LineID  PL SL  -----A-1-B-----2-----3-----4-----5-----6-----7-|-----8 Map and Cross Reference
0 000062          DATA DIVISION.
000064          FILE SECTION.
000066          FD XYZE501
000067          BLOCK                                          0
000068          RECORDING                                      F
000069          RECORD                                        200
000070          LABEL RECORD IS                                STANDARD.
000072          COPY 'XYZD501A'.
00118C          01 D501B-REG.                                BLF=0000+000
00119C          05 D501B-TIPO-REG                            PIC 9(001).                                BLF=0000+000,0000000 1C
00120C          05 D501B-DT-PROCES                          PIC 9(008).                                BLF=0000+001,0000001 8C
00121C          05 D501B-NUM-AG                             PIC 9(005)                                COMP-3.                                BLF=0000+009,0000009 3P
00122C          05 D501B-NUM-POSTO                          PIC 9(003)                                COMP-3.                                BLF=0000+00C,000000C 2P
00123C          05 D501B-CD-RCT                             PIC 9(005)                                COMP-3.                                BLF=0000+00E,000000E 3P
00124C          05 D501B-CD-IDENT                          PIC 9(015)                                COMP-3.                                BLF=0000+011,0000011 8P
00125C          05 D501B-DT-COMPET                         PIC 9(006).                                COMP-3.                                BLF=0000+019,0000019 6C
00126C          05 D501B-V-INSS                            PIC 9(013)V99                            COMP-3.                                BLF=0000+01F,000001F 8P
00127C          05 D501B-V-OUTRAS                          PIC 9(013)V99                            COMP-3.                                BLF=0000+027,0000027 8P
    
```

Figura 3.10: Código Fonte Expandido

Após a atuação do agente *AgMSeleção*, o meio é transformado em um texto mais limpo, facilitando sua análise posterior pelo *AgMAnalisador* [Figura 3.11].

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. XYZ711.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
SPECIAL-NAMES.  
DECIMAL-POINT IS COMMA.  
INPUT-OUTPUT SECTION.  
FILE-CONTROL.  
SELECT XYZE199  
ASSIGN XYZE199  
STATUS AS-STATUS.  
DATA DIVISION.  
FILE SECTION.  
FD XYZE199  
BLOCK 0  
RECORDING F  
RECORD 200  
LABEL RECORD IS STANDARD.  
COPY 'XYZD199A'.  
01 D199B-REG.  
05 D199B-TIPO-REG PIC 9(001).  
05 D199B-DT-PROCES PIC 9(008).  
05 D199B-NUM-AG PIC 9(005) COMP-3.  
05 D199B-NUM-POSTO PIC 9(003) COMP-3.  
05 D199B-CD-RCT PIC 9(005) COMP-3.  
05 D199B-CD-IDENT PIC 9(015) COMP-3.  
05 D199B-DT-COMPET PIC 9(006).  
05 D199B-V-INSS PIC 9(013)V99 COMP-3.  
05 D199B-V-OUTRAS PIC 9(013)V99 COMP-3.
```

Figura 3.11: Código Fonte selecionado

## b) *AgMAnalisador*

O agente *AgMAnalisador* tem um papel importantíssimo no que se refere à preparação do meio para extração dos termos e conceitos. Atua como um *parser*, fazendo uma análise da sintaxe da linguagem COBOL com o propósito de selecionar as linhas do código fonte essenciais para a atuação do agente *AgMConceito*.

Tabela 3.3: Definição Abstrata do AgMAnalisador

Parâmetros	Descrição
Competência	Análise gramatical: extrai do meio configurado de acordo com a linguagem utilizada (COBOL) as partes essenciais para extração dos termos e/ou conceitos
Informação de entrada	Meio preparado para análise dos conceitos
Informação de saída	Meio submetido ao analisador sintático

O produto resultante desse agente compreende as linhas de código necessárias para a construção dos conceitos e seleção de termos. De acordo com a tabela abaixo, são selecionados alguns comandos da linguagem COBOL para que produzam o efeito citado.

Tabela 3.4: Comandos COBOL selecionados

Tipo	Tipo	
Fonte	Parâmetro	Comando
EXP	INICIALIZA	IDENTIFICATION
EXP	SELECIONA	MOVE
EXP	SELECIONA	ADD
EXP	SELECIONA	SUBTRACT
EXP	SELECIONA	MULTIPLY
EXP	SELECIONA	DIVIDE
EXP	SELECIONA	COMPUTE

### c) AgMConceito

O agente *Conceito* atua no meio selecionado, preparado e submetido ao analisador sintático com o objetivo de gerar os termos e conceitos retirados dos CF. A proposta da metodologia, conforme descrição na seção 3.2.2, baseia-se nos metadados dos arquivos e banco de dados embutidos nos CF, particularmente nos códigos fontes expandidos [Figura 3.10]. Cabe ressaltar questões relacionadas à distinção dos termos e conceitos a serem extraídos, porque em CF COBOL é comum que grande parte dos termos, na programação



nominadas de variáveis, sejam operacionais e do negócio. Como é priorizada a Ontologia dos SIL voltada ao negócio, o agente descarta parte desses termos durante sua atuação.

Tabela 3.5: Definição abstrata do AgMConceito

Parâmetros	Descrição
Competência	Sugere a criação de novos conceitos no plano conceitual da <i>Ontologia</i> e sua associação a conceitos já existentes.
Informação de entrada	Meio submetido ao analisador sintático
Informação de saída	Sugestões de criação de novos conceitos e termos gerados pelo agente

Embora o agente não possua inteligência específica para distinguir termos ou conceitos operacionais ou de negócio, o agente Interface possibilita ao corpo técnico do SIL a opção da seleção dos termos e conceitos relevantes à *Ontologia*.

### 3.3.6. AgInterface

Os novos conceitos e termos sugeridos pelo AgMantenedor não são atualizados na Ontologia antes da aprovação pelo corpo técnico. Assim, o AgInterface recebe todas as sugestões de resultados provenientes do AgMantenedor. O AgInterface, caracterizado como um agente *interface* conforme seu nome já explicita, permite que o corpo técnico visualize todas as sugestões listadas, para poder selecionar e aprovar aquelas sugestões que possam auxiliar na complementação da *Ontologia* dos SIL. No momento em que o corpo técnico aprova as sugestões, elas são incorporadas automaticamente à estrutura da *Ontologia* do SIL.

Tabela 3.6: Definição Abstrata do AgInterface

Parâmetros	Descrição
Competência	Exibição dos Resultados sugeridos
Informação de entrada	Sugestões de criação de novos conceitos e termos gerados pelo agente
Informação de saída	Sugestões dos conceitos gerados pelo agente

O *AgInterface* tem dois componentes de configuração: a *expertise*, que compreende a exibição dos resultados obtidos por todos os agentes mantenedores e a efetivação, na Ontologia, dos resultados aprovados pelo corpo técnico. O outro é o de comunicação, que gerencia a troca de mensagens e informações entre os agentes [Figura 3.12].

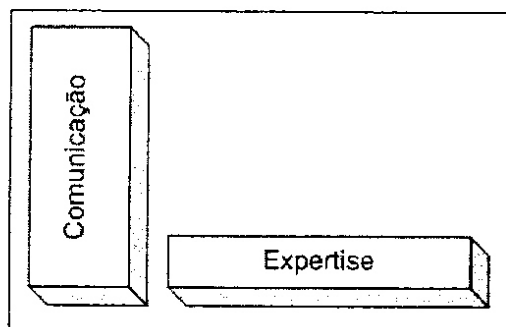


Figura 3.12: Configuração AgInterface

### 3.4. Considerações Finais

A metodologia proposta para gestão do conhecimento em SIL, baseado em *Ontologias*, norteia-se nas seguintes premissas:

- a) construção da *Ontologia* dos SIL;
- b) definição dos meios para extração dos termos e conceitos;
- c) associação da *Ontologia* dos SIL com os termos e conceitos extraídos; e
- d) compartilhamento do conhecimento dos SIL.

Conseqüentemente, ao se usar a metodologia, é necessário lançar mão de artefatos computacionais no auxílio da implementação da metodologia.

A ferramenta *Ontolegacy* é composta de três partes principais, a saber: *base de conhecimento*, composta das *Ontologias*; *repositório das informações explícitas*, e *sociedade de agentes*, atuador responsável pela complementação da *Ontologia* no repositório das informações explícitas. Na ocorrência dos fenômenos organizacionais, estas partes interagem na consecução dos objetivos pré-estabelecidos.

A ferramenta está apta a atuar com diversos domínios de conhecimento de SIL, apresentando, na sua interface, os locais onde estão os códigos fontes e o resultado das suas extrações de termos (entidades e atributos), bem como a localização destes termos.

O capítulo seguinte trata de um experimento aplicando a metodologia proposta com o objetivo de validar o funcionamento do protótipo da ferramenta *Ontolegacy*.

## Capítulo 4

# EXPERIMENTO

Com o intuito de validar a metodologia e testar o protótipo *Ontolegacy* como ferramenta, realizou-se um experimento baseado num Sistema de Informação Legado, no domínio do conhecimento, relativamente à *Arrecadação da Previdência Social*. O Sistema de Informação (SI) eleito é um caso prático de legado, ativo num ambiente de *mainframe* doravante denominado de sistema **XYZ**.

### **4.1. O Sistema de Informação Legado: arrecadação da Previdência Social**

O Sistema de Informação eleito para o experimento abrange todo o processo de arrecadação, referente ao numerário, pertencente ao Instituto Nacional da Previdência Social (INSS). A organização, com características de uma instituição financeira bancária, da qual faz parte o sistema nomina-se simplesmente de *BlackStone*. Esta organização, assim como outras existentes no país, prestam o serviço de arrecadação e os repassam ao INSS.

Além de repassarem numerários, enviam também informações dos contribuintes que fazem parte do INSS. O processo não envolve tão-somente o envio de informações e numerários; abrange também fenômenos no *inner* das organizações quer no sentido operacional quanto dos negócios.

Resumidamente, o processo inicia-se com o documento Guia da Previdência Social (GPS) [Figura 4.1]. No momento em que o contribuinte recolhe o valor do documento no caixa da instituição *BlackStone*, são gerados fenômenos que decorrem em processos dentro da organização. A entrada dos dados da GPS ocorre nos caixas, terminais de auto-atendimento ou *Internet Bank*. Na seqüência, tais dados são submetidos ao processamento, contabilização e, finalmente, é gerado o repasse dos valores e informações dos contribuintes para o INSS.


 MINISTÉRIO DA PREVIDÊNCIA SOCIAL - MPS INSTITUTO NACIONAL DO SEGURO SOCIAL - INSS <b>GUIA DA PREVIDÊNCIA SOCIAL - GPS</b>	3 CÓDIGO DE PAGAMENTO	1007
	4 COMPETÊNCIA	05/2004
	5 IDENTIFICADOR	12356789
1 NOME OU RAZÃO SOCIAL/FONE/ENDEREÇO:		6 VALOR DO INSS
FULANO DE TAL		148,50
(41) 333-4444		7
RUA DAS MANGABEIRAS, 55		8
CURITIBA - PR		9 VALOR DE OUTRAS ENTIDADES
2 Vencimento (Uso do INSS)		10 ATM, MULTA E JUROS
		1,50
ATENÇÃO: É vedada a utilização de GPS para recolhimento de receita de valor inferior ao estipulado em Resolução publicada pelo INSS. A receita que resultar valor inferior deverá ser adicionada à contribuição ou importância correspondente nos meses subsequentes, até que o total seja igual ou superior ao valor mínimo fixado.		11 TOTAL
		150,00
12 AUTENTICAÇÃO BANCÁRIA		

Figura 4.1: Guia GPS

#### 4.2. Aplicação da metodologia Ontolegacy

Para aplicação da metodologia proposta, conforme referenciado no capítulo 3, seção 3.2, um dos passos iniciais é a construção da *Ontologia*. Porém, alguns requisitos são necessários para que o corpo técnico possa desenvolvê-la, de modo que estejam disponíveis algumas formas de conhecimento tácito e/ou explícito.

No presente experimento, basicamente, o usuário recorreu:

- a) ao modelo de dados em ferramenta CASE [Figura 4.2];
- b) à documentação eletrônica disponível (conhecimentos explícitos);
- c) à verificação *in-loco* do funcionamento do SIL; e
- d) ao corpo técnico responsável (conhecimentos tácitos).



Figura 4.2: Modelo de dados do Sistema XYZ

#### 4.2.1. Construção da Ontologia

Ao utilizar como referência os meios citados anteriormente, o usuário criou uma *Ontologia* procurando seguir o modelo preconizado por Noy e McGuinness [NYG00].

Primeiramente determinou-se o escopo da *Ontologia*, a saber:

- auxiliar no processo de extração do conhecimento do SIL XYZ, contribuindo para capitalizar o conhecimento, permitindo a manutenção do XYZ e futuras atualizações;
- definir uma *Ontologia* de forma a atender ao domínio de conhecimento dos SIL legados, voltados para o ambiente de instituições bancárias;

- c) considerar que a granularidade da *Ontologia* proposta trata, num primeiro momento, dos aspectos macros, sem preocupação com os detalhes elementares de todo o sistema XYZ; e
- d) direcionar o uso a técnicos da área de sistemas e de negócios que necessitem conhecer o XYZ para dar manutenções, conhecer o negócio e atualizar módulos do sistema.

Espera-se que algumas questões, conforme abaixo, sejam respondidas pelas informações contidas na *Ontologia*, após seu desenvolvimento.

- a) Quais são os conceitos envolvidos no sistema XYZ?
- b) Qual é a estrutura de negócio do sistema XYZ?
- c) Quais as interfaces que o sistema XYZ possui com outros SI?
- d) A modificação de um determinado módulo afetará quais outros módulos do XYZ?

Em termos de reusabilidade, não se utilizou nenhuma outra *Ontologia*, visto que o ambiente não possuía nada semelhante; apenas recorreu-se ao dicionário de dados associado ao banco de dados corporativo.

No passo seguinte, foram determinados os *termos comuns*, para o entendimento da *Ontologia* proposta no domínio considerado. Desse modo, foi preciso definir alguns termos para compartilhar os conceitos do domínio relativo à contribuição previdenciária.


Tabela 4.1: Termos e Definições

Termos	Definições
GPS	Guia da Previdência Social, documento no qual são cadastrados os dados do contribuinte e valores a contribuir
Contribuinte	Pessoa física ou jurídica cadastrada no INSS, que contribui para a Previdência Social com o intuito de receber os devidos benefícios da Instituição
Nome do contribuinte	Como a pessoa do contribuinte é conhecida
Endereço do contribuinte	Local onde reside o contribuinte, logradouro, número do local e cep
Telefone do contribuinte	Identificação numérica do telefone do contribuinte
Razão social do contribuinte	Identificação nominal da organização do contribuinte
Código do pagamento	Identificação numérica do tipo do pagamento (Anexo A.2)
Competência	Data (mês e ano) referente ao objeto da contribuição
Identificador	Número do NIT ou PIS/Pasep do contribuinte
Valor do INSS	Valor devido ao INSS pelo contribuinte
Valor de ATM	Valor devido a título de atualização monetária e acréscimos legais, sobre recolhimentos em atraso
Valor de multas	Valor monetário devido sobre a falta de recolhimento das contribuições urbanas e rurais devidas ao INSS.

	correspondente àquela vigente à época de ocorrência, sobre o valor atualizado monetariamente até a data do efetivo recolhimento.
Valor de juros	Valor devido sobre as contribuições previdenciárias pagas após o vencimento, incidentes sobre o valor atualizado equivalentes a: 1% no mês de vencimento, Taxa referencial do SELIC (Sistema Especial de Liquidação e de Custódia) nos meses intermediários e 1% no mês de pagamento.
Valor total	Soma dos valores devidos (do INSS, ATM, Multas e Juros).
Autenticação bancária	Processo físico de indicação do pagamento efetuado em caixas bancários.

O próximo passo foi construir os conceitos estruturados hierarquicamente num editor de Ontologia, no caso o editor do KISF, conforme se demonstra, ou seja: a *Ontologia* do sistema XYZ [Figura 4.3].

Vale ressaltar que alguns conceitos foram reutilizados como, por exemplo, o de contribuinte conforme o conceito da GPS, assim como o conceito de pessoa foi reutilizado pelo conceito contribuinte e assim por diante.

Os conceitos em reusabilidade aparecem no editor com o ícone  após o nome, diferenciando-se dos demais. A partir desta distinção, pode-se visualizar vários conceitos sendo reutilizados.

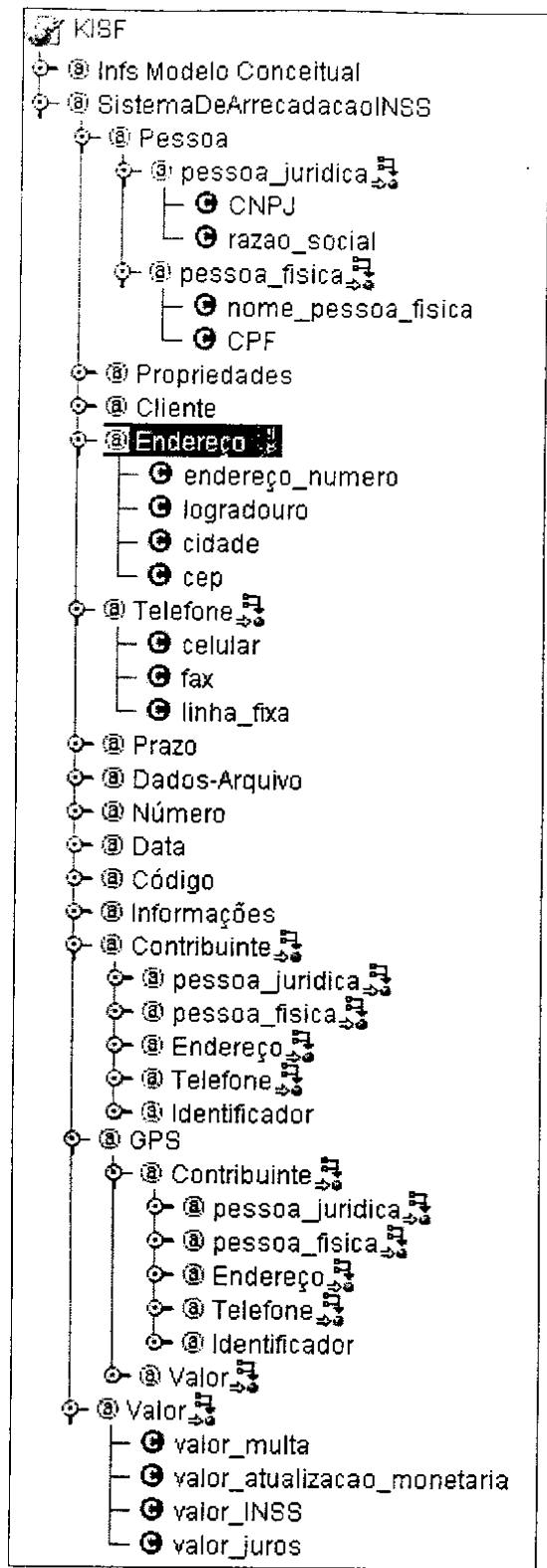


Figura 4.3: Ontologia com a hierarquia dos conceitos



#### 4.2.2. Definição dos meios para extração dos termos e conceitos

Após a conclusão do desenvolvimento da *Ontologia*, passou-se à determinação dos meios, objetos físicos, para extrair os termos e conceitos a serem a ela associados.

