TÉRCIO DE MORAIS SAMPAIO SILVA

EXTRAÇÃO DE INFORMAÇÃO PARA BUSCA SEMÂNTICA NA WEB BASEADA EM ONTOLOGIAS

FLORIANÓPOLIS
2003

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EXTRAÇÃO DE INFORMAÇÃO PARA BUSCA SEMÂNTICA NA WEB BASEADA EM ONTOLOGIAS

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

TÉRCIO DE MORAIS SAMPAIO SILVA

Florianópolis, julho de 2003.

EXTRAÇÃO DE INFORMAÇÃO PARA BUSCA SEMÂNTICA NA WEB BASEADA EM ONTOLOGIAS

Tércio de Morais Sampaio Silva

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

Guilherme Bittencourt, Dr.
Orientador

Evandro de Barros Costa, Dr.
Co-orientador

Edson Roberto de Pierri, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Guilherme Bittencourt, Dr.
Presidente

João Bosco Mangueira Sobral, Dr.

Marcelo Ricardo Stemmer, Dr.

Renata Wassermann, Dra.

Dedico este trabalho a quem mais se esforçou na sua realização: Eu, sem demagogia nem hipocrisia...

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus e meus pais que me apoiaram incondicionalmente durante minha vida. Eles são responsáveis diretos por esta conquista. A minha tão querida irmã, uma amiga muito importante na minha vida. As minhas tias que me incentivaram tanto quanto meus pais. À "dona Candinha" e "seu Nane" (*in memorian*) que faziam a alegria de minhas férias na infância e certamente estão orgulhosos de seu neto.

Quero agradecer as pessoas que direta ou indiretamente participaram desta etapa da minha vida:

Aos meus orientadores Guilherme Bittencourt e Evandro de Barros Costa pela orientação científica e extra-científica e ao meu "pseudo-orientador" Fred. Hoje posso considerá-los grandes amigos.

Aos amigos que ficaram na terra natal que também me incentivaram nesta jornada e que hoje esperam meu retorno ao "paraíso das águas", especialmente à Ana Paula e Rosário que, de certa forma, sacrificaram algo pela realização deste sonho. E aos amigos que conquistei aqui em Floripa. Posso dizer que são eles:

- Rafael que sempre estava disposto ao trabalho resolvendo os pepinos de programação;
- Colegas do DAS com quem convivi e tive bons momentos. Agradeço especialmente à "turma da magrinhagem" (destaque para a diretoria) pelas festas, trilhas, churrascos ...;
- Aos amigos que foram surgindo no decorrer destes dois anos e meio e se tornaram parte permanente da minha vida, inclusive aqueles que surgiram de última hora;
- A fina flor da sociedade paraense e amapaense que promoveram profundas discussões filosóficas sobre a malandragem e a boemia;
- Os amigos nordestinos e apreciadores de tal cultura que promoveram reuniões gastronômicas, onde o tema quase sempre era a culinária nordestina.

Agradeço também a FAPEAL, fundação de amparo a pesquisa do estado de Alagoas que prestou o suporte finaceiro a este curso.

Preferi não citar nomes desnecessariamente nestes agradecimentos. Acredito que quem o lê irá ver seu nome nas entrelinhas. Em suma, agradeço a você. Feliz Natal.

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

EXTRAÇÃO DE INFORMAÇÃO PARA BUSCA SEMÂNTICA NA WEB BASEADA EM ONTOLOGIAS

Tércio de Morais Sampaio Silva

julho/2003

Orientador: Guilherme Bittencourt
Co-orientador: Evandro de Barros Costa
Área de Concentração: Controle, Automação e Informática Industrial
Palavras-chave: Inteligência Artificial, Extração de Informação e Ontologias
Número de Páginas: xiii + 79

Sistemas de Recuperação de Informação (RI) prestam um papel fundamental na busca por páginas na Web. Entretanto, os resultados oferecidos por estes sistemas são pouco precisos, trazendo muitas informações que não condizem com o interesse do usuário. Isto ocorre devido à falta de semântica nas páginas da Web e nos critérios de busca adotados pelos sistemas de RI. Neste trabalho propomos um sistema de Extração de Informação (EI) baseado em ontologias. O objetivo é extrair informações de páginas previamente classificadas semanticamente pelo sistema MASTER-Web que é um sistema multiagente cognitivo para recuperação, classificação e extração de informação na Web. Ontologias são empregadas como formalismo de representação de conhecimento e permitem que o conhecimento seja discriminado em três tipos: conhecimento do domínio, conhecimento sobre a página Web e conhecimento sobre a informação a ser extraída. Regras de produção são usadas como representação do conhecimento sobre o processo de extração. A informação é tratada como um conjunto formado por dados que são extraídos individualmente e depois combinados de modo que componham uma informação consistente. Estes dois passos definem as duas fases da extração que são a extração individual e a integração. Na primeira fase os dados são extraídos individualmente e na segunda fase, os dados, que de alguma forma se relacionam, são unidos formando a informação. O sistema proposto permite portabilidade e reusabilidade do conhecimento, bem como flexibilidade na representação e manutenção do conhecimento sobre a extração. Experimentos foram feitos com o sistema visando avaliá-lo. Para validar os experimentos, os resultados obtidos foram confrontados com os resultados de um outro sistema de EI obtendo resultados bastante satisfatórios.

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

ONTOLOGY-BASED INFORMATION EXTRACTION FOR SEMANTIC SEARCHING ON WEB

Tércio de Morais Sampaio Silva

july/2003

Advisor: Guilherme Bittencourt
Co-advisor: Evandro de Barros Costa
Area of Concentration: Control, Automation and Industrial Computing
Key words: Artificial Intelligence, Information Extraction, Ontologies
Number of Pages: xiii + 79

Information Retrieval (IR) systems play a fundamental role in the search for pages in the Web. However, the results offered for these systems have a low precision, bringing many information that don't correspond to the interest of the user. This occurs due to the lack of semantics in the pages of the Web and to the searching criteria of searches used by the IR systems. This work proposes a system of Informatin Extration (IE) based on ontologies. The aim is to extract semantic information from pages previously semantically classified by the MASTER-Web system, a cognitive multiagent system for retrieval, classification and information extraction from the Web. Ontologies is used as knowledge representation formalism.They allow to decriminate the knowledge in three types of ontologies: knowledge of the domain, knowledge about the Web pages and knowledge about the information to be extracted . Production Rules are used as representation of the knowledge on the extration process. The information is treated as a set composed by combined data that is extracted individually and then composed into consistent information. Both steps define the two phases of the extration that are the individual extration and the integration. In the first phase data is extracted individually and in the second phase, the relationship among the data is verified in order to try forming the information. This system allows that the Knowledge used in the extraction be portable and reusable among some class of pages, as well as flexibile in the representation and maintenance of this knowledge about the extration. some experiments was played with the system in order to evaluate it. To validate the experiments, the obtained results have been compaired with the results a similar EI system, with satisfactory results.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O surgimento da Internet, em particular a Web (*Web Wide Word* ou, simplesmente, WWW), trouxe um crescimento exponencial na disposição de informações que acasionou no fenômeno chamado de "sobrecarga de informação" (do inglês, *information overload*) que trouxe problemas de gerenciamento de informação como localizar páginas na Web e de como usufruir de informações realmente relevantes ao interesse de quem as procura. De fato, a Web comporta um grande volume de informações distribuído que não segue critérios de organização quanto a sua estrutura e localização.

A Web surgiu com o objetivo de apresentar textos e imagens estáticas com ligações entre os documentos que permitisse a navegação por entre eles. Através do protocolo HTTP (*Hyper Text Transfering Protocol*) e a linguagem HTML (*Hyper Text Markup Language*) foi possível padronizar a forma de recuperação e apresentação de documentos [Decker et al., 2000]. Entretanto, as páginas traziam informações pré-definidas, pois as páginas eram construídas manualmente, o que não provia interação *ad hoc* com o usuário. Esta geração é chamada de *Web estática* e de acordo com seu crescimento surgiram problemas como os citados abaixo:

1. A informação estava disponível na Web, mas não havia meios que auxiliassem o usuário a encontrar a informação, se não através da divulgação por parte do servidor da informação[1] da página. Deduzir a provável localização da página era uma questão de sorte;

2. As informações dispostas em páginas nem sempre são as que o usuário está procurando. O usuário é então obrigado a depender de âncoras disponíveis nas páginas que já conhece;

3. A interação entre o usuário e o servidor da informação era num sentido único: do servidor da informação para o usuário, afinal as páginas estáticas não permitem uma interação direta entre os dois;

[1] Neste capítulo o termo "servidor da informação" ou simplesmente "servidor" representa a entidade provedora da informação, seja ela um simples autor, instituição, jornal, etc.

4. A própria natureza da informação que permite uma diversidade de contextos em que ela pode ser abordada. A linguagem HTML não provê recursos para definir o contexto semântico da informação apresentada.

Visando a solução destes problemas surgiram aplicações voltadas para Web que permitem a interação entre o usuário e o servidor permitindo que a informação fosse personalizada segundo os interesses do usuário. As páginas são construídas dinamicamente pelas aplicações de acordo com as informações enviadas pelo usuário através de formulários. Esta é a atual geração chamada de *Web dinâmica*.

Outra contribuição desta geração foi a dos mecanismos de Recuperação de Informação (RI) que localizam páginas na Web e indexam seu endereço baseando-se na freqüência de palavras-chave, frases, metadados, entre outros. Entretanto, as páginas indexadas pelos sistemas de RI não recebem nenhuma conotação semântica. Sistemas de RI desconhecem a respeito do conteúdo semântico da página indexada abrangendo assuntos diversos indiscriminadamente, sem levar em conta questões como sinonimia e polimorfismo das palavras, por exemplo. Questões estas que estão estritamente relacionadas a restrição de domínios. Como conseqüência, a RI peca quanto a precisão de seus resultados.

Como pode-se observar, apesar da evolução, a informação continuou sendo apresentada sem contexto semântico, seja por páginas estáticas ou dinâmicas. O contexto da informação é fundamental para o entendimento sobre uma informação. O que, sob a ótica do ser humano, é relativamente trivial definir, para sistemas computacionais não pode se dizer o mesmo. É necessário que um sistema disponha de um conhecimento prévio bem definido a respeito do domínio da informação (ou problema) que está tratando.

Como solução à falta de semântica da Web, grupos de pesquisa propuseram linguagens de representação de conhecimento (XML – *eXtensible Markup Language*, RDF – *Resource Description Framework*, etc) como ferramentas padrões para adição de contexto às páginas. A rede semântica (do inlgês, *Semantic Web*), a terceira geração da Web, tem como objetivo adicionar semântica às páginas da Web através da definição de conceitos atributos, relações, etc. O uso destas linguagens ainda é restrito na Web por ser uma idéia recente, abrangendo domínios restritos como o de comércio eletrônico, por exemplo. Durante um bom tempo muitas informações dispostas em páginas na Web continuarão sem contexto semântico. Resolver o problema da semântica na Web equivale em resolver os problemas de senso comum e de Processamento de Linguagem Natural (PLN) [Freitas, 2002]. Considerando este fato, os engenhos de busca ainda têm um papel fundamental na busca pela informação.

Baseado no que foi dito acima, a Extração de Informação (EI) surgiu no contexto da Web como uma alternativa ao aprimoramento nos resultados oferecidos pela RI. Sua proposta é extrair informações relevantes de um texto segundo algum contexto.

1.1 Extração de Informação na Web

A EI parte do princípio de que algumas páginas da Web que tratam de assuntos mais específicos tendem a apresentar regularidade quanto a formatação, estrutura e conteúdo podendo ser agrupadas formando *classes de páginas*, por exemplo, páginas de cinema, classificados ou eventos científicos. A EI extrai informações relevantes podendo tanto classificar uma página segundo um contexto de domínio como também extrair informações relevantes a este contexto estruturando as informações contidas na página e armazenando-as em bases de dados.

O fato de já ter pré-definido um domínio de assunto em que atua, permite uma definição semântica mais precisa da informação extraída. Por outro lado, os sistemas de EI estão limitados a domínios de assuntos conhecidos por eles. Tipicamente, estes domínios são pequenos criando classes de extratores muito específicos, como classificados [Embley et al., 1999b] e obituários [Embley et al., 1999a]. Esta desvantagem é acentuada pelo esforço dispendioso na construção e manutenção do conhecimento, diminuindo sensivelmente características do sistema como flexibilidade e reusabilidade, por exemplo. Sistemas que utilizam conhecimento de forma declarativa [Riloff, 1993, Wee et al., 1998, Laender et al., 2000, Freitas, 2002] ganham em flexibilidade e portabilidade por separarem o conhecimento do processo de extração. O uso de conhecimento declarativo é enriquecido quando este é representado por ontologias e quadros (do inglês, frames) [Minsky, 1975]. Apesar disto, o potencial oferecido por ontologias ainda não foi totalmente explorado como, por exemplo, a existência de informações comuns a mais de uma classe de páginas ou âncoras que apontam para páginas pertencente a outra classe. Estes dois exemplos definem relacionamentos entre classes de páginas que não são considerados pelos atuais sistemas de EI. Por exemplo, páginas de pesquisadores geralmente contêm âncoras para instituições de pesquisas. Já esta última é formada por pesquisadores podendo suas páginas apresentar dados sobre eles, como artigos publicados, que, por sua vez, peretenceria a uma outra classe.

Os sistemas de EI atuais tratam classes específicas sem considerar que estas classes de páginas se inter-relacionam através de âncoras ou mesmo por informações comuns às classes, formando conjuntos de classes que chamamos de agrupamentos (do inglês, *cluster*).

Em [Freitas, 2002] é proposto o MASTER-Web, um sistema de EI baseado na abordagem multiagente para classificação de páginas da Web segundo um contexto dentro de um dado domínio. O sistema utiliza ontologias e quadros para representar o conhecimento. O uso de ontologias adiciona características como portabilidade e flexibilidade na construção e manutenção do conhecimento. Além disto, o conhecimento é dividido em três partes: conhecimento sobre o domínio, conhecimento sobre a Web e conhecimento sobre a extração. Desta forma, é possível utilizar conhecimento sobre algum domínio que esteja disponível, evitando o labore da reconstrução. Por outro lado, conhecimentos construídos para fins de extração podem ser disponibilizados para outros fins, como é o caso da ontologia do domínio da ciência desenvolvida para este trabalho que está disponível no sítio do projeto Protégé [Freitas, 2001].

1.2 Contribuições deste trabalho

Este trabalho propõe a extração de informação no MASTER-Web com o objetivo de estruturar informações relevantes extraídas de páginas da Web classificadas pelo sistema e armazená-las em uma base de dados. Atualmente, os dados extraídos são utilizados para classificar páginas. A tarefa de EI se baseia em técnicas de extração dos *wrappers* e utiliza o conhecimento declarativo representado por ontologias.

A EI no MASTER-Web visa prover informações estruturadas e armazenadas em base de dados, oferecendo a usuários humanos e agentes, através de um serviço mediador [Freitas e Bittencourt, 2002], resultados de consultas menos ruidosos, reduzindo esforços do usuário no processo de filtragem de páginas de seu interesse.

A tarefa de EI proposta adiciona mais funcionalidade ao sistema MASTER-Web provendo, além da classificação, um conjunto de informações estruturadas que contextualizam a página segundo seu tema e garantem maior legibilidade da informação quando esta é utilizada por sistemas baseados em conhecimento. Além disto, o uso de ontologias permite uma estrutura clara para o conhecimento provendo diversos níveis de visão e inferência.

A contribuição mais significativa deste trabalho está relacionada à portabilidade e reusabilidade de regras de extração e das ontologias nele utilizadas. Isto permite que o sistema de EI seja portável entre vários domínios, necessitando de poucas modificações em nível de implementação. A reusabilidade se refere à aplicação de ontologias já desenvolvidas e disponíveis para a composição de outras ontologias. Por exemplo, especializar o conhecimento atual através do relacionamento de herança entre as classes da ontologia.

Outra contribuição importante é quanto a flexibilidade no processo de criação da representação do conhecimento, entrada de conhecimento e sua manutenção. O uso de ontologias permite que o conhecimento seja discriminado quanto ao seu papel no domínio no qual está inserido.

1.3 Organização do documento

Este documento está distribuído em sete capítulos assim organizados: *ontologias*, *extração de informação*, *MASTER-Web*, *extração de informação no MASTER-Web*, *estudo de caso e resultados* e *conclusão e trabalhos futuros*.

No capitulo dois discorremos sobre ontologias de um modo geral. Inicialmente falamos sobre os conceitos no campo da filosofia e da computação e sobre sua evolução durante as últimas décadas no campo da ciência da computação. Também são comentados os tipos de ontologias, suas aplicações,

linguagens de representação, ferramentas de construção de ontologias. O capítulo é finalizado com as contribuições de ontologias para a EI.

No capitulo de EI (capítulo três) definimos o que é a tarefa de extração de infomação e a comparamos com a recuperação de informação. Em seguida, as abordagens sobre as técnicas de extração são discutidas, bem como as formas de uso do conhecimento. Baseando-se nas abordagens discutidas é apresentada uma arquitetura genérica para um sistema de EI, onde cada etapa é discutida com mais detalhes. O capítulo segue com a análise comparativa de diversos trabalhos da área de EI e apresentamos brevemente nossa proposta de EI.

O sistema MASTER-Web é apresentado no capítulo quatro. Neste são apresentadas as visões da Web proposta pelo trabalho [Freitas, 2002] e utilizada para classificar conteúdo das páginas da Web. A arquitetura do sistema, baseada em multiagentes, é descrita, bem como a arquitetura de um agente do sistema. Na seção final justificamos a adição da tarefa de extração ao MASTER-Web.

A proposta do trabalho é apresentada no capítulo cinco, onde propomos uma nova arquitetura para um agente do MASTER-Web. Esta arquitetura é descrita e também as ontologias utilizadas no processo. As etapas do processo de extração são descritas. O capítulo é finalizado com os benefícios trazidos pela proposta.

O capítulo seis traz um estudo de caso onde o sistema é submetido a extrair infomações contidas em páginas de eventos científicos. São apresentados os requisitos do sistema, o conhecimento utilizado para extração no domínio de eventos científicos e exemplos de como se comportou cada etapa da extração. Para finalizar, os resultados da extração são apresentados e discutidos

O capítulo final traz as conclusões sobre o trabalho, discutindo suas contribuições tanto para o sistema MASTER-Web quanto para a área de EI. Trabalhos futuros também são propostos objetivando o enriquecimento da pesquisa.

Capítulo 2

Ontologias

Da Filosofia aos sistemas computacionais. Ontologias vêm ganhando espaço não apenas no campo da Inteligência Artificial (IA) como também nas áreas de Bancos de Dados e Engenharia de Software, deixando ser ser apenas um assunto para a Filosofia, Ciência da Informação, Lingüística e outras áreas das ciências humanas. Neste capítulo discutimos o que é ontologia, sua contribuição para a Ciência da Computação em especial para Inteligência Artificial. Também apresentamos algumas das principais linguagens de representação de conhecimento e ferramentas de edição. Por fim, apresentamos a ontologia do domínio científico utilizada em nosso estudo de caso.

2.1 Introdução

Nas últimas décadas ontologias vêm ganhando grande ênfase no domínio da Ciência da Informação e Inteligência Artificial como meio de representar, compartilhar e reusar conhecimento de forma legível para um computador. De fato, o que para nós, humanos, é simples de entender pode não ser para uma máquina.

Até então ontologias eram um tema pouco estudado pela comunidade da Ciência da Computação, estando mais relacionado à Filosofia. A partir do momento em que a IA focou esforços em simular o raciocínio humano e criar sistemas que "sabem" percebeu-se a necessidade de representar o conhecimento de maneira legível para a máquina. Sistemas especialistas são um exemplo desta necessidade: uma base de conhecimento construída por especialistas e submetida à inferência através de mecanismos de raciocínio automático gerando mais conhecimento a respeito do problema tratado.

Não apenas na IA como também nas áreas de Bancos de Dados e Engenharia de Software, por exemplo, notou-se que a separação entre o conhecimento sobre o problema e a estratégia de solução deste traria benefícios à construção e manutenção de um sistema de software, seja ele um sistema baseado em conhecimento, um modelo lógico-relacional ou a modelagem de um sistema de software.

Flexibilidade, portabilidade, reusabilidade, consistência semântica são vantagens trazidas pela abordagem do conhecimento declarativo. Por outro lado, desenvolvedores e pesquisadores de sistemas computacionais baseados apenas no procedimento de solução do problema tratam o conhecimento sobre o domínio do problema em questão apenas como "coadjuvante", dando ênfase à performance espacial e temporal. Apesar da divergência entre estas duas abordagens, cada uma delas tem seu espaço de aplicação.

Representações declarativas do conhecimento foram ganhando força no decorrer da evolução dos sistemas de informação, consolidando-se com o advento da Internet que disponibiliza um grande volume de informações heterogêneas e distribuídas. Junto a este volume, há a necessidade de integrar informações heterogêneas – um problema já conhecido pela comunidade de sistemas de bancos de dados [Jardine, 1997]. Outro fator importante foi o fato de que hardware e software tornaram-se mais sofisticados, o que permitiu que os desenvolvedores de software e pesquisadores se preocupassem mais sobre aspectos dos dados que seus sistemas operavam ao invés de aspectos procedimentais e funcionais [Smith e Welty, 2001].

2.2 O que são Ontologias?

No contexto filosófico ontologia é parte da ciência que estuda o ser e seus relacionamentos. Esta definição é bastante ampla permitindo diversas interpretações mais específicas de acordo com a área de aplicação, seja ela sistemas de informação, lingüística ou ciência da informação, por exemplo.

John Sowa [Sowa, 1984] se refere à ontologia de um mundo como "um catálogo de tudo que constitui tal mundo, como tudo é colocado junto e como funciona". Esta definição especializa mais o entendimento de ontologia no contexto dos sistemas de informação, em particular sistemas baseados em conhecimento, aproximando a ontologia filosófica e a ontologia formal ou matemática. Muitos outros significados foram apresentados com o passar dos tempos [van Heijst et al., 1997]. Um exemplo apresentado em [Alexander et al., 1986] mostra o quanto a idéia sobre ontologias foi enriquecida de acordo com as necessidades percebidas para modelagem do conhecimento. A figura 2.1 ilustra o exemplo onde são mostrados os diversos tipos de ontologias que variam de acordo com a complexidade.

O exemplo mais simples é o de um vocabulário controlado. Catálogos são exemplos de vocabulários onde os termos existentes no domínio são listados. Muitos não consideram este exemplo como uma ontologia, pois não provê semântica aos termos. Já glossários, além de uma lista de termos traz também uma lista de significados para estes termos. Isto provê uma semântica aos termos desde que seja legível para o leitor. O exemplo seguinte é o *Thesauri*. Thesauri adiciona mais semântica aos termos relacionando-os a termos sinônimos. Este relacionamento permite que os termos se tornem menos ambíguos, permitindo que sistemas computacionais possam interpretar informações em um conjunto limitado de casos.

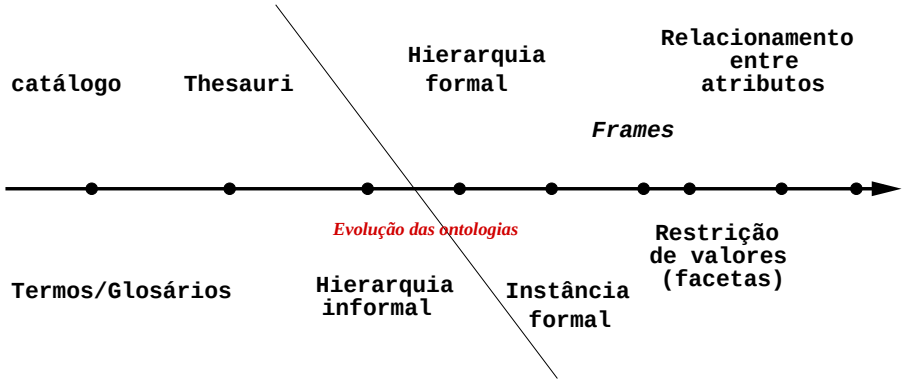


Figura 2.1: Evolução das ontologias, segundo [McGuinness, 2002].

Engenhos de busca como o *Yahoo* [Yahoo, 2003] introduziram uma noção de hierarquia entre os termos com a utilização de diretórios e sub-diretórios. Ainda que informalmente o uso de diretórios seja análogo ao uso de ontologias, onde uma noção de generalização e especialização é apresentada, esta analogia é considerada inconsistente. De fato, a idéia de generalização e especialização se baseia na analogia de classes onde uma instância de uma classe específica é necessariamente instância da classe mais genérica. Deborah McGuinness [McGuinness, 2002] usa como exemplo os termos "vestimenta", "mulher", "acessório" e "vestido", onde vestimenta é uma categoria mais genérica que contém a categoria mulher que por sua vez contém acessórios e vestidos. Neste tipo de relacionamento não é possível afirmar que instâncias da categoria mulher são também instâncias da categoria "vestimenta" (considerando categorias como classes). Já instâncias da categoria "vestido" também são instâncias da categoria vestimentas, mas não são instâncias da categoria "mulher".

A formalização do conceito de hierarquia é empregada em ontologias. Neste nível a relação hierárquica entre uma superclasse A e uma subclasse B segue a seguinte regra: dada uma instância b de uma subclasse B é possível afirmar que b também é instância da superclasse A. Como exemplo, consideremos as instâncias das classes humano e macaco que também são instâncias da classe mamífero, podemos afirmar que todo humano é um mamífero e todo macaco também é um mamífero. Já a recíproca não é verdadeira, onde nem todo mamífero é um macaco e nem todo mamífero é um humano. Nesta categoria de ontologia ainda não é possível agregar atributos às instâncias.

Num nível de complexidade maior, as classes são providas de atributos que especificam as características das instâncias. Por exemplo a classe pessoa tem atributos como sexo, altura e peso. Esta categoria é baseada no formalismo de representação conhecido por *quadros* [Minsky, 1975]. A restrição de valores foi em seguida inserida nos atributos das classes de uma ontologia. Agora é possível restringir valores atribuídos a uma instância evitando inconsistência na informação Por exemplo, considerando que a classe pessoa tenha um atributo que represente a idade, podemos retringir os valores deste atributo considerando que a idade de uma pessoa é sempre um inteiro natural menor que 200.

