

**A Abordagem POESIA para a Integração
de Dados e Serviços na Web Semântica**

Renato Fileto

Tese de Doutorado

A Abordagem POESIA para a Integração de Dados e Serviços na Web Semântica

Renato Fileto¹

Outubro de 2003

Banca Examinadora:

- Prof^a. Dr^a. Claudia Bauzer Medeiros (Orientadora)
- Prof^a. Dr^a. Ana Carolina Salgado
Centro de Informática—UFPE
- Prof. Dr. Caetano Traina Júnior
ICMSC—USP
- Prof. Dr. Calton Pu
College of Computing, Georgia Tech (USA)
- Prof. Dr. Célio Cardoso Guimarães
- Prof. Dr. Edmundo Madeira
- Prof. Dr. João Carlos Setúbal

¹Este trabalho recebeu apoio da Embrapa, CAPES, Funcamp e dos projetos MCT/PRONEX-SAI e Web-Maps do CNPq. Os co-orientadores do Georgia Tech foram parcialmente suportados pelos programas *Operating Systems* e *ITR* (divisão *CISE/CCR*) da *NSF*, por um contrato do programa *SciDAC* da *DoE* e um contrato do programa *PCES (IXO)* da *DARPA*.

Substitua pela ficha catalográfica

A Abordagem POESIA para a Integração de Dados e Serviços na Web Semântica

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Renato Fileto e aprovada pela Banca Examinadora.

Campinas, 1 de Dezembro de 2003.

Prof^ª. Dr^ª. Claudia Bauzer Medeiros
(Orientadora)

Tese apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Substitua pela folha com a assinatura da banca

© Renato Fileto, 2003.
Todos os direitos reservados.

*Quiero vivir en un mundo en el
que los seres sean solamente humanos,
sin mas títulos que esse,
sin darse en la cabeza con una regla,
con una palabra, con una etiqueta...*

*Quiero que la gran mayoría, la única mayoría,
todos puedan hablar, leer, escuchar, florecer.
No entendí la lucha
sino para que esta termine.
No entendí nunca el rigor,
sino para que el rigor no exista.*

*He tomado un camino porque ese camino
nos lleva a todos a la felicidad duradera...
Luchó por esa bondad ubicua, extensa, inexhaustible.*

*De todo lo vivido,
me queda una fe absoluta en el destino humano.
Una convicción cada vez más consciente
de que nos acercamos a una gran ternura.*

*En este minuto crítico,
en este parpadeo de agonía,
sabemos que entrar la luz definitiva
por los ojos entreabiertos.*

*Nos entenderemos todos.
Progresaremos juntos.
Y esta esperanza es irrevocable.*

Pablo Neruda – *Confieso que he vivido*

*Aos meus avós, Lázaro (in memoriam) e Benedita,
cujo exemplo de amor, serenidade e dedicação,
permite-nos ao menos contemplar
a forma mais sublime de sabedoria.*

Agradecimentos

Aos meus pais, minha irmã e todos os meus familiares e amigos, que souberam amar-me da forma como sou, apoiando-me para atingir meus próprios objetivos e orgulhando-se do que eu possa realizar de bom.

À Profa. Claudia Bauzer Medeiros, pessoa de admirável capacidade e do mais elevado caráter. Obrigado, de coração, pela atenção, paciência e oportunidades concedidos a todos os privilegiados em tê-la como orientadora.

Aos meus co-orientadores, Calton Pu e Ling Liu, pela atenção e estímulo que me dispensaram durante a estada no Georgia Tech e até os dias atuais.

Aos funcionários da Unicamp e do Georgia Tech, que são capazes de desempenhar suas funções com gentileza suficiente até para dissolver as minhas inquietudes.

Aos colegas da Unicamp e do Georgia Tech, que compartilham seus conhecimentos, empregam muito do seu tempo em benefício de todos e tornam esses ambientes acadêmicos mais agradáveis e divertidos.

Aos amigos com quem sempre posso contar: Evandro, Fátima, Nilza, Dna. Terezinha, Nair, Daniel, Mirja, Sandro, Zé Luiz, Arnold, Marcos, Felipe, Eric, Jean, Yoko, Emmanuel, Ellen, Cláudio, Ana, Hend, ... Todos têm me ensinado muito, tornado minha vida mais bela e demonstrado a grandeza e a similaridade da essência e das aspirações humanas, independentemente de raça, cultura, idade e outros atributos.