Inicialmente, a ferramenta Ontolegacy é direcionada aos CF em COBOL. Foram transferidas das bibliotecas, local onde ficam armazenados os CF no *mainframe*, as listagens expandidas em texto. Procurou-se utilizar apenas os CF ativos; isto é, que ainda estão em permanente uso pela organização. Estes CF estão subdivididos em duas variantes, dialetos, da linguagem COBOL, a saber:

##### a) Código Fonte VS COBOL II IBM com SQL

São os CF onde estão inseridos, além das instruções em COBOL, as instruções SQL e *layouts* das tabelas do banco de dados DB2<sup>6</sup> [Figura 4.3].

```
WORKING-STORAGE SECTION.
*** EXEC SQL
***   INCLUDE XYZBCPA
*** END-EXEC
*****
* COBOL DECLARATION FOR TABLE DEXYZ.CTL_PRO_XYZ
*****
01 DCLCTL-PRO-XYZ.
   10 CPA-NUH-SIST          PIC S9(3)V USAGE COMP-3.
   10 CPA-DT-PROCES        PIC S9(8)V USAGE COMP-3.
   10 CPA-DT-PROCES-ANT    PIC S9(8)V USAGE COMP-3.
   10 CPA-NUH-REHES-GPS    PIC S9(7)V USAGE COMP-3.
.

PROCEDURE DIVISION.

PERFORM AOO-INICIALIZA
PERFORM BOO-PROCESSA
PERFORM WOO-FINALIZA
GOBACK.

AOO-INICIALIZA SECTION.
MOVE 'XYZ - ARRECADACAO DA PREVIDENCIA FEDERAL'
      TO WOO1-SISTEMA
MOVE 'XYZS10'
      TO WOO1-PROGRAMA
MOVE 'DIARIA'
      TO WOO1-ROTINA
MOVE WHEN-COMPILED
      TO WOO1-DATA-CATALOGACAO
MOVE MSC-FERMINO-ANORMAL
      TO WOO1-MENSAGEM-5
MOVE 'AOO-INICIALIZA'
      TO WOO1-PARAGRAFO

EXEC SQL
SELECT  DT_PROCES,
        DT_PROCES_ANT
INTO    :CPA-DT-PROCES,
        :CPA-DT-PROCES-ANT
FROM    DEXYZ.CTL_PRO_XYZ
WHERE   NUM_SIST = :CPA-NUH-SIST
END-EXEC

EVALUATE SQLCODE
WHEN 0
```

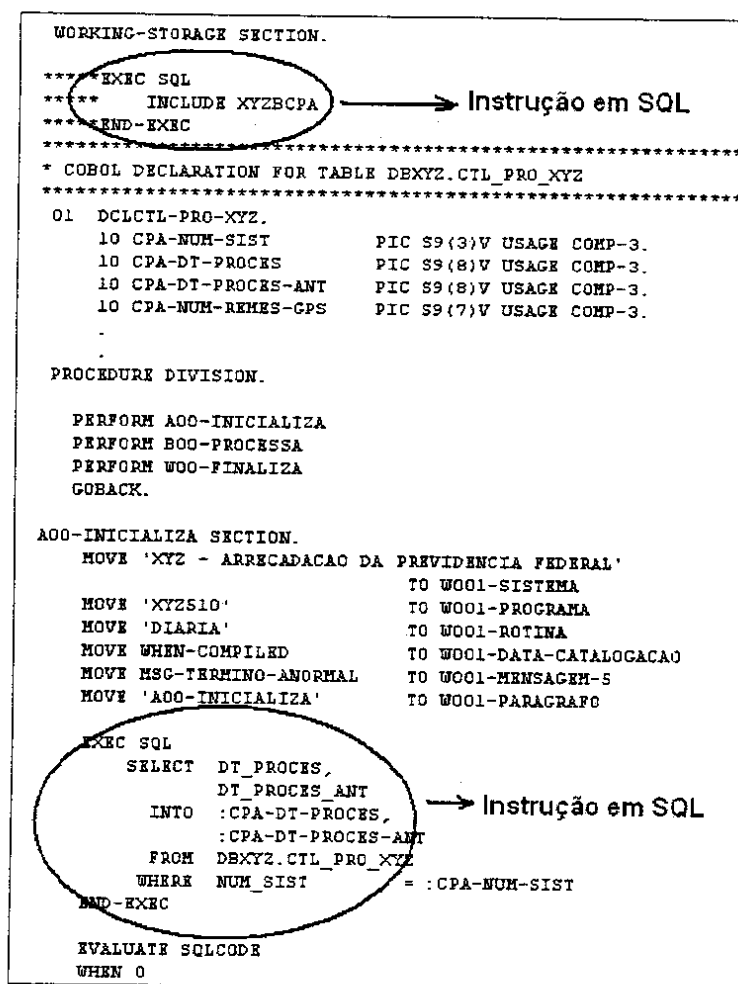
The image shows a snippet of COBOL code with two instances of SQL instructions circled in black. An arrow points from the label 'Instrução em SQL' to the first circled instance, which is the 'EXEC SQL INCLUDE XYZBCPA' block. Another arrow points from the label 'Instrução em SQL' to the second circled instance, which is a 'SELECT' query. The code includes a 'WORKING-STORAGE SECTION' with data declarations, a 'PROCEDURE DIVISION' with a 'PERFORM' statement, and an 'AOO-INICIALIZA SECTION' with 'MOVE' statements. The 'SELECT' query is followed by 'END-EXEC' and an 'EVALUATE SQLCODE' block.

Figura 4.4: Código Fonte VS COBOL II com SQL

<sup>6</sup> DB2 (*Universal Data Base*) é um produto de banco de dados relacional, da empresa IBM.

## b) Código Fonte VS COBOL II IBM CICS com SQL

Trata-se do código fonte que faz parte da aplicação XYZ, em que estão inseridas, além do código em COBOL, as instruções SQL, instruções CICS para transações *on-line* e *layouts* das tabelas do banco de dados DB2. O CICS é um *software* que gerencia e processa transações *on-line* para aplicações de negócios críticos. É usado em vários tipos de redes que variam no tamanho de alguns terminais a milhares de terminais. Abaixo são demonstrados alguns aspectos que diferem este dialeto do COBOL [Figura 4.4].

```
IDENTIFICATION DIVISION.
PROGRAM-ID.                XYZP101.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    DECIMAL-POINT          IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
01 FILLER                  PIC X(032)      VALUE
    'III WORKING-STORAGE SECTION III'.
    EXEC SQL
        INCLUDE XYZBCPA
    END-EXEC
* PARAMETERS CICS
01 ARRA-SYSID.
    05 AS-CICS.
        10 FILLER          PIC X(004)      VALUE 'CICS'.
        10 AS-NUM-CICS     PIC X(002)      VALUE SPACES.
        10 FILLER          PIC X(002)      VALUE SPACES.
    05 AS-SYSID.
        10 FILLER          PIC X(002)      VALUE 'IS'.
        10 AS-NUM-SYSID   PIC X(002)      VALUE SPACES.
LINKAGE SECTION.
01 DFHCOMMAREA             PIC X(100).
PROCEDURE DIVISION.
AOO-ROTINA-PRINCIPAL SECTION.
    MOVE 'PF11'             TO AS-TRANSACAO
    MOVE AS-LK-NUM-CICS     TO AS-NUM-CICS
    IF EIBCALEN             = ZEROS
        PERFORM AOO-10-PRIMEIRA-VEZ
    ELSE
        MOVE DFHCOMMAREA   TO AS-LINKAGE
        PERFORM AOO-20-RECEIVE
    END-IF
    EXEC CICS
        ASSIGN
        APPLID             (AS-CICS)
    END-EXEC
    MOVE AS-NUM-CICS       TO AS-LK-NUM-CICS
    EXEC CICS
        LINK
        PROGRAM             ('SRBSOOL')
        COMMAREA            (AS-SYS)
    END-EXEC
```

comando SQL

comando CICS

Figura 4.5: Código Fonte VS COBOL II CICS

Na tabela a seguir, são demonstradas as quantidades de CF utilizados no experimento.

Tabela 4.2: Estatística dos CF

Descrição dos CF	Quantidades de CF	Linhas de programas
VS COBOL II <i>batch</i> com <i>middleware</i> SQL DB2	21	26.770
VS COBOL II CICS on-line com <i>middleware</i> SQL DB2	28	102.014
Total de CF utilizados nos experimentos	49	128.784

#### 4.2.3. Associação da *Ontologia* dos SIL com os termos e conceitos extraídos

Nessa fase, associa-se a *Ontologia* criada aos termos e conceitos extraídos dos CF determinados na seção anterior, decorrentes da atuação da sociedade de agentes [Figura 4.4] contida na ferramenta *Ontolegacy*. Conforme os agentes atuaram entre si, houve a verificação, seleção, análise e liberação dos termos e conceitos encontrados nos códigos fontes.

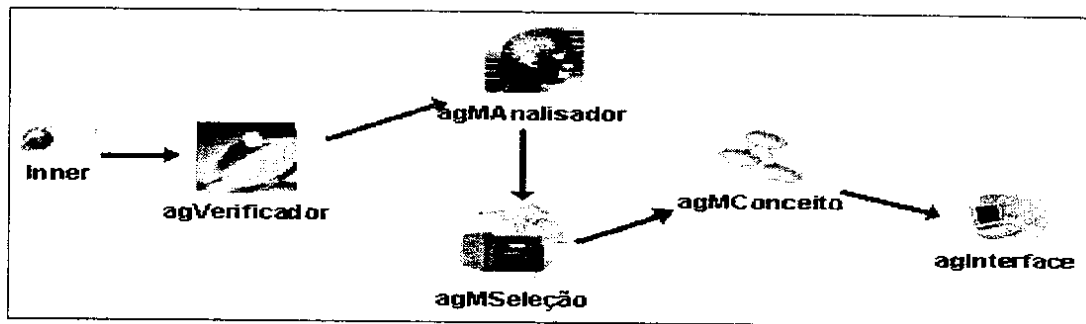


Figura 4.6: Sociedade de Agentes Ontolegacy

Os conceitos e termos foram gerados a partir das entidades e atributos presentes nos CF (*Anexo C.2*). A partir do código fonte XYZ510, foram identificadas as entidades e seus atributos, conforme [Tabela 4.3] (*Detalhes no apêndice C.1*).

Tabela 4.3: Entidade e atributos do CF XYZ510

Entidades	Atributos	Entidades	Atributos
Tabelas DB2		Arquivos Sequenciais	
MOV-ACM (Tabela DB2)	MAC-CD-BARRAS MAC-CD-IDENT MAC-DT-PROCES MAC-NUM-AG MAC-TP-DOC MAC-TP-PGTO MAC-V-ATRASSO MAC-V-ATRASSO MAC-V-OUTRAS MAC-V-PAGO	XYZE199	D199B-NUM-AG D199B-NUM-POSTO D199B-CD-RCT D199B-CD-IDENT D199B-DT-COMPET D199B-DT-PROCES D199B-V-OUTRAS D199B-V-ATRASSO D199B-V-PAGO D199B-VIA-CARTAO

MOV-IRG	MIR-CD-IDENT MIR-DT-COMPET MIR-V-OUTRAS MIR-V-ATRASSO MIR-V-PAGO	XYZS198	D199B-TITULARIDADE D199B-DT-PGTO-EFET D199B-HRR-PGTO D198B-TP-IDENT D198B-CD-IDENT D198B-DT-COMPET D198B-V-OUTRAS D198B-V-ATRASSO D198B-V-PAGO
CTL-PRO-KYZ (Tabela DB2)	CPA-DT-PROCES CPA-DT-PROCES-ANT	XYZS197	D197A-REG D197B-NUM-AG D197B-NUM-SQ D197B-NUM-POSTO D197B-DT-COMPET D197B-V-OUTRAS D197B-V-ATRASSO D197B-HRR-PGTO

O agente *AgInterface*, conforme a sociedade de agentes [Figura 4.6], tem o papel de permitir ao usuário a seleção dos conceitos e termos a serem incorporados na *Ontologia* criada anteriormente [Figura 4.3]. A seleção ocorre nas seguintes situações:

- as entidades do tipo, tabela, ou arquivo seqüencial, podem configurar inteiramente um conceito, dependendo do seu conteúdo;
- os atributos, campos, podem fazer parte dos conceitos como termos respectivos; e
- os campos podem ser associados a outros conceitos existentes na *Ontologia*.

Nesta fase do método, ocorre então a incorporação semi-automática dos conceitos como, por exemplo, o valor devido ao INSS que, no programa XYZ510, tem várias referências, de modo que podem ser associados ao conceito **valor\_INSS** da *Ontologia* construída [Figura 4.7].

Tabela 4.4: Atributos específicos

Entidades	Atributos
Tabelas DB2	
MOV-ACM	MAC-V-INSS
MOV-PEN	MPE-V-INSS
XYZE199	D199B-V-INSS

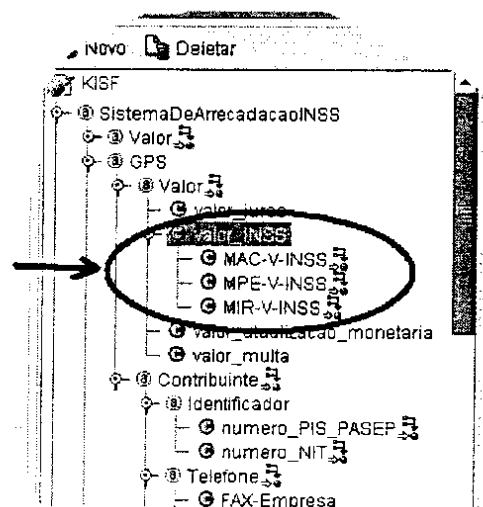
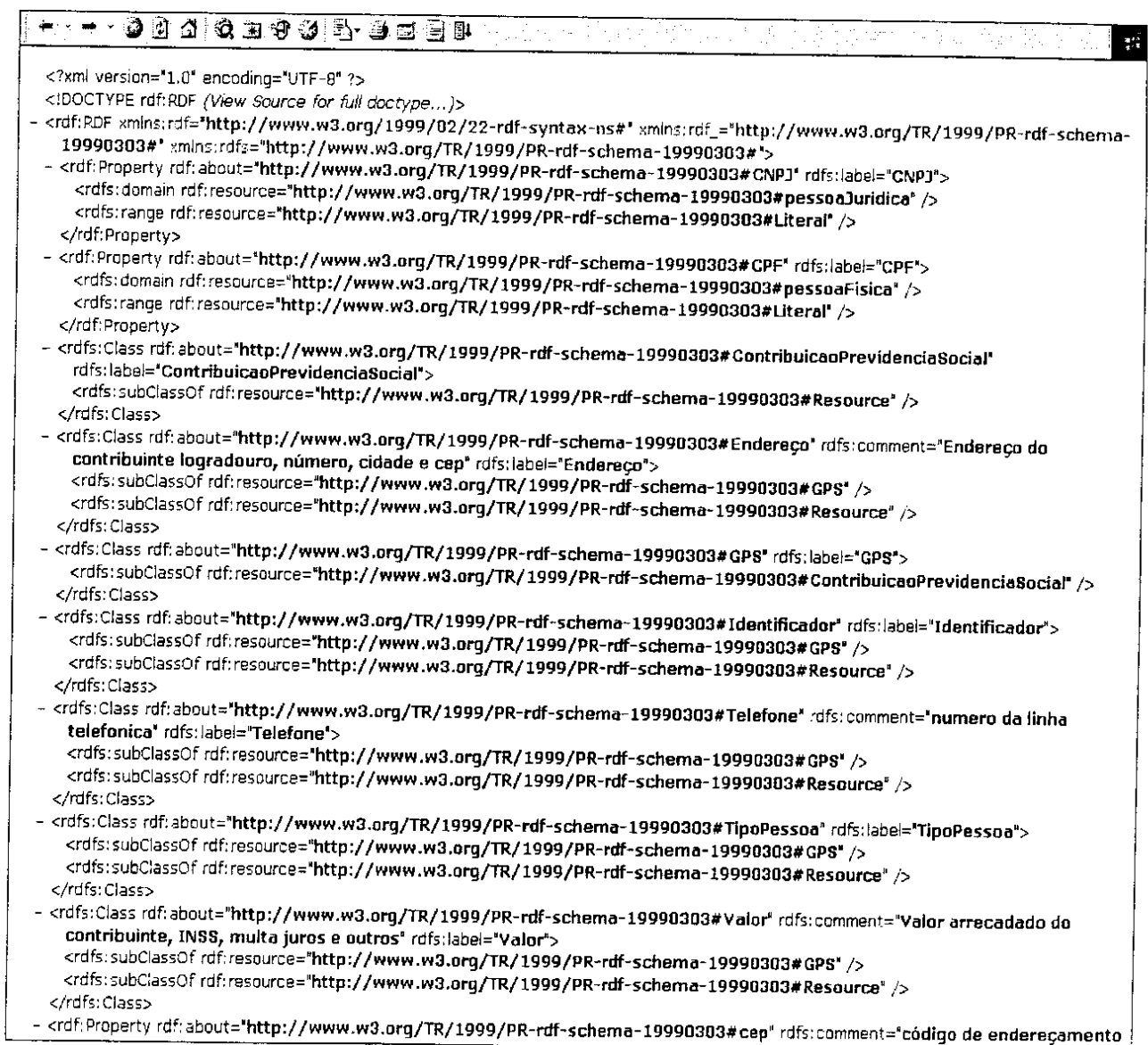


Figura 4.7: Associação dos conceitos

### 4.3. Compartilhamento do Conhecimento do SIL

Dentro das organizações, as *Ontologias* detêm uma importância estruturante em relação ao compartilhamento do conhecimento. Portanto, é fundamental que as ferramentas utilizadas permitam fazê-lo. No caso do uso do KISF, existe a opção da exportação no formato RDF. Esse formato permite compartilhar o conhecimento por meio de documentos gerados que proporcionam uma semântica da *Ontologia*. Conforme se verifica, a *Ontologia* foi exportada do Sistema XYZ para o formato RDF [Figura 4.8].



```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#" xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
- <rdf:Property rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#CNPJ" rdfs:label="CNPJ">
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#pessoaJuridica" />
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal" />
</rdf:Property>
- <rdf:Property rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#CPF" rdfs:label="CPF">
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#pessoaFisica" />
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal" />
</rdf:Property>
- <rdfs:Class rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#ContribuicaoPrevidenciaSocial"
  rdfs:label="ContribuicaoPrevidenciaSocial">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</rdfs:Class>
- <rdfs:Class rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Endereco" rdfs:comment="Endereço do
  contribuinte logradouro, número, cidade e cep" rdfs:label="Endereço">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#GPS" />
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</rdfs:Class>
- <rdfs:Class rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#GPS" rdfs:label="GPS">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#ContribuicaoPrevidenciaSocial" />
</rdfs:Class>
- <rdfs:Class rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Identificador" rdfs:label="Identificador">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#GPS" />
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</rdfs:Class>
- <rdfs:Class rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Telefone" rdfs:comment="numero da linha
  telefonica" rdfs:label="Telefone">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#GPS" />
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</rdfs:Class>
- <rdfs:Class rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#TipoPessoa" rdfs:label="TipoPessoa">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#GPS" />
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</rdfs:Class>
- <rdfs:Class rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Valor" rdfs:comment="Valor arrecadado do
  contribuinte, INSS, multa juros e outros" rdfs:label="Valor">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#GPS" />
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource" />
</rdfs:Class>
- <rdf:Property rdf:about="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#cep" rdfs:comment="código de endereçamento
```

Figura 4.8: Ontologia exportada para o RDF

Dentro das organizações, tecnologias como a *Intranet* funcionam como um facilitador para o compartilhamento das *Ontologias* dos SIL. A ferramenta *Ontolegacy* permite também a produção de páginas no formato HTML das entidades e atributos identificados durante a atuação do AgMAnalisador [Figura 4.9].