Outras propriedades foram adicionadas na medida em que se precisou expressar entidades com

mais propriedade. Por exemplo, a necessidade de preencher atributos com valores baseados nos valores de outros atributos. Neste caso, estamos considerando que existirão relacionamentos entre as classes e entre os atributos de uma classe. Por exemplo, se considerarmos uma classe de equação de segundo grau com atributos a, b e c que satisfazem a equação $ax^2 + bx + c = 0$, os atributos x' e x", que representam as raízes da equação, têm uma relação de dependência com aqueles três atributos.

Todos os exemplos apresentados anteriormente se adequam à idéia de ontologia. Uma definição proposta por Gruber [Gruber, 1993] expressa bem o que é ontologia, abrangendo todos as categorias aqui citadas: "Ontologia é uma especificação explícita de uma conceituação".

2.3 Tipos de Ontologia

van Heijst *et alli* sugerem em [van Heijst et al., 1997] que ontologias podem ser classificadas segundo duas dimensões. A primeira leva em consideração o "tamanho" e a estrutura da conceituação. A segunda dimensão trata sobre o assunto da conceituação. Consideraremos neste trabalho apenas a segunda dimensão, que expressa uma maior afinidade com as ontologias desenvolvidas.

Na dimensão em questão, quatro categorias de ontologias são definidas: ontologias de domínio, genéricas, de aplicação e de representação.

- Ontologias de Domínio expressam conceituações que são particulares a um tipo de domínio, por exemplo, eletrônica, medicina, mecânica, domínio digital. Em nosso trabalho representamos o domínio científico através deste tipo de ontologia;
- Ontologias de Aplicação contêm todas as definições necessárias para modelar o conhecimento específico de uma aplicação. Geralmente são compostas por conceitos contidos nas ontologias de domínio e genéricas. Por terem um propósito tão específico, ontologias de aplicação geralmente não são reusáveis;
- Ontologias Genéricas são similares às Ontologias de Domínio, mas seus conceitos são aplicáveis a vários campos, por exemplo, estados e processos. Geralmente os conceitos contidos nas ontologias de domínio são especializações dos conceitos das ontologias genéricas; e
- Ontologias de Representação dão suporte aos formalismos de representação. Estas ontologias não influênciam no domínio modelado. Elas provêem uma estrutura (*framework*) representacional para descrever as ontologias genéricas, de domínio e de aplicação.

2.4 Ontologias baseadas em Quadros

Quadros (do inglês *Frames*) é um formalismo de representação de conhecimento introduzido por Marvin Minsky [Minsky, 1975]. Em sua proposta original, Mynski sugere aplicar o formalismo em análise de cenas, modelagem de pecepção visual e compreensão de linguagem natural.

Em geral, um quadro consiste em um conjunto de atributos que, através de seus valores, descrevemos características do objeto representado pelo quadro . Os valores atribuídos aos atributos podem ser outros quadros formando uma relação de dependência ou composição entre eles. Os atributos possuem propriedades como restrições de tipos e quantidades de valores, conhecidas como facetas onde a estrutura do valor de um atributo é pré-definida. Outro relacionamento possível é o de generalização/especialização.

As características do formalismo de quadros adicionam mais recursos a uma linguagem de representação de ontologias. De fato, linguagens baseadas neste formalismo expressam conceitos contidos numa ontologia com mais propriedade. Através de quadros é possível adicionar características como atributos de um objeto e seus valores apropriados. Mais: a descrição de relacionamentos de hierearquia, dependência e composição também são possíveis. Estes relacionamentos compõem a topologia entre conceitos de uma ontologia. A figura 2.2 [Bittencourt, 1998] exemplifica uma ontologia baseada em quadros.

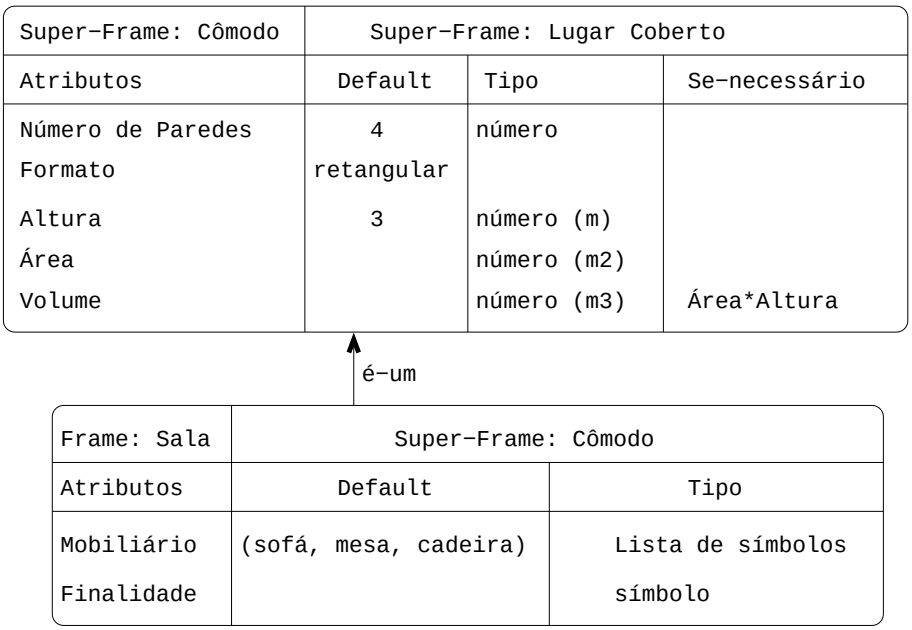


Figura 2.2: Exemplo de ontologia baseada em quadros [Bittencourt, 1998].

Nesta "amostra" de ontologia dois conceitos são descritos através de quadros. O primeiro quadro descreve o conceito de cômodo que é composto de atributos como número de paredes, formato, área, etc. Cada um destes atributos carrega facetas que "amarram" um determinado tipo de valor ao atributo,

por exemplo, o número de paredes deverá ser um tipo numérico, já o atributo `Formato` segue o tipo símbolo que aqui assume valores limitados e definidos previamente (retângulo, quadrado, losango, etc). Da mesma forma é definido outro quadro chamado `Sala`. Observe que sala é um tipo de `Cômodo` caracterizando uma especialização. Este relacionamento é expresso neste exemplo pela propriedade *Super-Frame* permitindo que as características e propriedades de `cômodo` sejam herdadas por `Sala`. Outra característica exemplificada aqui é o relacionamento entre atributos de um quadro. No quadro `Cômodo`, o atributo *Volume* depende dos valores dos atributos `Área` e `Altura`.

2.5 Aplicações de Ontologias

Ontologias estão sendo aplicadas em várias áreas onde o conhecimento explícito é desejável, por exemplo aplicações que envolvem Sistemas Multiagentes (SMA), Sistemas de Apoio à Decisão, análise de imagens, etc. A Internet, com suas características de heterogeneidade, distribuição e desestruturação, oferece à ontologias uma missão ousada: manipular informações contidas na Internet. Neste contexto podemos classificar o emprego de ontologias em três grupos segundo sua missão: (1) troca de informação; (2) estruturação da informação; e (3) busca da informação.

2.5.1 Troca de Informação

A troca de informação é bastante utilizada por SMA em suas interações. Para que isto seja possível, os agentes de uma sociedade devem compartilhar do mesmo conhecimento, o que pode ser obtido através de ontologias. Como exemplo, o MASTER-Web [Freitas, 2002] utiliza ontologias para representar o conhecimento tratado pelos seus agentes (mais detalhes no capítulo 4). Além disto, a vantagem oferecida pelas ontologias de permitir a especialização de um conhecimento também é aplicada em agentes. Cada agente detém o conhecimento geral sobre o domínio mais o conhecimento sobre o seu papel a ser desempenhado na sociedade.

2.5.2 Estruturação da Informação

A informação disponibilizada na Internet tinha como único objetivo alcançar as pessoas. Esta foi tida como a primeira geração da Web. Neste cenário, não havia preocupação em estruturar a informação contida nas páginas já que ela era legível para as pessoas. A segunda geração foi a Web dinâmica que agregou sistemas de acesso à bases de dados tornando a informação flexível ao usuário, ou seja, agora o usuário humano interage de modo a ter controle maior sobre as informações de seu interesse. Com o surgimento dos agentes (termo usado largamente na Web para definir aplicações que interagem com pessoas ou outras aplicações na Web) percebeu-se que, além da informação léxica era necessário compreender o contexto da informação. A terceira geração da Web visa adicionar

semântica às informações – a Web Semântica (do inglês *Semantic Web*). Ela tem como motivação a estruturação de informações segundo um contexto legível para agentes inteligentes, facilitando a troca de informações entre eles. Esta nova geração está trazendo benefícios a áreas como o comércio eletrônico (*business-to-business – b2b*). Na seção a seguir apresentamos linguagens de representação de conhecimento desenvolvidas visando suprir as necessidades da Web semântica. Apesar das vantagens, este projeto tem muitos desafios pela frente, onde alguns deles são mostrado em 2.5.3:

2.5.3 Recuperação de Informação

Apesar da proposta da Web semântica de compartilhar informação estruturada da Web, ela esbarra em alguns problemas:

- Muitas páginas na Web são desenvolvidas por pessoas não especializadas. De fato, o processo de dispobilizar uma página na Web é simples e rápido e não exige do autor conhecimentos especiais;
- Não existem ferramentas de estruturação automática da informação que abranjam todos os domínios e assim possam ser usadas por qualquer editor de páginas HTML;
- As páginas desenvolvidas ainda são direcionadas às pessoas. Mesmo que o autor disponha de conhecimento sobre Web semântica, será ele quem decidirá dispor ou não informações neste formato.
- A Web semântica exige coerência do conhecimento disponibilizado. Frequentemente, as páginas contém conteúdo vago e ambíguo não permitindo uma boa estruturação da informação;
- Páginas publicadas dificilmente serão alteradas para prover informação semântica;

Estes problemas dão espaço por um longo tempo aos sistemas de busca e recuperação de informação. Os engenhos de busca tradicionais não empregam totalmente, ou simplesmente não empregam, ontologias. Yahoo é um exemplo de uso parcial de ontologias, com sua estrutura montada em diretórios e sub-diretórios, porém não segue necessariamente uma hieraquia de classes. Mas engenhos especializados em domínios como *brint* (http://www.brint.com) adotam ontologias como representação do conhecimento. Nele o conhecimento é organizado em categorias bem definidas. O projeto MASTER-Web tem como objetivo buscar páginas no domínio de eventos científicos, classificá-las segundo seu conteúdo e extrair informações relevantes a sua classe. A extração da informação resulta num conjunto de informações estruturadas úteis para buscas mais precisas ou até provê semântica às páginas.

2.6 Linguagens de Representação

Até aqui foram discutidos conceitos, aplicações, vantagens e desvantagens sobre ontologias sem se referir à maneira como toda essa "especificação explícita de conceituações" será representada de modo legível em sistemas computacionais.

Ontologias, quando aplicadas em sistemas computacionais, precisam de uma linguagem de representação. Várias linguagens têm sido desenvolvidas nos últimos anos. Estas linguagens podem ser agrupadas segundo seu objetivo [Corcho e Pérez, 2000]: linguagens de representação de ontologias tradicionais e linguagens de representação de ontologias baseadas na Web. Esta última se desenvolveu juntamente com a proposta da terceira geração da Web – Web Semântica.

2.6.1 Linguagens de Representação de Ontologias Tradicionais

As ontologias tradicionais (termo usado em [Corcho e Pérez, 2000]) são tidas como aquelas que já existiam quando a terceira geração da Web surgiu. As linguagens mais conhecidas são a Ontolingua, LOOM, OCML, o formalismo FLogic e o protocolo OKBC.

Ontolingua [Farquhar et al., 1996] é uma linguagem desenvolvida para dar suporte a projetos e especificações de ontologias com uma semântica lógica clara. Ontolingua é baseada em KIF (***K**nowledge **I**nterchange **F**ormat*) que é uma linguagem formal para troca de conhecimento entre sistemas computacionais muito diferentes (isto é, escrito por vários programadores em épocas e linguagens diferentes) [Genesereth e Fikes, 1992]; é baseada em uma ontologia de representação de conhecimento (*Frame Ontology*) que define termos de linguagens baseadas em quadros e orientadas a objetos. A Ontolingua provê suporte explícito para a construção de módulos ontológicos e faz distinção entre uma ontologia de representação e de aplicação. Ontolingua permite a construção de ontologias de três formas: usando expressões do KIF; usando apenas o vocabulário definido em *Frame Ontology*; ou usando as duas formas simultaneamente.

LOOM [MacGregor, 1991] é uma linguagem e um ambiente para construção de aplicações inteligentes. Um sistema de representação do conhecimento provê suporte dedutivo para a parte declarativa da linguagem, ou seja LOOM é baseada na lógica de descrição o que permite integração entre os paradigmas de representação baseada em *frames* e regras de produção. O conhecimento declarativo em LOOM consiste de definições, regras, fatos e regras padrão. Seu mecanismo de inferência dedutiva chama um classificador que utiliza o encadeamento para frente. Um ambiente sucessor de LOOM é o **PowerLoom** que usa uma linguagem de representação lógica expressiva. Seu mecanismo de inferência utiliza encadeamento para trás e para frente.

OCML [Motta, 1998] (***O**peracional **C**onceptual **M**odeling **L**anguage*) é uma linguagem baseada em *frames* que provê mecanismos para expressar termos como relacionamentos, funções, regras,

classes e instâncias. Além disso permite alguns outros recursos como a especificação de módulos de estrtuturas (procedimentos) de controle interativo, seqüencial e condicional.

FLogic [Kifer et al., 1990] é um formalismo baseado em *frames* que descreve de modo claro e declarativo a maioria dos aspectos estruturais do paradigma de linguagens orientadas a objetos e baseadas em *frames*. Objetos, herança, tipos polimórficos, métodos de busca, encapsulamento são exemplos do que FLogic provê. FLogic está para o paradigma da orientação a objetos como cálculo de predicados clássicos está para a programação relacional.

OKBC (***O**pen **K**nowledge **B**ase **C**onnectivy*) é um protocolo de acesso a bases de conhecimento armazenadas em sistemas de representação de conhecimento. OKBC não é uma linguagem, ele complementa linguagens desenvolvidas para dar suporte ao compartilhamento de conhecimento. Há implementações do OKBC para várias linguagens de programação incluindo Java, C e Commom LISP, por exemplo. Outra característica é o acesso remoto a bases de conhecimento. OKBC inclui um modelo de conhecimento de sistemas de representação de conhecimento e também um conjunto de operações baseadas neste modelo. A versão atual do OKBC é orientada a objetos. Métodos numa linguagem orientada a objetos são usados para implementar as operações.

2.6.2 Linguagens de Representação de Ontologias Baseadas na Web

A proposta de atribuir semântica à Web fez com que surgissem novas linguagens de representação do conhecimento. Neste contexto, XML (***EX**tensible **M**arkup **L**anguage*) e RDF(S) (***R**esource **D**escription **F**ramework **S**quema*) são padrões desenvolvidos e sugeridos pelo W3C para desenvolvimento de linguagens ontológicas e construção de ontologias baseadas na Web.

XML [Bray et al., 1997] é uma linguagem simples e flexível derivada da SGML (***S**tandard **G**eneralized **M**arkup **L**anguage*) também desenvolvida pelo W3C. Sua proposta inicial era responder aos desafios de publicações eletrônicas em larga escala. Atualmente desempenha um papel importante no intercâmbio de dados na Web. XML traz vantagens como facilidade para análise (*parsing*) sintática bem definida e legível ao homem. As vantagens de XML como linguagem de especificação de Ontologias são:(1) definição de uma especificação sintática por meio de uma definição de tipo de documento (*Document Type Definition* – DTD); (2) facilidade de interpretação por humanos; (3) pode representar conhecimento distribuído por várias páginas; (4) é definida para suprir a falta de estrutura de páginas; (5) ferramentas são dispobilizadas para análise e manutenção de documentos descritos em XML.

Apesar das vantagens, XML apenas oferece uma especificação de sintaxe. XML não ofere nenhuma característica especial para especificação de ontologias. Para isto, foram desenvolvidas linguagens que extendem as funcionalidades de XML.

RDF [Lassila e Swick, 2003] é desenvolvido pela W3C com o objetivo de descrever recursos na

Web permitindo processamento automático. Este arcabouço provê interoperabilidade entre as aplicações que trocam informações na Web. RDF consiste de três tipos de objetos: recursos, que são todas as coisas descritas pelo RDF; propriedades, que são as características usadas para definir um recurso; e expressões (*statements*) que são os valores atribuídos às propriedades de um recurso específico. RDF não provê mecanismos que definam relacionamentos entre atributos nem entre recursos.

RDFSchema [Amann e Fundulaki, 1999] é uma linguagem declarativa para definição de esquemas em RDF. O modelo de dados do RDF(S) é baseado em *frames* e provê mecanismos para a definição de relacionamentos entre propriedades e recursos. Hierarquia e restrição de tipos podem ser definidas. RDF(S) é amplamente usada como formato de representação em muitas ferramentas e projetos

Baseando-se nestas duas linguagens são definidas outras como XOL, SHOE, OML, OIL e DAML+ OIL na intenção de aumentar os recursos daquelas linguagens. Podemos definir dois grupos para estas linguagens: linguagens baseadas em XML e RDF(S).

2.6.2.1 Linguagens Baseadas em XML

XOL (***X****ML-Based* ***O****ntology Exchange* ***L****anguage*) [Karp et al., 1999] foi desenvolvido pelo *Bio-Ontology Core Group* para a troca de definições ontológicas entre um conjunto de sistemas de software heterogênio em um domínio. Como base para criação do XOL foram usadas a sintaxe da XML e a semântica do OKBC-Lite, confirmando a alta expressividade do OKBC-Lite, um subconjunto do OKBC. Não há ferramentas que ofereçam um ambiente de implementação do XOL, mas ferramentas de edição para documentos XML podem ser usadas.

SHOE (***S****imple* ***H****TML* ***O****ntology* ***E****xtentions*) [Luke e Heflin, 2000] é uma extensão de HTML que permite aos autores de páginas HTML adicionar conhecimento semântico legível para sistemas computacionais. A sintaxe de SHOE foi adaptada à XML permitindo que agentes colham informações sobre páginas HTML aperfeiçoando mecanismos de busca e recuperação de informação

OML/CKML (***O****ntology* ***M****arkup* ***L****anguage*/*Conceptual Knowledge Markup Language*): OML representa estruturas esquemáticas e ontológicas. As estruturas ontológicas incluem classes, relacionamentos objetos e facetas. CKML provê um arcabouço de conhecimento conceitual para representação de informações distribuídas. Juntas, OML e CKML formam um par de linguagens baseadas em lógica de descrição e quadros.

2.6.2.2 Linguagens Baseadas em RDF(S)

As linguagens que se baseiam em RDF(S) propoem extender suas funcionalidades, como inferência sobre o conhecimento, por exemplo.

OIL [Fensel et al., 2000] é uma proposta de representação e inferência para ontologias baseadas na Web que combina a principais primitivas de modelagem de linguagens baseadas em *frames* com serviços de semântica formal e raciocínio providos pela lógica de descrição. OIL coincide em muitos aspectos com RDF-Schema permitindo que agentes que adotem RDF-Schema como linguagem sejam capazes de processar ontologias contruídas com OIL. OIL é composto de camadas que adicionam funcionalidade e complexidade à linguagem. As características acima se referem a camada núcleo (*Core* OIL) da linguagem. Ainda existem as camadas *Standard* e *Instance* que tem como objetivo adicionar mais poder de expressão e maior integração entre os termos de uma ontologia, respectivamente.

DAML+OIL é desenvolvida por um comitê formado pelos Estados Unidos e a União Européia (IST – ***I**nformation **S**ociety **T**echnologies*). DAML (***D**ARPA **A**gent **M**arkup **L**anguage*) e OIL substitui a antiga versão de DAML que também era baseada em RD(S). DAML+OIL é contruído baseando-se na linguagem RDF(S) e tem um rigoroso relacionamento com OIL.

2.7 Ferramentas para Construção de Ontologias

Ferramentas para contrução de ontologias visam minimizar o esfoço de autores durante sua criação. Estas ferramentas devem permitir a construção de ontologias desde a "estaca zero" ou a partir de ontologias reusáveis. Geralmente incluem documentação, importação e exportação de ontologias de diferentes formatos, visualização gráfica, bibliotecas e mecanismos de inferência. Várias ferramentas são disponibilizadas como Apollo, LinkFactory, Ontolingua, Protégé-2000, etc. O objetivo deste trabalho não é avaliar ambientes de edição de ontologias, portanto apenas apresentamos e justificamos a ferramenta escolhida Protégé-2000. Maiores detalhes sobre ferramentas para ontologias podem se encontradas em [Information, 2000].

Protégé-2000 é uma ferramenta desenvolvida pelo departamento de Informática Médica da Universidade de Stanford visando suprir as necessidades de ontologias médicas. No decorrer do tempo Protégé-2000 foi se firmando como ferrramenta para edição de ontologias. Os critérios de escolha de tal ferramenta foram: arquitetura da ferramenta; interoperabilidade; representação de conhecimento; serviços de inferência; e usabilidade.

1. *arquitetura da ferramenta*: visa avaliar a arquitetura do software, como ela pode ter sua funcionalidade extendida e como ontologias podem ser armazenadas. Protégé-2000 é uma ferramenta *standalone* que suporta outros mecanismos externos através de componentes (do inglês, *plugins*) como visualizadores gráficos, mecanismos de inferência, etc. O armazenamento pode ser feito de duas formas: arquivo e SGBD (**S**istema de **G**erenciamento de **B**anco de **D**ados), sendo este último através do JDBC (***J**ava **D**ata **B**ase **C**onectivy*);

2. *interoperabilidade*: trata da interação da ferramenta com outras e importação e da exportação de ontologias em outras linguagens. Protégé interage com outras ferramentas como OKBC e

Jess (***J**ava **E**xpert **S**ystem **S**hell*) que é uma API (***A**pplication **P**rogram **I**nterface*) que provê inferência sobre a ontologia. Também importa as linguagens XML, RDF(S) e XML Schema e exporta nas linguagens XML, RDF(S), XML Schema, DAML+OIL, FLogic, CLIPS, Java e HTML;

3. *representação de conhecimento*: determina o paradigmas de representação no qual a ferramenta se baseia e se ela provê alguma linguagem que suporte axiomas. *Quadros* é o formalismo de representação adotado pela ferramenta. Protégé utiliza PAL (***P**rotégé **A**xiom **L**anguage*) para construir axiomas.

4. *mecanismos de inferência*: verifica se a ferramenta provê algum mecanismo de inferência próprio ou suporta algum externo, se provê verificação de consistência e restrições (facetas – do inglês, *facets*), classificação automática e manipulação de exceções. Protégé-2000 provê PAL como mecanismo de inferência e ainda suporta componentes deste tipo como Jess e FLogic. A maioria das ferramentas provê verificação de consisência e facetas, o que no Protégé-2000 não é diferente. Protégé-2000 não tem classificação automática nem tratamento de exceções;

5. *usabilidade*: as características aqui analisadas são a existência de editor gráfico para criação de taxonomia de conceitos e relacionamentos, habilidade de navegação através de classes e seus relacionamentos, se a ferramenta permite algum tipo de trabalho colaborativo e se provê bibliotecas de ontologias. Protégé-2000 dispõe de recursos gráficos, tanto para criação de taxonomias quanto para navegação através destas taxonomias (por suas classes e relacionamentos). Além disso provê bibliotecas de ontologias. Entretanto, não pemite trabalho colaborativo. Protégé-2000 ainda permite gerar interface gráfica personalizada para entrada do conhecimento de uma ontologia, permitindo que a ontologia seja alimentada por usuários menos experiente.

As características acima apresentadas mostram a viabilidade do editor de ontologias Protégé-2000. As figuras 2.3 e 2.4 mostram as telas de construção e de entrada do conhecimento de uma ontologia, respectivamente.

2.8 A Ontologia do domínio da Ciência

A ontologia do domínio da ciência foi empregada no Projeto MASTER-Web como estudo de caso. Esta ontologia foi construída com base no projeto europeu $(KA)^2$ [Benjamins et al., 1998] utilizando-se a ferramenta Protégé-2000 e está disponível no repositório de ontologias do Protégé. Os conceitos mais usados em nossos experimentos são mostrados na figura 2.5. A classe abstrata `evento` decreve os tipos de eventos gerais com atributos como nome, data inicial e data final. Logo abaixo são descritas suas especializações que em nosso domínio são os eventos cientificos (`Scientific-Event`) que por sua vez têm especializações, como evento científico ao vivo e evento de publicação científica (`Live-Scientific-Event` e `Scientific-Publication-Event`, respectivamente). Os eventos

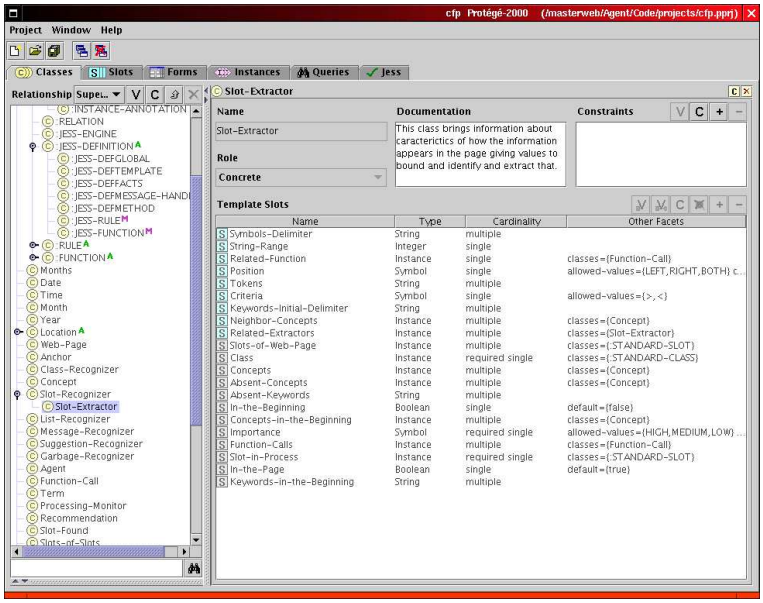


Figura 2.3: Tela de edição de classes e atributos (*slots*).

científicos ao vivo ainda podem ser: um `workshop`, conferência, evento educacional, etc. O mesmo pode ser dito dos eventos de pubicação científica (revista, jornal).