À Embrapa, pelo apoio e confiança, incluindo a manutenção dos vencimentos e a ajuda de colegas competentes, especialmente no entendimento das necessidades das aplicações. À CAPES pelo auxílio e bom atendimento durante o doutorado sanduíche. Ao CNPq, que viabilizou a participação em diversos eventos científicos e contribui para a manutenção da boa infraestrutura do laboratório LIS, através dos projetos MCT/PRONEX-SAI (Sistemas Avançados de Informação) e WebMaps.

Ao povo brasileiro, que continua sustentando tudo isso; especialmente aos mais humildes, que às vezes nem têm consciência disso. Que esta nação aprenda a distribuir oportunidades para todos. Que vá além do desenvolvimento econômico, tecnológico, científico e cultural, para dar sua singular contribuição à humanidade – mostrar caminhos para o convívio afetuoso das pessoas, dentro de toda diversidade.

Resumo

POESIA (*Processes for Open-Ended Systems for Information Analysis*), a abordagem proposta neste trabalho, visa a construção de processos complexos envolvendo integração e análise de dados de diversas fontes, particularmente em aplicações científicas. A abordagem é centrada em dois tipos de mecanismos da Web semântica: workflows científicos, para especificar e compor serviços Web; e ontologias de domínio, para viabilizar a interoperabilidade e o gerenciamento semânticos dos dados e processos.

As principais contribuições desta tese são: (i) um arcabouço teórico para a descrição, localização e composição de dados e serviços na Web, com regras para verificar a consistência semântica de composições desses recursos; (ii) métodos baseados em ontologias de domínio para auxiliar a integração de dados e estimar a proveniência de dados em processos cooperativos na Web; (iii) implementação e validação parcial das propostas, em uma aplicação real no domínio de planejamento agrícola, analisando os benefícios e as limitações de eficiência e escalabilidade da tecnologia atual da Web semântica, face a grandes volumes de dados.

Abstract

POESIA (Processes for Open-Ended Systems for Information Analysis), the approach proposed in this work, supports the construction of complex processes that involve the integration and analysis of data from several sources, particularly in scientific applications. This approach is centered in two types of semantic Web mechanisms: scientific workflows, to specify and compose Web services; and domain ontologies, to enable semantic interoperability and management of data and processes.

The main contributions of this thesis are: (i) a theoretical framework to describe, discover and compose data and services on the Web, including rules to check the semantic consistency of resource compositions; (ii) ontology-based methods to help data integration and estimate data provenance in cooperative processes on the Web; (iii) partial implementation and validation of the proposal, in a real application for the domain of agricultural planning, analyzing the benefits and scalability problems of the current semantic Web technology, when faced with large volumes of data.

Conteúdo

Resumo	x
Abstract	xi
1 Introdução	1
1.1 Motivação e Contexto do Trabalho	1
1.2 Organização da Tese	3
2 A Survey on Information Systems Interoperability	7
2.1 Introduction	7
2.2 Information Systems Interoperability	8
2.2.1 Viewpoints of Systems Interoperability	9
2.2.2 Technologies addressing Interoperability	10
2.3 Database Systems Interoperability	11
2.3.1 Centralized Database Systems	11
2.3.2 Heterogeneous Database Systems	12
2.3.3 Integrated Access to Multiple Databases	13
2.3.4 Web Databases	14
2.4 Data Integration	14
2.4.1 Data Structuring	15
2.4.2 Characterizing Data Heterogeneity	16
2.4.3 Solving Syntactic and Structural Conflicts	16
2.4.4 Reconciling Semantics	17
2.4.5 The Data Integration Steps	17
2.5 Building Blocks to Integrate Data in Cooperative Systems	18
2.5.1 Gateways	18
2.5.2 Wrappers and Mediators	19
2.5.3 Data Warehouses	20
2.5.4 The View Approach	20

2.6	The Semantic Web	21
2.6.1	XML	23
2.6.2	RDF	25
2.6.3	Ontologies	28
2.7	Web Services	30
2.7.1	Architecture and Basic Standards	30
2.7.2	Cooperative Distributed Processes enabled by Web Services	32
2.8	Applications and Supporting Environments	34
2.8.1	Scientific Workflows	35
2.8.2	Geographic Information Systems Interoperability	35
2.9	Conclusions	36
3	POESIA: An Ontological Workflow Approach for Composing Web Services in Agriculture	38
3.1	Introduction	38
3.2	Application scenario	39
3.2.1	Agricultural zoning	39
3.2.2	Case study	41
3.2.3	Technical challenges	43
3.3	Ontological delineation of utilization scopes	44
3.3.1	Semantic relationships between words	44
3.3.2	POESIA ontologies and ontological coverages	47
3.4	The POESIA activity model	49
3.4.1	Overview	49
3.4.2	Activity pattern	51
3.4.3	Activity pattern aggregation	53
3.4.4	Activity pattern specialization	55
3.4.5	The combined refinement mechanism	57
3.4.6	Process framework	59
3.5	Implementation issues	62
3.5.1	Checking specifications	62
3.5.2	Composing Web services: an implementation perspective	64
3.5.3	Architecture	65
3.6	Related work	67
3.7	Conclusions	69
4	Using Domain Ontologies to Help Track Data Provenance	71
4.1	Introduction	71
4.2	Motivating Example	72