Importa ressaltar o outro produto resultante desse agente, relativo à transitividade dos dados. Este recurso permite identificar como os conteúdos dos atributos são repassados dentro do CF em questão, facilitando as suas manutenções.

OntoLegacy  
Transitividade dos Dados

Codigo Fonte : XYZ510.EXP

Entidades	Atributos
<b>ACUMULADORES</b>	
->AC-ACUM	->MSG-TOT-ACUM
->AC-E501	->MSG-TOT-E501
->AC-NUM-SQ	->D503B-NUM-SQ
->AC-PEND	->MSG-TOT-PEND
->AC-S502	->MSG-TOT-S502
->AC-S503	->MSG-TOT-S503
<b>AREAS-DE-SALVAMENTO</b>	
->AS-DT-PROCES	->D502A-DT-PROCES
->AS-DT-PROCES	->D503A-DT-PROCES
->AS-DT-PROCES	->MPE-DT-PROCES
->AS-DT-PROCES-ANT	->MAC-DT-PROCES
->AS-NUM-SIST	->CPR-NUM-SIST
->SRT-DT-PROCES	->D503B-DT-PROCES
->SRT-DT-PROCES	->D503B-DT-PROCES
<b>CTL-PRO-XYZ</b>	
->CPA-DT-PROCES	->AS-DT-PROCES
->CPA-DT-PROCES-ANT	->AS-DT-PROCES-ANT
->MPE-DT-PROCES	->SRT-DT-PROCES
->MAC-DT-PROCES	->SRT-DT-PROCES
->SRT-DT-PROCES	->D503B-DT-PROCES
->SRT-DT-PROCES	->D503B-DT-PROCES
<b>MOV-ACM</b>	
->MAC-CD-BARRAS	->SRT-CD-BARRAS
->MAC-CD-IDENT	->SRT-CD-IDENT
->MAC-CD-RCT	->SRT-CD-RCT
->MAC-DT-COMPET	->SRT-DT-COMPET
->MAC-DT-PGTO	->SRT-DT-PGTO
->MAC-DT-PROCES	->SRT-DT-PROCES
->MAC-NSU	->SRT-NSU
->MAC-NUM-AG	->SRT-NUM-AG
->MAC-NUM-POSTO	->SRT-NUM-POSTO
->MAC-NUM-TERM-CX	->SRT-NUM-TERM-CX
->MAC-ORIG-LANC	->SRT-ORIG-LANC
->MAC-TP-CAPT-DRB	->SRT-TP-CAPT-DRB
->MAC-TP-DOC	->SRT-TP-DOC
->MAC-TP-PGTO	->SRT-TP-PGTO
->MAC-V-ATRASSO	->SRT-V-ATRASSO
->MAC-V-INSS	->SRT-V-INSS
->MAC-V-OUTRAS	->SRT-V-OUTRAS
->MAC-V-PAGO	->SRT-V-PAGO
->XYZ800-CD-BARRA	->XYZ800-CD-IDENT
->XYZ800-CD-RCT	->XYZ800-DT-COMP
->XYZ800-DT-PGTO	->D503B-DT-PROCES
->D503B-NSU	->XYZ800-NUM-AG
->XYZ800-NUM-POSTO	->D503B-NUM-TERM-CX
->XYZ800-ID-ORIG	->XYZ800-TP-CAPT
->D503B-TP-DOC	->D503B-TP-PGTO
->XYZ800-V-ATRASSO	->XYZ800-V-INSS
->XYZ800-V-OUTRAS	->XYZ800-V-TOTAL
<b>MOV-IRG</b>	
->MIR-NUM-SQ	->AC-NUM-SQ
->D503B-NUM-SQ	

Figura 4.9: Páginas HTML Gerada pelo *Ontolegacy*

## 4.4. Considerações Finais

A partir do sistema XYZ, o experimento que versa sobre o processo de arrecadação das contribuições destinadas à Previdência Social do INSS, com o meio selecionado, códigos fontes em VS COBOL II SQL e CICS, possibilitou a obtenção de resultados e conclusões com a aplicação do método.

### 4.4.1. Ambiente do Experimento

O ambiente do experimento utilizado se caracterizou como um sistema de porte pequeno, para os padrões de um ambiente de *mainframe*, sendo 49 programas no total dos quais: 21 em COBOL *batch* e 28 em COBOL *on-line* CICS.

Para leitura dos CF da aplicação utilizada no experimento, adotaram-se alguns padrões aplicados no desenvolvimento do *software*, na organização de origem, tais como:

- a) sigla do sistema com três dígitos;
- b) determinação da finalidade do programa, código fonte em seu início; e
- c) adoção de prefixo, sigla do sistema em nomes de arquivos seqüenciais.

### 4.4.2. Resultados Obtidos

Seguindo-se os critérios pré-estabelecidos, a *Ontologia* desenvolvida permitiu maior clareza e entendimento do Sistema XYZ. A incorporação de novos conceitos e associações com a ferramenta *Ontolegacy*, baseada em agentes, repercutiu na melhora sensível da qualidade da *Ontologia*, tendo em vista alguns aspectos iniciais de sua criação pelo usuário. Nesse estágio inicial, há uma deficiência em abranger todos os detalhes pertinentes aos conceitos da *Ontologia*, reforçados com o uso de um SMA e, sobretudo, em associar o que foi teoricamente idealizado no âmbito conceitual, com o plano sensível traduzido pelos CF que realmente é o SIL.

O uso da ferramenta possibilitou identificar pelo menos 29 entidades comuns em todos os CF analisados, mas com a ajuda do engenheiro do conhecimento, no caso o analista de sistemas, esta quantidade decresceu para 6 por meio da sua intervenção [Figura 4.2]. Embora se associem as entidades como possíveis macroconceitos, neste experimento se verificou que

tal associação não foi muito forte em função da não-normalização<sup>7</sup> contida nas tabelas. Isso ocorre pela natureza da metodologia empregada na construção do SIL e também, em alguns casos, pela ausência de qualquer metodologia, verificada na prática em situações ocorridas no ambiente de trabalho.

Cabe aqui fazer um parêntese, trazendo um pouco da experiência prática, pois muito do que ocorre é consequência de dois fatores principais:

- a) o despreparo dos próprios técnicos, em função da formação acadêmica muitas vezes ser dissociada da área da Análise de Sistemas; e
- b) a escassez de tempo a que são submetidos os técnicos na condução do desenvolvimento de um novo SI ou sua manutenção.

É importante observar também que as entidades identificadas nos CF, e que seriam associadas a macroconceitos, podem e devem ser divididas em partes distintas. Por exemplo, conceitos do negócio e operacionais porque em todos os SIL tem-se estas partes distintas e claras e que não podem ser desconsideradas por traduzirem conhecimentos de suma importância para a compreensão dos SIL.

Outra característica observada no experimento foi a quantidade de termos similares gerados com a aplicação da ferramenta [Figura 4.9]. Analisando-se os resultados, vê-se que decorre da necessidade de se identificar os mesmos termos com suas diversidades de origens, os quais, por sua vez, são consequência:

- a) da natureza da linguagem COBOL e;
- b) da necessidade própria de se trabalhar com o termo, conforme a lógica do CF.

É importante enfatizar que o aspecto humano, sua participação na metodologia, tem um papel preponderante no contexto, pois ainda não nos é permitida a extração do conhecimento automático sem a participação do elemento humano, qualquer que seja o processo aplicado para esse fim [SHM01].

A proposta da metodologia junto com a ferramenta poderia ter um acréscimo de muitos outros recursos, sobretudo se trabalhados diretamente com os técnicos que estão no dia-a-dia, detectando novas necessidades e deficiências na busca do conhecimento dos SIL no qual estão inseridos.

---

<sup>7</sup> A normalização de dados é uma série de passos que se segue no desenho de uma base de dados, permitindo-lhes o armazenamento consistente e um eficiente acesso a eles, em uma base de dados relacional. Esses passos reduzem a redundância de dados e suas chances de se tornarem inconsistentes. Ou seja, a não-normalização seria o processo inverso.



# Capítulo 5

## Conclusão

A gestão do conhecimento é fator preponderante para a inovação contínua e a competitividade das organizações, as quais estão inseridas num ambiente constante de mudanças econômicas, sociais e tecnológicas, em âmbito global, de modo que a única certeza a ser verificada é a da incerteza nessas relações de negócios [NNT97] [TRR00].

Um dos princípios da organização baseada no conhecimento preconiza que o gargalo da produção, na visão do paradigma da era industrial, seria o capital financeiro e as habilidades humanas. Entretanto, no paradigma da era do conhecimento, sua base está no tempo e no conhecimento [SVB00] [TRR00].

Solução de problemas e manutenções de SIL, com a intencionalidade da inovação e competitividade, em geral demandam tempo e conhecimento, de maneira que a metodologia proposta – *Ontolegacy* – concorre para estes aspectos.

Sem a pretensão de se apresentar como uma solução pronta e acabada, reconhece-se que a metodologia e ferramenta *Ontolegacy* se depara com algumas deficiências, pela natureza do sistema analisado no experimento, quais sejam:

- a) apesar das metodologias disponíveis para o desenvolvimento de SI na organização, do estudo de caso, devido às exigências e dinâmicas com que os negócios acontecem no ambiente financeiro, há uma aplicação mínima destas metodologias formais. Tal situação configura dificuldades extremas quanto ao compartilhamento do conhecimento na organização, com corpo técnico que necessita assimilar o conhecimento dos SIL;
- b) a explicitação do conhecimento é bem reduzida no ambiente da organização do estudo de caso, o que se reflete não somente nessa organização como em outras de mesmo porte. Modelos de dados e algumas documentações eletrônicas são exemplos disponíveis do conhecimento explícito;

- c) recorrer ao conhecimento tácito pelo processo da socialização é prática muito comum. No estudo de caso, a facilidade e a assimilação do conhecimento ocorre num tempo menor, tempo este na maioria das vezes, escasso, embora muitas vezes esta prática não seja possível para alguns SIL porque o corpo técnico não faz mais parte do ambiente organizacional;
- d) a metodologia e a ferramenta propostas possuem um ponto crítico para ser incorporado ao ambiente dos SIL, na questão da criação da *Ontologia*, que deve ser conduzida pelo corpo técnico que detenha um conhecimento prévio do SIL a ser analisado e aliado à motivação para inserção das informações necessárias; e
- e) a extração automática do conhecimento em quaisquer que sejam os meios, manuais, códigos fonte, ferramentas CASE, documentação eletrônica e outros, ainda se torna impraticável devido ao estado da tecnologia em que nos encontramos. Dadas as características e particularidades do processo de aquisição do conhecimento, envolve não só o conhecimento em si, como também os fatores culturais, crenças e ambientes específicos.

## 5.1. Contribuição

A metodologia proposta e a ferramenta *Ontolegacy* contribuem de modo a propiciar à área de SI elementos que envolvem ampliar a obtenção do conhecimento explícito e tácito dos SIL, problema este que emerge não somente nas organizações financeiras bancárias, detentoras de uma tecnologia ultrapassada em algumas áreas específicas de TI, como em outras organizações, intensas usuárias de sistemas corporativos centralizados em *mainframe*.

Assim, espera-se que tenha sido possível demonstrar a ferramenta proposta como facilitador dos processos de mudanças e compartilhamento do conhecimento dos SIL, tônica principal desta dissertação. A diminuição do tempo e a otimização na aquisição do conhecimento têm força na disciplina da *Ontologia*, a considerar que o mapeamento e a localização de pontos específicos dentro dos SIL se apóia na solução de um SMA.

Portanto, o contributo da metodologia e da ferramenta proposto é dar suporte à gestão do conhecimento, sobretudo em aspectos que a norteiam, tais como: o aprendizado, o conhecimento e a inovação. Assim, vale destacar que é com o conhecimento que se faz o aprendizado e a inovação.

## 5.2. Estado da Arte

Atualmente, observa-se haver um direcionamento da aplicação das ontologias e SMA para a Semântica *Web*, de modo que se identifica um esforço concentrado para o desenvolvimento de ferramentas e métodos que propiciem e facilitem o uso e a geração de uma Semântica comum na *Web*, em função da aplicabilidade característica da *Ontologia*, que é o compartilhamento do conhecimento [MGP04].

Os sítios na *Web* que permitem uma linguagem comum e compartilhamento de informações permitem também a interoperabilidade com outros SI e usuários. De fato, há um interesse por parte das organizações em compartilhar informações e conhecimentos comuns relativos ao comércio eletrônico (B2B), recorrendo, assim, a linguagens de Ontologia voltadas para *Web*, como o WOL [W3C04]. Isso porque as informações e os conhecimentos ficam disponíveis na Internet e, portanto, são compartilhados nas relações entre as organizações.

Todo esse artefato se complementa com a utilização de SMA, para executar o processo de comunicação, bem como com a busca e a sensibilização do ambiente virtual no qual as organizações estão relacionadas.

Cabe ressaltar projetos como o EsperOnto [ESP04] (2001-2005), desenvolvido em parceria com universidades e empresas européias, com o intuito de desenvolver uma ponte entre o espaço da *Web* atual e a *Web* Semântica, fornecendo um serviço para “atualizar” conteúdos existentes ao conteúdo semântico da web. Sob tal perspectiva, a Ontologia assume papel preponderante nesse esforço, unificando e integrando conteúdos digitais, heterogêneos, internacionais e multilingual.

## 5.3. Trabalhos Futuros

Como proposta para trabalhos futuros, tem-se a recomendar a abertura para outros tipos de meios e, principalmente, CF de outras linguagens legadas. O ambiente de *mainframe* se caracteriza, sobretudo, pela quantidade de informações e dados em detrimento da qualidade. Conseqüentemente, os benefícios que proporcionarem uma maior automação dos processos, com a implementação de novos agentes computacionais, tornarão o dia-a-dia dos profissionais na área de SI mais ágil, eficaz e confortável, na solução dos problemas decorrentes dos processos de manutenção.

## Referências

- [ABE98] ANDRÉAS A.; BERNARDI, A.; HINKELMANN, K.; KUHN, O.; SINTEK, M. *Knowledge Management*, IEEE Intelligent Systems, 1998.
- [ABX03] ALMEIDA, M. B.; BAX, M. P. *Uma visão geral sobre Ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção*. n. 3, Ci. Inf., Brasília, v. 32, set./dez. 2003, p. 7-20.
- [AFH04] Site sobre *Anatomia e fisiologia humanas*, Vilela, A. M. Acessado em: 12/6/2004, [www.afh.bio.br/basicos/Sentidos1.htm](http://www.afh.bio.br/basicos/Sentidos1.htm).
- [ARL99] FERREIRA, A. B. H.. *O dicionário da língua portuguesa*. Rio de Janeiro, Ed. Nova Editora, 1999.
- [ARR00] ARRANGA, E. C.; ARCHBELL, I.; BRADLEY, J.; COKER, P.; LANGER, R.; TOWNSEND, C.; WHEATLEY, M.. *Roundtable - In COBOL's Defense*. IEEE Software 17(2), (2000).
- [AVL91] ÁVILA, B. C. *Representação do conhecimento utilizando frames*. Dissertação de Mestrado, Universidade de São Paulo, São Paulo, 1991.
- [BHG01] BECHHOFFER, S.; HORROCKS, I.; GOBLE, C., STEVENS R. *Oiled: a reasonable ontology editor for the semantic web*. In Working Notes of the 2001 Int. Description Logics Workshop (DL-2001), 2001. <http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/>.
- [BRD97] BRAND, M. G. J. van den; SELLINK, A.; VERHOEF, C.. *Obtaining a COBOL grammar from legacy code for reengineering purposes* in: ASF+SDF'97: Participants Proceedings of the 2nd International Workshop on the Theory and Practice of Algebraic Specifications (ed. M.P.A. Sellink), Electronic Workshops in Computing (series under auspices of the British Computer Society, series editor: K. van Rijsbergen), Springer-Verlag, 1997.
- [BRS97] BRADSHAW, J. *An introduction to software agents*", pages 3-46. In, 1997a.