Ontologias genéricas de tempo, locais e turismo foram reusadas visando complementar a ontologia de domínio e de aplicação (esta última é apresentada no capítulo 5).

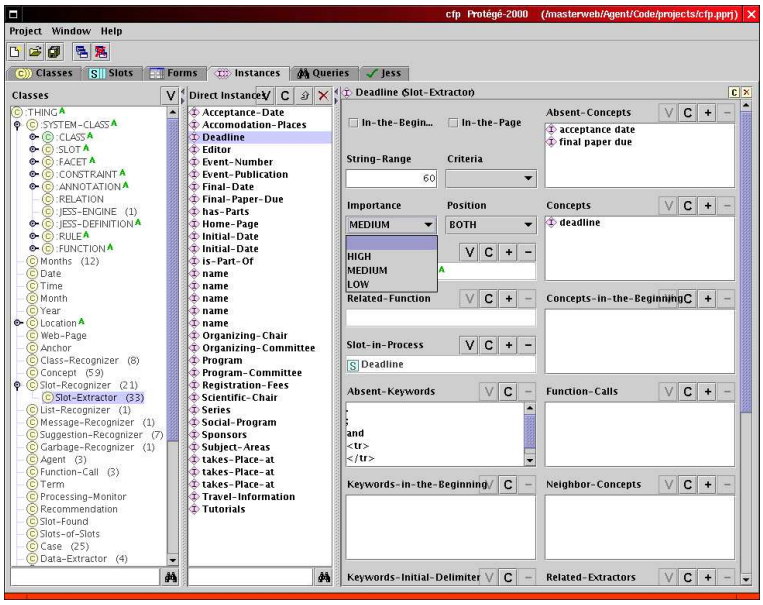


Figura 2.4: Tela de entrada de conhecimento

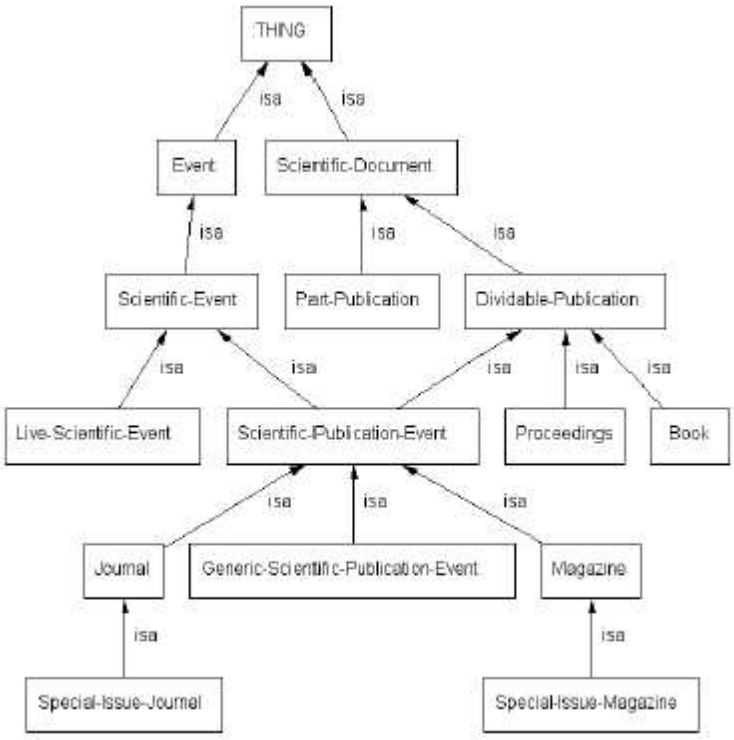


Figura 2.5: Algumas classes e seus relacionamentos da ontologia Ciência.

Capítulo 3

Extração de Informação

A Web Semântica tem desafios que fogem ao domínio técnico de sua proposta, como a intenção do autor de uma página adicionar ou não informação semântica a ela. Este problema, já existente desde a primeira geração da Web, é tratado pelos sistemas de buscas. Sistemas de Recuperação de Informação (RI) tradicionais foram desenvolvidos neste intuito, mas pecam quanto à precisão de seus resultados. A falta de semântica é a principal causa dessa falta. Neste capítulo apresentamos a Extração de Informação (EI) e suas vertentes como alternativa de tratamento deste problema, suas contribuições à RI, técnicas de EI, seu relacionamento com ontologias e trabalhos relacionados a esta área.

3.1 Introdução

A EI identifica dados relevantes a um tema contido em um texto e os extrai convertendo-os para uma estrutura tabular. Esta estrutura tem o objetivo de sumarizar o conteúdo do assunto abordado no documento numa forma legível, tanto para o usuário quanto para uma aplicação.

As primeiras idéias de estruturação da informação em linguagem natural datam da década de 50 sugerida por Zelling Herris [Grishman, 1997]. Mas apenas no final da década de 80 a EI se destacou no meio científico através de desafios lançados pela *Message Understand Conference* (MUC). O desafio constava de entender o conteúdo de mensagens contidas em textos. Algoritmos propostos teriam de ser eficientes no entendimento do significado das mensagens.

No início, técnicas de processamento de linguagem natural (PLN) eram aplicadas no desenvolvimento destes algoritmos. Entretanto, a complexidade algorítmica, a carga de processamento e o fato de não haver no PLN sistemas que analisem textos com estrutura livre indiferente ao nível de complexidade foram fatores para que pesquisadores da área de análise de lingugem natural se voltassem para uma alternativa mais simples, porém também eficiente: a Extração de Informação.

Técnicas tradicionais de PLN têm como objetivo a completa análise do texto. Para um grande volume de textos isto se torna inviável. Ao contrário, a EI tem interesse apenas em partes específicas do texto onde há a possibilidade da existência de informações relevantes. Isto traz vantagens como menor esforço computacional e menor complexidade no desenvolvimento.

A EI adaptou técnicas de PLN visando a análise de um grande volume de documentos. Esta adaptação se tornou muito própria diante de um outro problema abordado pela RI: a indexação da informação na Web.

Sendo a Web um ambiente composto por informações distribuídas e livre de qualquer padrão de formatação, a RI se propõe a indexar suas páginas através de palavras-chave. O objetivo da RI é recuperar um conjunto de documentos relevantes baseando-se em cálculos estatísticos sobre os termos que ocorrem na página. O resultado obtido ainda passa por uma triagem manual feita pelo usuário para selecionar as páginas de seu interesse. A EI se propõe a minimizar o trabalho manual fazendo uma filtragem baseada em correspondência de padrões (do inglês, *pattern macthing*) que identifiquem informações relevantes ao tema abordado na página. Estas duas tecnologias se complementam podendo-se afirmar que RI recupera documentos importantes enquanto que a EI extrai informações relevantes de um documento [Eikvil, 1999].

3.2 Extração de informação vs. recuperação de informação

A RI aplicada na Internet tem como objetivo recuperar páginas e indexá-las baseado-se em dados estatísticos sobre a ocorrência de palavras-chave. Entretanto, alguns problemas da RI podem ser enumerados [Riloff e Lehnert, 1994]:

1. *Palavras sinônimas*: palavras diferentes podem expressar o mesmo significado. Por exemplo, as expressões *data de notificação* e *data de aceitação* expressam o mesmo significado quando tratamos de eventos científicos;

2. *Polissemia*: uma palavra pode ter mais de um significado dependo de seu contexto. Por exemplo a palavra *latex* pode se referir à matéria prima da borracha ou ao software de geração de textos;

3. *Frase*: alguns conceitos são representados por um conjunto de palavras. Estas palavras mudam o significado quando analisadas individualmente. A expressão *submissão de artigos* é um exemplo. As palavras *submissão* e *artigos* têm sentidos diferentes quando são analisadas sozinhas;

4. *Contexto local*: a relevância de uma palavra, varia de acordo com o contexto abordado. Numa busca sobre roubos de banco, por exemplo, a palavra *roubo* é bastante relevante para a busca, pois especifica o tipo de informação que será buscada sobre bancos;

5. *Contexto global*: alguns documentos não contêm palavras ou frases boas para indexação. A relevância do documento depende do contexto da sentença como um todo. Por exemplo, a frase "um homem armado tomou o dinheiro e fugiu", as palavras isoladas não denotam o contexto da frase, entretanto, todas juntas descrevem claramente um roubo.

Os problemas apresentados acima tornam a RI pouco precisa em suas respostas de buscas feitas por usuários. Os resultados de buscas através de RI apresentam duas características: (1) resultados de busca ruidosos, ou seja, muitas páginas que não condizem com o interesse do usuário; (2) muitas páginas perdidas durante a busca ou porque foram indexadas de modo errado ou porque sequer foram indexadas. O exemplo 3.1 mostra um trecho de classificados de uma página que pode ser perdida durante a recuperação:

Exemplo 3.1 *Trecho de um anuncio de venda de carros*

```
'97 CHEV Cavalier, Red, 5 spd, only 7,000 miles on her.
Previous owner heart broken!  Asking only $ 11,995.  # 1415.
JERRY SEINER MIDVALE, 566-38006-3800
```

Neste exemplo não há referência de contexto através de palavras isoladas. O texto como um todo expressa corretamente seu significado. Este texto seria indexado erradamente a contextos divergentes.

Estes problemas foram amenizados, porém não resolvidos. Dicionários de termos ou *Thesauri* são aplicados para resolver o problema de palavras sinônimas. Quanto à polissemia, algumas técnicas foram desenvolvidas [Sanderson, 1994], embora não tenha resolvido o problema. Os três últimos problemas requerem um tratamento mais complexo como indexação por mais de uma palavra [Croft e Lewis, 1991, Dillon, 1983, Fagan, 1989].

Apesar do aumento na precisão dos resultados, a RI ainda não provê contexto às páginas indexadas. De fato, o uso de palavras-chaves e suas combinações não são suficientes para que um sistema de RI recupere e indexe páginas segundo seus contextos. Além da análise estatística dos termos relevantes seria necessário conhecer a semântica dos termos.

Como visto acima a EI busca por termos pré-definidos que caracterizam o assunto de interesse do usuário. Desta forma as palavras e termos que são encontrados ganham um significado segundo o domínio de assunto considerado. Entretanto, sistemas de EI estão limitados a domínios de assuntos conhecidos por eles. Enquanto que a RI desconhece o assunto dos documentos recuperados, a EI tem palavras, termos e relacionamentos que descrevem conceitos que compõem um domínio de conhecimento. Baseados nestes elementos, sistemas de EI identificam e extraem informações das páginas. Neste caso podemos dizer que sistemas de EI detêm um conhecimento sobre o domínio das informações que devem ser extraídas.

O fato da EI já ter pré-definido um domínio de conhecimento permite que problemas de RI sejam tratados com mais eficiência. A restrição de domínio permite uma definição semântica mais precisa

das palavras [Appelt e Israel, 1999]. Estes sistemas de EI tipicamente atuam em domínios pequenos e normalmente são bastante portáveis [Zechner, 1997]. Por outro lado, os sistemas de RI, por desconhecerem domínios de conhecimento, são de uso irrestrito.

Outra desvantagem da EI é o esforço exigido na construção e manutenção do conhecimento. O esforço computacional é outro contra-ponto à EI. Sistemas de EI exigem mais do processamento que os sistemas de RI. Por outro lado, a precisão dos resultados de um sistema de EI é maior que de sistemas de RI. Isto minimiza tempo e esforço do usuário na seleção de páginas de seu interesse. Como um dos resultados de um processo de EI pode ser a informação estruturada, linguagens de consultas semelhantes à linguagem natural como SQL (***S**tructured **Q**uery **L**anguage*) podem ser usadas para efetuar buscas mais eficientes.

A distinção entre RI e EI pode ser definida em termos de funcionalidade. Os objetivos das duas são distintos e complementares. Enquanto a RI coleta documentos relevantes baseando-se apenas em seu conteúdo sintático, a EI extrai informações relevantes que caracterizam o provável domínio de conhecimento do documento. A EI é aplicada como um meio de filtrar o resultado de uma tarefa de RI.

3.3 Abordagens de desenvolvimento de sistemas de EI

Basicamente, existem duas abordagens para construir sistemas de EI: abordagem de treinamento automático e engenharia de conhecimento.

A abordagem de treinamento automático permite que o sistema aprenda a extrair informações submetendo um algoritmo de treinamento a um *corpus* de textos, onde as informações a serem extraídas são destacadas com padrões de extração. Ao final do treinamento, regras de extração são geradas como resultado. Estas regras serão aplicadas para inferir sobre um novo *corpus* de texto. Outra forma de treinamento é interagindo com o usuário durante o processo de treinamento. O sistema depende da concordância de usuário para definir se suas regras de extração são válidas ou não. A cada fase do treinamento o resultado é mostrado ao usuário que decidirá quais são os resultados corretos. O treinamento é finalizado quando um resultado satisfatório é alcançado. Este método se baseia em dados estatísticos para o treinamento.

A engenharia de conhecimento é caracterizada pela construção artesanal de gramáticas. Estas gramáticas são construídas por especialistas com bom conhecimento em sistemas de EI, no formalismo para construção de regras e sobre o domínio de aplicação do sistema. A construção de regras de extração são baseadas na observação de padrões de um *corpus* de textos. As regras são testadas em um *corpus* de teste e o resultado analizado para fazer alterações necessárias nas regras.

As duas abordagens exigem a interferência humana durante seu desenvolvimento. Na engenharia de conhecimento o especialista desempenha um papel fundamental que exige conhecimento e ex-

periência tanto sobre o domínio de aplicação quanto no desenvolvimento de regras. Além disto, o especialista humano é submetido a tarefas artesanais exigindo maior esforço. O custo para manter profissionais deste tipo é muito alto e muitas vezes estes não estão disponíveis devido à experiência exigida. Já no treinamento automático o especialista humano não precisa dispor de tais requisitos, sendo apenas nescessário um conhecimento suficiente sobre o domímio de aplicação e como operar as ferramentas de anotação e treinamento.

Por outro lado, a precisão dos resultados obtidos com a abordagem de engenharia de conhecimento é maior do que a de treinamento automático. O desenvolvimento artesanal de regras, por explorar convenções e formatos padronizados, permite uma precisão maior nos resultados obtidos.

Ainda na abordagem da engenharia de conhecimento é possível aumentar significativamente a precisão dos resultados utilizando uma representação de conhecimento mais expressiva. Como visto no capítulo anterior, a mudança de foco de densenvolvimento para os dados ao invés de se ater aos procedimentos de solução do problema torna os sistemas baseados em conhecimento mais precisos. Outro fator importante é a discriminação do meta-conhecimento sobre o domínio do problema, tornando-o explícito. Discriminar e expor o conhecimento permite maior clareza no processo de desenvolvimento e manutenção do conhecimento. Ontologias representadas por quadros [Minsky, 1975] adicionam estas características na abordagem da engenharia de conhecimento.

Neste trabalho adotamos a abordagem de engenharia de conhecimento utilizando ontologias para o desenvolvimento de um sistema de EI. A precisão nos resultados e o uso de ontologias para representar conhecimentos agregam vantagens como portabilidade entre domínios, necessidade de menos experiência e trabalho na concepção e manutenção do sistema e reusabilidade e compartilhamento das ontologias.

3.4 Taxonomia de sistemas de EI

Sistemas de EI podem ser classificados segundo seu método de extração. Dois grupos principais podem ser definidos: sistemas de EI baseados em conhecimento e sistemas de EI baseados em *wrappers*. A taxonomia dos métodos de extração é mostrada na figura 3.1.

3.4.1 Sistemas de EI baseados em conhecimento

Estas ferramentas relacionam informações do texto a bases de conhecimento. Um dos métodos mais usados se baseia em técnicas de PLN (**P**rocessamento de **L**inguagem **N**atural) [Califf, 1998, Freitag, 1998, Soderland, 1999, Riloff e Lehnert, 1994]. Estas técnicas são aplicadas à EI com a diferença de que somente partes específicas do texto, que contêm informações relevantes, são consideradas. Sistemas de EI baseados em PLN usam aprendizado para aquisição de conhecimento aplicando

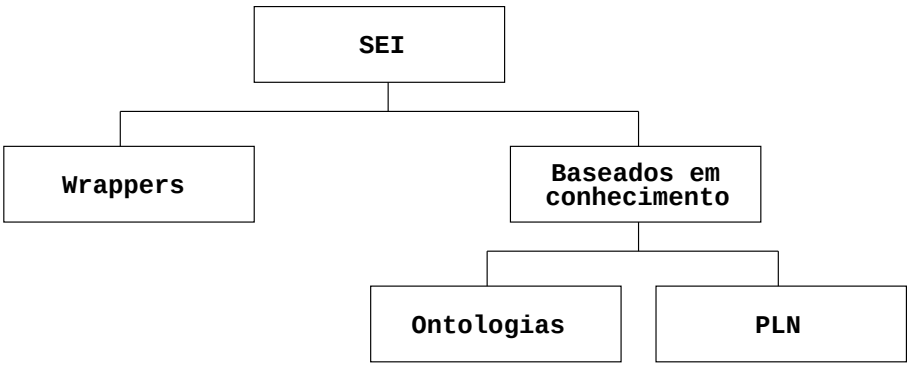


Figura 3.1: Taxonomia de sistemas de EI.

técnicas como filtragem, análise de classes de palavras (substantivos, verbos, predicados, etc.), análise léxica e semântica e relacionamento entre termos e sentenças. Dos resultados destas análises são produzidas as regras de extração. A desvantagem do PLN esta na complexidade de desenvolvimento e manuteção tendo como conseqüência um alto custo em sua concepção. O PLN também exige um maior esforço computacional. Um resultado de extração usando técnicas de PLN é mostrado no exemplo 3.2.

Exemplo 3.2 *Sentença: "The Parliament was bombed by the guerrillas."*

```
Name:                  target-subject-passive-verb-bombed
Trigger:               bombed
Variable Slots:       (target (*SUBJECT* 1))
Constraints:          (class phys-target *SUBJECT*)
Constant Slots:       (type bombing)
Enabling Conditions:  ((passive))
```

A palavra "*bombed*" disparou o reconhecimento dessa frase como sendo do domínio de terrorismo, do tipo bombardeio (*bombing*) e a frase, por seu padrão sintático de voz passiva, reconhece que o alvo (*target*) do bombardeio é o sujeito e agente da passiva da frase (*The Parliament*).

Outro sub-grupo desta abordagem é o uso de um domínio de conhecimento declarativo estruturado em ontologias [Embley et al., 1998, Silva et al., 2002], onde conceitos de um domínio podem ser descritos. Em ontologias representadas por quadros, o conhecimento é constituído por classes, subclasses, instâncias das classes e relacionamentos entre as classes. Isso provê uma estrutura bem definida permitindo inferência de regras e diversos níveis de granularidade do conhecimento. Como consequência ganha-se em reusabilidade de conhecimento e de regras, modularidade, abstração e herança [Noy e MacGuinness, 2001].

3.4.2 Sistemas de EI baseados em *wrappers*

Esta abordagem é a mais usada na Web. Ela surgiu independentemente da EI tradicional sendo aplicada baseando-se nos marcadores da linguagem HTML [Muslea, 1998]. Apesar disto, *wrappers* são considerado sistemas de EI que atuam em textos semi-estruturados, onde identificam dados de seu interesse e os mapeiam para um formato estruturado. Um *wrapper* consiste de um conjunto de regras e uma coleção de expressões para aplicar às regras. Geralmente esta coleção contém marcadores da linguagem HTML. Considerando o exemplo 3.1, o resultado obtido pelo *wrapper* proposto em [Embley et al., 1998] seria o conjunto de informações estruturadas a seguir:

Exemplo 3.3 *Resultado gerado por um wrapper proposto em [Embley et al., 1998].*

```
Descriptor/String/Position(start/end)
Year|97|1|3
Make|CHEV|5|8
Model|Cavalier|10|17
Feature|Red|20|22
Feature|5 spd|25|29
Mileage|7,000|37|41
KEYWORD(Mileage)|miles|43|47
Price|11,995|108|114
Owner|JERRY SEINER MIDVALE|124|143
PhoneNr|556-3800|146|153
```

Este *wrapper* extrai as informações contidas no texto do exemplo 3.1 indicando seu conteúdo e sua posição. Entretanto, este resultado não traz nenhuma conotação semântica. Outro problema é a restrição do domínio de aplicação que é definido pela estrutura de formatação textual utilizada na construção das páginas. Mudanças na formatação acarretam alterações em nível de implementação do *wrapper*. Além disto, páginas relevantes mas com estruturas um pouco diferentes são descartadas.

Os primeiros *wrappers* eram linguagens desenvolvidas especialmente para buscas na Web [Laender et al., 2002] como alternativa às linguagens tradicionais como perl e java. Alguns exemplos de liguagens são TSIMMIS [Hammer et al., 1997] e Minerva [Crescenzi et al., 2001].

3.5 Arquitetura de um sistema de EI

Nesta seção apresentamos uma arquitetura básica para sistemas de EI. Apesar da diversidade dos sistemas de EI existem elementos básicos presentes no processo de extração da maioria dos sistemas.

Em [Grishman, 1997], Grishman divide o processo de extração em dois blocos principais: no primeiro são extraídos fatos individuais e no segundo estes fatos são integrados gerando outros fatos.

A extração de fatos individuais é feita a partir de um conjunto de padrões que expressam a possível existência de um fato no texto. No caso de sistemas baseados em PLN não é prático usar diretamente a correspondência de padrões. Neste caso é feita uma estruturação identificando-se vários níveis de termos e seus relacionamentos. Tipicamente, esta fase começa com uma análise léxica através da identificação de nomes próprios (locais, datas, empresas, etc) e análise de classes de palavras. Este último é executado apenas no PLN .

Na fase de integração, os fatos são analisados e combinados considerando todo o contexto. Esta etapa se refere à análise dos relacionamentos possíveis entre os fatos. No PLN estas relações envolvem classes de palavras. Regras de inferência são usadas para integrar os fatos. Ainda pode-se deduzir informações implícitas no texto através da inferência de regras, baseando-se nos fatos já existentes.

Finalmente, as informações extraídas são inferidas para classificar o texto no contexto do assunto do documento. A figura 3.2 mostra a arquitetura básica de um sistema de EI. A seguir discutimos fases de extração considerando o contexto de sistemas de EI baseados em conhecimento.

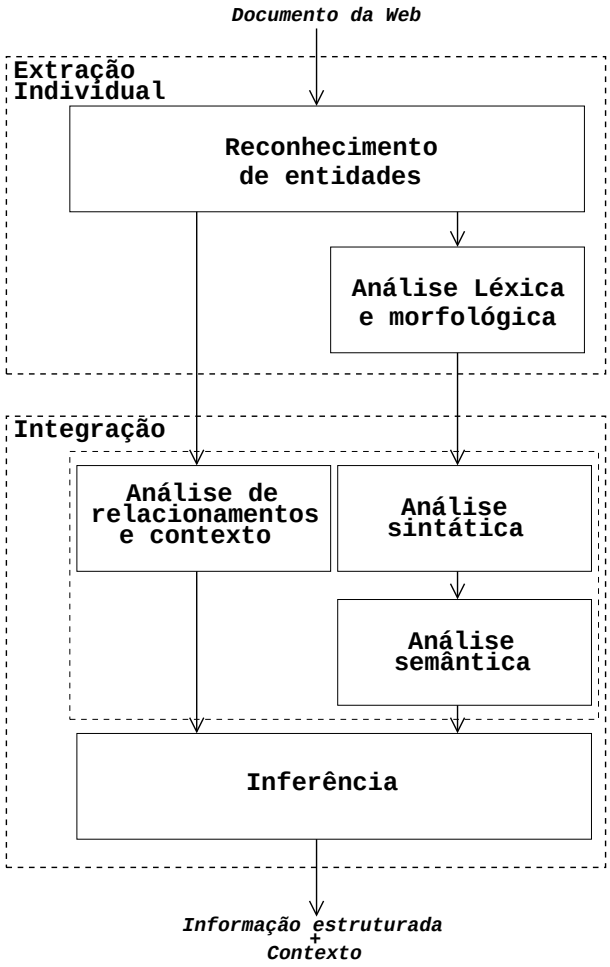


Figura 3.2: Arquitetura de sistemas de EI.

3.5.1 Reconhecimento de entidades

Este módulo tem o objetivo de identificar nomes próprios, lugares, organizações, datas. É a tarefa mais simples de extração e a mais confiável [Cunningham, 1997]. Estes nomes geralmente são declarados em glossários. No exemplo 3.4 é mostrado trecho de código HTML.

Exemplo 3.4 *Trecho de página em HTML descrevendo uma tabela de produtos eletrônicos.*

```
<table>
<tr><td> <b>Olympus Camedia</b></td></tr>
<tr><td> <b>Panasonic Mini-dv</b></td></tr>
<tr><td> <b>Sony Cybershot </b></td></tr>
<tr><td> <b>Sony Hi 8</b></td></tr>
</table>
```

Um glossário de marcas de produtos eletrônicos pode ser usado por um algoritmo simples de busca retornando o nome encontrado e sua posição. No exemplo seriam encontrados os nomes `Olympus`, `Panasonic` e `Sony`, sendo este último em duas posições diferentes no texto. Observe que um bom reconhecedor de nomes dependerá da abrangência do Glossário usado. Outro problema que não é tratado aqui é o de polissemia. Por exemplo, um termo qualquer no texto (e.g. *body*) que coincidisse com a sintaxe de algum marcador HTML (e.g. *<body>* ou *</body>*) seria reconhecido como uma marca. Para este caso específico a solução é simples através da dinstinção dos marcadores da linguagem HTML. Porém, em casos mais complexos, a solução não é tão trivial considerando-se que não há contexto relacionado ao termo.