4.3	POESIA Ontologies and Ontological Coverages	74
4.4	Ontological Estimation of Data Provenance	75
4.5	Ontological Nets for Data Integration	77
4.5.1	Data Integration Operators	78
4.5.2	Data Reconciling through Articulation of Ontologies	81
4.5.3	Semantic Workflows	82
4.6	Related Work	83
4.7	Conclusions	84
5	Applying Semantic Web Technology in Agricultural Sciences	86
5.1	Introduction	86
5.2	Motivation: Agricultural Zoning	87
5.3	Solution Context	90
5.3.1	The POESIA Approach	90
5.3.2	POESIA Ontologies as Web Services	91
5.4	An Ontology for the Agriculture Realm	92
5.4.1	The Ontology Design	92
5.4.2	The Ontology on Protégé	93
5.5	Exploiting Ontological Relationships	94
5.5.1	Ontological Coverages to Express and Interrelate Scopes	94
5.5.2	Representing Ontological Relationships	98
5.5.3	Defining Ontology Views	99
5.6	Engineering Considerations and Systems Evaluation	100
5.6.1	Architecture and Design Tradeoffs	100
5.6.2	Constructing Ontology Views	102
5.6.3	Experimental Evaluation	103
5.7	Related Work	106
5.8	Conclusions	108
6	Conclusões	110
6.1	Contribuições	110
6.2	Extensões	111
	Bibliografia	113
I	Formal Definitions and Properties for POESIA	134
II	POESIA Architecture and Implementation Issues	148

Capítulo 1

Introdução

“The only abnormality is the incapacity to love.”

Anaïs Nin

1.1 Motivação e Contexto do Trabalho

A motivação deste trabalho é a construção de sistemas computacionais para a coleta, integração e processamento de dados, visando a extração de informação em aplicações científicas na agricultura. A aplicação utilizada como estudo de caso é o *zoneamento agrícola* – determinação das terras mais apropriadas para o cultivo de diferentes culturas em uma dada região geográfica. Um processo de zoneamento agrícola classifica as terras em parcelas de acordo com o seu grau de aptidão para uma determinada cultura e as épocas do ano mais indicadas para a realização dos tratamentos culturais (tais como, plantio e adubação). O objetivo é determinar as melhores opções para o uso produtivo e sustentável das terras.

As informações resultantes do zoneamento agrícola são fundamentais para o planejamento e gerenciamento de toda a logística da produção e distribuição. Órgãos governamentais e instituições financeiras, por exemplo, baseiam-se nessas informações para definir e executar políticas de concessão de empréstimos agrícolas. Essas políticas visam direcionar os fazendeiros para práticas que contribuam para minimizar os riscos e aumentar a produtividade de seus empreendimentos. Experiências em diversos setores da agricultura brasileira nos últimos anos comprovam os benefícios desse tipo de abordagem.

O zoneamento agrícola envolve a análise de diversos fatores tais como clima, relevo e tipos de solo, de modo a compatibilizar as necessidades das culturas, nas diversas fases do seu desenvolvimento, com as condições ambientais esperadas nas diferentes regiões ao longo do ano. Os dados necessários ao processamento e análise de informações são obtidos de fontes hetero-

gêneas, incluindo sensores para coletar dados de fenômenos físicos e biológicos (por exemplo, estações meteorológicas, satélites e dispositivos de automação laboratorial). Muitas vezes é necessário integrar dados oriundos de sistemas legados e de diferentes instituições, a fim de minimizar os custos com coleta de dados e conseguir volume e amostragem espacial e temporal suficientes para a obtenção de resultados confiáveis.