- [DMM99] DOMINGUE, J. *Tadzebao and WebOnto: Discussing, browsing, and editing ontologies on the Web*. Proceedings of the 11th Banff Knowledge Acquisition Workshop, Banff, Alberta, Canada, April 18-23, 1998.
- [DRC93] DRUCKER, P. *Conhecimento empresarial – Como as organizações gerenciam o seu capital intelectual*. 7 ed. Rio de Janeiro, Ed. Campus, 1998.
- [DVP98] DAVENPORT, T. H.; PRUSAK, L. *Conhecimento empresarial – Como as organizações gerenciam o seu capital intelectual*. 7. ed., Rio de Janeiro, Ed. Campus, 1998.
- [ESP04] *Esperanto*. Sítio acessado em 15/7/2004, <http://www.esperanto.net>.
- [FFR96] FARQUHAR, A.; FIKES, R.; RICE, J. *The ontolingua server: a tool for collaborative ontology construction*. In the 10th Knowledge Acquisition for Knowledge-Based Systems (KAW'96), Canada, 1996.
- [FLM97] FININ, T., LABROU, Y., AND MAYFIELD, J. *KQML as an agent communication language*, page 291-316. In (Bradshaw, 1997 b).
- [FNM01] NOY, N. F.; SINTEK, M.; DECKER, S; CRUBÉZY, M.; FERGERSON, R. W.; MUSEN, M. A., *Creating semantic Web contents with Protégé-2000*, Stanford University computer.org/intelligent, IEEE, March / April 2001.
- [FQH97] FARQUHAR, A. *Ontolingua tutorial*, Knowledge Systems Lab Stanford University, 1997.
- [FRB99] FERBER, Jacques. *Multi-Agent Systems – an introduction to distributed artificial intelligence*. Addison Wesley Longman, 1999.
- [FRG97] FRANKLIN, S.; GRAESSER, A. *Is it an agent, or just a program? - Taxonomy for autonomous agents*. Pages 21-35. Third International Workshop on Agent Theories, Springer-Verlag, 1997.
- [FXG98] FOX, M.S.; GRUNINGER, M., (1998), *Enterprise modeling*. AI Magazine, AAAI Press, Fall 1998.
- [GRB93] GRUBER; T. R. *A translation approach to portable ontology specifications*. Knowledge Systems Laboratory, September 1992, Technical Report KSL 92-71 Revised April 1993 Manchester Business School, Stanford University, 1993.
- [GRB95] \_\_\_\_\_. *Towards principles for the design of ontologies used for knowledge sharing*. Int. J. Human-Computer Studies, v. 43, n. 5/6, 1995.

- [GPZ01] GÓMEZ-PÉREZ, A.; ARPÍREZ, J. C.; CORCHO, O.; FERNÁNDEZ-LÓPEZ, M. *WebODE: a workbench for ontological engineering*. First International Conference on Knowledge Capture (K-CAP'01). Victoria B. C. (Canada). 2001
- [GUA98] GUARINO, N. *Formal ontology and information systems*. Amended version of a paper appeared in N. Guarino (ed.), *Formal ontology in information systems*, proceedings of FOIS'98, Trento, Italy, 6-8 June 1998, Amsterdam, IOS Press, pp. 3-15.
- [HEN96] HENDLER, J. A. *Intelligent agents: where AI meets information technology*. IEEE Expert, pages 20-23, 1996.
- [HLS99] HOLLAND, C. P.; KELLY, S.; GIBSON, N.; LIGHT, B. *Focus issue on legacy information systems and business process change: a business perspective of legacy information systems*, Manchester Business School, University of Manchester, 1999.
- [HUN97] HUHNS, M. N.; SINGH, M. P. *Readings in Agents*. New York: Morgan Kaufmann, 1997.
- [INS04] Sítio do Instituto Nacional da Previdência Nacional, acessado em: 18/6/2004  
[http://www.inss.gov.br/03\\_01\\_01\\_03.asp](http://www.inss.gov.br/03_01_01_03.asp)
- [JEN98] JENNINGS, N.R.; SYCARA, K.; Wooldridge, M. *A Roadmap of Agent Research and Development*. Journal of Autonomous Agents and Multi-Agent Systems. Vol.1, No. 1, 1998, p. 276-306.
- [KNT04] KANT, I. *Crítica da Razão Pura*. Ed. Acrópolis, xxxx. Trad.: J. Rodrigues de Meringe, sítio acessado em: 20/5/2004.  
<http://www.ebooksbrasil.com/eLibris/critica.html>.
- [KNO97] KNOBLOCK, C. A.; AMBITE, J. L. *Agents for information gathering*. Pages 347-373. In (Bradshaw, 1997b), 1997.
- [LNG90] LENAT, D. B.; GUHA, R. V. *Building large knowledge-based systems*. Massachusetts: Addison-Wesley 372 p., 1990.
- [LPZ99] FERNÁNDEZ-LÓPEZ, M. et al. *Building a chemical ontology using methodology and the ontology design environment*. [S. l.] IEEE Intelligent systems & their applications, p. 37-46, Jan./Feb., 1999.
- [MAE94] MAES, P. *Agents that Reduce Work and Information Overload*. Communications of the ACM, 37(7): 31-40, 1994.
- [MAE96] MAES, P. *Modeling adaptive autonomous agents*. Artificial Life Journal, 1(1): 135-162, 1996.

- [MGP04] MCGUINNESS, D. L.; PINHEIRO DA SILVA, P. *Explaining Answers from the Semantic Web*. Technical Report, Knowledge Systems Laboratory, Stanford University., 2004.
- [MOU96] MOULIN, M.; CHALIB-DRAA, B. *An overview of distributed artificial intelligence*. Foundations of Distributed Artificial Intelligence, John Wiley & Sons, 1996.
- [MSS00] A. Maedche, H. P. Schnurr, S. Staab, and R. Studer. *Representation language-neutral modeling of ontologies*. In workshop: Modellierung 2000, St. Goar, Germany, 2000.
- [NNT97] NONAKA, I; TAKEUCHI, H. *Criação de conhecimento na empresa: como as empresas japonesas geram a dinâmica da inovação*, Rio de Janeiro, Ed. Campus, 1997.
- [NKK00] KROGH, GEORGE VON; ICHIJO, KAZUO; NONAKA, IKUJIRO. *Facilitando a criação de conhecimento: reinventando a empresa com o poder da inovação contínua*. Rio de Janeiro, Ed. Campus, 2000.
- [NYH97] NOY, NATALYA F.; HAFNER, C. *The state of the art in ontology design: a survey and comparative review by C*. In AI Magazine, 18 (3), p 53-74 (1997).
- [NYG00] NOY, N. F.; MCGUINNESS, D. L. *Ontology development 101: A guide to creating your first ontology*. Stanford University, Standford, 2000.
- [NFM00] NOY, N. F.; FERGERSON, R. W.; MUSEN, M. A. *The knowledge model of protégé-2000: combining inter-operability and flexibility*. In proceeding of the 2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France, 2000.
- [NWA96] NWANA, H. S. *Software agents: an overview*. *Knowledge Engineering Review*, 11(3): p. 205-244, 1996.
- [PAC03] PACHECO, E. J. *KISF - Knowledge Integration & Sharing Framework*. Dissertação de Mestrado em Informática Aplicada. Pontifícia Universidade Católica do Paraná, Curitiba, 2003.
- [PRR02] PROBST, G.; RAUB, S.; ROMHARDT, K. *Gestão do conhecimento: Os elementos construtivos do sucesso*. Porto Alegre, Ed. Bookman, 2002.
- [QNN90] QUINN, J. B. et al. *Technology in services: rethinking strategic focus*. Sloan Management Review, Winter, 1990, p. 79-87.

- [RZD98] REZENDE, S. O. *Aquisição de conhecimento explícito ou manual*, Notas do ICMC, São Carlos, 1998.
- [RSN95] RUSSEL, S.; NORVIG, P. *Artificial intelligence: a modern approach*. Prentice Hall, Englewoods Cliff, NJ, 1995.
- [SRL00] SEARLE, J. R. *Mente, linguagem e sociedade: filosofia no mundo real*. Rio de Janeiro, Ed. Rocco, 2000.
- [SHM99] SHMEIL, M. A. H. *Sistemas multiagentes na modelação da estrutura e relações de contratação de organizações*. Tese de Doutorado em Engenharia Electrotécnica e de Computadores. Faculdade de Engenharia, Universidade do Porto, Portugal, 1999.
- [SHM01] \_\_\_\_\_. *Notas de Aulas da disciplina Gestão do Conhecimento*. Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná. 2001.
- [STB01] STAAB, S. et al. *Knowledge processes and ontologies*. Intelligent systems. *IEEE*, v. 16, n. 1, p. 26-34, Jan./Feb. 2001.
- [STL01] STANO, R. C.; LEITE, V. *A teoria da gestão do conhecimento sob um olhar construtivista*. Anais do 4.º Simpósio Internacional de Gestão do Conhecimento/Gestão de documentos, Curitiba, 2001.
- [SUI02] SU, X.; ILEBREKKE, L. *A comparative study of ontology languages and tools*. In proceeding of Conference on Advanced Information System Engineering (CAiSE' 02). Toronto, Canada, 2002, Springer.
- [SVB00] SVEIBY, K. *What is knowledge management?* Sítio acessado em: 20/6/2004 <http://www.sveiby.com>.
- [SWT96] SWARTOUT, B. et al. *Toward distributed use of large-scale ontologies*. In: *Proceedings of AAAI97, Spring Symposium Series Workshop on Ontological Engineering*, 1997.[S. l.] : AAAI Press, 1997. p. 138-148.
- [SWA04] SOWA, J. *Ontology definition and scope*. Sítio acessado em : 17/6/2004 <http://www.jfsowa.com/ontology/index.htm>
- [TRR00] TERRA, J. C. C. *Gestão do conhecimento*. São Paulo, Ed. Negócio Editora, 2000.
- [TFL84] TOFFLER, A. *A terceira onda*. Rio de Janeiro, Ed. Nova Fronteira, 1984.
- [TXR00] TEIXEIRA FILHO, J. *Gerenciando conhecimento: como a empresa pode usar a memória organizacional e a inteligência competitiva no desenvolvimento de negócios*. Rio de Janeiro, Ed. Senac, 2000.



- [UCG96] USCHOLD, N.; GRUNINGER, M. *Ontologies: principles, methods and applications*, AIAI-TR-91, February, 1996.
- [USC98] USCHOLD, M. et al. *The Enterprise Ontology. The Knowledge Engineering Review*. Vol. 13, Special Issue on Putting Ontologies to Use, 1998, p. 31-89.
- [W3C04] *Web-Ontology (WebOnt) Working Group (Closed)*. Sítio acessado em 15/6/2004, <http://www.w3.org/2001/sw/WebOnt/>.
- [WDJ94] WOOLDRIDGE, M.; JENNINGS, N. R. *Intelligent agents: theory and practice*. Knowledge Engineering Review, October 1994, p. 115-152.
- [WDJ98] \_\_\_\_\_ *Pitfalls of agent-oriented development*. Pages 385-391. 2<sup>nd</sup> International Conference on Autonomous Agents (Agents' 98), 1998.

# Apêndice A

## Linguagens para desenvolvimento de Ontologias

Tabela A.1 – Linguagens para Ontologias

Linguagem	Características
<i>KIF</i> ( <i>Knowledge</i> <i>Interchange</i> <i>Format</i> )	Linguagem formal construída para trabalhar como meio de comunicação de conhecimento entre bases construídas usando diferentes linguagens. É basicamente uma notação prefixa para lógica de predicados de primeira ordem com termos funcionais e igualdade, sobre a qual várias Ontologias adicionais (de conjuntos, números, seqüências etc.) foram construídas.
<i>Ontolândia</i>	Linguagem formal e um sistema projetado para o propósito específico de expressar Ontologias. <i>Ontolândia</i> foi construída sobre a linguagem <i>KIF</i> , adicionando mecanismos para expressar classes, relações e hierarquias de classes.
<i>CycL</i> ( <i>Cyc</i> <i>Language</i> )	O <i>CycL</i> é uma linguagem formal cuja sintaxe deriva da lógica de predicado de primeira ordem, e foi utilizado inicialmente no projeto <i>Cyc</i> , que visa provar uma ontologia geral para o conhecimento do senso comum. O <i>Cycorp</i> criou e controla uma grande base de conhecimento comum e detecta o conhecimento criado com essa língua. Para expressar conceitos do mundo real, a linguagem tem um vocabulário dos termos (aproximadamente 160 deles) que podem ser combinados em expressões significativas do <i>CycL</i> . Alguns dos conceitos principais do <i>CycL</i> são: constantes, variáveis, fórmulas, predicados e microteorias.
<i>F-logic</i> ( <i>Frame Logic</i> )	Foi desenvolvida no final dos anos 80. É uma linguagem de lógica integrada com o paradigma da orientação a objetos e baseado em <i>Frames</i> . Alguns conceitos fundamentais das linguagens de orientação a objetos têm uma representação direta na <i>F-logic</i> , para a classe do exemplo, o método, os tipos e a herança, e outros aspectos secundários, como o polimorfismo, podem ser facilmente modelados também. Um dos problemas principais com a aproximação da orientação a objetos é a falta da semântica da lógica, superada aqui pela fundação lógica da <i>F-logic</i> . Há muitas similaridades entre a <i>F-logic</i> e a <i>Ontolândia</i> , visto que ambos tentam integrar frames na estrutura lógica.

Porém, os modelos primitivos baseados em *Frames* estão definidos explicitamente na semântica da *F-logic*, quando a *Ontolinguagem* o trata como termos de segunda ordem definidos com os axiomas do KIF.

*OCML* (Operational Conceptual Modeling Language) foi desenvolvido e é mantido pelo Knowledge Media Institute (KMI) no contexto do projeto VITAL. Sua finalidade preliminar é fornecer o conhecimento operacional que modela facilidades. Para conseguir isso, inclui interpretes para termos funcionais e do controle. *OCML* fornece mecanismos definindo relações, funções, classes, exemplos, réguas e procedimentos. Pode ser visto como extensão da "ontolinguagem operacional", fornecendo o teorema da prova, e mecanismos da avaliação da função construída para a Ontolinguagem. A natureza operacional de *OCML* possibilita suportar rapidamente prototipação, importante para validação do modelo. O *OCML* fornece uma série de ontologias básicas que dão forma a uma plataforma para construir outras ontologias: metafunções, relações, séries, números, listas, cordas, quadros, inferências, ambientes e métodos para tarefa. A funcionalidade das ontologias básicas é aproximadamente análoga às classes da linguagem Java.

*LOOM* O Loom é um sistema da representação e do raciocínio do conhecimento baseado na lógica da descrição. A Universidade do Instituto das Ciências de Informação de Califórnia do Sul (ISI) começou o desenvolvimento no final dos anos 80, sob o patrocínio da DARPA. O sistema foi mantido continuamente nos anos 90, e a versão fornecida mais atual é a 4. Uma distinta característica da lógica da descrição é que as classes (conceitos) podem ser definidas nos termos das descrições que especificam as propriedades ou as limitações, a que os objetos devem satisfazer a fim de pertencer ao conceito. Isso significa que as relações da sociedade da classe podem ser determinadas pela inferência. Uma das tarefas preliminares da lógica da descrição baseou o sistema para computar relações de subassuntos entre descrições, organizando-as em taxonomias. Para conseguir a derivação automática das taxonomias, o tear oferece uma língua para a descrição dos objetos e das relações e uma linguagem de afirmação para especificar confinamentos nos conceitos e nas relações. O tear fornece o raciocínio dedutivo poderoso com a produção subjacente e as potencialidades das classificações baseadas em inferência.

*Telos* *Telos* foi construído para suportar diversos modelos e as idéias da representação do conhecimento, das bases de dados dedutivas e das línguas da exigência para construir sistemas de informação. Basicamente é uma linguagem de representação do conhecimento com foco orientado a objetos. Um objeto é um indivíduo (entidades, conceitos) ou um atributo (relacionamentos, *attributelinks*). *Telos* tem poder significativamente elevado, sobretudo porque suas entidades e atributos são tratados uniformemente. As regras da integridade, dedutivas e do confinamento podem ser especificadas com uma sublinguagem de afirmação, e as construções para indicações temporais também estão presentes. Essa linguagem não foi mantida na última década, conseqüentemente, não está apropriada à interoperabilidade e à distribuição.

Tabela A.2 – Linguagens baseadas em WEB

Linguagem	Descrição
<i>OIL</i> ( <i>Ontology</i> <i>Inference</i> <i>Layer</i> )	Foi desenvolvido no projeto On-To-Knowledge, sendo uma representação e linguagem de troca para Ontologias. A linguagem é uma combinação das linguagens primitivas baseadas em frames e os serviços formais da semântica e do raciocínio lógico da descrição. Para permitir o uso do OIL na <i>Web</i> , é baseada nos padrões W3C, XML e RDF(S). A descrição da Ontologia é dividida em três camadas: nível de objeto (exemplos concretos), primeiro meta-nível (definições ontológicas) e segundo meta-nível (descreve características da Ontologia). O OIL fornece definições para classes e relações, e uma série limitada de axiomas. Os <i>Slots</i> são tratados como cidadãos de primeira classe, e podem ser representados nas hierarquias. Há diversas limitações relacionadas aos axiomas, e também limites significativos decorrentes de sua linguagem. O OIL tem uma semântica formal precisa, necessária para a sustentação efetiva do raciocínio.
<i>DAML+OIL</i>	<i>DAML+OIL</i> é uma linguagem semântica de marcação com recursos da <i>Web</i> , e é um padrão proposto de W3C para a representação ontológica e do metadado. <i>DAML</i> (agente da DARPA que modela a linguagem) foi transformado a <i>DAML+OIL</i> incluindo alguns aspectos do OIL na língua. <i>DAML+OIL</i> foi construído no esquema do RDF, mas fornece modelos primitivos mais ricos, encontrados geralmente em lógica de descrição. A maioria das idéias baseadas em frames fornecidos no OIL foi removida, e as afirmações são feitas nos termos de uma limitada série de axiomas. O resultado é uma linguagem que trabalhe melhor como uma plataforma para Ontologias do que RDF e XTM, mas tem limitações como linguagem para o desenvolvimento das Ontologias, por causa da remoção das construções baseadas em frames.
<i>XOL (XML-</i> <i>Based</i> <i>Ontology</i> <i>Exchange</i> <i>Language)</i>	<i>XOL</i> foi criado originalmente para trocar Ontologias da biologia molecular. Fornece uma definição geral que o fazem apropriado para a troca de outras Ontologias. As primitivas e o modelo semântico são baseados em OKBC-Lite (um formulário simplificado do modelo do conhecimento do OKBC). Em <i>XOL</i> , os <i>Slots</i> como extensão, são tratados como entidades de segunda classe, resultando na falta de sustentação para hierarquias, possui fraca especificação dos relacionamentos.

# Apêndice B

## Informações da GPS

No capítulo 4 foi eleito o sistema XYZ como estudo de caso que versa sobre a Arrecadação da Previdência Social. Este apêndice apresenta as informações sobre o documento gerador, GPS [INS04], de todo o processo do SIL XYZ.

### Trabalhador com Previdência

#### Guia da Previdência Social - GPS

A Guia da Previdência Social (GPS) é o documento hábil para o recolhimento das contribuições sociais dos contribuintes individuais da Previdência Social.