Conceitos de um domínio de conhecimento também podem ser identificados nesta tarefa por termos que os representam. Em ontologias é possível expressar conceitos através de seus termos. Por exemplo, o conceito `Evento científico` pode ser identificado pelos termos `workshop`, `simpósio` ou `conferência`.

3.5.2 Análise léxica e morfológica

Esta tarefa é aplicada em sistemas baseados em PLN. Nela, cada palavra é analizada e associada a uma classe de palavras. Características de gênero, número e tempo também são consideradas aqui para o caso de verbos [Barros e Robin, 1996]. Dicionários léxicos são usados na análise morfológica para identificar o sentido em que a palavra é empregada de acordo com sua posição no texto. Juntas, estas duas análises dão contexto gramatical às palavras quando analisadas isoladamente.

3.5.3 Análise de relacionamento e contexto

Esta é a principal tarefa na etapa de integração dos fatos. Fatos são gerados a partir de fatos encontrados nas etapas anteriores. Se existir alguma relação entre fatos, um novo fato é gerado. Enquanto existirem relacionamentos entre fatos ainda não relacionados, o processo se repete. Com a entrada de novos fatos no conhecimento, as relações entre estes fatos e os já existentes são verificadas. O processo se repete enquanto existirem novos fatos. No exemplo 3.3 podemos dizer que os *slots* encontrados expressam uma classe que neste caso seria `Carro`. Os *slots* `phone` e `price` mantêm uma relação com a classe `Carro` que expressam a intenção de venda.

Em PLN, a análise sintática trata as palavras como componentes de frase. Ali, é determinado o papel de cada uma na frase através de gramáticas (bases de conhecimento) portáveis e independentes de domínios. O processo se extende até a definição dos papéis de orações e partes de orações. Aqui, o contexto gramatical é agregado às palavras considerando a frase como um todo.

A análise semântica usa uma base de conhecimento chamada de modelo de domínio para interpretar o conteúdo das frases. Esta fase não se aplica a todos os sistemas de EI baseados em PLN. Em alguns casos, a análise sintática é suficiente para o processo de extração.

3.5.4 Inferência

Esta tarefa visa encontrar informações, *a priori*, ocultas no texto através de inferências de regras sobre os fatos já analisados. Esta tarefa depende do contexto ao qual o texto foi relacionado. Na PLN, esta etapa é feita pela análise do discurso. A análise do discurso é a tarefa seguinte à análise sintática. Nela são aplicadas regras de inferência para analisar relacionamentos entre frases. Um componente de uma frase pode representar a mesma entidade em outra frase, como podemos ver no exemplo 3.5

Exemplo 3.5 *"O policial perseguiu a ladra. Ele estava motorizado".*

No exemplo 3.5 [Freitas, 2002], o sujeito da segunda frase `Ele` representa a mesma entidade que o sujeito da primeira `Policial`.

3.6 Trabalhos relacionados

As características principais de um sistema de EI estão relacionadas a sua abordagem de desenvolvimento (treinamento automático ou engenharia de conhecimento) e sua estratégia de extração (baseada em conhecimento ou *wrappers*). A seguir apresentamos algumas ferramentas de EI que expressam o estado da arte.

Na área de PLN podemos citar WHISK [Soderland, 1999]. WHISK se baseia em um conjunto de páginas de treinamento para induzir sua regras de extração. Seu treinamento é feito interagindo com o usuário. A cada fase do treinamento o usuário seleciona os atributos de interesse inserindo marcações através de uma interface gráfica. O treinamento é encerrado quando todos os resultados da extração são satisfatórios. Os padrões de extração do WHISK são um tipo especial de expressões regulares que têm dois componentes [Muslea, 1998]: um que descreve o contexto da informação e que especifica os delimitadores da frase que a ser extraída. Portanto, WHISK extrai informações tanto de textos semi-estruturados quanto textos não estruturados, ou seja, ele adota os dois métodos de extração: PLN e *wrapper*. A desvantagem desta ferramenta é sua dependência quanto à estrutura do texto analizado.

Na área de *wrappers* destacam-se ferramentas de geração automática de *wrappers*. A maioria destas ferramentas se baseiam no aprendizado automático para gerar regras de extração. STALKER [Muslea et al., 2001] é uma ferramenta que usa técnicas de indução de *wrappers* desenvolvidas em ferramentas anteriores. Seu treinamento se baseia em um conjunto de exemplo de treinamento na forma de marcadores que representam a informação a ser extraída e uma descrição da estrutura das páginas chamada de árvore de catálogo embutido (do inglês, *Embedded Catalog Tree* – ECT). A performance de STALKER depende do número de exemplos utilizados no treinamento.

Linguagens para geração de *wrappers* foram as primeiras iniciativas de extração baseadas em *wrappers*. Minerva [Crescenzi e Mecca, 1998] combina uma abordagem baseada em gramática declarativa com caraterísticas de linguagens de programação procedimental. A gramática usada em Minerva é definida no estilo EBNF (*Extended Backus Normal Form* [Wirth, 1977]). Minerva é complementada por uma linguagem de busca e reestruturação de documentos. Minerva também permite a manipulação de exceções onde é adicionada uma claúsula de exceção a cada *produção* de uma gramática EBNF. Como em qualquer linguagem de programação, o desenvolvedor de *wrappers* precisa conheçer a linguagem de programação e a EBNF.

DEADLINER [Kruger et al., 2000] é um sistema de EI similar ao sistema de EI apresentado em nosso estudo de caso. Este sistema monitora a Web, grupos de notícias (do inglês, *newsgroups*), mensagens enviadas a listas de dicussão em busca de anúncios de conferências, das quais são extraídos os atributos data inicial, final e limite, comitê de programa, afiliação de cada membro do comitê, temas, nome dos eventos e país. Cada atributo é extraído através de um grande número de filtros de extração posicionais, similares aos dos *wrappers*. Técnicas de aprendizado são utilizadas com o objetivo de integrar estes filtros. O conhecimento utilizado no processo de extração está armazenado em bases de dados, como pesquisadores, por exemplo.

Em [Embley et al., 1998] é proposto um *wrapper* baseado em ontologias. Ontologias são utilizadas para descrever a informação de interesse incluindo relacionamentos e constantes. A partir de uma ontologia são geradas regras de extração e o esquema normalizado do banco de dados. Apesar da economia de desenvolvimento esta abordagem traz limitações. As ontologias utilizadas especifi-

cam apenas o conhecimento estritamente necessário à extração e, portanto, não exploram uma das principais vantagens de ontologias que é a reusabilidade.

Outra proposta baseada em ontologias é a de [Wee et al., 1998]. Um arcabouço é proposto para representar a informação a ser extraída de um texto. O sistema baseado em quadros é chamado *Frame Extracting Information from Messages* (FEIM). Este quadro é composto por *slots* (atributos) que descrevem características da informação a ser extraída e processos de extração e validação desta informação. O conhecimento sobre o domínio é representado por uma ontologia. FEIM permite descrever como conceitos (classes), definidos na ontologia, serão extraídos do texto e tratados.

3.7 Proposta de EI

Nosso trabalho baseia-se na abordagem de engenharia de conhecimento, utilizando ontologias como formalismo de representação. Em nossa estratégia de extração utilizamos técnicas baseadas em *wrappers* agregando conotação semântica aos padrões de extração. Desta forma, os padrões de extração são expressos não apenas por palavras-chaves, delimitadores, marcadores e valores estatísticos, como também por características que determinam seu contexto.

O sistema de EI proposto é um módulo de tratamento da informação do sistema MASTER-Web (***M**ultiAgent **S**ystem for **T**ext **E**xtracting from Web*) [Freitas, 2002] que é um sistema de manipulação integrada de informação (vide capitulo 4). Este módulo de extração recebe como entrada uma instância de classe que representa a página previamente classificada pelo MASTER-Web, o conhecimento sobre a extração e o conhecimento sobre o domínio. Um conjunto de regras é utilizado para inferir sobre estes conhecimentos. O resultado obtido é o conjunto de informações extraídas segundo o contexto em que a página foi classificada.

Este trabalho se diferencia de outros por prover reusabilidade tanto para a base de regras quanto para o conhecimento sobre o domínio, através do uso de ontologias, além de ser possível extrair informações de interesse para outras classes de páginas promovendo cooperação entre os agentes do MASTER-Web.

Capítulo 4

MASTER-Web

O MASTER-Web (***M**ultiagent **S**ystem for **T**ext **E**xtracting from **Web***) é produto da pesquisa de doutorado de Frederico Freitas [Freitas e Bittencourt, 2003] foi desenvolvido na Universidade Federal de Santa Catarina (UFSC). O sistema tem como objetivo a extração integrada de informação na Web empregando um sistema multiagente e conhecimento explícito através de ontologias. Neste capítulo descrevemos a arquitetura e funcionamento do MASTER-Web.

4.1 Introdução

MASTER-Web é um sistema multiagente de manipulação integrada de informação em que agentes fazem busca e classificação de páginas da Web e extração de informações relevantes.

MASTER-Web parte do princípio de que algumas classes de páginas se interrelacionam, por exemplo, instâncias da classe de páginas de eventos científicos (*Call for Papers*) podem conter informações ou âncoras que levem à páginas de pesquisadores através do atributo *chairman* do evento. Ao conjunto de classes assim relacionadas damos o nome de agrupamento (do inglês, *Cluster*). Estes relacionamentos atualmente são desprezados pelos sistemas de EI por desconhecerem a respeito de outras classes. Outra característica é a forma como páginas são tratadas. Dois tipos de visão são adotados pelo MASTER-Web para tratar as páginas: visão por conteúdo e visão por funcionalidade (seção 4.2). Isto permite categorizar páginas segundo seu conteúdo e seu papel aumentando a eficiência do sistema.

O uso de sistemas multiagentes beneficia o relacionamento entre as classes através da cooperação entre os agentes.

4.2 Visões da Web

Uma signifitiva porção da Web corresponde a textos semi-estruturados e estruturados que compartilham muitas características em comum como padrão de ligações das páginas, terminologia e estilo da página. Estas páginas obedecem minimamente esquemas pré-definidos. Estes esquemas podem ser considerados como categorias de páginas. Categorias podem ser definidas usando como critério modelos de categorias pré-definidos ou analisando-se semelhanças entre páginas. Este último é chamado de agrupamento (*clustering*) [Sahami et al., 1997]. No MASTER-Web, um conjunto destas páginas caracteriza uma classe de páginas (e.g., chamada de trabalhos, pesquisadores e instituições de pesquisa). O uso de esquemas pré-definidos é largamente usado na tarefa de EI.

Dois tipos de visões da Web facilitam a extração integrada do MASTER-Web: visão por conteúdo e visão por funcionalidade. A combinação de duas visões complementares otimiza a recuperação de informação.

4.2.1 Visão por conteúdo

A visão por conteúdo é caracterizada por identificar numa página elementos que expressem o contexto de seu conteúdo. A visão por conteúdo permite agrupar páginas segundo o conceito de classes de páginas.

4.2.1.1 Classes de páginas

Páginas são agrupadas segundo seu conteúdo. Muitos engenhos de busca oferecem este tipo de serviço. A construção e manutenção das categorias são feitas geralmente de maneira artesanal, o que motiva o desenvolvimento de classificadores automáticos utilizando técnicas de aprendizado [Cohen e Singer, 1996]. As categorias estão geralmente organizadas hierarquicamente e sua características principais são:

1. *Similaridade estrutural*: o estilo de composição de páginas é bastante considerado na classificação automática. Algumas categorias tendem a seguir um padrão de formatação textual. A similaridade auxilia na identificação de páginas que pertencem a uma categoria.

2. *Conteúdo*: as categorias são organizadas em hierarquias, onde uma categoria pode ser melhor especificada partindo de um conjunto de sub-categorias que pertençam à referida categoria. Uma categoria pode ser composta por páginas e sub-categorias. A figura 4.1 esboça graficamente a disposição de categorias e sub-categorias.

3. *Entidade*: as categorias mais especializadas, chamadas *folhas*, são compostas apenas por páginas que trazem infomações específicas sobre alguma entidade. Estas páginas podem ser ditas como instâncias da categoria. Estas entidades e seus respectivos atributos caracterizam a visão por conteúdo.

As páginas que apresentam *entidades* compartilham muitas características comuns entre si, tais como estilo de editoração, padrões de conexão a outras páginas, terminologia e, principalmente, o conjunto de atributos, que podem definir classes de páginas. Sistemas de EI se baseiam em classes de páginas. Para que uma página seja relacionada a uma classe é necessário um número mínimo de atributos que caracterizem tal entidade, além da verificação da consistência dos valores destes atributos. Por exemplo, páginas de chamadas de trabalhos para eventos científicos ("calls for papers") devem portar pelo menos uma data ou um ponteiro para datas e na existência de mais de uma data, a diferença entre elas deve ser menor que um ano.

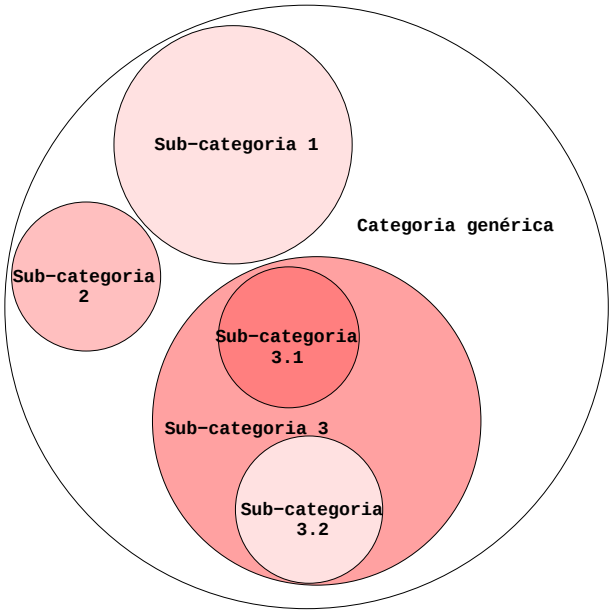


Figura 4.1: Relacionamento entre categorias e sub-categorias.

4.2.1.2 Grupo de classes – *Clusters*

Classes de páginas podem estar interrelacionadas através de ponteiros que apontam para outras páginas cujo conteúdo se refere a outra classe. Por exemplo, em páginas de pesquisadores, com certeza serão encontrados ponteiros para páginas de artigos, de chamadas de trabalho de eventos científicos, e outras classes.

4.2.2 Visão por funcionalidade

A visão por funcionalidade categoriza páginas segundo seu papel na ligação entre páginas e na representação e armazenamento de dados relevantes. Inspirada no trabalho de Pirolli *et alii* [Pirolli et al., 1996], esta visão baseia-se no exame de listas de resultados retornados pelos mecanismos de busca para o processamento de uma classe. Assim, dado este propósito e visando a extração integrada, as categorias funcionais estão assim divididas [Freitas, 2002]:

1. *Páginas conteúdo*: pertencem à classe que está sendo processada, e de onde serão extraídos os atributos da entidade em questão;
2. *Páginas auxiliares*: apontadas exclusivamente por páginas conteúdo, hospedam atributos específicos da(s) entidade(s) da página que a aponta;
3. *Listas de páginas conteúdo*: diretórios ou índices de ponteiros para outras páginas. Listas permitem a localização segura e contextualizada de páginas de conteúdo;
4. *Mensagem ou lista de mensagens*: são mensagens ou listas de mensagens sobre assuntos relacionados à entidade que está sendo extraída. Por exemplo, páginas de perguntas freqüentes (*Frequently Asked Questions* – FAQ);
5. *Recomendações*: se refere a páginas de conteúdo pertencentes a outras classes;
6. *Lixo*: são as páginas sem valor para o processo de EI.

Estas duas visões são usadas simultaneamente pelo MASTER-Web. A visão por conteúdo é responsável por trazer da Web páginas potencialmente pertencentes às classes processadas, garantindo cobertura sobre a Web e a visão por funcionalidade preocupa-se em selecionar com rigor as páginas que contêm as entidades de onde serão extraídas informações relevantes tanto para a contextualização do conteúdo da página (página conteúdo) quanto na otimização da busca de mais entidades (listas e páginas auxiliares). Por exemplo, o resultado de busca apresentado por um robô (mais detalhes na seção 4.4) traz um conjunto de páginas relacionadas a um domínio de conhecimento (visão por conteúdo), sendo que neste conjunto de páginas podem ser encontradas páginas conteúdo, páginas auxiliares, listas, mensagens ou lixo (visão por funcionalidade). Estas paginas são selecionada e tratadas segundo sua categoria funcional.

4.3 Sistemas multiagentes cognitivos

Sistemas multiagentes (SMA) cognitivos são baseados em organizações sociais humanas como gurpos, hierarquias e mercados [Bittencourt, 1998]. Os agentes possuem uma representação explícita do ambiente e de outros agentes, dispõem de memória e são capazes de planejar suas ações futuras.

Considerando o tamanho e heterogeneidade da Internet, qualquer aplicação com processamento centralizado está fadada a baixa eficiência. A abordagem multiagente se adequa bem ao problema da Internet por sua característica de resolução distríbuida problemas (RDP). De fato, cada agente de um SMA tem uma habilidade particular.

A estrutura de um problema no contexto de SMA é análoga à estrutura de um agrupamento de páginas. Um problema é basicamente tratado por um SMA da seguinte forma:

- dividir o problema em subproblemas;
- definir agentes aptos a resolver o problema;
- solicitar a cooperação destes agentes; e
- gerar uma solução sintetizada do problema a partir das soluções dos subproblemas recebidas dos agentes envolvidos na cooperação.

Já os agrupamentos são compostos de um conjunto de classes de páginas com algum tipo de relacionamento. Podemos, em nosso contexto, considerar um agrupamento como um problema complexo e dividi-lo em subproblemas. A aplicação de SMA em EI permite que classes de páginas não sejam tratadas isoladamente, compartilhando informações e se relacionando com outras classes. A seguir apresentamos as características de SMA aplicáveis ao tratamento da Extração da Informação.

4.4 Arquitetura do MASTER-Web

A arquitetura do MASTER-Web é baseada em Sistemas Multiagentes (SMA) com o objetivo de recuperar e extrair dados de páginas da Web pertencentes a classes de um *cluster*. A concepção teve como pilar o príncipio de torná-la o mais reusável possível, tanto num nível *macro*, permitindo a portabilidade do sistema entre domínios da Web, quanto no nível *micro*, permitindo o reuso de seus agentes em diversos contextos de um domínio. A figura 4.2 mostra a arquitetura do MASTER-Web. Cada agente é especialista no reconhecimento de páginas que correspondem à instâncias das classes que ele processa. Além do reconhecimento, os agentes são especilistas em extrair atributos da entidade em questão, procurando também ponteiros úteis a outros agentes. A cooperação entre os agentes é representada pela estrela que os interliga.

Um agente pode ser responsável pela extração de uma classe de páginas ou de um conjunto delas dependendo da similaridade entre os padrões de suas classes. Por exemplo, a classe de eventos científicos tem especializações como *workshop*, colóquio e conferência. Um único agente poderá ser usado no processo de extração já que estas classes compartilham atributos. Caso contrário, se os padrões diferem, é preferível que seja incluído um novo agente que trate independentemente parte deste conjunto de classes.

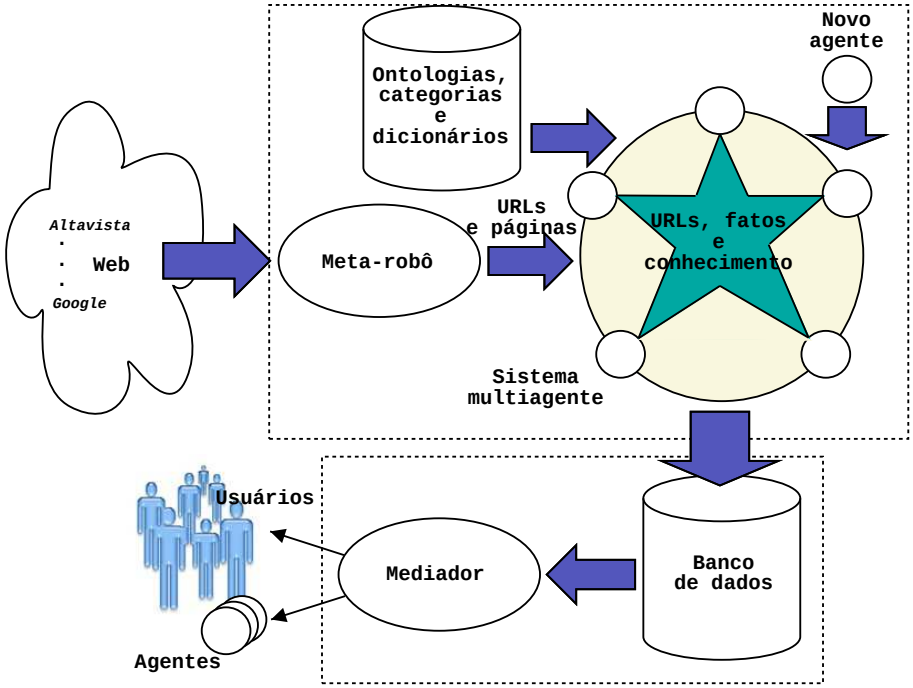


Figura 4.2: Arquitetura do MASTER-Web.

4.4.1 Componentes

Como podemos ver na figura 4.2 componentes da arquitetura do MASTER-Web são:

1. *Meta-robô*: o meta-robô se conecta a vários engenhos de busca como *Altavista*, *Infoseek*, etc aproveitando seus índices. Este meta-robô segue as diretrizes de bom comportamento de um robô na Web [Koster, 1993]. Seu funcionamento procede da seguinte forma: consultas são feitas em engenho de busca utilizando palavras-chave que abranjam o máximo possível de páginas relacionadas à classe de páginas em processo. O robô faz uma busca baseando-se na visão por conteúdo recuperando um conjunto de páginas de várias categorias funcionais. O meta-robô alimenta uma fila de baixa prioridade do agente que tratará o conjunto de páginas. Apesar da arquitetura apresentar apenas um meta-robô, cada agente acessa a fila de páginas do meta-robô ao qual está relacionado.

2. *Categorias, dicionários e ontologias*: formam o conhecimento estático que dá suporte à extração. Categoria são armazenadas num banco de dados e possuem tabelas gerais como de países e cidades, e específicas , como áreas de pesquisa e as hierarquias de cargos em centros de pesquisa para o grupo científico (domínio considerado no estudo de caso). O terceiro é conhecimento do agente sobre o domínio representado por ontologias

3. *Mediador*: é a interface do sistema com outras entidades externas as quais interagem com ele através de consultas. Estas consultas podem ser feitas por outros agentes ou por usuários. O

mediador deverá estar disponível com a função de ajudar às consultas, provendo visões não-normalizadas permitindo a qualquer usuário beneficiar-se do acesso a dados extraídos.

A sociedade de agentes cognitivos forma o núcleo do sistema. Alí, as páginas serão processadas como veremos na seção a seguir.

4.4.2 Características de um agente MASTER-Web

Algumas características comportamentais e estruturais do agente devem ser consideradas para avaliar sua interação social.

Quanto ao modelo comportamental, o agente da arquitetura se registra ao entrar na sociedade e divulga seus interesses através de regras gerais e específicas (enviadas apenas para agentes específicos) entre todos os agentes da sociedade. Este novo agente também recebe regras dos agentes já pertencentes à sociedade. Estas regras pemitem a identificação de páginas do interesse do agente relacionado a elas, iniciando a cooperação entre eles. As regras trocadas são disparadas quando algum agente identifica informações relevantes a outros agentes. Quando um novo agente entra na sociedade todos os outros mudam seu comportamento, tentando identificar informações úteis para ele.

Quanto a estrutura, os agentes mantêm a mesma estrutura e código, permitindo reuso e flexibilidade.

4.5 Arquitetura de um agente MASTER-Web

A arquitetura do agente MASTER-Web é mostrada na figura 4.3. Cada agente é composto de módulos que executam tarefas específicas no processo de classificação de páginas.

Cada agente acessa a duas filas de páginas, uma de baixa prioridade que é alimentada por um meta-robô e outra de alta prioridade que é fruto de sugestões oriundas da cooperação entre os agentes. Esta última contém páginas que já foram parcialmente avaliadas e foram sugeridas por outros agentes o que as tornam mais confiáveis quanto ao seu conteúdo. As páginas, então, são subimetidas à tarefas que são descritas a seguir.

4.5.1 Validação

Aqui, as páginas são analisadas segundo o formato e protocolo permitido pelo agente, se já estão contidas no banco de dados, evitando redudância, se já foram processadas, se está acessível, entre outras regras. A validação acelera o processamento, além de evitar trabalho redundante economizando

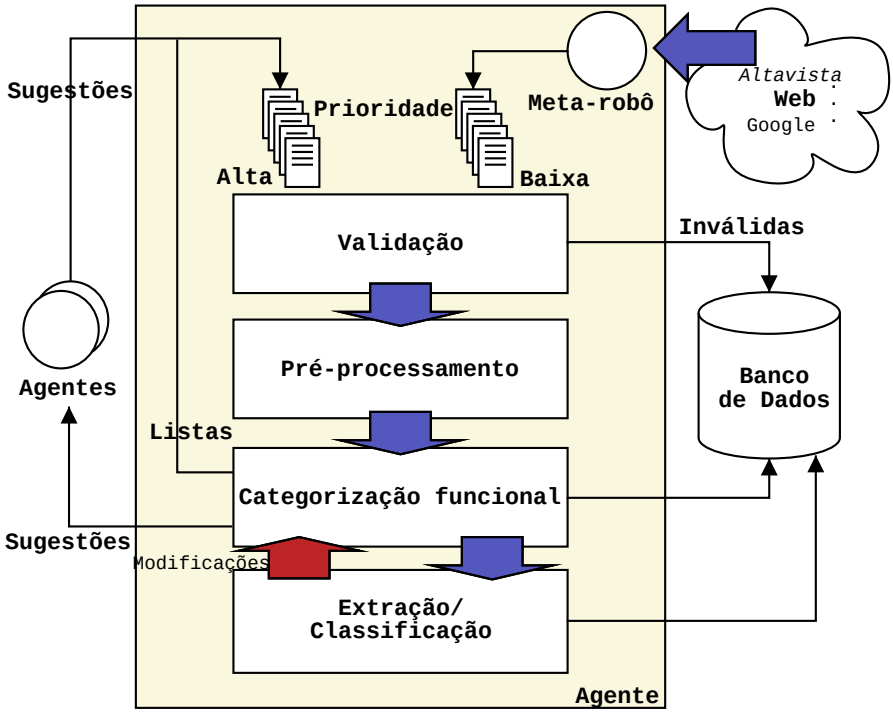


Figura 4.3: Arquitetura do MASTER-Web.

tempo de de processamento e gastos com a recuperação de páginas inúteis. Ao final, o endereço da página é armazenado no banco de dados, bem como a data e o estado da página Mesmo páginas inúteis são armazenadas.

4.5.2 Pré-processamento

Nesta fase uma página é representada em diversas formatações. Uma Classe chamada *Web-Page* é utilizada na representação da página (figura 4.4). A partir dela, é possível criar uma instância de uma página com *slots* com o texto original, sem marcadores HTML, em listas geradas a partir de técnicas de *stop-list* (lista de palavras irrelevantes), com todas as palavras minúsculas, etc. Estes formatos auxiliam nas tarefas posteriores. Por exemplo, a busca por nomes próprio como o de cidades e pessoas deve ser realizadas no texto original, enquanto que palavras-chave como *deadline* serão procuradas no texto com todas as letras em minúsculo.

4.5.3 Categorização funcional

Nesta fase as páginas são classificadas segundo a visão funcional. Páginas reconhecidas como de conteúdo passam para a fase seguinte do processamento. As páginas reconhecidas como listas têm seus endereços relevantes extraídos e enviados para a fila de alta prioridade para serem processados.

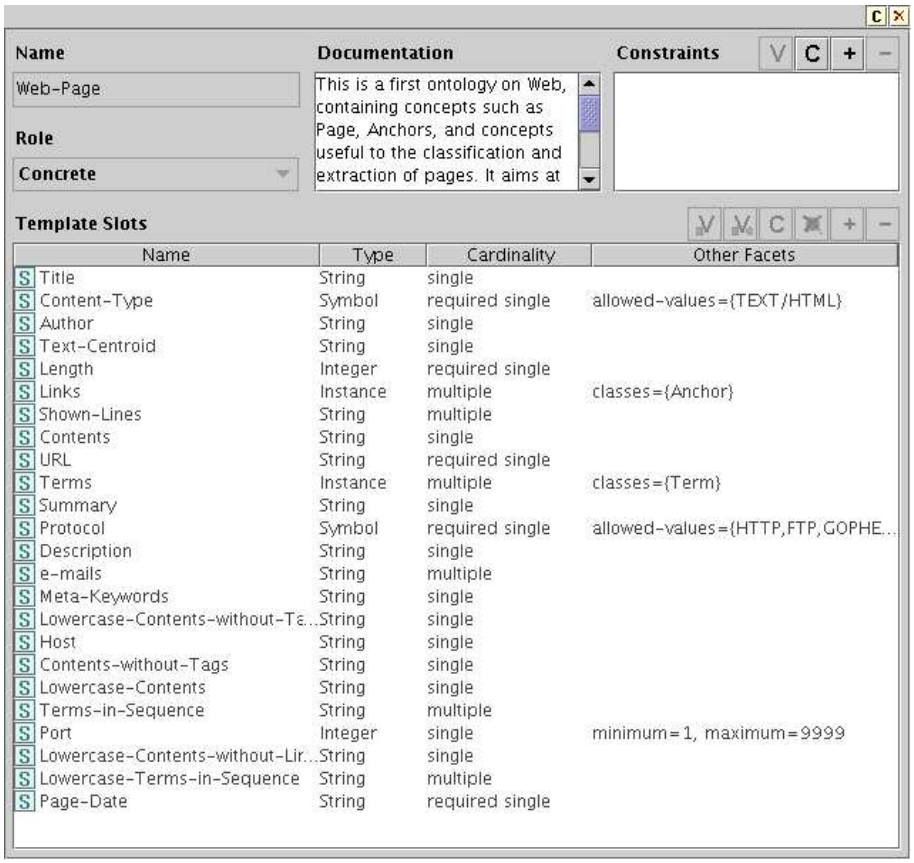


Figura 4.4: Classe *Web-Page* para representação de páginas.

Páginas auxiliares são sugeridas para outros agentes através da cooperação, pois são identificadas como pertencente a outras classes. Outros tipos de páginas são considaderas como lixo.

Em particular, listas devem ser identificadas de forma precisa a fim de evitar perda de informações relevantes, envio de falsas listas ou de conteúdo duvidoso, compromentendo a qualidade dos resultados obtidos pelo sistema.

4.5.4 Classificação

Nesta etapa a extração é executada a fim de classificar a página segundo seu conteúdo. Cada agente abrange um conjunto de classes as quais serão utilizadas como referência para a classificação. Dado um conjunto de atributos extraídos de uma página, é possível identificar a qual classe ela pertence. Regras são utilizadas na extração dos atributos, que pode ser feita de três maneiras: diretamente da página, através de inferência ou através de categorização. Um número mínimo de atributos essenciais extraídos, bem como a ausência de atributos não desejados, são declarados através de *casos* e utilizados para classificar a página.

4.6 Conhecimento do agente

O conhecimento dos agentes é representado através de ontologias mais base de regras e funções. A biblioteca de ontologias é formada por cinco ontologias distribuídas nas categorias de ontologias de domínio, ontologias genéricas e ontologias de aplicação. A primeira modela o domínio do meio científico, a segunda é composta pela ontologia da Web, de tempo, de turismo, etc. A terceira se refere a ontologia específica de cada agente, são elas: ontologias de manipulação integrada de informação, *templates* classificadores e extratores.

O conjunto de regras é dividido da seguinte forma: regras de reconhecimento e extração e regras de recomendações.

4.7 Extração de informação no MASTER-Web

Apesar de se falar bastante em extração, o MASTER-Web executa esta tarefa visando apenas o reconhecimento da página, classificando-a de acordo com os atributos extraídos. Entretanto, o endereço da página e a classe a qual pertence não são informações suficientes para pesquisas feitas por usuários ou agentes.

Como visto no capítulo 3, a extração pode ser realizadas com o objetivo de classificar um texto ou estruturar informações relevantes ao entendimento do texto. O MASTER-Web extrai "partes da informação" que confirmam a presença de entidades que representam o texto, por exemplo, o atributo *deadline* (pertencente a classes de *workshop*, conferência, colóquio, etc) pode ser evidenciado pela presença de termos que o representam, como *submitted by*, *paper due*, *proposal due*, etc. A constatação da existência de atributos é suficiente para a classificação, mas para a estruturação da informação é necessário ainda atribuir valores para estes atributos.

No capítulo 5 propomos a extração de informação com fins de estruturação da informação contida numa página previamente classificada pelo MASTER-Web. As informações agora serão extraídas por completo, ou seja, para cada atributo encontrado haverá a tentativa de encontrar seu referido valor. Voltando ao exemplo anterior, além da constatação da existência do atributo *deadline*, datas serão procuradas como valor relacionado a ele. Com dados estrutrados é possível oferecer buscas mais precisas a usuários e outros agentes através do mediador.

Capítulo 5

Extração de Informação no MASTER-Web

Como visto no capítulo anterior, o MASTER-Web extrai informações com o objetivo de classificar páginas da Web que tratem de assuntos do meio científico. Neste capítulo apresentamos um sistema de Extração de Informação (EI) com o objetivo de estruturar e armazenar as informações extraídas das páginas. O processo de EI baseia-se no resultado da classificação para extrair atributos relevantes ao tema abordado.

5.1 Introdução

O sistema de EI desenvolvido para o MASTER-Web tem como objetivo extrair informações relevantes ao contexto no qual a página foi classificada. Este sistema compõe o módulo de extração do agente do MASTER-Web. Com adição da extração visando a estruturação das informações contidas no texto, a arquitetura do agente MASTER-Web é expandida conforme mostrado na figura 5.1. A tarefa de extração agora é composta de dois módulos. O primeiro a ser executado trata de extrair informações para classificar a página e o segundo se baseia no resultado do primeiro para identificar quais atributos serão extraídos da página.

Este módulo de EI ainda poderá confirmar se uma página pode ser dita pertencente a uma classe ou não. Por exemplo, considerando que uma página de chamada de trabalhos (*Call for papers*) tem datas e conceitos como *deadline* e *workshop* extraídos de seu corpo, a fase de classificação poderá considerá-la como pertencente a classe de *Workshop*. Porém, é necessário que, para o atributo representado pelo conceito *deadline*, existam valores consistentes, como datas. As datas encontradas também podem não estar relacionadas ao conceito extraído.

Conceitos e outras entidades (data e locais) encontradas na fase de classificação são utilizadas para a próxima etapa da extração. Isto evita que o sistema refaça as tarefas desnecessariamente. A tarefa de extração é aplicada na busca dos dados restantes. A partir destes dados extraídos, a informação será constituída.

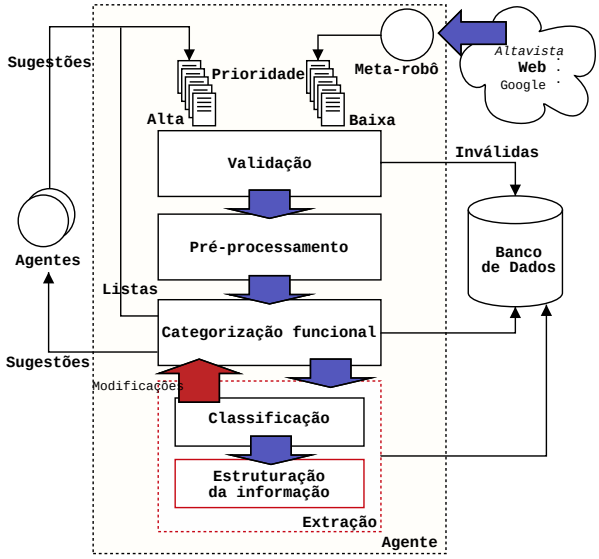


Figura 5.1: Nova estrutura do agente MASTER-Web.

5.2 Proposta de EI para o MASTER-Web

A proposta do sistema de EI foi baseada em abordagens e estratégias discutidas no capítulo 3. Nossa abordagem é a de engenharia de conhecimento e usamos ontologias e regras de produção para representar o conhecimento. Quanto à estratégia de extração, as técnicas desenvolvidas se baseiam nos *wrappers*. Apesar de discutirmos que *wrappers* não traziam informação semântica nos seus resultados, mostramos neste trabalho que é possível adicionar esta característica a eles utilizando ontologias que usam quadros como formalismo de representação. A figura 5.2 mostra a taxonomia de sistemas de EI considerando esta abordagem.

A união destas duas estratégias (*wrappers* e baseada em ontologias) é baseada na proposta de Li Wee [Wee et al., 1998], onde se usa ontologias para representar o conhecimento do domínio e quadros para representar padrões da informação a ser extraída. Li Wee propõe representar três tipos diferentes de conhecimento: (1) conhecimento sobre a informação de entrada (texto da MUC); (2) conhecimento sobre o domínio da aplicação; e (3) conhecimento sobre a informação a ser extraída. A estratégia de EI adotada neste caso se baseia em técnicas de processamento de linguagem natural (PLN).

O processo de EI é executado através de inferência [Villain, 1999]. Cada etapa deste processo é composta por um conjunto de regras. Estas etapas de EI são *reconhecimento de entidades, análise de*

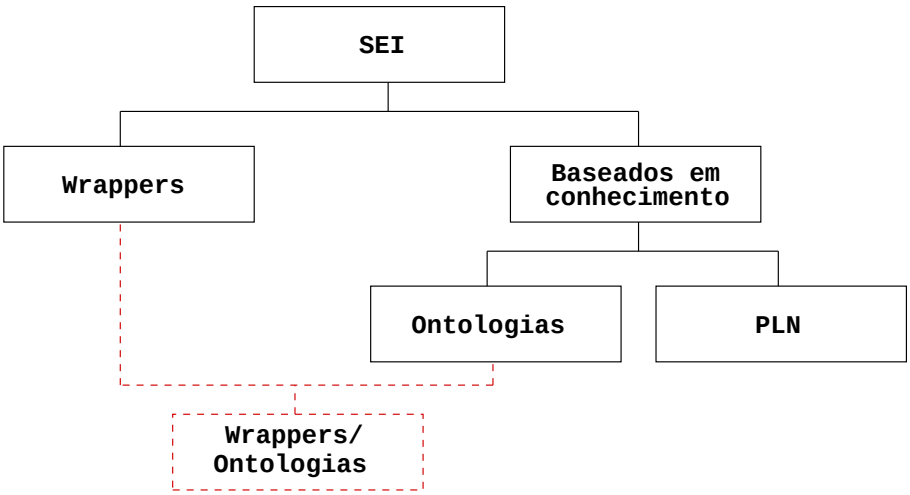


Figura 5.2: União das estratégias de EI – *wrappers* e baseada em ontologias.

relacionamentos e contexto, *inferência* e *reclassificação* e estão descritas na seção 5.4.

5.3 Conhecimento do sistema

O conhecimento representado em nossa abordagem utiliza ontologias baseada em quadros e regras de produção. Três tipos de conhecimentos são representados por ontologias: *conhecimento sobre o domínio*, *conhecimento sobre a página* e *conhecimento sobre as informações a serem extraídas*.

5.3.1 Conhecimento sobre o domínio

É o conjunto de classes que compõem um agrupamento. Estas classes representam os conceitos do domínio em questão. Cada agente MASTER-Web é responsável por uma ou mais classes (dependendo da similaridade entre elas) para extrair dados tanto para a classificação quanto para a estruturação da informação. Esta ontologia é também composta por ontologias auxiliares como turismo, tempo e Web, por exemplo.

5.3.2 Conhecimento sobre a página

Um conjunto de classes foi definido para representar a página processada, os dados extraídos dela e para monitorar seu comportamento durante o processo de EI. A página de onde os dados são extraídos é representada através de uma instância da classe `Web-Page` (figura 4.4). Esta classe é composta por *slots* que representam a página em diversos formatos (texto nomal, texto com letras minúsculas, texto com marcadores HTML, etc). As informações extraídas da página são guardadas

em instâncias das classes `Data-Found` e `Information-Found`. Esta última é composta por uma ou mais instâncias de `Data-Found`. As duas classes permitem uma maior granularidade da informação e a extração através da integração de entidades componentes da informação no sentido de baixo para cima (do inglês, *bottom-up*). A figura 5.3 mostra as classes que representam o conhecimento sobre a página e seus relacionamentos.

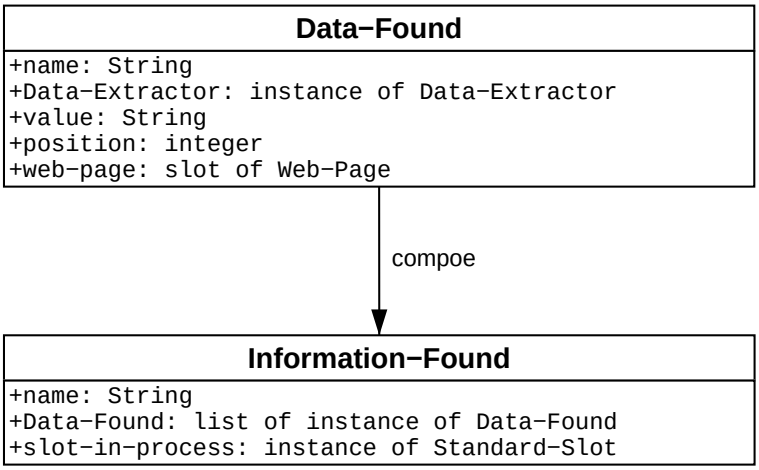


Figura 5.3: Diagrama de relacionamento entre as classes de representação da informação

O processo de EI é monitorado por uma instância da classe `Processing-Monitor` que é composta de *slots* que representam o estado da página. Na figura 5.4 mostramos as classes `Processing-Monitor` e `Web-Page`. Os *slots* `Page-Status` e `URL-in-Process` definem a relação entre as duas classes.

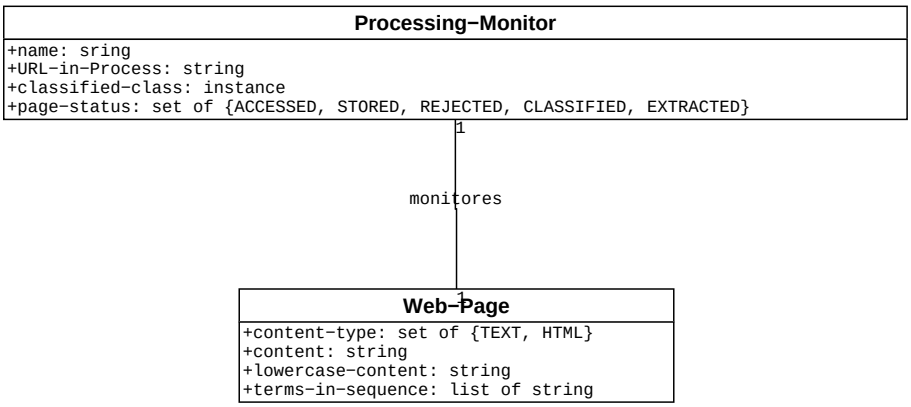


Figura 5.4: Diagrama de relacionamento entre a classe de representação da página e a classe de monitoramento da EI

5.3.3 Conhecimento sobre a informação a ser extraída

Este conhecimento é representado por um conjunto de classes que descrevem os padrões das informações a serem extraídas. A figura 5.5 mostra as classes envolvidas neste conhecimento e seus

relacionamentos.

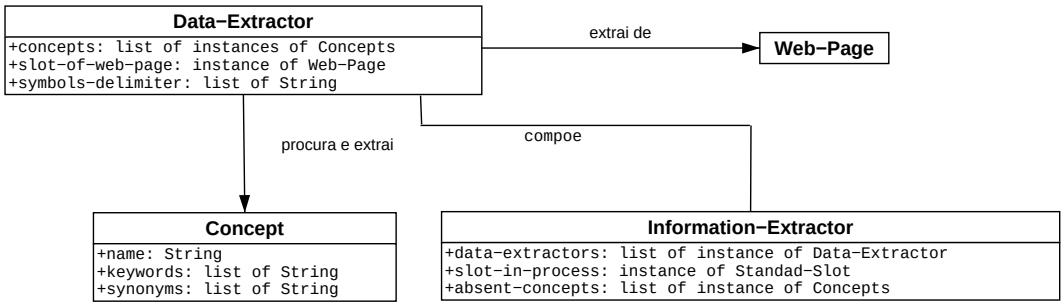


Figura 5.5: Diagrama de relacionamento entre a classe de EI

As classes representam características sobre a extração de entidades e seus relacionamentos. Cada *slot* da classe de classificação[1] pode ter sua representação descrita através de conceitos (classe `Concept`) e informações de como extraí-los através de instâncias das classes `Data-Extractor` e `Information-Extractor`. A informação é tratada como um conjunto de uma ou mais entidades na qual a classe `Data-Extractor` descreve como extrair cada uma destas entidades enquanto que `Information-Extractor` que descreve como integrar estas entidades. Informações como posição na página, conceitos relacionados, conceitos e símbolos ausentes são fundamentais para definir relacionamentos entre as instâncias destas classes.

5.3.4 Regras de produção

Regras de produção são compostas por um conjunto de premissas e uma conclusão. Quando existem fatos que suportam as premissas, a regra é disparada gerando uma conseqüência (conclusão).

As regras são usada para representar o conhecimento de como extrair informações de uma página. Enquanto que as ontologias representam o conhecimento sobre a estrutura sintática e semântica da informação, as regras representam o conhecimento comportamental, ou seja, como extrair, analisar e classificar a informação. A regra mostrada em 5.1 é um exemplo conhecimento sobre o processo de extração.

Exemplo 5.1 *Regra básica para EI.*

```
(defrule r_444_slots_me_ccpt_term
      (object (Importance MEDIUM) (is-a Slot-Extractor)
            (Slot-in-Process ?s) (Concepts $ ?cb) (Slots-of-Web-Page $ ?sw))
      (not (object(is-a Slot-Found) (Slot-in-Process ?s)))
      (test (member-number $ ?cb (beginning (slot-get [PAGE] $ ?sw)))
```

[1] Chamamos de classe de classificação as possíveis classes em que uma página processada pode ser classificada

```
=>
(make-instance of Slot-Found (Slot-in- Process ?s)))
```

5.4 Processo de Extração de Informação

A EI está dividida em quatro tarefas, cada uma delas reponsável pela extração de partes da informação ou sua composição. A EI é executada de modo que a informação seja construída a partir de entidades extraídas da página, pode-se dizer que a informação é extraída no sentido de baixo para cima (do inglês *bottom-up*). As tarefas estão agrupadas em duas fases: a EI individual e a integração das entidades. A figura 5.6 mostra as etapas de EI para o MASTER-Web.

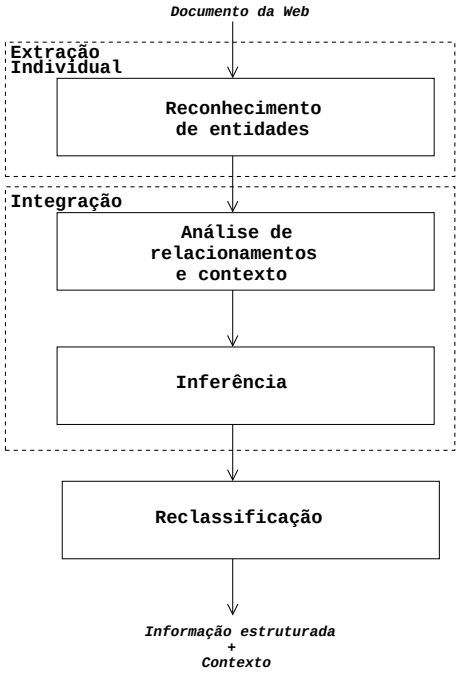


Figura 5.6: Etapas de EI para o MASTER-Web

Na fase de EI individual, é executada a tarefa de reconhecimento de entidades, onde os componentes da informação são identificados e extraídos. Na fase de integração estes componentes são associados formando a informação.

O processo de EI no MASTER-Web recebe como entrada uma instância da classe `Web-Page` que representa a página em processamento e uma instância da classe `Processing-Monitor` que monitora seu estado. Os *slots* `Page-Status` e `Classified-Class` desta instância disparam regras que iniciam o processo de EI. O primeiro indica o estado da página e o segundo indica em que classe a página foi enquadrada. No início da extração, a página tem como estado `ARMAZENADA` e o processo de classificação é iniciado. Após a classificação, a página pode assumir dois estados: `REJEITADA` ou

`CLASSIFICADA`. No primeiro caso a página não foi reconhecida como pertencente a nenhuma das classes sob responsabilidade do agente. No segundo, a página é classificada e seu estado é alterado para `CLASSIFICADA`. Além disso o *slot* `Classified-Class` é preenchido com a classe em que a página foi classificada. Estes dois valores de *slots* dão início a uma nova etapa de EI objetivando extrair os *slots* restantes.

Outros *slots* da instância da classe `Web-Page` como `Lowercase-Contents`, `Terms-in-Sequence` e `Contents` guardam o conteúdo da página. Na maioria das etapas da extração, é usado o conteúdo original da página com todas as letras minúsculas, exceto na busca por palavras que são diferenciadas pelas letras maiúsculas e minúsculas, como é o caso de nomes próprios (nome de pessoas, cidades, países, etc) e siglas.

5.4.1 Reconhecimento de entidades

Nesta etapa são procuradas entidades que podem compor uma informação. As entidades são identificadas através de padrões declarados em instâncias da classe `Concept`. Esta classe é definida por *slots* que representam palavras-chave e sinônimos de um determinado conceito. Inicialmente, um conjunto de palavras-chave e sinônimos definidos nas instâncias de `Concept` são usados para executar buscas por conceito que compõem a informação a ser extraída. No exemplo 5.2 é mostrado um trecho de página com sinônimos que representam os conceitos "*deadline*" e "data de notificação" (*submitted by* e *Notification of acceptance*, respectivamente).

Exemplo 5.2 *Trecho de página de chamada de trabalho onde o conceito "deadline"e "data de notificação"são encontrados.*

```
"Paper deadline
Technical Paper must be submitted by:  July 17, 1995
(Papers must be complete for review with all references, figures etc.)
Notification of acceptance:  September 4, 1995 (Reviewers may suggest
modifications.)"
```

Outra situação é mostrada no exemplo 5.3 onde partes do sinônimo aparecem separadas. Neste caso o termo deverá ser procurado por partes sob a condição de estarem próximos um do outro e da inexistência de sinais que representem a quebra do seu significado, como "."e ";".

Exemplo 5.3 *Trecho de página de chamada de trabalho onde o sinônimo notify by aparece em partes separadas.*

```
"The Academic Programme Committee intends to complete its work and notify
contributors by mid-February 2001 if possible."
```

Para isto, sinônimos e palavras-chave são declarados na instância da seguinte forma:

$$termo = p_1 * p_2 * p_3 * ... * p_n,$$

onde *termo* representa um sinônimo ou uma palavra-chave e p_i é um componente do termo. O símbolo "*" (asterisco) indica que entre as partes do termo pode haver palavras não relacionadas ao conceito. No exemplo 5.3, a palavra não relacionada é "*contributors*". Se a busca fosse feita com termos do exemplo 5.2 o conceito não seria reconhecido.

Além da busca por conceitos, também é feita uma busca utilizando listas de termos que não compõem os conceitos, como meses e localidades. Outros dados como números ordinais, endereços eletrônicos, etc. são procurados por funções executadas por regras. Esta busca é mais simples pois os termos costumam ser invariáveis.