Trata-se de documento simplificado instituído pela Resolução INSS/PR nº 657 de 17/12/98 utilizável obrigatoriamente desde 23/07/99.

- [Veja e imprima o Modelo da GPS](#)

#### Informações básicas:

- **CAMPO 1 - Nome do contribuinte, Fone e Endereço**

Dados para identificação do contribuinte.

- **CAMPO 3 - Código de pagamento**

Relação de Códigos de Pagamento

**Código**

**Descrição**

- 1007** Contribuinte Individual - Recolhimento Mensal - NIT/PIS/PASEP
- 1104** Contribuinte Individual - Recolhimento Trimestral NIT/PIS/PASEP
- 1120** Contribuinte Individual - Recolhimento Mensal - Com dedução de 45 % (Lei nº 9.876/99) - NIT/PIS/PASEP
- 1147** Contribuinte Individual - Recolhimento Trimestral - Com dedução de 45 % (Lei nº 9.876/99) - NIT/PIS/PASEP
- 1201** GRC Contribuinte Individual - DEBCAD (Preenchimento exclusivo pelo INSS)
- 1406** Segurado Facultativo – Recolhimento Mensal - NIT/PIS/PASEP
- 1457** Segurado Facultativo – Recolhimento Trimestral - NIT/PIS/PASEP
- 1197** Segurado Especial Recolhimento Mensal NIT/PIS/PASEP
- 1554** Segurado Especial Recolhimento Trimestral - NIT/PIS/PASEP
- 1600** Empregado Doméstico - Recolhimento Mensal - NIT/PIS/PASEP
- 1651** Empregado Doméstico - Recolhimento Trimestral - NIT/PIS/PASEP
- 1708** Ação Trabalhista - NIT/PIS/PASEP

- **CAMPO 4 - Competência**

Informação no formato MM/AAAA da competência objeto do recolhimento.

- **CAMPO 5 - Identificador**

Número do NIT ou PIS/PASEP do contribuinte.

- **CAMPO 6 - Valor do INSS**

- Valor devido ao INSS pelo contribuinte, já considerados:
- os valores de eventuais compensações; e

- **CAMPO 9 - Valor de Outras Entidades**

Não preencher.

- **CAMPO 10 - Atualização Monetária, Multa e Juros**

Valor devido a título de atualização monetária e acréscimos legais, quando for o caso, sobre recolhimentos em atraso.

Ver acréscimos legais.

- **CAMPO 11 - Total**

- Valor total a recolher ao INSS.

## **Preenchimento**

A GPS deve ser preenchida em duas vias com a seguinte destinação:

- A primeira via, destinada à guarda e comprovação do recolhimento junto ao INSS;
- A segunda via, destinada ao controle do agente arrecadador.

**Observação:** Para comprovar o exercício de atividade remunerada, com vistas à concessão de benefícios, será exigido do contribuinte individual, a qualquer tempo, o recolhimento das correspondentes contribuições.

### **Prazos**

Os prazos para recolhimento das contribuições previdenciárias em GPS são:

**No dia 15 do mês seguinte** àquele a que as contribuições se referirem, prorrogando-se o vencimento para o dia útil subsequente, quando não houver expediente bancário, **para os contribuintes individuais, facultativos e domésticos;**

**Até o dia 20 de dezembro**, antecipando-se o vencimento para o dia útil imediatamente anterior, quando não houver expediente bancário, **para as contribuições incidentes sobre o 13º salário**, para domésticos.

### **GPS – Valor inferior a R\$ 29,00**

A Resolução INSS/DC nº 39 de 23/11/00 determinou o valor mínimo de R\$ 29,00 (vinte e nove reais) para recolhimento de contribuições previdenciárias junto à rede arrecadadora, à partir de 1º de dezembro de 2000.

O contribuinte que eventualmente possuir recolhimento inferior a R\$ 29,00 deverá acumular este valor com os próximos futuros até que a soma atinja este mínimo, para então proceder ao recolhimento, utilizando a última competência como base de informação no campo 4 da GPS.

### **GPS - Trimestral**

Os contribuintes individuais e facultativos que estiverem contribuindo sobre o valor mínimo de contribuição (hoje 20% x R\$ 260,00 = R\$ 52,00), poderão optar pelo recolhimento trimestral.

O contribuinte poderá efetuar o recolhimento, agrupando os valores das competências por trimestre civil, ou seja:

- Janeiro, fevereiro e março;
- Abril, maio e junho;
- Julho, agosto e setembro; e
- Outubro, novembro e dezembro.

**Observações:** Para o recolhimento trimestral, o contribuinte deverá utilizar código de pagamento específico, conforme o caso:

<b>Código</b>	<b>Descrição</b>
<b>1104</b>	Contribuinte Individual - Recolhimento Trimestral NIT/PIS/PASEP
<b>1147</b>	Contribuinte Individual - Recolhimento Trimestral - Com dedução de 45 % (Lei nº 9.876/99) - NIT/PIS/PASEP
<b>1457</b>	Segurado Facultativo – Recolhimento trimestral – NIT/PIS/PASEP
<b>1554</b>	Segurado Especial – Recolhimento trimestral – NIT/PIS/PASEP
<b>1651</b>	Empregado Doméstico – Recolhimento trimestral – NIT/PIS/PASEP

O vencimento será no dia 15 do mês seguinte ao de cada trimestre civil, prorrogando-se para o dia útil subsequente, quando não houver expediente bancário.

No caso desta opção (trimestralidade), nas GPS's serão consignadas as competências março, junho, setembro e dezembro, mesmo que a inscrição do segurado tenha ocorrido no segundo ou terceiro mês do trimestre civil.

Aplica-se ao empregador doméstico, relativamente aos empregados domésticos a seu serviço, cujos salários-de-contribuição sejam iguais ao valor do salário-mínimo, ou inferiores, nos casos de admissão, dispensa ou fração do salário em razão de gozo de benefício, o mesmo dispositivo da trimestralidade facultada aos contribuintes individuais e facultativos, exceto no que concerne ao recolhimento sobre remuneração de 13º salário, que segue a regra geral.

### **GPS Eletrônica para contribuinte individual**

O recolhimento da contribuição individual poderá ser efetuado por intermédio da GPS Eletrônica, através de débito em conta, comandado por meio da rede Internet ou por aplicativos eletrônicos disponibilizados pelos bancos. O próprio contribuinte fará a digitação dos campos obrigatórios, sendo gerado comprovante de recolhimento com layout estabelecido pelos bancos, que conterá as seguintes informações:

- Campo 3 - Código de pagamento
- Campo 4 - Competência
- Campo 5 - Identificador
- Campo 6 - Valor do INSS
- Campo 7 - Valor de outras Entidades
- Campo 10 - At. Mon/Multa e Juros
- Campo 11 - total
- Campo 12 - Autenticação bancária



# Apêndice C

## CF do Sistema de Informação XYZ

No Capítulo 4, foi apresentado o resultado da aplicação da ferramenta *Ontolegacy*, no Sistema de Informação XYZ no domínio Arrecadação. Este apêndice apresenta as entidades e atributos e o CF do qual foi derivado, um dos utilizados no experimento.

### C.1 - Entidade e atributos do CF XYZ510

Entidades Tabelas DB2	Atributos	Entidades Arquivos Seqüenciais	Atributos
MOV-ACM (Tabela DB2)	MAC-CD-BARRAS MAC-CD-IDENT MAC-CD-RCT MAC-DT-COMPET MAC-DT-PGTO MAC-DT-PROCES MAC-NSU MAC-NUM-AG MAC-NUM-POSTO MAC-NUM-TERM-CX MAC-ORIG-LANC MAC-TP-CAPT-DAD MAC-TP-DOC MAC-TP-PGTO MAC-V-ATRASSO MAC-V-INSS MAC-V-OUTRAS MAC-V-PAGO	XYZE199	D199A-REG D199A-TIPO-REG D199A-DT-PROCES D199B-REG D199B-TIPO-REG D199B-DT-PROCES D199B-NUM-AG D199B-NUM-POSTO D199B-CD-RCT D199B-CD-IDENT D199B-DT-COMPET D199B-V-INSS D199B-V-OUTRAS D199B-V-ATRASSO D199B-V-PAGO D199B-CD-BARRAS D199B-DT-PGTO D199B-TP-CAPT-DAD D199B-ORIG-LANC D199B-TP-PGTO D199B-TP-DOC D199B-NUM-TERM-CX D199B-NSU D199B-AG-DEB D199B-CTA-DEB D199B-DV-CTA-DEB D199B-TP-CTA-DEB D199B-NUM-CARTAO D199B-TP-CARTAO

MOV-IRG

MIR-NUM-SQ  
MIR-DT-PROCES  
MIR-NUM-AG  
MIR-NUM-POSTO  
MIR-DT-REPAS-  
DOC  
MIR-CD-RCT  
MIR-TP-IDENT  
MIR-CD-IDENT  
MIR-DT-COMPET  
MIR-V-INSS  
MIR-V-OUTRAS  
MIR-V-ATRASSO  
MIR-V-PAGO  
MIR-CD-BARRAS  
MIR-DT-PGTO  
MIR-TP-CAPT-DAD  
MIR-ORIG-LANC  
MIR-TP-PGTO  
MIR-TP-DOC  
MIR-NUM-TERM-CX  
MIR-NSU  
MIR-SIT-DOC  
MIR-DCC-IRREG  
MIR-ID-CTB

XYZS198

D199B-VIA-CARTAO  
D199B-TITULARIDADE  
D199B-DT-PGTO-EFET  
D199B-HRR-PGTO  
D198A-REG  
D198A-TIPO-REG  
D198A-DT-PROCES  
D198B-REG  
D198B-TIPO-REG  
D198B-DT-PROCES  
D198B-NUM-AG  
D198B-NUM-POSTO  
D198B-CD-RCT  
D198B-TP-IDENT  
D198B-CD-IDENT  
D198B-DT-COMPET  
D198B-V-INSS  
D198B-V-OUTRAS  
D198B-V-ATRASSO  
D198B-V-PAGO  
D198B-CD-BARRAS  
D198B-DT-PGTO  
D198B-TP-CAPT-DAD  
D198B-ORIG-LANC  
D198B-TP-PGTO  
D198B-TP-DOC  
D198B-NUM-TERM-CX  
D198B-NSU  
D198B-AG-DEB  
D198B-CTA-DEB  
D198B-DV-CTA-DEB  
D198B-TP-CTA-DEB  
D198B-NUM-CARTAO  
D198B-TP-CARTAO  
D198B-VIA-CARTAO  
D198B-TITULARIDADE  
D198B-DT-PGTO-EFET  
D198B-HRR-PGTO

CTL-PRO-XYZ  
(Tabela DB2)

CPA-DT-PROCES  
CPA-DT-PROCES-  
ANT

XYZS197

D197A-REG  
D197A-TIPO-REG  
D197A-DT-PROCES  
D197B-REG  
D197B-TIPO-REG  
D197B-DT-PROCES  
D197B-NUM-AG  
D197B-NUM-SQ  
D197B-NUM-POSTO  
D197B-CD-RCT  
D197B-TP-IDENT  
D197B-CD-IDENT  
D197B-DT-COMPET  
D197B-V-INSS  
D197B-V-OUTRAS  
D197B-V-ATRASSO  
D197B-V-PAGO  
D197B-CD-BARRAS  
D197B-DT-PGTO  
D197B-TP-CAPT-DAD

D197B-ORIG-LANC  
D197B-TP-PGTO  
D197B-TP-DOC  
D197B-NUM-TERM-CX  
D197B-NSU  
D197B-SIT-DOC  
D197B-DCC-IRREG  
D197B-AG-DEB  
D197B-CTA-DEB  
D197B-DV-CTA-DEB  
D197B-TP-CTA-DEB  
D197B-NUM-CARTAO  
D197B-TP-CARTAO  
D197B-VIA-CARTAO  
D197B-TITULARIDADE  
D197B-DT-PGTO-EFET  
D197B-HRR-PGTO

## C.2 - Código Fonte XYZ510

```

IDENTIFICATION DIVISION.
PROGRAM-ID.                                XYZ510.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES.
    DECIMAL-POINT                           IS COMMA.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT  XYZE199
           ASSIGN  XYZE199
           STATUS  AS-STATUS.
    SELECT  XYZS198
           ASSIGN  XYZS198
           STATUS  AS-STATUS.
    SELECT  XYZS197
           ASSIGN  XYZS197
           STATUS  AS-STATUS.
    SELECT  SORTWK01
           ASSIGN  SORTWK01.
DATA DIVISION.
FILE SECTION.
FD  XYZE199
   BLOCK          0
   RECORDING      F
   RECORD         200
   LABEL RECORD IS STANDARD.
   COPY 'XYZD199A'.
01  D199A-REG.
   05  D199A-TIPO-REG      PIC  9(001).
   05  D199A-DT-PROCES    PIC  9(008).
   05  FILLER              PIC  X(191).
   COPY 'XYZD199B'.
01  D199B-REG.
   05  D199B-TIPO-REG      PIC  9(001).
   05  D199B-DT-PROCES    PIC  9(008).
   05  D199B-NUM-AG       PIC  9(005)          COMP-3.
   05  D199B-NUM-POSTO    PIC  9(003)          COMP-3.
   05  D199B-CD-RCT       PIC  9(005)          COMP-3.
   05  D199B-CD-IDENT     PIC  9(015)          COMP-3.
   05  D199B-DT-COMPET    PIC  9(006).
   05  D199B-V-INSS       PIC  9(013)V99        COMP-3.
   05  D199B-V-OUTRAS     PIC  9(013)V99        COMP-3.
   05  D199B-V-ATRASSO    PIC  9(013)V99        COMP-3.
   05  D199B-V-PAGO       PIC  9(013)V99        COMP-3.
   05  D199B-CD-BARRAS    PIC  X(044).
   05  D199B-DT-PGTO      PIC  9(008).
   05  D199B-TP-CAPT-DAD  PIC  9(002).
   05  D199B-ORIG-LANC    PIC  9(002).
   05  D199B-TP-PGTO      PIC  9(002).
   05  D199B-TP-DOC       PIC  9(001).
   05  D199B-NUM-TERM-CX  PIC  9(015)          COMP-3.
   05  D199B-NSU          PIC  9(007)          COMP-3.
   05  D199B-AG-DEB       PIC  9(005)          COMP-3.
   05  D199B-CTA-DEB     PIC  9(007)          COMP-3.
   05  D199B-DV-CTA-DEB  PIC  9(001).

```

05	D199B-TP-CTA-DEB	PIC	9(001).	
05	D199B-NUM-CARTAO	PIC	9(017)	COMP-3.
05	D199B-TP-CARTAO	PIC	9(001).	
05	D199B-VIA-CARTAO	PIC	9(002).	
05	D199B-TITULARIDADE	PIC	9(002).	
05	D199B-DT-PGTO-EFET	PIC	9(008).	
05	D199B-HRR-PGTO	PIC	9(006).	
05	FILLER	PIC	X(029).	
FD	XYZS198			
	BLOCK		0	
	RECORDING		F	
	RECORD		200	
	LABEL RECORD IS		STANDARD.	
	COPY 'XYZD198A'.			
01	D198A-REG.			
05	D198A-TIPO-REG	PIC	9(001).	
05	D198A-DT-PROCES	PIC	9(008).	
05	FILLER	PIC	X(191).	
	COPY 'XYZD198B'.			
01	D198B-REG.			
05	D198B-TIPO-REG	PIC	9(001).	
05	D198B-DT-PROCES	PIC	9(008).	
05	D198B-NUM-AG	PIC	9(005)	COMP-3.
05	D198B-NUM-POSTO	PIC	9(003)	COMP-3.
05	D198B-CD-RCT	PIC	9(005)	COMP-3.
05	D198B-TP-IDENT	PIC	9(002).	
05	D198B-CD-IDENT	PIC	9(015)	COMP-3.
05	D198B-DT-COMPET	PIC	9(006).	
05	D198B-V-INSS	PIC	9(013)V99	COMP-3.
05	D198B-V-OUTRAS	PIC	9(013)V99	COMP-3.
05	D198B-V-ATRASSO	PIC	9(013)V99	COMP-3.
05	D198B-V-PAGO	PIC	9(013)V99	COMP-3.
05	D198B-CD-BARRAS	PIC	X(044).	
05	D198B-DT-PGTO	PIC	X(008).	
05	D198B-TP-CAPT-DAD	PIC	9(002).	
05	D198B-ORIG-LANC	PIC	9(002).	
05	D198B-TP-PGTO	PIC	9(002).	
05	D198B-TP-DOC	PIC	9(001).	
05	D198B-NUM-TERM-CX	PIC	9(005)	COMP-3.
05	D198B-NSU	PIC	9(007)	COMP-3.
05	D198B-AG-DEB	PIC	9(005)	COMP-3.
05	D198B-CTA-DEB	PIC	9(007)	COMP-3.
05	D198B-DV-CTA-DEB	PIC	9(001).	
05	D198B-TP-CTA-DEB	PIC	9(001).	
05	D198B-NUM-CARTAO	PIC	9(017)	COMP-3.
05	D198B-TP-CARTAO	PIC	9(001).	
05	D198B-VIA-CARTAO	PIC	9(002).	
05	D198B-TITULARIDADE	PIC	9(002).	
05	D198B-DT-PGTO-EFET	PIC	9(008).	
05	D198B-HRR-PGTO	PIC	9(006).	
05	FILLER	PIC	X(032).	
FD	XYZS197			
	BLOCK		0	
	RECORDING		F	
	RECORD		200	
	LABEL RECORD IS		STANDARD.	
	COPY 'XYZD197A'.			
01	D197A-REG.			
05	D197A-TIPO-REG	PIC	9(001).	