Para cada conceito encontrado é gerado um novo fato na base de conhecimento. Este fato é representado por uma instância da classe `Data-Found`. Esta instância traz o texto extraído, sua posição e suas relações com outros *slots* e instâncias como as de `Concept` e `Data-Extractor`.

Alguns dados extraídos ainda passam por uma formatação e análise crítica de seu conteúdo como é o caso das datas. Por exemplo, as seguintes datas são prazos de submissão (*deadline*) e de realização de um *workshop* : 30/02/2003 e 12/08/2003, respectivamente. A data referente ao prazo de submissão[2] é inconsistente, pois o mês de fevereiro deve ter no máximo 29 dias.

O reconhecimento de entidades termina quando todos os conceitos são identificados na página, ou seja, quando toda a página é analisada. Ao final, os fatos extraídos são utilizados na próxima etapa.

5.4.2 Análise de relacionamentos e contexto

Esta etapa é a principal no processo de EI. Novos fatos são gerados a partir de fatos encontrados na etapa anterior. Consideramos aqui que os fatos podem ser de dois tipos: *fatos independentes* e *fatos dependentes*. Os fatos independentes não precisam de complementos para expressar seu significado. Localidade (cidades, países, etc), no domínio de eventos científicos, é um exemplo deste tipo de fato. Datas encontradas no início da página e na vizinhança de alguma localidade podem também ser consideradas fatos independentes. O segundo tipo depende de algum relacionamento que o complemente para definir uma informação.

A composição da informação através de fatos dependentes pode ser feita de três maneiras: (1) através da integração dos fatos dependentes; (2) através de funções; e (3) por delimitação de regiões.

Considerando que

[2]Nos referimos aqui ao termo conhecido no meio científico por *deadline* como data de submissão, prazo de submissão e data limite.

$$\text{informação} = \textit{contexto} + \textit{dado},$$

onde `contexto` e `dado` são entidades que compõem a `informação`, fatos representados por entidades extraídas da página (Instâncias de `Data-Found`) podem manter um relacionamento do tipo descrito acima, ou seja, um fato representa o contexto da informação e o outro representa o dado ao qual o contexto é atribuído, constituíndo a informação. O exemplo 5.4 mostra um trecho de página onde os termos sublinhados representam fatos relacionados.

Exemplo 5.4 *Trecho de página de chamada de trabalho com fatos relacionados.*

```
"Paper deadline
Technical Paper must be submitted by:  July 17, 1995
(Papers must be complete for review with all references, figures etc.)
Notification of acceptance:  September 4, 1995 (Reviewers may suggest
modifications.)"
```

Os termos *submitted by* e *Notification of acceptance* representam conceitos que evidenciam a existência da informação procurada e estão relacionados com as respectivas datas *July 17, 1995* e *September 4, 1995* que são dados correspondentes àqueles *slot*. Os critérios para relacionar fatos são descritos por instâncias da classe `Data-Integrator`. O relacionamento entre o fato *contexto* e o fato *dado* pode ser assegurado pela proximidade entre os termos e a ausência de sinais que quebrem de alguma forma esta relação, por exemplo ".", ";", "</tr>", etc.

Na segunda maneira, um fato dependente que representa o contexto da informação reconhece seu respectivo dado através de funções específicas. Os fatos que não foram integrados podem formar outros fatos executando estas funções. A figura 5.7 mostra uma página onde o conceito `areas` é identificado a partir do sinônimo `Subject` e seu complemento é uma lista de tópicos. Uma função de delimitação é executada para identificar uma lista de tópicos correspondente ao dado da informação. Apesar de serem funções específicas de extração, é possível reusar algumas delas que são representadas por quadros. Uma instância da classe `Function-Call`, que representa uma função, é mostrada na figura 5.8. Na instância são especificados o nome da função e os argumentos necessário para a sua execução. O usuário da função só precisa saber o objetivo da função para utilizá-la.

Fatos também podem ser extraídos através da delimitação de uma região do texto. Nesta forma de extração, fatos são utilizados para delimitar uma provável área onde uma informação esteja. Um novo fato é gerado a partir do trecho delimitado. A figura 5.9 mostra a página de um evento científico onde o título é delimitado pelos termos que representam fatos dos conceitos evento científico e localidade.

Os novos fatos gerados podem disparar novamente as regras de integração, sejam elas para integrar novos fatos, para executar funções que complementam a informação ou delimitar fatos ainda não descobertos. Este processo se repitirá enquanto houver fatos e relacionamentos entre eles que ainda não foram inferidos. Ao fim desta etapa, um número significativo de informações já estarão extraídas.

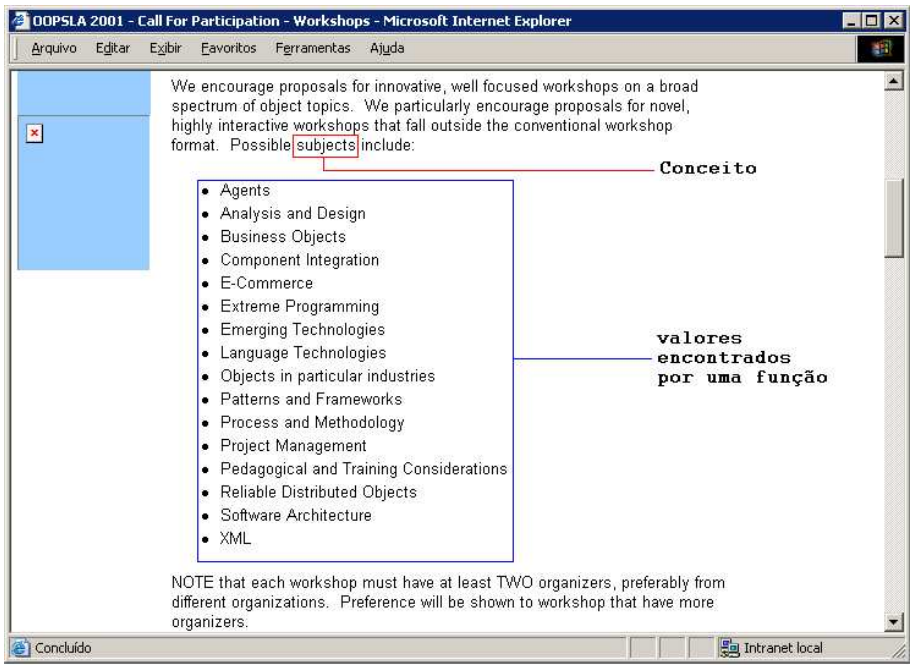


Figura 5.7: O conceito `area` representado por `subjects` está relacionado a lista de tópicos da página.

Para cada informação completa é criada uma instância da classe `Information-Found` onde a informação é guardada, bem como os fatos relacionados a ela, sua posição no texto e qual *slot* da classe de classificação está relacionada.

5.4.3 Inferência

Nesta etapa parte das informações já foram extraídas, entretanto, os fatos ainda podem ser úteis na extração de informação que não está explícita no texto. Estas novas informações são deduzidas através de inferências de regras que se baseiam apenas nos fatos extraídos e não mais no texto da página. Um trecho de uma página é mostrado na figura 5.10. As datas são extraídas sem o ano. O ano poderá ser deduzido através de alguma data que tenha esta informação. Considerando que a página é de um evento científico, a diferença entre as datas não poderá ultrapassar um ano. Apesar da importância, foram poucas as amostras encontradas com datas com esta característica, não permitindo que fossem feitos experimentos mais consistentes.

5.4.4 Reclassificação

Esta etapa tem como objetivo confirmar a classificação da página processada. Agora as informações utilizadas são mais consistentes do que no início do processo de EI. A reclassificação pode ser feita através da existência de informações relevantes no texto. O número de *slots* extraídos também é relevante para a classificação.

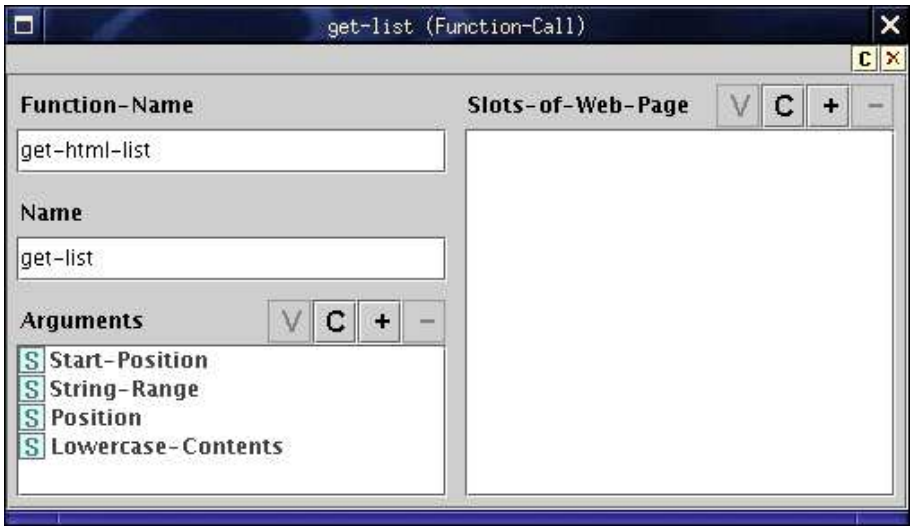


Figura 5.8: Instância da classe `Function-Call` no Protégé-2000.

A diferença entre a primeira classificação e esta é que no primeiro caso a classificação baseia-se em instâncias de conceitos enquanto que a segunda classificação baseia-se em informações mais completas e consistentes. Entretanto, a classificação por conceitos é necessária para definir os *slots* que serão extraídos, caso contrário, seria necessário tentar extrair todos os *slots* de todas as classes do domínio do problema.

5.5 Benefícios da Extração de Informação para o MASTER-Web

A EI adiciona mais funcionalidade ao sistema MASTER-Web trazendo, além da classificação da página, um conjunto de informações estruturadas. As vantagens da estruturação são:

- Reconhecer o contexto do conteúdo de uma página através de informações mais consistentes. Ao invés de utilizar apenas conceitos como critério de classificação, informações consistentes também poderão ser usadas;
- Representar o conteúdo da página de forma estruturada provendo maior legibilidade da informação por parte de sistemas baseados em conhecimento. Isto permite um resultado de busca mais preciso que reduz os esfoços do usuário no processo de filtragem de páginas de seu interesse;
- A informação estruturada também permite clareza e legibilidade dos dados por parte dos sistemas baseados em conheciemnto.

Nossa abordagem traz benefícios como o uso de uma estrutura clara do conhecimento através de classes e seus relacionamentos provendo diversos níveis de visão e inferência, e a extração de infor-

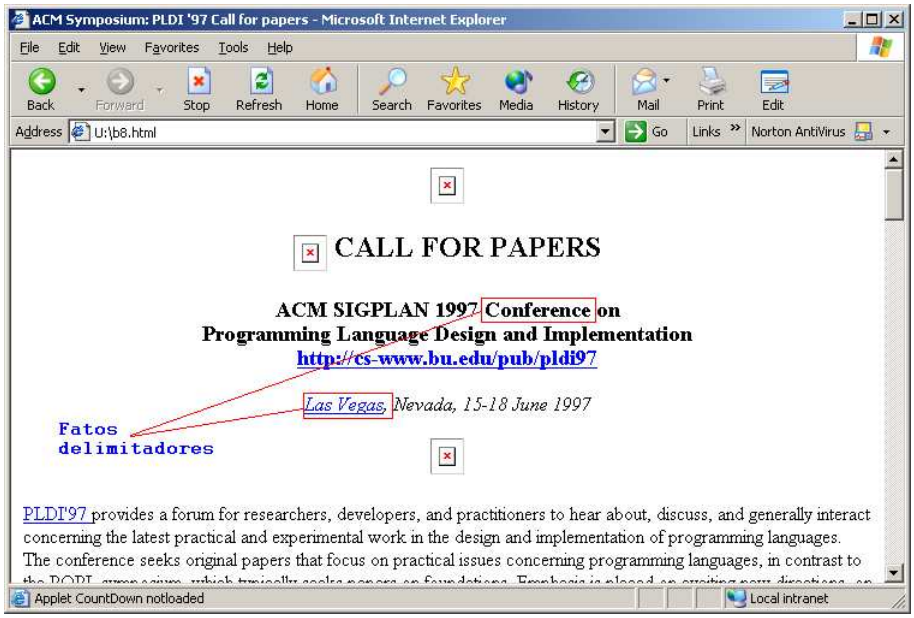


Figura 5.9: Página onde dois termos delimitam o título do evento científico.

mações relevantes a outros agentes como âncoras e dados (seção 5.5.1), que podem ser comunicadas a eles por meio de uma linguagem de comunicação baseada em conhecimento.

5.5.1 Cooperação

Algumas informações extraídas podem coincidir com o interesse de outros agentes da sociedade. Neste momento o processo de comunicação entre agentes é iniciado onde o conteúdo do *slot* é sugerido para o agente interessado. O vocabulário de uma ontologia torna a comunicação mais rica e expressiva [Freitas e Bittencourt, 2002]. Por exemplo, a mensagem mostrada em 5.5 descreve uma sugestão que pode ser feita pelo agente CFP (*Call for Paper*) a respeito de informações sobre um *chairman* para o agente de pesquisadores. Neste exemplo é usada a linguagem KQML (***K**nowledge **Q**uery and **M**anipulation **L**anguage*) para compartilhamento de conhecimento entre agentes [Finin et al., 1994].

Exemplo 5.5 *Mensagem trocada por agentes.*

```
(tell :sender cfp
      :receiver researchers
      :language Jess
      :content(researcher (First-Name Peter)
               (Last-Name Sweeney)
               (chair(event (name OOPSLA)
                            (year 2001)))))
```

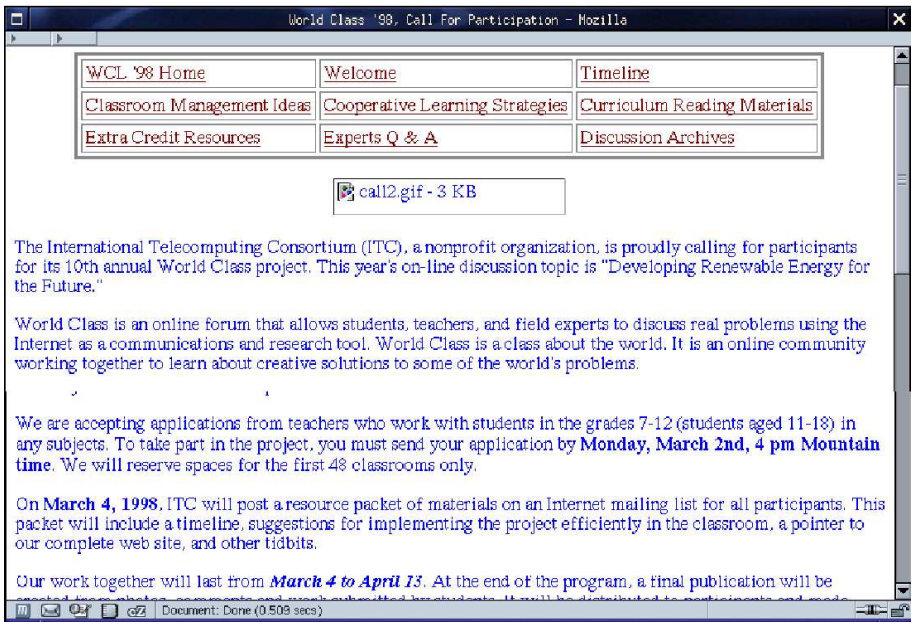


Figura 5.10: Datas sem o ano. O ano pode ser deduzido a partir da data de realização do evento.

Atualmente, informações sugeridas por um agente MASTER-Web são endereços de páginas da Web. Com o processo de extração, um conjunto de informações estruturadas poderá ser sugerida entre os agentes. Para isso será necessário que cada agente, ao entrar no sistema divulge regras que expressem seu interesse sobre determinados tipo de infomação. A cooperação é um tema abordado por sistemas multiagentes, o que não é da alçada desta proposta, entretanto, comentamos a cooperação neste trabalho como benefício e trabalho futuro para a EI e SMA.

5.5.2 Portabilidade e Reusabilidade

A principal vantagem deste trabalho é a portabilidade do sistema de EI. O uso de conhecimento declarativo, separando dados e operações (regras), permite que o sistema seja aplicado em domínios diferentes, sendo necessárias poucas mudanças no código do sistema. A discriminação do conhecimento (conhecimento sobre o domínio, sobre a página e sobre o processo de extração) também permite a portabilidade das ontologias desenvolvidas. A ontologia do domínio científico [Freitas, 2001] desenvolvida para o projeto MASTER-Web está disponível no repositório do Protégé. Por outro lado, ontologias disponíveis para o reuso podem ser usadas para compor o conhecimento da extração, se necessário especializando-a para adaptá-la aos objetivos do sistema. Esta caracterísa provida por ontologias ainda não é bem explorada pelos sistemas atuais [Embley et al., 1999a]. De fato, a ontologia desenvolvida para este sistema é muito específica à aplicação perdendo características como descriminação de tipos de conhecimento e reusabilidade.

Capítulo 6

Estudo de caso e resultados

A proposta de Extração de Informação (EI) descrita no capítulo anterior foi aplicada à páginas que tratam de eventos científicos, utilizando o sistema MASTER-Web instanciado para extrair informações de páginas de eventos científicos considerando o domínio da ciência.

6.1 Introdução

O estudo de caso segue as bases e requisitos para a construção do sistema experimental do MASTER-Web em [Freitas, 2002]. Como o sistema aqui proposto está inserido num sistema maior, a escolha por ferramentas de desenvolvimento, tanto de regras como de ontologias seguiram o que já fora definido naquele trabalho. O mesmo é dito a respeito do conhecimento sobre o domínio. Levando-se em conta que a tarefa de extração para estruturar informações se baseia no resultado de classificação da página, o mesmo conhecimento sobre o domínio foi utilizado.

Por outro lado, o conhecimento sobre a estrutura da informação a ser extraída, bem como o conhecimento sobre como extrair, foram representados construindo-se ontologias e regras. O conhecimento sobre a página e sobre informações extraídas foi reusado através da adição de novas classes.

6.2 Requisitos

Os recursos exigidos para implementação do sistema de EI proposto devem oferecer suporte quanto:

- à construção de ontologias;

- à instanciação de classes que compõem uma ontologia;
- ao desenvolvimento de regras de extração;
- à inferência de regras de extração; e
- ao armazenamento de dados.

Para os dois primeiros itens acima, a ferramenta de edição Protégé-2000 foi escolhida. Protégé-2000 é um ambiente para construção de ontologias desenvolvido pelo Departamento de Informática Médica da Universidade de Stanford e inicialmente visava a construção de ontologias do domínio da medicina. O Protégé permite o desenvolvimento de ontologias independente do domínio a que ela se refere e também dá suporte a diversas linguagens de representação como XML e RDF(S) (maiores detalhes no capítulo 2). As principais características deste ambiente são [Noy et al., 2000]:

- Extensibilidade: uma ontologia é construída a partir da redefinição de classes através da herança. Protégé-2000 contém um conjunto de classes primitivas – as metaclasses – que são redefinidas para a construção das ontologias de aplicação. As metaclasses dão suporte à criação de classes, slots, facetas, entre outros recursos.
- Flexibilidade: a ferramenta armazena as ontologias em formatos que permitem que outros componentes interpretem e manipulem o conhecimento ali declarado. São exemplos destes componentes: Jess, F-Logic, RDF, DAML+OIL, XML, Topic Maps. Outro formato de armazenamento são tabelas de bancos de dados relacionais;
- Interface: a interface do sistema é bastante amigável provendo recursos gráficos para a construção de ontologias, navegação pelas classes e instâncias através de seus relacionamentos. Além disto, a ferramenta dispõe de um gerador automático de formulários para a entrada do conhecimento. Neste formulário são criadas as instâncias das classes que compõem o conhecimento;
- Adaptabilidade: por intermédio de componentes (*plugins*), diversas aplicações podem ser conectadas ao Protégé-2000. Estes componentes são desenvolvidos por grupos de pesquisas usuários da ferramenta para prover maior funcionalidade à ferramenta em termos da Engenharia de Software. Jambalaya [SHriMP, 2003] e Ontoviz [Sintek, 2003] são exemplos de componentes. O primeiro é um utilitário de animação que permite vários recursos de animação e o segundo é um componente que permite a geração de gráficos com instâncias.

O motor de inferência utilizado para inferir sobre o conhecimento do sistema é o Jess (Java Expert System Shell) [Friedman-Hill, 2000]. Jess é um motor de inferência com encadeamento para frente que tem um bom índice de aceitação por parte dos usuários devido às suas características:

- Boa integração entre os objetos Java e os formalismos de representação de regras;

- Implementa boa parte das funcionalidades do sistema de produção CLIPS (C Language Integrated Production Systems) [Riley, 1999] para Java;
- Linguagem de representação do conhecimento é semelhante ao LISP;

Entretanto, Jess foi projetado sob a ótica da orientação a objetos, utilizando componentes Java (Java beans). Isto impede a representação de quadros. De fato, em Jess, instâncias de classes declarativas não podem ser atribuídas como valor de um slot. Esta é uma das principais características de ontologias baseada no formalismo de quadros. Este problema é solucionado através de um componente que integra o Jess ao Protégé-2000 chamado JessTab [Eriksson, 2000]. Este componente permite que o Jess manipule ontologias através do Protégé-2000. JessTab implementa classes declarativas seguindo a expressividade de quadros.

Outra vantagem é a integração entre Java e Jess. Jess é capaz de manipular diretamente objetos, métodos e variáveis Java dentro de regras do Jess. A recíproca também é verdadeira: regras e fatos Jess podem ser criados dentro do código Java. Entidades criadas tanto no Jess quanto no Java podem ser passadas entre eles em ambos os sentidos.

Quanto ao armazenamento das informações sobre a página (endereço da página, classificação e informações extraídas), foi utilizado o banco de dados relacional MySQL [Widenius e Axmark, 2002], onde são armazenados dados relacionados à classificação e extração.

6.3 Construção do conhecimento

Como descrito no capítulo anterior, o conhecimento está dividido em quatro partes: conhecimento sobre o domínio, conhecimento sobre a página e informações extraídas, conhecimento sobre a estrutura da informação a ser extraída e conhecimento sobre como extrair informações.

Os três primeiros tipos de conhecimentos foram representados através de ontologias, sendo que o conhecimento sobre o domínio foi apenas reusado sem a necessidade de qualquer alteração. O domínio aqui considerado foi sobre a ciência, no qual o sistema de extração se baseia para classificar e estruturar as páginas da Web.

Quanto ao conhecimento sobre a estrutura das informações a serem extraídas e sobre as informações extraídas, este sofreu modificações com a adição de novas classes e relacionamentos.

6.3.1 Criação de classes

Novas classes foram adicionadas à ontologia sobre a página e informações extraídas visando uma representação da informação mais detalhada. No capítulo anterior consideramos que a informação é

composta de um *contexto* e um *dado* que são representados pela classe `Data-Found`. A informação extraída é representada pela classe `Information-Found` que contém *slots* relacionados à instâncias que representam dados extraídos da página, seja ela referente ao contexto ou ao dado. As figuras 6.1 e 6.2 mostram o código que definem as classes `Data-Found` e `Information-Found`. Este trecho de código é gerado automaticamente pelo Protégé-2000 quando as classes são criadas. Nela temos a especificação da classe e seus *slots*.

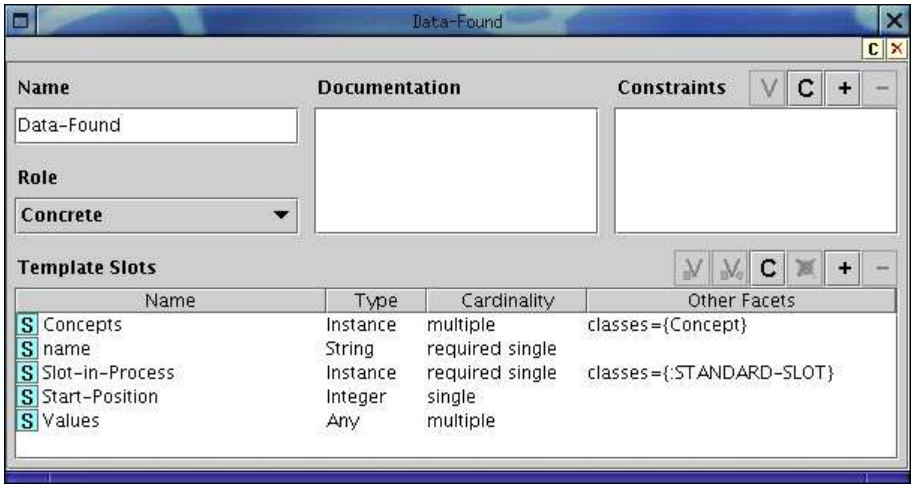


Figura 6.1: Classe `Data-Found` criada no Protégé.

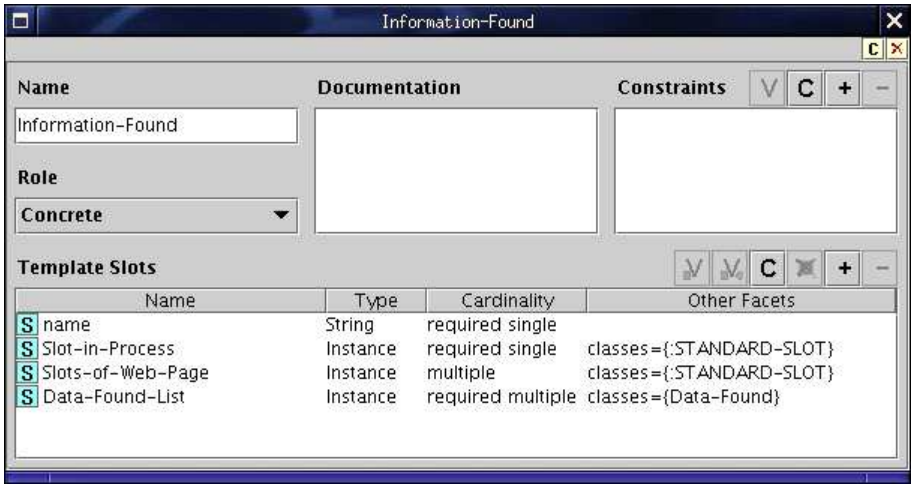


Figura 6.2: Classe `Information-Found` criada no Protégé.

Estas duas classes permitem que a informação seja extraída no sentido de baixo para cima (do inglês, *bottom-up*), permitindo um nível de detalhamento maior da informação e a atribuição de semântica a ela.

O conhecimento sobre a estrutura da informação expressa características sobre os dados a serem extraídos e sobre quais dados compõem a informação. As classes `Data-Extractor` e `Information-Extractor` foram criadas para definir características sobre a representação dos dados no texto e o

relacionamento entre estes dados, respectivamente. A classe `Data-Extractor` é composta por *slots* (a figura 6.3 mostra a classe criada no Protégé-2000) com características estruturais do dado, como a posição do dado no texto, delimitadores, *tokens*, palavras-chave e sinônimos que representam o dado procurado, por exemplo. Este último é representado através de instâncias da classe `Concept`. No exemplo 6.1 é mostrado o trecho de código gerado pelo Protégé que define uma classe (neste caso, a classe é `Concept`).

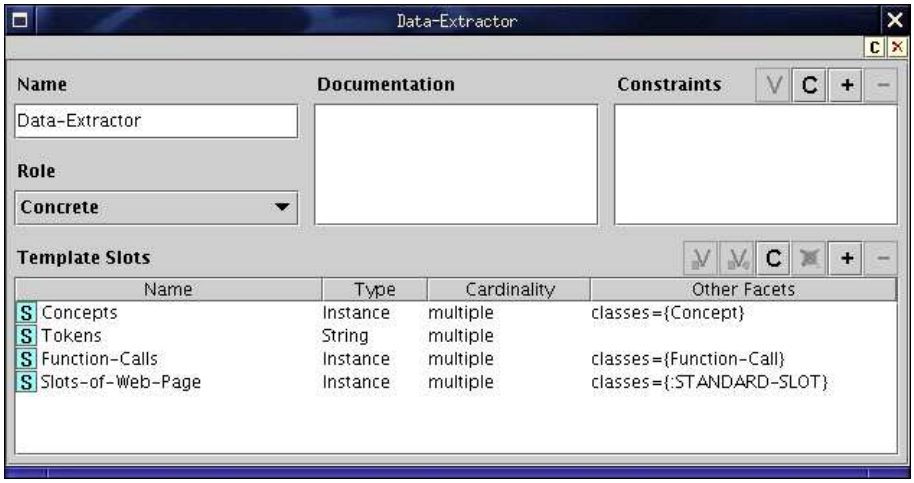


Figura 6.3: Classe `Information-Extractor` criada no Protégé-2000.

Exemplo 6.1 *Trecho de código gerado pelo Prótégé (simplificado) que define uma classe*

```
(defclass Concept
   (single-slot name (type STRING))
   (multislot Synonyms (type STRING))
   (multislot Keywords (type STRING)))
```

A classe `Information-Extractor` define os relacionamentos entre instâncias da classe `Data-Extractor`, bem como a distância mínima entre eles no texto, possíveis termos e sinais que não podem estar entre eles, etc. A figura 6.4 mostra os *slots* da classe criada no Protégé-2000.

Este conjunto de classes forma a ontologia sobre a informação a ser extraída e a informação extraída de uma página. O relacionamento entre estas classes é mostrado na figura 6.5. As classes `Data-Extractor` e `Information-Extractor` descrevem como dados e informações são representados na página. Quando estes são extraídas, passam a ser representados por instâncias das classes `Data-Found` e `Information-Found`.

6.3.2 Definição de instâncias das classes

Até este momento, o conhecimento foi apenas representado através de classes, mas não foi definido ao que ele se refere, por exemplo, quais informações serão extraídas, quais conceitos representam

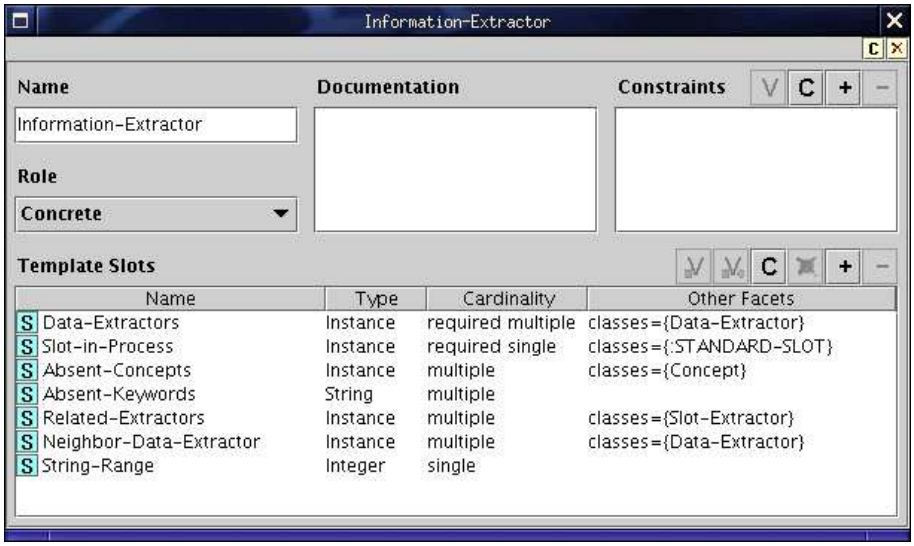


Figura 6.4: Classe `Information-Extractor` criada no Protégé-2000.

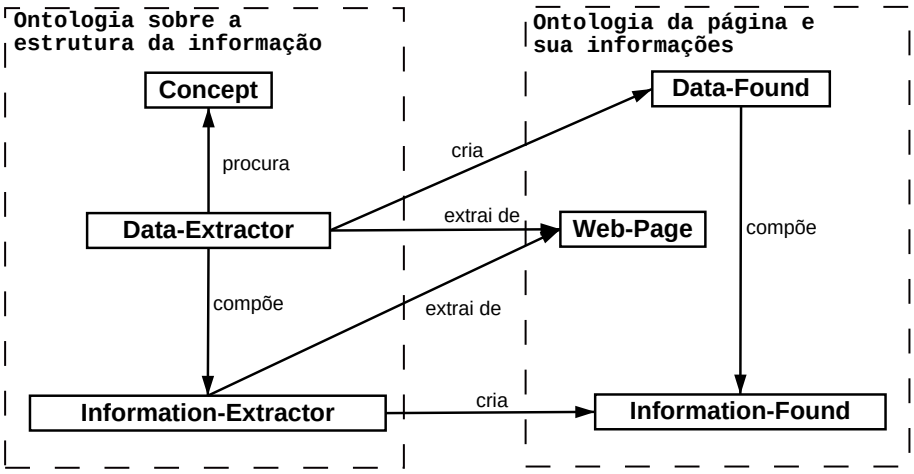


Figura 6.5: Diagrama de relacionamento entre as classes das ontologias de extração.

um dado e quais dados compõem uma informação. A instanciação das classes se refere ao processo de "entrada do conhecimento" na ontologia. Definido o domínio em que as páginas serão classificadas e estruturadas, instâncias das classes `Concept`, `Data-Extractor` e `Information-Extractor` são criadas indicando as características da informação a ser extraída.

Em nosso estudo de caso foi considerado o domínio da ciência, onde páginas sobre eventos científicos são classificadas como *conferência*, *workshop*, *jornal*, etc. Conceitos que definem os dados foram instanciados adicionando-se aos *slots* sinônimos e palavras-chave. No exemplo 6.2 mostramos a definição do conceito `acceptance-date`. Este conceito pode ser identificado através de seus sinônimos (*date of acceptance*, *date of notification*, *notification*, etc.).

Exemplo 6.2 *Definição de uma instância da classe Concept*

```
([Web_00284] of Concept
   (name "acceptance date")
   (Synonyms
      "date of acceptance"
      "date of notification"
      "notification"
      "acceptance"
      "feedback"
      "notified * by"))
```

Instâncias de `Data-Extractor` definem como a informação poderá estar disposta no texto e quais conceitos representam o dado procurado. Em outros casos, quando o dado é procurado por função, a função é definida pelo *slot* `function-call`. O exemplo 6.3 mostra um trecho de código que é uma instância de Data-Extractor. A instância se refere ao que será procurada usando-se como critério para a busca o conceito declarado no *slot* `Concepts` (O valor deste *slot* – "[Web_00274]" – se refere a uma instância da classe `Concept`). No outro exemplo (exemplo 6.4), temos uma instância que expressa um dado que é procurado por funções declaradas em `function-call` que, neste caso, é *find-month*. Observe que o *slot* `Function-Calls` tem, como valor, um nome de função, que no caso é *find-month*.

Exemplo 6.3 *Instância de* `Data-Extractor` *onde o dado é extraído baseando-se em conceitos.*

```
([cfp_01275] of Data-Extractor
      (Slots-of-Web-Page [Lowercase-Contents])
      (name "Deadline")
      (Concepts [Web_00274]))
```

Exemplo 6.4 *Instância de* `Data-Extractor` *onde o dado é extraído baseando-se em funções.*

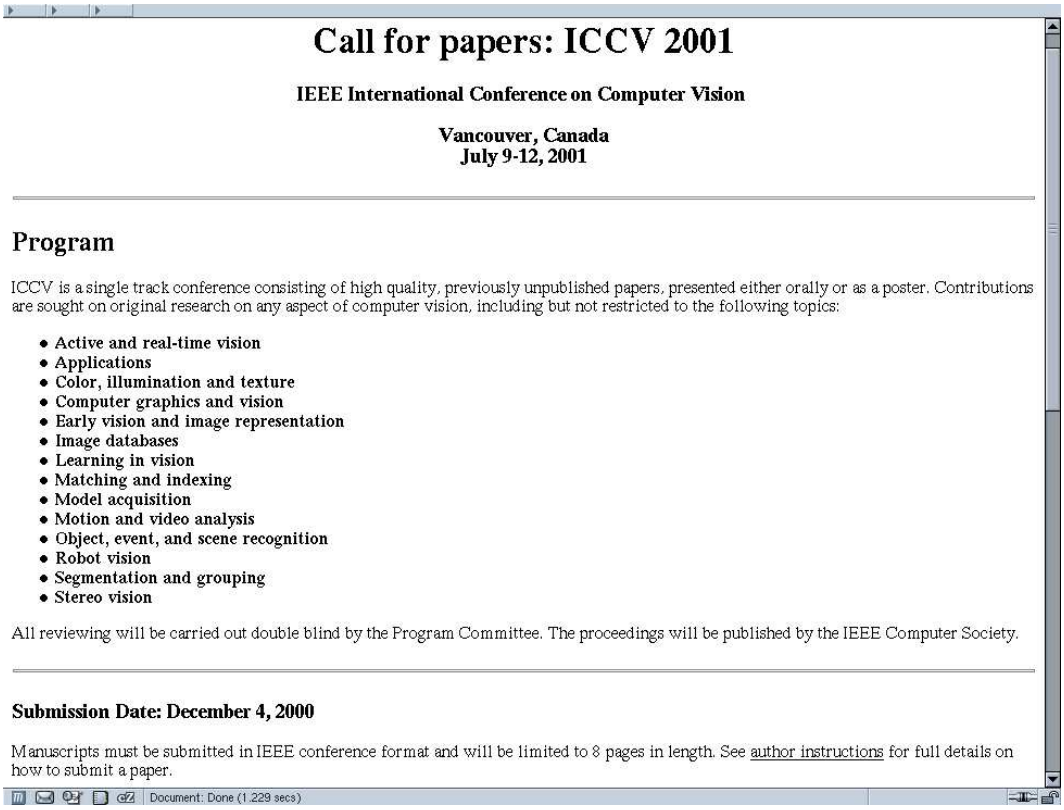
```
([cfp_01276] of Data-Extractor
      (Slots-of-Web-Page [Lowercase-Contents])
      (name "Date")
      (Function-Calls "find-month"))
```

As instâncias de `Information-Extractor` são criadas associando-se as diversas instâncias de `Data-Extractor` com o objetivo de expressar como tal informação é composta. Nestas instâncias pode-se definir quais dados compõem a informação, a distância máxima entre eles, conceitos e termos que não podem ocorrer entre eles, funções de extração e tratamento de dados, etc. A seguir mostramos um exemplo (exemplo 6.5) de instância desta classe. A informação declarada nela se refere à "data de notificação" que tem como dados componentes, aqueles extraídos pelas instâncias mostradas nos exemplos 6.3 e 6.4. Estas instâncias são atribuídas como valores ao *slot* `Data-Extractor`.

Exemplo 6.5 *Instância de* `Information-Extractor` *que é composta por instâncias de* `Data-Extractor`.

```
([cfp_01277] of Information-Extractor
      (Data-Extractors
            [cfp_01276]
            [cfp_01275])
      (Slot-in-Process [Deadline])
      (name "Deadline")
      (String-Range 100))
```

As instâncias criadas neste estudo de caso foram baseadas num *corpus* de páginas da Web relacionadas a eventos científicos onde os padrões foram estudados para a geração destas instâncias. A figura 6.6 é um exemplo de página que compõe o *corpus*. Esta página se refere a um evento científico na área de ciência da computação que foi classificada como referente a um "Simpósio". As informações relevantes que se deseja extrair dela são o título do evento, local e data do evento e datas importantes como *deadline* e data de notificação.



Call for papers: ICCV 2001

IEEE International Conference on Computer Vision

Vancouver, Canada
July 9-12, 2001

Program

ICCV is a single track conference consisting of high quality, previously unpublished papers, presented either orally or as a poster. Contributions are sought on original research on any aspect of computer vision, including but not restricted to the following topics:

- **Active and real-time vision**
- **Applications**
- **Color, illumination and texture**
- **Computer graphics and vision**
- **Early vision and image representation**
- **Image databases**
- **Learning in vision**
- **Matching and indexing**
- **Model acquisition**
- **Motion and video analysis**
- **Object, event, and scene recognition**
- **Robot vision**
- **Segmentation and grouping**
- **Stereo vision**

All reviewing will be carried out double blind by the Program Committee. The proceedings will be published by the IEEE Computer Society.

Submission Date: December 4, 2000

Manuscripts must be submitted in IEEE conference format and will be limited to 8 pages in length. See author instructions for full details on how to submit a paper.

Document: Done (1.229 secs)

Figura 6.6: Exemplo de página classificada como Simpósio

6.3.3 Criação de regras de extração

As regras de extração representam o conhecimento de como extrair dados baseando-se nas instâncias das ontologias. As regras foram implementadas segundo as etapas de extração definidas no capítulo 5: reconhecimento de entidades, análise de relacionamentos e contexto, inferência e classificação.

O reconhecimento de entidades baseia-se nas instâncias de `Data-Extractor` e `Concept` para identificar e extrair entidades que compõem o dado. Por exemplo, a instância do exemplo 6.3 apresenta termos que são procurados na página. Esta instância é premissa das regras de reconhecimento. Se for encontrado algum conceito, a regra pode ser disparada gerando um novo fato representado pela instância da classe `Data-Found`. O exemplo 6.6 mostra uma regra implementada em Jess que pode ser disparada por uma instância de `Concept`. A função `find-concepts` executa a busca do conceito na página web processada.

Exemplo 6.6 *Regra para reconhecer entidades que representam um dado.*

```
(defrule r_444_slots_me_ccpt_term
   ?fact-data ← (object (Importance MEDIUM) (is-a Data-Extractor)
                        (Slot-in-Process ?s) (Concepts $ ?cb)
                        (Slots-of-Web-Page $ ?sw))
   (test (find-concepts $ ?cb (beginning (slot-get [PAGE] $?sw)))
   ⇒
   (make-instance of Data-Found (Data-Extractor ?fact-data)))
```

A regra do exemplo acima foi simplificada para melhor compreensão de seu objetivo. Na instância de `Data-Found` criada também são guardadas informações sobre a posição do conceito encontrado, a instância de `Data-Extractor` que disparou a regra e outras informações de caráter operacional.

A análise de relacionamentos e contextos é feita de três maneiras: extração de dados independentes, dados relacionado e dados complementados por funções de extração. Dados como localidades e períodos de datas, quando localizados no início de página sobre eventos científicos, não precisam de complementos para indicar ao que se referem, na maioria dos casos. Um exemplo de regra que extrai este tipo de informação é mostrado em 6.7.

Exemplo 6.7 *Regra para extrair dados independentes de complemento.*

```
(defrule ex_complete_inf_hi
   ?fact-info ← (object (is-a Slot-Extractor) (Slot-in-Process ?slt-ip)
                        (In-the-Beginning TRUE) (Data-Extractor $?data))
   ?fact-data ← (object (is-a Data-Found) (Start-Position ?sip-fnd)
                        (Slots-of-Web-Page ?swp) (Data-Extractor (nth$ 1 $?data))
   (test (< ?sip-fnd 1000))
```

```
⇒
(make-instance of Information-Found (Slot-in-Process ?)
               (Data-Extractor ?fact-data)))
```

No exemplo acima observe que o fato `?fact-info` expressa que o dado encontrado deve estar no início da página. Esta verificação é feita através do teste de função (`(test (< ?sip-fnd 1000)`). Quando a regra é disparada, uma instância de `Information-Found` é criada.

O segundo caso de extração procura associar os fatos com o objetivo de montar a informação. Dois fatos relacionados podem compor uma informação dependendo de suas características estruturais dentro da página, como posição conceitos em sua vizinhaça, etc. O exemplo 6.8 mostra uma regra de extração de dados relacionados.

Exemplo 6.8 *Regra para associar instâncias de* `Data-Found` *que se relacionam.*

```
(defrule r_444_slots_me_ccpt_term
   ?fact-info ← (object (Importance MEDIUM) (is-a Information-Extractor)
                        (Slot-in-Process ?s) (Slots-of-Web-Page $ ?sw)
                        (Absent-Concepts $?abs-cpt) (distance ?dist))
   ?fact-data-1 ← (object (is-a Data-Found) (Data-Extractor (nth$ 1 $?data))
                        (Position ?fact-pos-1))
   ?fact-data-2 ← (object (is-a Data-Found) (Data-Extractor (nth$ 2 $?data))
                        (Position-fact-pos-2))
   (test (and (< (- ?fact-pos-2 ?fact-pos-1) ?dist))
            (not-occurs $?abs-cpt (substring ?fact-pos-1 ?fact-pos-2
                                    (Slot-get [PAGE] $?sw))))
   ⇒
   (make-instance of Information-Found (Slot-in-Process ?s)
      (Information-Extractor ?fact-info)))
```

Esta regra é disparada quando existem fatos que se enquadram com as premissas descritas nela. A instância do exemplo 6.5 preenche os requisitos considerados no fato `?fact-info` expresso em algumas premissas da regra. Os exemplos 6.3 e 6.4 representam os fatos `?fact-data-1` e `?fact-data-2`, respectivamente. Se a distância no texto entre as posições dos dois destes dois últimos fatos for menor que o especificado por `?dist` (Valor da distância máxima entre dois fatos relacionados) e não houver nenhum conceito indesejável entre eles (`(Absent-Concepts $?abs-cpt)`).

Outro caso de extração é quando um dado é complementado por dados extraídos por função. a função de extração é especificada pelo *slot* `function-call`. Um dado deste tipo tem seu complemento procurado em suas redondezas. É o caso da expressão "tópicos abordados" num evento científico que geralmente vem seguida de uma lista de assuntos. Uma função é usada para extrair estes itens da lista. Em 6.9 mostramos um exemplo desta regra.

Exemplo 6.9 *Regra para complementar um dado de* `Data-Found` *para compor a informação.*

```
(defrule ex_703_data_functions
   ?info-extr ← (object (is-a Information-Extractor) (Slot-in-Process ?slt-ip)
                        (Data-Extractor ?data) (Related-Function ?slt-rf)
                        (Absent-Keywords $?abs-keys))
   ?fact-data ← (object (is-a Data-Found) (Values ?vl-fnd)
                        (Start-Position ?sp-fnd) (Slots-of-Web-Page ?swp-fnd))
   (test (and (neq ?slt-rf nil)
            (run-function (create$ ?slt-extr ?slt-pg ?slt-fnd))))
   ⇒
   (make-instance of Information-Found (Slot-in-Process ?slt-ip)
            (Information-Extractor ?fact-info)))
```

Nesta regra, um teste é feito verificando se exite alguma função a ser executada e se seu resultado retorna verdadeiro disparando a regra e criando a instância sobre a informação extraída. Na figura 6.6 temos um exemplo de informação que pode ser extraída através de uma função.

Exemplo 6.10 *Regra que delimita uma informação baseando-se em fatos já encontrados.*

```
(defrule ex_704_delimited_by_concepts_bgn
   ?fact-extr ← (object (is-a Information-Extractor) (Slot-in-Process ?slt-ip)
                        (Neighbor-Data-Extractor $?ngbr-de)
                        (String-Range ?extr-sr) (Absent-Concepts ?ac-extr))
   ?fact-data-1 ← (object (is-a Data-Found) (Start-Position ?fact-pos-1))
                        (Data-Extractor (nth$ 1 $?extr-nc))
   ?fact-data-2 ← (object (is-a Data-Found) (Data-Extractor (nth$ 2 $?extr-nc))
                        (Start-Position ?fact-pos-2))
   (test (< (abs (- fact-pos-1 ?fact-pos-2)) ?extr-sr)))
   ⇒
   (make-instance of Information-Found (Slot-in-Process ?slt-ip)
                (Information-Extractor ?fact-extr)))
```

No exemplo acima dois fatos delimitam a informação a ser extraída. Este fatos devem ser dados extraídos através de instâncias definidas no *slot* `Neighbor-Data-Extractor`. A informação extraída também tem que ter seu tamanho máximo verificado.

6.3.4 Reusabilidade do conhecimento

As regras descritas acima baseiam-se em fatos representados em instâncias das classes das ontologias. Isto permite que as regras sejam usadas para extrair diversos tipos de informações sem que

informação	no texto	extração	correto?
data inicial	July 9-12, 2001	9, junho, 2001	Sim
Data final	July 9-12, 2001	12, junho, 2001	Sim
Local	Vancouver, Canada	Canada	Sim
Nome do evento	International Conference on Computer Vision	Conference on Computer Vision	Sim
Data de submissão	Submission date: December 4, 2000	"Submission date" 4, december, 2000	Sim
subjects	vide figura 6.6	topics Applications Color, illumination and texture Computer graphics and vision Computer graphics and vision Image databases Learning in vision Matching and indexing Model acquisition Motion and video analysis Object, event, and scene recognition Robot vision Segmentation and grouping Stereo vision	Sim

Tabela 6.1: Amostra de resultado da extração na página da figura 6.6.

seja necessário alterar seu código. Apenas a entrada de novas instâncias sobre o conhecimento é necessária para a extração de novas informações, facilitando a construção de novos extratores e sua manutenção.

Outra característica destacada aqui é que as instâncias criadas abrangem apenas o conhecimento sobre a extração, tornando a ontologia sobre a ciência independente. Esta última apenas descreve as possíveis classes de páginas e seus atributos a serem extraídos. Quando um extrator é construído as instâncias de `Information-Extractor` indicam a qual classe e *slot* a informação a ser extraída se refere. Isto caracteriza a discriminação do conhecimento que permite que partes do conhecimento sejam construídos ou atualizados de forma independente e reusados para outros fins como é o caso da ontologia sobre o domínio.

6.4 Amostra de extração

Nesta seção, apresentamos, como exemplo, um resultado de extração de informações da página apresentada na figura 6.6. A página apresenta informações sobre uma chamada de trabalho para uma conferência que ocorreu no Canadá no período de 9 a 12 de julho de 2001. A página traz o nome do evento, o local e a data de realização, uma lista de tópicos que devem ser abordados pelos trabalhos e uma data limite para a entrega dos trabalhos submetidos. É desejável que o sistema extraia todas as informações relevantes descritas acima.

Inicialmente a página é classificada pelo MASTER-Web como uma conferência. A condição "classificada" da página dá início à extração das informações pré-definidas nas instâncias da ontologia (classes `Concept`, `Data-Extractor` e `Information-Extractor`). O resultado obtido no processo de extração é mostrado na tabela 6.1.

Os resultados foram extraídos por regras como as exemplificadas acima. Para cada data extraída, é criada uma instância da classe `Date` pertencente a uma ontologia auxiliar. Esta classe contém *slots* do dia, mês e ano e dia da semana. Desta forma, as datas extraídas não seguem um formato específico. A informação extraída sobre o local do evento é o país relacionado. O período do evento é desmembrado em data inicial e data final. A data de submissão dos trabalhos é extraída através da composição dos dados representados por "Submission Date" e "December 4, 2000". O título foi extraído a partir da delimitação da informação por dados relevantes. Apesar de ter extraído no exemplo acima, esta forma de delimitar a informação tem sua eficiência dependente do número de dados que se está extraíndo da página e a regularidade com que a informação se apresenta. Títulos de eventos científicos, geralmente, são delimitados por siglas, cidades, tipo do evento ou a data do evento e a ordem em que estes termos são dispostos na vizinhança do título. Durante os testes este tipo de extração não apresentou um bom resultado devido ao número de informações extraídas durante os experimentos. Seria necessário um número maior de atributos extraídos para delimitar este tipo de informação, como as siglas dos eventos e cidades.

6.5 Experimentos e resultados

Os experimentos foram realizados inicialmente utilizando um composto de 133 páginas classificadas pelo MASTER-Web. Este *corpus* foi capturado da Web utilizando robôs de busca. Este *corpus* foi utilizado para a aquisição de conhecimento sobre a extração, onde foram verificados os padrões de apresentação da informação no texto. Cinco informações foram consideradas para executar os experimentos: *localidade, período, data de submissão, data de notificação* e *tópicos*. Para cada uma delas foram criadas instâncias de `Concept`, `Data-Extractor` e `Information-Extractor`, como as dos exemplos mostrados no decorrer deste capítulo.