	05	D197A-DT-PROCES	PIC	9(008).	
	05	FILLER	PIC	X(191).	
		COPY 'XYZD197B'.			
01		D197B-REG.			
	05	D197B-TIPO-REG	PIC	9(001).	
	05	D197B-DT-PROCES	PIC	9(008).	
	05	D197B-NUM-AG	PIC	9(005)	COMP-3.
	05	D197B-NUM-SQ	PIC	9(007)	COMP-3.
	05	D197B-NUM-POSTO	PIC	9(003)	COMP-3.
	05	D197B-CD-RCT	PIC	9(005)	COMP-3.
	05	D197B-TP-IDENT	PIC	9(002).	
	05	D197B-CD-IDENT	PIC	9(015)	COMP-3.
	05	D197B-DT-COMPET	PIC	9(006).	
	05	D197B-V-INSS	PIC	9(013)V99	COMP-3.
	05	D197B-V-OUTRAS	PIC	9(013)V99	COMP-3.
	05	D197B-V-ATRASSO	PIC	9(013)V99	COMP-3.
	05	D197B-V-PAGO	PIC	9(013)V99	COMP-3.
	05	D197B-CD-BARRAS	PIC	X(044).	
	05	D197B-DT-PGTO	PIC	X(008).	
	05	D197B-TP-CAPT-DAD	PIC	9(002).	
	05	D197B-ORIG-LANC	PIC	9(002).	
	05	D197B-TP-PGTO	PIC	9(002).	
	05	D197B-TP-DOC	PIC	9(001).	
	05	D197B-NUM-TERM-CX	PIC	9(005)	COMP-3.
	05	D197B-NSU	PIC	9(007)	COMP-3.
	05	D197B-SIT-DOC	PIC	9(001).	
	05	D197B-DCC-IRREG	PIC	X(030).	
	05	D197B-AG-DEB	PIC	9(005)	COMP-3.
	05	D197B-CTA-DEB	PIC	9(007)	COMP-3.
	05	D197B-DV-CTA-DEB	PIC	9(001).	
	05	D197B-TP-CTA-DEB	PIC	9(001).	
	05	D197B-NUM-CARTAO	PIC	9(017)	COMP-3.
	05	D197B-TP-CARTAO	PIC	9(001).	
	05	D197B-VIA-CARTAO	PIC	9(002).	
	05	D197B-TITULARIDADE	PIC	9(002).	
	05	D197B-DT-PGTO-EFET	PIC	9(008)	COMP-3.
	05	D197B-HRR-PGTO	PIC	9(006)	COMP-3.
	05	FILLER	PIC	X(002).	
SD		SORTWK01.			
01		RG-SORTWK01.			
	05	SRT-CHAVE.			
	10	SRT-CHAVE-PRINCIPAL.			
	15	SRT-NUM-AG	PIC	9(005).	
	15	SRT-NUM-POSTO	PIC	9(003).	
	15	SRT-CD-RCT	PIC	9(005).	
	15	SRT-CD-IDENT	PIC	9(015).	
	15	SRT-DT-COMPET	PIC	9(006).	
	15	SRT-NSU	PIC	9(007).	
	15	SRT-V-PAGO	PIC	9(013)V99.	
	10	SRT-ORIGEM	PIC	9(001).	
	05	SRT-DT-PGTO	PIC	9(008).	
	05	SRT-DT-PROCES	PIC	9(008).	
	05	SRT-V-INSS	PIC	9(013)V99.	
	05	SRT-V-OUTRAS	PIC	9(013)V99.	
	05	SRT-V-ATRASSO	PIC	9(013)V99.	
	05	SRT-CD-BARRAS	PIC	X(044).	
	05	SRT-TP-CAPT-DAD	PIC	9(002).	
	05	SRT-ORIG-LANC	PIC	9(002).	
	05	SRT-TP-PGTO	PIC	9(002).	

05	SRT-TP-DOC	PIC	9(001).	
05	SRT-NUM-TERM-CX	PIC	X(015).	
05	SRT-DT-PGTO-EFET	PIC	9(008).	
05	SRT-AG-DEB	PIC	9(005).	
05	SRT-CTA-DEB	PIC	9(007).	
05	SRT-DV-CTA-DEB	PIC	9(001).	
05	SRT-TP-CTA-DEB	PIC	9(001).	
05	SRT-HRR-PGTO	PIC	9(007).	
05	SRT-TITULARIDADE	PIC	9(001).	
05	SRT-VIA-CARTAO	PIC	9(001).	
05	SRT-TP-CARTAO	PIC	9(001).	
05	SRT-NUM-CARTAO	PIC	9(017).	
WORKING-STORAGE SECTION.				
01	FILLER	PIC	X(032)	VALUE
	'III WORKING-STORAGE SECTION III'.			
01	DCLCTL-PRO-XYZ.			
10	CPA-NUM-SIST	PIC	S9(3)V USAGE	COMP-3.
10	CPA-DT-PROCES	PIC	S9(8)V USAGE	COMP-3.
10	CPA-DT-PROCES-ANT	PIC	S9(8)V USAGE	COMP-3.
10	CPA-NUM-REMES-GPS	PIC	S9(7)V USAGE	COMP-3.
01	DCLMOV-PEN.			
10	MPE-DT-PROCES	PIC	S9(8)V USAGE	COMP-3.
10	MPE-NUM-SQ	PIC	S9(7)V USAGE	COMP-3.
10	MPE-NUM-AG	PIC	S9(5)V USAGE	COMP-3.
10	MPE-NUM-POSTO	PIC	S9(3)V USAGE	COMP-3.
10	MPE-CD-RCT	PIC	S9(5)V USAGE	COMP-3.
10	MPE-TP-IDENT	PIC	S9(2)V USAGE	COMP-3.
10	MPE-CD-IDENT	PIC	S9(15)V USAGE	COMP-3.
10	MPE-DT-COMPET	PIC	S9(6)V USAGE	COMP-3.
10	MPE-V-INSS	PIC	S9(13)V9(2) USAGE	COMP-3.
10	MPE-V-OUTRAS	PIC	S9(13)V9(2) USAGE	COMP-3.
10	MPE-V-ATRASSO	PIC	S9(13)V9(2) USAGE	COMP-3.
10	MPE-V-PAGO	PIC	S9(13)V9(2) USAGE	COMP-3.
10	MPE-CD-BARRAS	PIC	X(44).	
10	MPE-DT-PGTO	PIC	S9(8)V USAGE	COMP-3.
10	MPE-DT-INCL	PIC	S9(8)V USAGE	COMP-3.
10	MPE-HRR-INCL	PIC	S9(6)V USAGE	COMP-3.
10	MPE-ID-USU	PIC	X(3).	
10	MPE-TP-CAPT-DAD	PIC	S9(2)V USAGE	COMP-3.
10	MPE-TP-PGTO	PIC	S9(2)V USAGE	COMP-3.
10	MPE-TP-DOC	PIC	S9(1)V USAGE	COMP-3.
10	MPE-NUM-TERM-CX	PIC	X(15).	
10	MPE-NSU	PIC	S9(7)V USAGE	COMP-3.
01	DCLMOV-IRG.			
10	MIR-NUM-SQ	PIC	S9(7)V USAGE	COMP-3.
10	MIR-DT-PROCES	PIC	S9(8)V USAGE	COMP-3.
10	MIR-NUM-AG	PIC	S9(5)V USAGE	COMP-3.
10	MIR-NUM-POSTO	PIC	S9(3)V USAGE	COMP-3.
10	MIR-DT-REPAS-DOC	PIC	S9(8)V USAGE	COMP-3.
10	MIR-CD-RCT	PIC	S9(5)V USAGE	COMP-3.
10	MIR-TP-IDENT	PIC	S9(2)V USAGE	COMP-3.
10	MIR-CD-IDENT	PIC	S9(15)V USAGE	COMP-3.
10	MIR-DT-COMPET	PIC	S9(6)V USAGE	COMP-3.
10	MIR-V-INSS	PIC	S9(13)V9(2) USAGE	COMP-3.
10	MIR-V-OUTRAS	PIC	S9(13)V9(2) USAGE	COMP-3.
10	MIR-V-ATRASSO	PIC	S9(13)V9(2) USAGE	COMP-3.
10	MIR-V-PAGO	PIC	S9(13)V9(2) USAGE	COMP-3.

10	MIR-CD-BARRAS	PIC X(44).	
10	MIR-DT-PGTO	PIC S9(8)V USAGE COMP-3.	
10	MIR-TP-CAPT-DAD	PIC S9(2)V USAGE COMP-3.	
10	MIR-ORIG-LANC	PIC S9(2)V USAGE COMP-3.	
10	MIR-TP-PGTO	PIC S9(2)V USAGE COMP-3.	
10	MIR-TP-DOC	PIC S9(1)V USAGE COMP-3.	
10	MIR-NUM-TERM-CX	PIC X(15).	
10	MIR-NSU	PIC S9(7)V USAGE COMP-3.	
10	MIR-SIT-DOC	PIC S9(1)V USAGE COMP-3.	
10	MIR-DCC-IRREG	PIC X(30).	
10	MIR-ID-CTB	PIC X(1).	
01 DCLMOV-ACM.			
10	MAC-DT-PROCES	PIC S9(8)V USAGE COMP-3.	
10	MAC-NUM-AG	PIC S9(5)V USAGE COMP-3.	
10	MAC-NUM-SQ	PIC S9(7)V USAGE COMP-3.	
10	MAC-NUM-POSTO	PIC S9(3)V USAGE COMP-3.	
10	MAC-DT-REPAS-DCC	PIC S9(8)V USAGE COMP-3.	
10	MAC-CD-RCT	PIC S9(5)V USAGE COMP-3.	
10	MAC-TP-IDENT	PIC S9(2)V USAGE COMP-3.	
10	MAC-CD-IDENT	PIC S9(15)V USAGE COMP-3.	
10	MAC-DT-COMPET	PIC S9(6)V USAGE COMP-3.	
10	MAC-V-INSS	PIC S9(13)V9(2) USAGE COMP-3.	
10	MAC-V-OUTRAS	PIC S9(13)V9(2) USAGE COMP-3.	
10	MAC-V-ATRASSO	PIC S9(13)V9(2) USAGE COMP-3.	
10	MAC-V-PAGO	PIC S9(13)V9(2) USAGE COMP-3.	
10	MAC-CD-BARRAS	PIC X(44).	
10	MAC-DT-PGTO	PIC S9(8)V USAGE COMP-3.	
10	MAC-TP-CAPT-DAD	PIC S9(2)V USAGE COMP-3.	
10	MAC-ORIG-LANC	PIC S9(2)V USAGE COMP-3.	
10	MAC-TP-PGTO	PIC S9(2)V USAGE COMP-3.	
10	MAC-TP-DOC	PIC S9(1)V USAGE COMP-3.	
10	MAC-NUM-TERM-CX	PIC X(15).	
10	MAC-NSU	PIC S9(7)V USAGE COMP-3.	
01 AREAS-DE-SALVAMENTO.			
05	AS-RETURN-CODE-4000	PIC S9(004) COMP	VALUE +4000.
05	AS-DV-AG	PIC X(001)	VALUE SPACES.
05	AS-NUM-SIST	PIC 9(003)	VALUE 370.
05	AS-NUM-POSTO	PIC 9(003)	VALUE ZEROES.
05	AS-NUM-AG	PIC 9(005)	VALUE ZEROES.
05	AS-DT-PROCES	PIC 9(008)	VALUE ZEROES.
05	AS-DT-PROCES-ANT	PIC 9(008)	VALUE ZEROES.
05	AS-XYZS800	PIC X(008)	VALUE
	'XYZS800'.		
05	AS-DSNTIAR	PIC X(008)	VALUE
	'DSNTIAR'.		
05	AS-SUB002	PIC X(008)	VALUE
	'SUB002'.		
05	AS-STATUS	PIC X(002)	VALUE '00'.
	88 STATUS-OK		VALUE '00'.
05	AS-DT-MV	PIC 9(008)	VALUE ZEROES.
05	FILLER	REDEFINES AS-DT-MV.	
	10 AS-ANO-MV	PIC 9(004).	
	10 AS-MES-MV	PIC 9(002).	
	10 AS-DIA-MV	PIC 9(002).	
05	CH-MVTO	PIC X(001)	VALUE 'N'.
05	AS-SRT-CHAVE.		
	10 AS-SRT-CHAVE-PRINCIPAL.		



	15	AS-SRT-NUM-AG	PIC 9(005)	COMP-3.
	15	AS-SRT-NUM-POSTO	PIC 9(003)	COMP-3.
	15	AS-SRT-CD-RCT	PIC 9(005)	COMP-3.
	15	AS-SRT-CD-IDENT	PIC 9(015)	COMP-3.
	15	AS-SRT-DT-COMPET	PIC 9(006).	
	15	AS-SRT-NSU	PIC 9(007)	COMP-3.
	15	AS-SRT-V-PAGO	PIC 9(013)V99	COMP-3.
	10	AS-SRT-ORIGEM	PIC 9(001).	
01		ACUMULADORES	COMP-3.	
	05	AC-E199	PIC 9(005)	VALUE ZEROES.
	05	AC-PEND	PIC 9(005)	VALUE ZEROES.
	05	AC-ACUM	PIC 9(005)	VALUE ZEROES.
	05	AC-S198	PIC 9(005)	VALUE ZEROES.
	05	AC-S197	PIC 9(005)	VALUE ZEROES.
	05	AC-NUM-SQ	PIC 9(007)	VALUE ZEROES.
01		MSG-ERRO-SQL1.		
	05	FILLER	PIC X(023)	VALUE
		'ERRO OPERACAO TABELA = '.		
	05	MSG-TABELA	PIC X(011)	VALUE SPACES.
	05	FILLER	PIC X(011)	VALUE
		' COMANDO = '.		
	05	MSG-COMANDO	PIC X(008)	VALUE SPACES.
	05	FILLER	PIC X(012)	VALUE
		' SQLCODE = '.		
	05	MSG-SQLCODE	PIC -9999.	
	05	FILLER	PIC X(019)	VALUE
		' CODIGO DATACOM = '.		
	05	MSG-SQLCA-DBCODE-EXT	PIC X(003)	VALUE SPACES.
01		MSG-ERRO-SUBROTINA.		
	05	FILLER	PIC X(020)	VALUE
		'ERRO NA SUBROTINA '.		
	05	MSG-SUB-NOME	PIC X(010)	VALUE SPACES.
	05	FILLER	PIC X(012)	VALUE
		' STATUS = '.		
	05	MSG-SUB-STATUS	PIC X(002)	VALUE SPACES.
	05	FILLER	PIC X(020)	VALUE
		' DB ERROR CODE = '.		
	05	MSG-SUB-DB-ERROR-CODE	PIC X(002)	VALUE SPACES.
01		MSG-ERRO-OPEN.		
	05	FILLER	PIC X(028)	VALUE
		'ERRO NA ABERTURA DO ARQUIVO'.		
	05	MSG-OPEN-ARQUIVO	PIC X(009)	VALUE SPACES.
	05	FILLER	PIC X(010)	VALUE
		' STATUS - '.		
	05	MSG-OPEN-STATUS	PIC X(004)	VALUE SPACES.
01		MSG-ERRO-READ.		
	05	FILLER	PIC X(027)	VALUE
		'ERRO NA LEITURA DO ARQUIVO'.		
	05	MSG-READ-ARQUIVO	PIC X(009)	VALUE SPACES.
	05	FILLER	PIC X(009)	VALUE
		'STATUS - '.		
	05	MSG-READ-STATUS	PIC X(004)	VALUE SPACES.
01		MSG-ERRO-WRITE.		
	05	FILLER	PIC X(028)	VALUE
		'ERRO NA GRAVACAO DO ARQUIVO'.		
	05	MSG-WRITE-ARQUIVO	PIC X(009)	VALUE SPACES.
	05	FILLER	PIC X(010)	VALUE
		' STATUS - '.		

01	05	MSG-WRITE-STATUS	PIC X(004)	VALUE SPACES.
	05	FILLER	PIC X(030)	VALUE
		'ERRO NO FECHAMENTO DO ARQUIVO'.		
	05	MSG-CLOSE-ARQUIVO	PIC X(009)	VALUE SPACES.
	05	FILLER	PIC X(010)	VALUE
		' STATUS -'.		
01	05	MSG-CLOSE-STATUS	PIC X(004)	VALUE SPACES.
	05	FILLER	PIC X(012)	VALUE
		'DATA PROCES'.		
	05	MSG-DIA-PRO	PIC 99/.	
	05	MSG-MES-PRO	PIC 99/.	
	05	MSG-ANO-PRO	PIC 9999.	
	05	FILLER	PIC X(015)	VALUE
		' DIF. DO MOVTO'.		
	05	MSG-DIA-DIF	PIC 99/.	
	05	MSG-MES-DIF	PIC 99/.	
	05	MSG-ANO-DIF	PIC 9999.	
01	05	MSG-MOVTO.		
	05	FILLER	PIC X(017)	VALUE
		'MOVIMENTO DO DIA '.		
	05	MSG-DIA-MV	PIC 99/.	
	05	MSG-MES-MV	PIC 99/.	
	05	MSG-ANO-MV	PIC 9999.	
01	05	MSG-TOT-LIDO.		
	05	FILLER	PIC X(027)	VALUE
		'LIDOS => E199 (AUTOM):'.		
	05	MSG-TOT-E199	PIC ZZ.ZZ9.	
	05	FILLER	PIC X(023)	VALUE
		' - PENDENTES :'.		
	05	MSG-TOT-PEND	PIC ZZ.ZZ9.	
	05	FILLER	PIC X(023)	VALUE
		' - ACUMULADO :'.		
	05	MSG-TOT-ACUM	PIC ZZ.ZZ9.	
01	05	MSG-TOT-GRAV.		
	05	FILLER	PIC X(027)	VALUE
		'GRAVADOS => S198 (OK) :'.		
	05	MSG-TOT-S198	PIC ZZ.ZZ9.	
	05	FILLER	PIC X(023)	VALUE
		' - S197 (IRREG):'.		
	05	MSG-TOT-S197	PIC ZZ.ZZ9.	
01	05	MSG-SEM-MOVTO.		
	05	FILLER	PIC X(024)	VALUE
		'NAO HOUVE MOVIMENTO DIA '.		
	05	MSG-DIA	PIC 99/.	
	05	MSG-MES	PIC 99/.	
	05	MSG-ANO	PIC 9999.	
01	05	ERROR-MESSAGE.		
	05	ERROR-LEN	PIC S9(004) COMP VALUE +300.	
	05	ERROR-TXT	PIC X(100) OCCURS 3 TIMES	
			INDEXED BY ERROR-IND.	
77	05	ERROR-TXT-LEN	PIC S9(009) COMP VALUE +100.	
01	05	MSG-TERMINO-ANORMAL	PIC X(031)	VALUE
		'PROGRAMA ENCERRADO ANORMALMENTE'.		
01	05	MSG-TERMINO-NORMAL	PIC X(030)	VALUE
		'PROGRAMA ENCERRADO COM SUCESSO'.		
		COPY 'CPGW01'.		
01		CPGW01.		

05	FILLER	PIC X(120)	VALUE ALL '*'.
05	FILLER	PIC X(119)	VALUE '*'.
05	FILLER	PIC X(011)	VALUE '***'.
05	FILLER	PIC X(010)	VALUE 'SISTEMA -'.
05	W001-SISTEMA	PIC X(030)	VALUE SPACES.
05	FILLER	PIC X(027)	VALUE SPACES.
05	FILLER	PIC X(009)	VALUE 'ROTINA -'.
05	W001-ROTINA	PIC X(010)	VALUE SPACES.
05	FILLER	PIC X(023)	VALUE SPACES.
05	FILLER	PIC X(120)	VALUE '***'.
05	FILLER	PIC X(010)	VALUE '***'.
05	FILLER	PIC X(011)	VALUE
	'PROGRAMA -'.		
05	W001-PROGRAMA	PIC X(008)	VALUE SPACES.
05	FILLER	PIC X(020)	VALUE
	'		
	CATALOGADO EM -'.		
05	W001-DATA-CATALOGACAO	PIC X(020)	VALUE SPACES.
05	FILLER	PIC X(006)	VALUE SPACES.
05	FILLER	PIC X(012)	VALUE
	'PARAGRAFO -'.		
05	W001-PARAGRAFO	PIC X(031)	VALUE SPACES.
05	FILLER	PIC X(122)	VALUE ' ***'.
05	FILLER	PIC X(011)	VALUE '***'.
05	W001-MENSAGEM-1	PIC X(100)	VALUE SPACES.
05	FILLER	PIC X(009)	VALUE SPACES.
05	FILLER	PIC X(011)	VALUE '***'.
05	W001-MENSAGEM-2	PIC X(100)	VALUE SPACES.
05	FILLER	PIC X(009)	VALUE SPACES.
05	FILLER	PIC X(011)	VALUE '***'.
05	W001-MENSAGEM-3	PIC X(100)	VALUE SPACES.
05	FILLER	PIC X(009)	VALUE SPACES.
05	FILLER	PIC X(011)	VALUE '***'.
05	W001-MENSAGEM-4	PIC X(100)	VALUE SPACES.
05	FILLER	PIC X(009)	VALUE SPACES.
05	FILLER	PIC X(011)	VALUE '***'.
05	W001-MENSAGEM-5	PIC X(100)	VALUE SPACES.
05	FILLER	PIC X(009)	VALUE SPACES.
05	FILLER	PIC X(120)	VALUE '***'.
05	FILLER	PIC X(121)	VALUE ALL '*'.
	COPY 'XYZW800'.		
01	LG-XYZ800.		
05	XYZ800-DADOS-DE-ENTRADA.		
10	XYZ800-ID-ORIG	PIC S9(002)	COMP-3.
10	XYZ800-TP-CAPT	PIC S9(002)	COMP-3.
10	XYZ800-NUM-AG	PIC S9(005)	COMP-3.
10	XYZ800-DT-PGTO	PIC S9(008)	COMP-3.
10	XYZ800-GPS	PIC X(044).	
10	FILLER	REDEFINES XYZ800-GPS.	
15	XYZ800-CD-RCT	PIC S9(005)	COMP-3.
15	XYZ800-DT-COMP	PIC S9(006)	COMP-3.
15	XYZ800-CD-IDENT	PIC S9(015)	COMP-3.
15	XYZ800-V-TOTAL	PIC S9(013)V99	COMP-3.
15	FILLER	PIC X(021).	
10	XYZ800-V-INSS	PIC S9(013)V99	COMP-3.
10	XYZ800-V-OUTRAS	PIC S9(013)V99	COMP-3.
10	XYZ800-V-ATRASSO	PIC S9(013)V99	COMP-3.
10	XYZ800-V-PAGO	PIC S9(013)V99	COMP-3.
10	FILLER	PIC X(025).	
10	XYZ800-CD-BARRA	PIC X(044).	

```

05 XYZ800-DADOS-DE-SAIDA.
   10 XYZ800-TP-IDENT          PIC S9(002)      COMP-3.
   10 XYZ800-CD-ERRO          PIC 9(002).
   10 FILLER                   PIC X(020).
05 XYZ800-DADOS-DE-CONTROLE.
   10 XYZ800-STATUS           PIC X(002).
   10 XYZ800-SQLCODE          PIC S9(004).
   10 XYZ800-SQLCA-DBCODE-EXT PIC X(003).
   10 XYZ800-DCC-STATUS       PIC X(100).
01 FILLER                      PIC X(032)      VALUE
   'FFF WORKING-STORAGE SECTION FFF'.

```

PROCEDURE DIVISION.

```

PERFORM A00-INICIALIZA
SORT SORTWK01
   ASCENDING KEY              SRT-CHAVE
   INPUT PROCEDURE            B00-CARREGA-SORT
   OUTPUT PROCEDURE           C00-PROCESSA-SORT
IF SORT-RETURN                NOT = ZEROES
   MOVE 'ERRO NO SORT'        TO W001-MENSAGEM-2
   PERFORM Z00-ERROR
END-IF
PERFORM W00-FINALIZA
GOBACK.

```

A00-INICIALIZA SECTION.

```

MOVE 'XYZ - ARRECADACAO DA PREVIDENCIA FEDERAL'
                                     TO W001-SISTEMA
MOVE 'XYZ510' TO W001-PROGRAMA
MOVE 'DIARIA'                       TO W001-ROTINA
MOVE WHEN-COMPILED                  TO W001-DATA-CATALOGACAO
MOVE MSG-TERMINO-ANORMAL             TO W001-MENSAGEM-5
MOVE 'A00-INICIALIZA'               TO W001-PARAGRAFO
MOVE AS-NUM-SIST                    TO CPA-NUM-SIST
EVALUATE SQLCODE
  WHEN 0
    MOVE CPA-DT-PROCES               TO AS-DT-PROCES
    MOVE CPA-DT-PROCES-ANT           TO AS-DT-PROCES-ANT
  WHEN 100
    MOVE 'NAO ENCONTROU CONTROLE DO PROCESSAMENTO (XYZ/370)'
                                     TO W001-MENSAGEM-1
    PERFORM Z00-ERROR
  WHEN OTHER
    MOVE 'SELECT'                   TO MSG-COMANDO
    MOVE 'CTL_PRO_XYZ'               TO MSG-TABELA
    PERFORM Y00-ERRO-SQL
END-EVALUATE
IF SQLCODE                          NOT = ZEROES
  MOVE 'OPEN'                       TO MSG-COMANDO
  MOVE 'MOV_IRG'                    TO MSG-TABELA
  PERFORM Y00-ERRO-SQL
END-IF
EVALUATE SQLCODE
  WHEN 0
    MOVE MIR-NUM-SQ                  TO AC-NUM-SQ
  WHEN 100
    MOVE ZEROES                     TO AC-NUM-SQ
  WHEN OTHER
    MOVE 'FETCH'                    TO MSG-COMANDO
    MOVE 'MOV_IRG'                  TO MSG-TABELA

```

```

        PERFORM Y00-ERRO-SQL
END-EVALUATE
IF  SQLCODE                                NOT = ZEROES
    MOVE 'CLOSE'                            TO MSG-COMANDO
    MOVE 'MOV_IRG'                          TO MSG-TABELA
    PERFORM Y00-ERRO-SQL
END-IF
OPEN INPUT XYZE199
IF  NOT STATUS-OK
    MOVE AS-STATUS                          TO MSG-OPEN-STATUS
    MOVE 'XYZE199'                          TO MSG-OPEN-ARQUIVO
    MOVE MSG-ERRO-OPEN                      TO W001-MENSAGEM-1
    PERFORM Z00-ERROR
END-IF
OPEN OUTPUT XYZS198
IF  NOT STATUS-OK
    MOVE AS-STATUS                          TO MSG-OPEN-STATUS
    MOVE 'XYZS198'                          TO MSG-OPEN-ARQUIVO
    MOVE MSG-ERRO-OPEN                      TO W001-MENSAGEM-1
    CLOSE XYZE199
    PERFORM Z00-ERROR
END-IF
OPEN OUTPUT XYZS197
IF  NOT STATUS-OK
    MOVE AS-STATUS                          TO MSG-OPEN-STATUS
    MOVE 'XYZS197'                          TO MSG-OPEN-ARQUIVO
    MOVE MSG-ERRO-OPEN                      TO W001-MENSAGEM-1
    CLOSE XYZE199 XYZS198
    PERFORM Z00-ERROR
END-IF
INITIALIZE                                D198A-REG
MOVE ZEROES                               TO D198A-TIPO-REG
MOVE AS-DT-PROCES                         TO D198A-DT-PROCES
WRITE D198A-REG
IF  NOT STATUS-OK
    MOVE AS-STATUS                          TO MSG-WRITE-STATUS
    MOVE 'XYZS198'                          TO MSG-WRITE-ARQUIVO
    MOVE MSG-ERRO-WRITE                     TO W001-MENSAGEM-1
    CLOSE XYZE199 XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF
ADD 1                                      TO AC-S198
INITIALIZE                                D197A-REG
MOVE ZEROES                               TO D197A-TIPO-REG
MOVE AS-DT-PROCES                         TO D197A-DT-PROCES
WRITE D197A-REG
IF  NOT STATUS-OK
    MOVE AS-STATUS                          TO MSG-WRITE-STATUS
    MOVE 'XYZS197'                          TO MSG-WRITE-ARQUIVO
    MOVE MSG-ERRO-WRITE                     TO W001-MENSAGEM-1
    CLOSE XYZE199 XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF
ADD 1                                      TO AC-S197
MOVE AS-DT-PROCES(7:2)                   TO MSG-DIA
                                                MSG-DIA-MV
                                                MSG-DIA-PRO
MOVE AS-DT-PROCES(5:2)                   TO MSG-MES
                                                MSG-MES-MV

```

```

MSG-MES-PRO
MOVE AS-DT-PROCES(1:4)      TO MSG-ANO
                             MSG-ANO-MV
                             MSG-ANO-PRO.

A00-EXIT.
EXIT.

B00-CARREGA-SORT SECTION.
PERFORM B00-10-LER-AUTOMACAO
PERFORM B00-20-LER-PENDENTE
PERFORM B00-30-LER-ACUMULADO.

B00-EXIT.
EXIT.

B00-10-LER-AUTOMACAO SECTION.
MOVE 'B00-10-LER-AUTOMACAO' TO W001-PARAGRAFO
READ XYZE199
    AT END GO TO B00-10-B-FIM
END-READ
IF NOT STATUS-OK
    MOVE AS-STATUS          TO MSG-READ-STATUS
    MOVE 'XYZE199'          TO MSG-READ-ARQUIVO
    MOVE MSG-ERRO-READ      TO W001-MENSAGEM-1
    CLOSE XYZE199 XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF
ADD 1                      TO AC-E199
IF D199A-TIPO-REG          NOT = ZEROES
    MOVE 'PRIMEIRO REGISTRO NAO E HEADER'
                                TO W001-MENSAGEM-1
    CLOSE XYZE199 XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF
IF D199A-DT-PROCES        NOT = AS-DT-PROCES
AND D199A-DT-PROCES        NOT = AS-DT-PROCES-ANT
    MOVE D199A-DT-PROCES    TO AS-DT-MV
    MOVE AS-DIA-MV          TO MSG-DIA-DIF
    MOVE AS-MES-MV          TO MSG-MES-DIF
    MOVE AS-ANO-MV          TO MSG-ANO-DIF
    MOVE MSG-DT-DIF         TO W001-MENSAGEM-1
    CLOSE XYZE199 XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF

B00-10-A-LOOP.
READ XYZE199
    AT END GO TO B00-10-B-FIM
END-READ
IF NOT STATUS-OK
    MOVE AS-STATUS          TO MSG-READ-STATUS
    MOVE 'XYZE199'          TO MSG-READ-ARQUIVO
    MOVE MSG-ERRO-READ      TO W001-MENSAGEM-1
    CLOSE XYZE199 XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF
ADD 1                      TO AC-E199
IF D199A-TIPO-REG          EQUAL ZEROES
    GO TO B00-10-A-LOOP
END-IF
MOVE D199B-DT-PROCES      TO SRT-DT-PROCES
MOVE D199B-NUM-AG        TO SRT-NUM-AG

```

```

MOVE D199B-NUM-POSTO          TO SRT-NUM-POSTO
MOVE D199B-CD-RCT             TO SRT-CD-RCT
MOVE D199B-CD-IDENT          TO SRT-CD-IDENT
MOVE D199B-DT-COMPET         TO SRT-DT-COMPET
MOVE D199B-V-INSS            TO SRT-V-INSS
MOVE D199B-V-OUTRAS          TO SRT-V-OUTRAS
MOVE D199B-V-ATRASSO         TO SRT-V-ATRASSO
MOVE D199B-V-PAGO            TO SRT-V-PAGO
MOVE D199B-TP-CAPT-DAD       TO SRT-TP-CAPT-DAD
MOVE D199B-ORIG-LANC         TO SRT-ORIG-LANC
MOVE D199B-AG-DEB            TO SRT-AG-DEB
MOVE D199B-CTA-DEB           TO SRT-CTA-DEB
MOVE D199B-DV-CTA-DEB        TO SRT-DV-CTA-DEB
MOVE D199B-TP-CTA-DEB        TO SRT-TP-CTA-DEB
MOVE D199B-DT-PGTO           TO SRT-DT-PGTO
MOVE D199B-DT-PGTO-EFET      TO SRT-DT-PGTO-EFET
MOVE D199B-HRR-PGTO          TO SRT-HRR-PGTO
MOVE D199B-TITULARIDADE      TO SRT-TITULARIDADE
MOVE D199B-VIA-CARTAO        TO SRT-VIA-CARTAO
MOVE D199B-TP-CARTAO         TO SRT-TP-CARTAO
MOVE D199B-NUM-CARTAO        TO SRT-NUM-CARTAO
IF D199B-CD-BARRAS           = ZEROES
    MOVE SPACES                TO SRT-CD-BARRAS
ELSE
    MOVE D199B-CD-BARRAS       TO SRT-CD-BARRAS
END-IF
IF D199B-TP-PGTO              = 3
    MOVE 9                      TO SRT-TP-PGTO
ELSE
    MOVE D199B-TP-PGTO         TO SRT-TP-PGTO
END-IF
MOVE D199B-TP-DOC             TO SRT-TP-DOC
MOVE D199B-NUM-TERM-CX       TO SRT-NUM-TERM-CX
MOVE D199B-NSU               TO SRT-NSU
MOVE 3                        TO SRT-ORIGEM
RELEASE RG-SORTWK01
GO TO B00-10-A-LOOP.
B00-10-B-FIM.
CLOSE XYZE199
IF NOT STATUS-OK
    MOVE AS-STATUS              TO MSG-CLOSE-STATUS
    MOVE 'XYZE199'              TO MSG-CLOSE-ARQUIVO
    MOVE MSG-ERRO-CLOSE         TO W001-MENSAGEM-1
    CLOSE XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF.
B00-10-EXIT.
EXIT.
B00-20-LER-PENDENTE SECTION.
MOVE 'B00-20-LER-PENDENTE'    TO W001-PARAGRAFO
MOVE AS-DT-PROCES             TO MPE-DT-PROCES
IF SQLCODE                     NOT = ZEROES
    MOVE 'OPEN'                 TO MSG-COMANDO
    MOVE 'MOV PEN'              TO MSG-TABELA
    CLOSE XYZS198 XYZS197
    PERFORM Y00-ERRO-SQL
END-IF.
B00-20-A-LOOP.
EVALUATE SQLCODE

```

```

WHEN 0
  CONTINUE
WHEN 100
  GO TO B00-20-B-FIM
WHEN OTHER
  MOVE 'FETCH' TO MSG-COMANDO
  MOVE 'MOV_PEN' TO MSG-TABELA
  CLOSE XYZS198 XYZS197
  PERFORM Y00-ERRO-SQL
END-EVALUATE
MOVE MPE-DT-PROCES TO SRT-DT-PROCES
MOVE MPE-NUM-AG TO SRT-NUM-AG
MOVE MPE-NUM-POSTO TO SRT-NUM-POSTO
MOVE MPE-CD-RCT TO SRT-CD-RCT
MOVE MPE-CD-IDENT TO SRT-CD-IDENT
MOVE MPE-DT-COMPET TO SRT-DT-COMPET
MOVE MPE-V-INSS TO SRT-V-INSS
MOVE MPE-V-OUTRAS TO SRT-V-OUTRAS
MOVE MPE-V-ATRASSO TO SRT-V-ATRASSO
MOVE MPE-V-PAGO TO SRT-V-PAGO
MOVE MPE-CD-BARRAS TO SRT-CD-BARRAS
MOVE MPE-DT-PGTO TO SRT-DT-PGTO
MOVE MPE-TP-CAPT-DAD TO SRT-TP-CAPT-DAD
IF MPE-TP-PGTO = 3
  MOVE S TO SRT-TP-PGTO
ELSE
  MOVE MPE-TP-PGTO TO SRT-TP-PGTO
END-IF
MOVE MPE-TP-DOC TO SRT-TP-DOC
MOVE MPE-NUM-TERM-CX TO SRT-NUM-TERM-CX
MOVE MPE-NSU TO SRT-NSU
MOVE 1 TO SRT-ORIG-LANC
MOVE 2 TO SRT-ORIGEM
RELEASE RG-SORTWK01
ADD 1 TO AC-PEND
GO TO B00-20-A-LOOP.
B00-20-B-FIM.
IF SQLCODE NOT = ZEROES
  MOVE 'B00-20-B-FIM' TO W001-PARAGRAFO
  MOVE 'CLOSE' TO MSG-COMANDO
  MOVE 'MOV_PEN' TO MSG-TABELA
  CLOSE XYZS198 XYZS197
  PERFORM Y00-ERRO-SQL
END-IF.
B00-20-EXIT.
EXIT.
B00-30-LER-ACUMULADO SECTION.
MOVE 'B00-30-LER-ACUMULADO' TO W001-PARAGRAFO
MOVE AS-DT-PROCES-ANT TO MAC-DT-PROCES
IF SQLCODE NOT = ZEROES
  MOVE 'OPEN' TO MSG-COMANDO
  MOVE 'MOV_ACM' TO MSG-TABELA
  CLOSE XYZS198 XYZS197
  PERFORM Y00-ERRO-SQL
END-IF.
B00-30-A-LOOP.
EVALUATE SQLCODE
WHEN 0
  CONTINUE

```



```

WHEN 100
  GO TO B00-30-B-FIM
WHEN OTHER
  MOVE 'FETCH' TO MSG-COMANDO
  MOVE 'MOV_ACM' TO MSG-TABELA
  CLOSE XYZS198 XYZS197
  PERFORM Y00-ERRO-SQL
END-EVALUATE
MOVE MAC-DT-PROCES TO SRT-DT-PROCES
MOVE MAC-NUM-AG TO SRT-NUM-AG
MOVE MAC-NUM-POSTO TO SRT-NUM-POSTO
MOVE MAC-CD-RCT TO SRT-CD-RCT
MOVE MAC-CD-IDENT TO SRT-CD-IDENT
MOVE MAC-DT-COMPET TO SRT-DT-COMPET
MOVE MAC-V-INSS TO SRT-V-INSS
MOVE MAC-V-OUTRAS TO SRT-V-OUTRAS
MOVE MAC-V-ATRASSO TO SRT-V-ATRASSO
MOVE MAC-V-PAGO TO SRT-V-PAGO
MOVE MAC-CD-BARRAS TO SRT-CD-BARRAS
MOVE MAC-DT-PGTO TO SRT-DT-PGTO
MOVE MAC-TP-CAPT-DAD TO SRT-TP-CAPT-DAD
MOVE MAC-TP-PGTO TO SRT-TP-PGTO
MOVE MAC-TP-DOC TO SRT-TP-DOC
MOVE MAC-NUM-TERM-CX TO SRT-NUM-TERM-CX
MOVE MAC-NSU TO SRT-NSU
MOVE MAC-ORIG-LANC TO SRT-ORIG-LANC
MOVE 1 TO SRT-ORIGEM
RELEASE RG-SORTWK01
ADD 1 TO AC-ACUM
GO TO B00-30-A-LOOP.
B00-30-B-FIM.
IF SQLCODE NOT = ZEROES
  MOVE 'B00-30-B-FIM' TO W001-PARAGRAFO
  MOVE 'CLOSE' TO MSG-COMANDO
  MOVE 'MOV_ACM' TO MSG-TABELA
  CLOSE XYZS198 XYZS197
  PERFORM Y00-ERRO-SQL
END-IF.
B00-30-EXIT.
EXIT.
C00-PROCESSA-SORT SECTION.
INITIALIZE AS-SRT-CHAVE.
C00-A-LOOP.
RETURN SORTWK01
AT END GO TO C00-B-FIM
END-RETURN
IF SRT-ORIGEM = 2 OR 3
  IF SRT-CHAVE-PRINCIPAL = AS-SRT-CHAVE-PRINCIPAL
    MOVE 80 TO XYZ800-CD-ERRO
    MOVE ZEROES TO XYZ800-TP-IDENT
    PERFORM C00-30-GRAVA-IRREGULAR
  ELSE
    PERFORM C00-10-CONSISTE
  END-IF
  MOVE 'S' TO CH-MVTO
END-IF
MOVE SRT-CHAVE TO AS-SRT-CHAVE
GO TO C00-A-LOOP.
C00-B-FIM.