O critério de avaliação do sistema de extração foi baseado no descrito em [Grishman, 1997]. Três variáveis são consideradas aqui: N_{pagina} expressa o número de informações apresentadas na página; $N_{extraido}$ é o número de informações extraídas da página; e $N_{correto}$ se refere ao número de informações extraídas corretamente.

A partir deste critério é possível avaliar a cobertura e precisão do sistema. Em EI, cobertura se refere à relação pecentual entre a quantidade de informações extraídas corretamente e a quantidade de informações apresentadas na página e é expressa pela fórmula:

$$cobertura = \frac{N_{extraido}}{N_{pagina}}.$$

A precisão se refere à relação entre a quantidade de informações extraídas corretamente e o número de informações estraídas, como podemos ver a seguir:

$$precisão = \frac{N_{correto}}{N_{pagina}}.$$

Durante os testes cada forma de extração foi avaliada individualmente. Os resultados são mostradros na tabela 6.2. A medida em que se processava a extração ajustes eram feitos para efeito de correção. O resultado global deste teste é mostrado na tabela 6.3.

A avaliação individual se refere a cada informação considerada neste experimento. Períodos e países, quando encontrados juntos e, de preferência no início da página, refererem-se ao local e a data do evento científico (na maioria dos casos), portanto, sua semântica é pré-definida, já que a página foi classificada como evento científico. Em alguns casos os períodos não foram reconhecidos devido à sua formatação (datas com padrões peculiares) e à formatação da página (âncoras e marcadores HTML distanciavam as duas informações, apesar de aparecerem no texto muito próximas).

Informações compostas por mais de um dado, no caso data de submissão e data de notificação que indicam ao que se referem as datas relacionas a elas. Cada dado referente a uma data tem seus relacionamentos verificados na intenção de detectar seu contexto. Neste caso também houve conceitos não encontrados e extraídos incorretamente. Entretanto, a precisão do sistema se mostrou bastante satisfatória. A escolha dos atributos data limite e data de notificação não se deu ao acaso. Nossa intenção foi analisar se as regras relacionariam as datas aos seus respectivos contextos de maneira correta.

Listas de tópicos foram extraídas com o auxílio de funções de extração de texto. Esta função baseia-se em marcadores da linguagem HTML para extrair listas. Esta extração apresentou níveis satisfatórios de extração.

	Cobertura	***Precisão***
Local	74,07%	74,07%
Período	86,49%	81,08%
Data limite	78,04%	70,13%
Data de aceitação	93,75%	81,25%
Lista de tópicos	66,67%	59,56%

Tabela 6.2: Resultados individuais da extração do primeiro *corpus* de teste.

Cobertura	***Precisão***
79,85%	72,66%

Tabela 6.3: Resultados globais da extração do primeiro *corpus* de teste.

Um segundo teste foi executado em um novo *corpus* de página, sendo que, desta vez, nenhum ajuste foi feito. O objetivo deste teste é fazer uma avaliação final do sistema de extração. Os resultados

obtidos não diferem muito do primeiro teste. Em alguns casos a cobertura e precisão aumentaram. Isto ocorreu devido a ajustes feitos nas instâncias de classes das ontologias após o primeiro teste, tendo em vista que este foi realizado apenas visando aumentar a eficiência do sistema. A tabela 6.4 mostra os resultados de cada forma de extração utilizada no experimento. Os valores estatísticos globais deste segundo teste são mostrados na tabela 6.5.

	Cobertura	***Precisão***
Local	68,75%	68,75%
Período	78,49%	57,89%
Data limite	75%	71,43%
Data de aceitação	88,89%	77,78%
Lista de tópicos	70%	59,99%

Tabela 6.4: Resultados individuais da extração considerando o segundo *corpus* de teste.

Cobertura	***Precisão***
75,6%	67,06%

Tabela 6.5: Resultados globais da extração considerando o segundo *corpus* de teste.

6.6 Discussão

Estes resultados comprovam a viabilidade do sistema de EI para MASTER-Web. O uso de ontologias pemitiu que, durante a aquisição de conhecimento, fossem feitas mínimas alterações em regras e função (poucos casos). A grande maioria das modificações estava relacionada à instâncias criadas durante a entrada de conhecimento. Vale ressaltar que este processo não aplica técnicas de aprendizado nem usa dicionário de termos (exceto pela lista de países e estados americanos) que aumentaria significativamente a precisão do sistema.

Como referência comparamos os resultados obtidos com os resultados apresentados pelo sistema DEADLINER [Kruger et al., 2000] que faz a extração de informação em páginas de chamadas de trabalho. Os atributos extraídos[1] são data inicial e final do evento, data limite e país. A tabela 6.6 mostra os resultados da extração. A eficiência deste sistema é uma das melhores entre os sistemas de EI alcançando índices de reconhecimento acima de 95% e índices de extração de no mínimo 70%, podendo chegar a 86%.

O DEADLINER é um sistema de EI que classifica e extrai dados de páginas oriundas de grupos

[1]Outros atributos são extraídos, mas consideramos aqui apenas os que são de nosso interesse para a comparação.

Data inicial	*Data final*	*Data limite*	*País*
73%	71%	87%	77,5%

Tabela 6.6: Resultados obitidos pelo sistema DEADLINER.

de notícias (do inglês, *newsgroups*) e *e-mails* enviados a listas. Para a tarefa de extração, o DEADLINER utiliza técnicas de aprendizado. Em nosso caso, toda entrada de conhecimento é feita de maneira artesanal. Entretanto, como foi mostrado neste trabalho, o uso de ontologias permite que o conhecimento seja adquirido de maneira simples evitando desenvolvimento e manutenção em nível de implementação. O MASTER-Web tem a abrangência como vantagem sobre o DEADLINER. Enquanto que sua busca é limitada a grupos de notícias, universidades e e organizações profissionais, o MASTER-Web aceita quaisquer páginas de chamada de trabalhos, jornais, *workshops*, etc. Isto é possível devido à precisão deste sistema durante a classificação.

Capítulo 7

Conclusões e trabalhos futuros

A Web levanta questões relacionadas ao gerenciamento de informação devido à sua natureza aberta e distribuída, a como usufruir ao máximo das informações disponibilizadas nela e a como adicionar contexto à informação. Um dos principais motivos destes problemas se dá pela falta de semântica das páginas da Web. Engenhos de busca baseados em Recuperação de Informação (RI) não indexam páginas pelo contexto do assunto abordado, ou seja, não provêem semântica à Web que é algo desejado tanto para usuários humanos quanto para sistemas baseados em conhecimento.

Este trabalho apresentou um sistema de Extração de Informação (EI) baseado em ontologias e *wrappers* como o objetivo de extender as funcionalidades do Sistema MASTER-Web que tinha como objetivo extrair dados para classificar uma página considerando um domínio de assunto específico no qual a página estaria inserida. A adição desta tarefa de extração de informação permite, além da classificação, que as informações desejadas sejam extraídas de maneira completa e correta na maioria dos casos, tornando páginas, inicialmente não estruturadas, em páginas estruturadas.

A tarefa de extração de informação desenvolvida neste trabalho foi baseada na abordagem de Engenharia de Conhecimento usando ontologias como formalismo de representação de conhecimento. Técnicas baseadas em *wrappers* foram adotadas para executar a extração. A combinação de ontologias e *wrappers* permitiu que o sistema proposto extraísse informações baseando-se não apenas em palavras-chaves, marcadores HTML e valores estatísticos (no caso de *wrappers* tradicionais) como também baseando-se em características que determinam o contexto da informação que se deseja extrair. Em suma, o trabalho aqui desenvolvido adiciona semântica à informação extraída por um *wrapper*.

O uso de ontologias permitiu a representação de conhecimento declarativo, separando a informação do processo de extração e tornando o conhecimento sobre a informação independente do código que a manipula. O conhecimento sobre como extrair informações foi representado por regras de produção. Isto permitiu uma melhor compreensão sobre a estrutura da infomação a ser extraída, bem

como uma maior facilidade na construção e manutenção do conhecimento. O conhecimento representado por ontologias foi dividido em três tipos: conhecimento sobre o domínio, conhecimento sobre a página e informações extraídas, e conhecimento sobre a informação a ser extraída. A discriminação do conhecimento adicionou características ao sistema de extração como flexibilidade, portabilidade e reusabilidade, tanto da parte de ontologias como também de regras. De fato, as regras aqui desenvolvidas inferem sobre fatos representados por instâncias das classes que compõem as ontologias independente dos valores que elas carreguem.

O processo de extração foi dividido em etapas que extraem a informação no sentido de baixo para cima, ou seja, ela é composta por dados que juntos constituem uma informação. Para esta etapa, consideramos que a informação tem a seguinte definição (seção 5.4.2):

$$\text{informação} = \mathit{contexto} + \mathit{dado}.$$

Um estudo de caso foi realizado com o objetivo de avaliar o desempenho do sistema proposto. Para isto, o sistema foi utilizado para extração de informação de páginas referentes a chamadas de trabalhos científicos. Os resultados obtidos mostram que o sistema é viável. O uso de ontologias permitiu a representação e entrada de conhecimento sobre a extração de atributos relevantes sobre chamadas de trabalho. Como comparativo, apresentamos resultados semelhantes alcançados pelo sistema DEADLINER, bem como algumas de suas características de extração, diferenciando os dois sistemas.

Este trabalho trouxe contribuições considerando que sua principal meta foi desenvolver um sistema de EI que usufruísse das vantagens da representação declarativa do conhecimento. A seguir os benefícios oferecidos pelo trabalho:

1. O sistema extrai informações de páginas previamente classificadas pelo MASTER-Web, permitindo que tal página, antes contendo informações não estruturadas, seja representada por informações estruturadas. Estas informações oferecem maior legibilidade da informação para sistemas baseados em conhecimento;

2. Portabilidade e reusabilidade são duas contribuições significativas deste trabalho. Com o uso de ontologias foi possível separar conhecimento do processo de manipulação do conhecimento, permitindo maior clareza na representação do conhecimento ainda podendo ser reusado para outros fins (como é o caso da ontologia do domínio);

3. O sistema provê maior facilidade tanto na configuração do sistema para a extração em classes de páginas diferentes, como na entrada de conhecimento nas ontologias. Isto foi possível por causa da discriminação do conhecimento usado no processo de extração;

4. A extração de informação funciona como tarefa complementar dos sistemas de recuperação de informação, adicionando contexto às páginas recuperadas através da extração de informações semânticas das páginas;

5. É possível prover ao usuário final um sistema de busca de um domínio de assunto específico onde se poderá, através de um mediador, oferecer resultados de consultas menos ruidosos e mais precisos. Estas consultas seriam realizadas pelo usuário através de especificações de atributos como em uma linguagem estruturada.

Tendo em vista que ainda existem muitos tópicos em aberto na área de EI e que podem ser explorados a partir deste trabalho, sugerimos algumas extensões como trabalhos futuros:

1. O uso de linguagens axiomáticas permitirão a definição de relacionamentos mais complexos entre as classes e seus atributos criadas no Protégé-2000. A linguagem PAL (Protégé Axiomatic Language) permite a definição de axiomas no Protégé-2000. Desta forma é possível definir relacionamentos e restrições na própria ontologia, como a verificação de consistência entre datas;

2. Cooperação de informações entre os agentes permitirá que, além de âncoras apresentadas nas páginas, informações interessantes sejam enviadas a outros agentes de extração;

3. Bancos de dados relacionais não dão suporte ao armazenamento de estruturas baseadas em quadros, pois não permitem que listas sejam armazenadas em suas tabelas. Representação baseada em quadros fere a primeira forma normal de relacionamento entre tabelas. Como trabalho futuro é sugerida a pesquisa de outras formas de armazenamento da informação, como banco de dados orientados a objetos, por exemplo;

4. Adição de técnicas de PLN visando abranger textos menos estruturados. Com a adição da PLN será possível extrair informações de páginas que não são "bem comportadas"quanto a sua formatação. Neste caso, é interessante a aplicação de técnicas de aprendizado;

5. O uso de técnicas de aprendizado permitirá maior eficiência na aquisição de conhecimento, bem como no processo de extração. A inclusão de aprendizado automático pode auxiliar a avaliação das regras de extração e regras de sugestão trocadas entre agentes;

6. O MASTER-Web pode auxiliar na construção de estruturas semânticas para páginas de uma determinada classe. Com o surgimento da rede semântica (do inglês, *Semantic Web* semântica), ferramentas de geração automática de semântica são desejáveis. O MASTER-Web, além de extrair informações é portável entre domínios de assuntos, cuja apresentação em páginas segue padrões semi-estruturados.

Referências Bibliográficas

[Alexander et al., 1986] Alexander, J. H., Freiling, M. J., Shulman, S. J., e J. L. Staley, S. Rehfus, S. L. M. (1986). Knowledge level engineering: Ontological analysis. Em *Proceedings of the 5th National Conference on Artificial Intelligence – AAAI–86*, páginas 963–968. Morgan Kaufmann Publishers. Philadelphia.

[Amann e Fundulaki, 1999] Amann, B. e Fundulaki, I. (1999). Integrating ontologies and thesauri to build RDF schemas. Em *European Conference on Digital Libraries*, páginas 234–253.

[Appelt e Israel, 1999] Appelt, D. E. e Israel, D. J. (1999). Introduction to information extraction tecnology. Em *Proceedings of the 16th International Joint Conference on Artificial Intelligence.*

[Barros e Robin, 1996] Barros, F. e Robin, J. (1996). Processamento de linguagem natural. Em *Jornada de Atualização em Informática – Sociedade Brasileira de Computação – JAI/SBC*, Recife–Brasil.

[Benjamins et al., 1998] Benjamins, V., Fensel, D., e Perez, A. (1998). Knowledge management through ontologies.

[Berners-Lee e Fischetti, 1999] Berners-Lee, T. e Fischetti, M. (1999). Weaving the web: the original design and ultimate destiny of the world wide web by its inventor. Haper, San Francisco.

[Bittencourt, 1998] Bittencourt, G. (1998). *Inteligência Artificial: ferramentas e teorias*. Ed. da UFSC.

[Bray et al., 1997] Bray, T., Paoli, J., e Sperberg-McQueen, C. (1997). Extensible Markup Language (XML). *The World Wide Web Journal*, 2(4):29–66.

[Califf, 1998] Califf, M. E. (1998). Relational learning techniques for natural language extraction. Relatório Técnico AI98-276, University of Texas, Austin, TX.

[Cohen e Singer, 1996] Cohen, W. e Singer, Y. (1996). Learning to query the web. Em *Workshop on Internet-Based Information Systems – AAAI-96*.

[Corcho e Pérez, 2000] Corcho, O. e Pérez, A. G. (2000). A roadmap to ontology specification languages. ECAW2000 – http://delicias.dia.fi.upm.es/articulos/ocorcho/ekaw2000-corcho.pdf.

[Crescenzi e Mecca, 1998] Crescenzi, V. e Mecca, G. (1998). Grammars have exceptions. *Information Systems*, 23(8):539–565.

[Crescenzi et al., 2001] Crescenzi, V., Mecca, G., e Merialdo, P. (2001). Roadrunner: Towards automatic data extraction from large web sites. Em *Proceedings of 27th International Conference on Very Large Data Bases*, páginas 109–118, Roma, Itália.

[Croft e Lewis, 1991] Croft, W. B. e Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. Em *Proceedings of SIGIR 1991*, páginas 32–45, New York.

[Cunningham, 1997] Cunningham, H. (1997). Information extraction - a user guide. "Research Memo CS-97-02, University of Sheffield, Sheffield".

[Decker et al., 2000] Decker, S., Fensel, D., van Harmelen, F., Horrocks, I., Melnik, S., Klein, M. C. A., e Broekstra, J. (2000). Knowledge representation on the web. Em *Description Logics*, páginas 89–97.

[Dillon, 1983] Dillon, M. (1983). A fully automatic syntactically based indexing system. J. Am Soc Infi Sci. 34, 2, 99-108.

[Eikvil, 1999] Eikvil, L. (1999). Information extraction from world wide web - a survey. Relatório Técnico 945, Norweigan Computing Center.

[Embley et al., 1999a] Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Ng, Y.-K., Quass, D., e Smith, R. D. (1999a). Conceptual-model-based data extraction from multiple-record web pages. *Data Knowledge Engineering*, 31(3):227–251.

[Embley et al., 1998] Embley, D. W., Campbell, D. M., Smith, R. D., e Liddle, S. W. (1998). Ontology-based extraction and structuring of information from data-rich unstructured documents. Em *CIKM*, páginas 52–59.

[Embley et al., 1999b] Embley, D. W., Fuhr, N., Klas, C.-P., e Rölleke, T. (1999b). Ontology suitability for uncertain extraction of information from multi-records web documents. *Datenbank Rundbrief*, 24:48–53.

[Eriksson, 2000] Eriksson, H. (2000). Jesstab plugin for protégé. Dept. of Computer and Information Science, Linköping University. http://www.ida.liu.se/ her/JessTab.

[Fagan, 1989] Fagan, J. (1989). The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. J. Arm Sac. Inj! Sci. 40, 2, 115-132.

[Farquhar et al., 1996] Farquhar, A., Fikes, R., e Rice, J. (1996). The ontolingua server: A tool for collaborative ontology construction.

[Fensel et al., 2000] Fensel, D., Horrocks, I., van Harmelen, F., Decker, S., Erdmann, M., e Klein, M. C. A. (2000). OIL in a nutshell. Em *Knowledge Acquisition, Modeling and Management*, páginas 1–16.

[Finin et al., 1994] Finin, T., Fritzson, R., McKay, D., e McEntire, R. (1994). KQML as an Agent Communication Language. Em Adam, N., Bhargava, B., e Yesha, Y., editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, páginas 456–463, Gaithersburg, MD, USA. ACM Press.

[Freitag, 1998] Freitag, D. (1998). *Machine Learning for Information Extraction in Informal Domains*. Tese de Doutorado, Carnegie Mellon University.

[Freitas, 2001] Freitas, F. (2001). Ontology of science. disponível em http://protege.stanford.edu/plugins/ontologyOfScience/ontology_of_science.htm.

[Freitas, 2002] Freitas, F. (2002). *Sistemas multiagentes cognitivos para a recuperação e extração integradas de informação da web*. Tese de Doutorado, Programa de pós graduação em Engenharia Életrica/UFSC, Florianópolis.

[Freitas e Bittencourt, 2002] Freitas, F. e Bittencourt, G. (2002). Comunicação entre agentes em ambientes distribuídos abertos: o modelo peer-to-peer. *Revista eletrônica de iniciação científica (REIC)*, II(II).

[Freitas e Bittencourt, 2003] Freitas, F. e Bittencourt, G. (2003). An ontology-based architecture for cooperative information agents. Em *Proceedings of International Joint Conferences on Artificial Intelligence 2003 – IJICAI'03*, Alacapuco, Mexico. Artigo aceito com publicação prevista para agosto de 2003.

[Friedman-Hill, 2000] Friedman-Hill (2000). Jess, the java expert system shell. http://herzberg.ca.sandia.gov/Jess.

[Genesereth e Fikes, 1992] Genesereth, M. R. e Fikes, R. E. (1992). Knowledge Interchange Format, Version 3.0 Reference Manual. Relatório Técnico Logic-92-1, Stanford, CA, USA.

[Grishman, 1997] Grishman, R. (1997). Information extraction: Techniques and challenges. Em *SCIE*, páginas 10–27.

[Gruber, 1993] Gruber, T. (1993). A translation approach to portable ontology specifications. Relatório técnico, Knowledge Systems Laboratory. Computer Science Department, Stanford University Stanford.

[Hammer et al., 1997] Hammer, J., McHugh, J., e Garcia-Molina, H. (1997). Semistructured data: The tsimmis experience. Em *Advances in Databases and Information Systems*, páginas 1–8.

[Information, 2000] Information, O. O.-B. (2000). Ist project ist-2000-29243 ontoweb ontoweb: Ontology-based information exchange for knowledge management and electronic commerce d21 successful scenarios for ontology-based applications v1.0.

[Jardine, 1997] Jardine, D. A. (1997). The ansi/sparc dbms model. Em *Proceedings of the Second SHARE Working Conference on Data Base Management Systems*, páginas 26–30, Montreal, Canada.

[Karp et al., 1999] Karp, P. D., Chaudhri, V. K., e Thomere, J. (1999). Xol: An xml-based ontology exchange language. ftp://smi.stanford.edu/pub/bio-ontology/xol.doc.

[Kifer et al., 1990] Kifer, M., Lausen, G., e Wu, J. (1990). Logical foundations of object-oriented and frame-based languages. Relatório Técnico TR-90-003.

[Koster, 1993] Koster, M. (1993). Guidelines for robot writers. Electronically available at http://www.robotstxt.org/wc/guidelines.html.

[Kruger et al., 2000] Kruger, A., Giles, C. L., Coetzee, F., Glover, E., Flake, G., Lawrence, S., e Omlin, C. (2000). DEADLINER: Building a new niche search engine. Em *Ninth International Conference on Information and Knowledge Management, CIKM 2000*, Washington, DC.

[Laender et al., 2002] Laender, A., Ribeiro-Neto, B., Silva, A., e Teixeira, J. (2002). A brief survey of web data extraction tools. Em *SIGMOD Record*, volume 31.

[Laender et al., 2000] Laender, A. H. F., Ribeiro-Neto, B. A., da Silva, A. S., e Silva, E. E. (2000). Representing web data as complex objects. Em *EC-Web*, páginas 216–228.

[Lassila e Swick, 2003] Lassila, O. e Swick, R. (2003). Resource description framework (rdf) model and syntax specification. W3C Working Draft WD-rdf-syntax-19981008. http://www.w3.org/TR/WD-rdf-syntax.

[Luke e Heflin, 2000] Luke, S. e Heflin, J. (2000). Shoe 1.01. proposed specification. shoe project. http://www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm.

[MacGregor, 1991] MacGregor, R. (1991). Inside the loom description classifier. SIGART Bulletin, 2(3):88 – 92.

[McGuinness, 2002] McGuinness, D. (2002). Ontologies come of age. Em Fensel, D. e Hendler, J., editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, chapter 6. MIT Press.

[Minsky, 1975] Minsky, M. (1975). A framework for representing knowledge. Em *Psicology of Computer Vision*, páginas 211–281. McGraw-Hill.

[Motta, 1998] Motta, E. (1998). An overview of the ocml modelling language. Em *8th Workshop on Knowledge Engineering: Methods & Languages KEML 98*, Karlsruhe, Germany.

[Muslea, 1998] Muslea, I. (1998). Extraction patterns: from information extraction to wrapper generation. Relatório técnico, Information Sciences Institute, University of Southern California.

[Muslea et al., 2001] Muslea, I., Minton, S., e Knoblock, C. A. (2001). Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):93–114.

[Noy et al., 2000] Noy, N. F., Fergerson, R., e Musen, M. (2000). The knowledge model of protege-2000: Combining interoperability and flexibility.

[Noy e MacGuinness, 2001] Noy, N. F. e MacGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

[Pirolli et al., 1996] Pirolli, P., Pitkow, J., e Rao, R. (1996). Silk from a sow's ear: Extracting usable structures from web. Em *proceedings of chi96 – ACM*.

[Riley, 1999] Riley, G. (1999). Clisp: A tool for building expert systems. http://www.ghg.net/clips/CLIPS.html.

[Riloff, 1993] Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. Em *National Conference on Artificial Intelligence*, páginas 811–816.

[Riloff e Lehnert, 1994] Riloff, E. e Lehnert, W. (1994). Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333.

[Sahami et al., 1997] Sahami, M., Yusufali, S., e Baldonado, M. (1997). Real-time full-text clustering of networked documents. Em *Proceedings da Conferência do AAAI-97 –EUA*.

[Sanderson, 1994] Sanderson, M. (1994). Word sense disambiguation and information retrieval. Em *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, páginas 49–57, Dublin, IE.

[SHriMP, 2003] SHriMP (2003). Jambalaya. SHriMP Research Group , Department of Computer Science, University of Victoria. http://shrimp.cs.uvic.ca/downloads_jambalaya_clear.htm.

[Silva et al., 2002] Silva, T. M. S., Freitas, F., e Bittencourt, G. (2002). Extração de informação no master-web baseada em ontologias. Em *Anais do XIII Simpósio Brasileiro de Informática em Educação*, São Leopoldo, Brasil. Sociedade Brasileira de Computação – SBC.

[Sintek, 2003] Sintek, M. (2003). Ontoviz tab: Visualizing protégé ontologies. http://protege.stanford.edu/plugins/ontoviz/ontoviz.html.

[Smith e Welty, 2001] Smith, B. e Welty, C. (2001). Ontology: Towards a new synthesis. Em *Proceedings of the international conference on Formal Ontology in Information Systems*, páginas 3–9, Ogunquit, Maine. ACM Press, New York, NY, USA.

[Soderland, 1999] Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272.

[Sowa, 1984] Sowa, F. J. (1984). *Conceptual Structures. Information Processing in Mind and Machine*. Addison Wesley.

[van Heijst et al., 1997] van Heijst, G., Schreiber, A., e Wielinga, B. (1997). Using explicit ontologies in kbs development. Department of Social Science Informatics, University of Amsterdam.

[Villain, 1999] Villain, M. (1999). Inferential information extraction. Em Springer-Verlag, editor, *Information Extraction – Towards Scalable, Adaptable Systems*.

[Wee et al., 1998] Wee, A., Tong, L. C., e Tan, C. L. (1998). Knowledge representation issues in information extraction. Em *5th Pacific Rim International Conference on Artificial Intelligence*, páginas 448–458.

[Widenius e Axmark, 2002] Widenius, M. e Axmark, D. (2002). *MySQL Reference Manual*. O'Reilly and Associates.

[Wirth, 1977] Wirth, N. (1977). What can we do about the unnecessary diversity of notation for syntactic definitions. Comm. ACM 20:11 pp. 822-823.

[Yahoo, 2003] Yahoo (2003). Engenho de busca yahoo. http://www.yahoo.com.

[Zechner, 1997] Zechner, K. (1997). A literature survey on information extraction and text summarization.