```

```

IF CH-MVTO NOT = 'S'
MOVE 'NAO HOUVE MOVIMENTO NESTA DATA'
TO W001-MENSAGEM-1
END-IF.
C00-EXIT.
EXIT.
C00-10-CONSISTE SECTION.
MOVE SRT-ORIG-LANC TO XYZ800-ID-ORIG
MOVE SRT-TP-CAPT-DAD TO XYZ800-TP-CAPT
MOVE SRT-NUM-AG TO XYZ800-NUM-AG
MOVE SRT-DT-PGTO TO XYZ800-DT-PGTO
MOVE SRT-CD-RCT TO XYZ800-CD-RCT
MOVE SRT-DT-COMPET TO XYZ800-DT-COMP
MOVE SRT-CD-IDENT TO XYZ800-CD-IDENT
MOVE SRT-V-INSS TO XYZ800-V-INSS
MOVE SRT-V-OUTRAS TO XYZ800-V-OUTRAS
MOVE SRT-V-ATRASO TO XYZ800-V-ATRASO
MOVE SRT-V-PAGO TO XYZ800-V-PAGO
XYZ800-V-TOTAL
MOVE SRT-CD-BARRAS TO XYZ800-CD-BARRA
CALL AS-XYZS600 USING LG-XYZ800
IF XYZ800-STATUS NOT = '00'
PERFORM C00-30-GRAVA-IRREGULAR
ELSE
PERFORM C00-20-GRAVA-MOVTO-OK
END-IF.
C00-10-EXIT.
EXIT.
C00-20-GRAVA-MOVTO-OK SECTION.
MOVE 'C00-20-GRAVA-MOVTO-O' TO W001-PARAGRAFO
INITIALIZE D198B-REG
MOVE 1 TO D198B-TIPO-REG
MOVE SRT-DT-PROCES TO D198B-DT-PROCES
MOVE SRT-NUM-AG TO D198B-NUM-AG
MOVE SRT-NUM-POSTO TO D198B-NUM-POSTO
MOVE SRT-CD-RCT TO D198B-CD-RCT
MOVE XYZ800-TP-IDENT TO D198B-TP-IDENT
MOVE SRT-CD-IDENT TO D198B-CD-IDENT
MOVE SRT-DT-COMPET TO D198B-DT-COMPET
MOVE SRT-V-INSS TO D198B-V-INSS
MOVE SRT-V-OUTRAS TO D198B-V-OUTRAS
MOVE SRT-V-ATRASO TO D198B-V-ATRASO
MOVE SRT-V-PAGO TO D198B-V-PAGO
MOVE SRT-CD-BARRAS TO D198B-CD-BARRAS
MOVE SRT-DT-PGTO TO D198B-DT-PGTO
MOVE SRT-TP-CAPT-DAD TO D198B-TP-CAPT-DAD
MOVE SRT-ORIG-LANC TO D198B-ORIG-LANC
MOVE SRT-TP-PGTO TO D198B-TP-PGTO
MOVE SRT-TP-DOC TO D198B-TP-DOC
MOVE SRT-NUM-TERM-CX TO D198B-NUM-TERM-CX
MOVE SRT-NSU TO D198B-NSU
MOVE SRT-AG-DEB TO D198B-AG-DEB
MOVE SRT-CTA-DEB TO D198B-CTA-DEB
MOVE SRT-DV-CTA-DEB TO D198B-DV-CTA-DEB
MOVE SRT-TP-CTA-DEB TO D198B-TP-CTA-DEB
MOVE SRT-NUM-CARTAO TO D198B-NUM-CARTAO
MOVE SRT-TP-CARTAO TO D198B-TP-CARTAO
MOVE SRT-VIA-CARTAO TO D198B-VIA-CARTAO
MOVE SRT-TITULARIDADE TO D198B-TITULARIDADE

```

```

MOVE SRT-DT-PGTO-EFET          TO D198B-DT-PGTO-EFET
MOVE SRT-HRR-PGTO              TO D198B-HRR-PGTO
WRITE D198B-REG
IF NOT STATUS-OK
    MOVE AS-STATUS              TO MSG-READ-STATUS
    MOVE 'XYZS198'              TO MSG-READ-ARQUIVO
    MOVE MSG-ERRO-READ          TO W001-MENSAGEM-1
    CLOSE XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF
ADD 1                          TO AC-S198.
C00-20-EXIT.
EXIT.
C00-30-GRAVA-IRREGULAR SECTION.
MOVE 'C00-30-GRAVA-IRREGUL' TO W001-PARAGRAFO
INITIALIZE                     D197B-REG
ADD 1                          TO AC-NUM-SQ
MOVE 1                          TO D197B-TIPO-REG
MOVE SRT-DT-PROCES             TO D197B-DT-PROCES
MOVE SRT-NUM-AG                TO D197B-NUM-AG
MOVE AC-NUM-SQ                 TO D197B-NUM-SQ
MOVE SRT-NUM-POSTO            TO D197B-NUM-POSTO
MOVE SRT-CD-RCT               TO D197B-CD-RCT
MOVE XYZ800-TP-IDENT          TO D197B-TP-IDENT
MOVE SRT-CD-IDENT             TO D197B-CD-IDENT
MOVE SRT-DT-COMPET            TO D197B-DT-COMPET
MOVE SRT-V-INSS               TO D197B-V-INSS
MOVE SRT-V-OUTRAS             TO D197B-V-OUTRAS
MOVE SRT-V-ATRASO             TO D197B-V-ATRASO
MOVE SRT-V-PAGO               TO D197B-V-PAGO
MOVE SRT-CD-BARRAS            TO D197B-CD-BARRAS
MOVE SRT-DT-PGTO              TO D197B-DT-PGTO
MOVE SRT-TP-CAPT-DAD          TO D197B-TP-CAPT-DAD
MOVE SRT-ORIG-LANC            TO D197B-ORIG-LANC
MOVE SRT-TP-PGTO              TO D197B-TP-PGTO
MOVE SRT-TP-DOC               TO D197B-TP-DOC
MOVE SRT-NUM-TERM-CX          TO D197B-NUM-TERM-CX
MOVE SRT-NSU                  TO D197B-NSU
MOVE SRT-AG-DEB               TO D197B-AG-DEB
MOVE SRT-CTA-DEB              TO D197B-CTA-DEB
MOVE SRT-DV-CTA-DEB           TO D197B-DV-CTA-DEB
MOVE SRT-TP-CTA-DEB           TO D197B-TP-CTA-DEB
MOVE SRT-NUM-CARTAO           TO D197B-NUM-CARTAO
MOVE SRT-TP-CARTAO            TO D197B-TP-CARTAO
MOVE SRT-VIA-CARTAO           TO D197B-VIA-CARTAO
MOVE SRT-TITULARIDADE         TO D197B-TITULARIDADE
MOVE SRT-DT-PGTO-EFET         TO D197B-DT-PGTO-EFET
MOVE SRT-HRR-PGTO             TO D197B-HRR-PGTO
EVALUATE TRUE
    WHEN XYZ800-CD-ERRO        = 1 OR 2
        MOVE 3                  TO D197B-SIT-DOC
    WHEN XYZ800-CD-ERRO        = 6 OR 7 OR 8 OR 9
        AND SRT-V-PAGO          = ZEROES
        MOVE 3                  TO D197B-SIT-DOC
    WHEN XYZ800-CD-ERRO        = 80
        MOVE 2                  TO D197B-SIT-DOC
        MOVE 'GPS DUPLICADA'    TO XYZ800-DCC-STATUS
    WHEN OTHER
        MOVE 1                  TO D197B-SIT-DOC

```

```

END-EVALUATE
MOVE XYZ800-DCC-STATUS      TO D197B-DCC-IRREG
WRITE D197B-REG
IF NOT STATUS-OK
    MOVE AS-STATUS          TO MSG-READ-STATUS
    MOVE 'XYZS197'         TO MSG-READ-ARQUIVO
    MOVE MSG-ERRO-READ     TO W001-MENSAGEM-1
    CLOSE XYZS198 XYZS197
    PERFORM Z00-ERROR
END-IF
ADD 1                      TO AC-S197.
C00-30-EXIT.
EXIT.
W00-FINALIZA SECTION.
MOVE 'W00-FINALIZA'       TO W001-PARAGRAFO
IF AC-S197
    MOVE AC-E199           TO MSG-TOT-E199
    MOVE AC-PEND          TO MSG-TOT-PEND
    MOVE AC-ACUM          TO MSG-TOT-ACUM
    MOVE AC-S198         TO MSG-TOT-S198
    MOVE AC-S197         TO MSG-TOT-S197
    MOVE MSG-MOVTO       TO W001-MENSAGEM-1
    MOVE MSG-TOT-LIDO    TO W001-MENSAGEM-2
    MOVE MSG-TOT-GRAV    TO W001-MENSAGEM-3
ELSE
    MOVE MSG-SEM-MOVTO   TO W001-MENSAGEM-1
END-IF
CLOSE XYZS198
IF NOT STATUS-OK
    MOVE AS-STATUS        TO MSG-CLOSE-STATUS
    MOVE 'XYZS198'       TO MSG-CLOSE-ARQUIVO
    MOVE MSG-ERRO-CLOSE  TO W001-MENSAGEM-1
    CLOSE XYZS197
    PERFORM Z00-ERROR
END-IF
CLOSE XYZS197
IF NOT STATUS-OK
    MOVE AS-STATUS        TO MSG-CLOSE-STATUS
    MOVE 'XYZS197'       TO MSG-CLOSE-ARQUIVO
    MOVE MSG-ERRO-CLOSE  TO W001-MENSAGEM-1
    PERFORM Z00-ERROR
END-IF
MOVE MSG-TERMINO-NORMAL   TO W001-MENSAGEM-5
DISPLAY CPGW01.
W00-EXIT.
EXIT.
Y00-ERRO-SQL SECTION.
MOVE SQLCODE             TO MSG-SQLCODE
MOVE MSG-ERRO-SQL1     TO W001-MENSAGEM-1
PERFORM Z00-ERROR.
Y00-EXIT.
EXIT.
Z00-ERROR SECTION.
DISPLAY CPGW01
Z00-EXIT.
EXIT.

```

## Apêndice D

### Ontologia do Sistema de Informação Legado XYZ

Para a construção da *Ontologia* do Sistema XYZ, foram levantadas informações junto à documentação existente na organização e aos técnicos que mantêm o sistema. Na seqüência, mostra-se a *Ontologia* estruturada de forma apenas a ser inserida num sistema computacional destinada para este fim.

Identificador	Definição do Dicionário	É um	Associações	Sinónimos
Endereço	Local onde o contribuinte está estabelecido		(Contribuinte)	
Logradouro	Nome do local onde está estabelecido à pessoa	(Endereço)		
Número do Logradouro	Número do local no logradouro onde está estabelecida a pessoa	(Endereço)		
Cidade	Cidade onde a pessoa está estabelecida	(Endereço)		
Cep	Código de endereçamento postal do logradouro da pessoa	(Endereço)		
Pessoa	Tipo de pessoa estabelecida em lei		(Contribuinte)	
Pessoa Física	Ser humano considerado singularmente como sujeito de direitos e obrigações	(Pessoa)		
Nome Pessoa Física	Palavra que designa a pessoa física	(Pessoa Física)		
CPF	Cadastro de pessoa física	(Pessoa Física)		
Pessoa Jurídica	Entidade jurídica resultante de um agrupamento humano, organizada e estável com fins de utilidade pública ou privada (distinta dos indivíduos que a compõem), podendo exercer direitos e contrair obrigações	(Pessoa)		
Nome Pessoa Jurídica	Palavra que designa a pessoa jurídica	(Pessoa Jurídica)		
CNPJ	Cadastro Nacional da Pessoa Jurídica	(Pessoa Jurídica)		
Enviar		(propriedade)		Mandar, encaminhar, levar

Identificador	Definição do Dicionário	É um	Associações	Sinónimos
KISF	Nó raiz da Ontologia			
Cliente	Pessoa para a qual um produto ou serviço é ofertado			
Telefone	Meio de comunicação		(contribuinte)	Fone
Celular	Número do celular da pessoa	(Telefone)		
Fax	Número do Fax da pessoa	(Telefone)		
Linha Fixa	Número do telefone linha fixa	(Telefone)		
Prazo	Quantidade de tempo			
Dados de um arquivo	Informações de controle de um arquivo para envio ao INSS			
Data de Remessa	Data de envio do arquivo			
Número da Remessa	Número de envio da remessa	(Dados Arquivo)		
Endereço de Rede	Endereço IP da rede p/ envio	(Dados Arquivo)		

Identificador	Definição do Dicionário	É um	Associações	Sinônimos
Identificação da Rede	Identificação da Rede. Pode ser: 1) STM400,2) Office Bank, 3) Proceda,4) RVS, 5) Interch, 6) NJE, 7) NDM, 8) USTF	(Dados Arquivo)		
Label do arquivo	Identificador do arquivo Label do Arquivo. Pode ser: 1) NL, 2) SL, 3) RJ, 4) SN	(Dados Arquivo)		
Densidade do arquivo	Densidade do Arquivo. Pode ser: 1) 0800, 2) 1600, 3) 6250	(Dados Arquivo)		
Tamanho do registro do arquivo	Tamanho em <i>bytes</i> do registro	(Dados Arquivo)		
Blocagem do arquivo	Número de blocos do arquivo	(Dados Arquivo)		
Meio de envio do arquivo	Meio de Envio. Pode ser: 1) Fita 2) Cartucho 3) Disquete 4) Tele Processamento 5) Papel	(Dados Arquivo)		
Número de dias de retenção	Quantidade de dias que o arquivo deve ser retido	(Dados Arquivo)		
Número da versão	Número da versão do arquivo de remessa	(Dados Arquivo)		
Nome do arquivo	Nome de identificação do arquivo	(Dados Arquivo)		
Identificador do Contribuinte	Identificação de objetos ou entidades		(Contribuinte)	
Numero do NIT	Número do NIT	(Número)		
Número PIS PASEP	Código de identificação PIS ou PASEP	(Número)		



Identificador	Definição do Dicionário	É um	Associações	Sinônimos
Data	Dia, mês, ano da ocorrência de um fato			
Controle de Informação	Controle de datas, destino e duplicação			
Controle de datas	Informação que determina como deve ser o Controle das datas do Convênio. Pode ser: 1) Única, 2) Várias	(Controle Informação)		
Nome CPD destino	Nome do CPD de destino da remessa	(Controle Informação)		
Controle da duplicação	Descrição de como devem ser tratadas as duplicações de arquivos. Pode ser: 1) Não consiste duplicação, 2) Rejeita todos duplicados, 3) Rejeita parcial duplicados, 4) Rejeita relatórios duplicados	(Controle Informação)		
Valor	Valor arrecadado do INSS		(GPS)	
Valor de Multa	Valor monetário devido sobre a falta de recolhimento das contribuições urbanas e rurais devidas ao INSS, correspondente àquela vigente à época de ocorrência, sobre o valor atualizado monetariamente até a data do efetivo recolhimento	(Valor)	(Valor)	
Valor Atualização Monetária	Valor devido a título de atualização monetária e acréscimos legais, sobre recolhimentos em atraso	(Valor)	(Valor)	
Valor do INSS	Valor devido ao INSS	(Valor)	(Valor)	
Valor de Juros	Valor devido sobre as contribuições previdenciárias pagas após o vencimento, incidentes sobre o valor atualizado equivalentes a: 1% no mês de vencimento, Taxa referencial do SELIC (Sistema Especial de Liquidação e de Custódia) nos meses intermediários e 1% no mês de pagamento	(Valor)	(Valor)	

Identificador	Definição do Dicionário	É um	Associações	Sinônimos
GPS	Guia da Previdência Social, documento no qual são cadastrados os dados do contribuinte e valores a contribuir			
Contribuinte	Pessoa física ou jurídica cadastrada no INSS que contribua para a Previdência Social com o intuito de receber os devidos benefícios da Instituição		(GPS)	
Identificador	Números de identificação do contribuinte (NIT, PIS, Pasep)		(Contribuinte)	
Endereço	Endereço da pessoa		(Contribuinte)	
Pessoa	Dados da pessoa que contribui com o INSS		(Contribuinte)	
Telefone	Telefones da pessoa contribuinte		(Contribuinte)	
Valor	Valores repassados ao INSS, referentes à contribuição, juros, multas e atualizações monetárias		(GPS)	