Um processo de zoneamento agrícola envolve a cooperação de diversas especialistas, trabalhando em organizações distintas e utilizando uma grande variedade de plataformas computacionais e ferramentas de análise de dados. Por exemplo, agrônomos contribuem com técnicas de plantio e modelos de gerenciamento de lavouras e biólogos fornecem os requisitos nutricionais para o bom desenvolvimento das plantas. Estatísticos fazem a análise de riscos de perdas nas lavouras (por exemplo, devido a seca ou geadas). Ambientalistas avaliam o impacto da seleção de cultura agrícola sobre o meio ambiente, a curto e longo prazo. Em suma, diversos especialistas combinam a sua experiência e uma gama de recursos computacionais para construir modelos de zoneamento agrícola. Esses modelos e os processos computacionais que eles originam variam com a cultura agrícola, região geográfica e práticas dos especialistas e instituições envolvidos.

O desafio, do ponto de vista de sistemas de informação, é organizar e conectar os recursos computacionais (dados e serviços) necessários. Além disso, é fundamental promover o reuso de tais recursos, permitindo também sua composição. A importância do reuso neste tipo de domínio pode ser avaliada usando um exemplo simples. Considere o desenvolvimento de processos de zoneamento agrícola para as 20 principais culturas agrícolas no Brasil, e 10 variedades distintas de cada cultura (com diferentes requisitos climáticos e nutricionais). Dividir o Brasil de acordo com as fronteiras estaduais (27 estados) resulta em 5400 modelos. Todavia, grande parte dos recursos computacionais utilizados e mesmo da estrutura dos processos resultantes pode ser compartilhada. A dificuldade em promover o reuso reside em gerenciar o acervo de modelos e recursos computacionais, de modo a promover sua composição em processos cada vez mais sofisticados. Métodos sistemáticos e automatizados para gerenciar tais recursos e processos são cruciais, pois o gerenciamento manual é caro e sujeito a erros. Para responder a este desafio, são necessários resultados em integração de dados, interoperabilidade e composição de serviços na Web.

Integração de dados consiste em produzir uma visão unificada de dados heterogêneos, de modo a permitir o seu intercâmbio e processamento conjunto. Propostas para solucionar esse problema, na maioria das vezes, partem do pressuposto de mundo fechado, e requerem a estipulação de um esquema único, para compatibilizar as necessidades de dados de uma organização. Visões do esquema global permitem restringir o acesso e contemplar necessidades específicas. No entanto, o pressuposto de mundo fechado frequentemente se mostra impraticável, especialmente no contexto de aplicações distribuídas na Internet.

Esta tese apresenta POESIA (*Processes for Open-Ended Systems for Information Analysis*) para fazer frente a tais desafios. POESIA é uma abordagem para a composição de da-

dos e serviços em processos cooperativos na Web semântica. Em POESIA, o intercâmbio de informações e a cooperação de sistemas autônomos no processamento de dados envolve integração de dados em diversos pontos e em múltiplos estágios. A abordagem POESIA combina ontologias de domínio, modelos de atividades e workflows para a composição de serviços na Web. Esta abordagem complementa outras propostas para a recuperação, seleção e composição de serviços, com novas facilidades para o gerenciamento dos recursos utilizados em processos cooperativos.

As principais contribuições da tese são:

- descrição dos requisitos de processos de zoneamento agrícola e elaboração de propostas para contemplá-los;
- desenvolvimento de um arcabouço teórico, baseado em ontologias de domínio, modelos de atividades e workflows científicos, para a descrição, organização, recuperação e composição de serviços na Web, com regras para verificar a consistência semântica de composições de recursos;
- combinação de uma ontologia de domínio e descrições de fluxos de dados para avaliar a proveniência de dados e auxiliar a integração de dados em processos distribuídos na Web;
- validação parcial do arcabouço teórico, através da implementação de alternativas para lidar com grandes volumes de dados em um domínio específico, demonstrando as deficiências da tecnologia atual da Web semântica e propondo alternativas, que incluem a combinação de tal tecnologia com métodos convencionais de gerenciamento de dados.

1.2 Organização da Tese

Os capítulos centrais desta tese são artigos publicados ou submetidos para publicação. As definições e a notação utilizadas em cada artigo foram as que melhor se enquadravam aos resultados apresentados e/ou trabalhos relacionados. Assim, o leitor deve ficar atento a algumas variações.

O Capítulo 2 é uma revisão bibliográfica sobre interoperabilidade de sistemas de informação, submetida ao corpo editorial da série relatórios técnicos do IC/Unicamp. Ela cobre trabalhos em interconexão de bancos de dados relacionais, classificação de problemas de integração de dados, principais padrões e arquiteturas, além dos mais recentes progressos em Web semântica, serviços Web e workflows científicos. Esta revisão descreve alguns dos problemas em aberto abordados pela tese. Além disso, detalha conceitos teóricos apenas mencionados nos capítulos subsequentes, facilitando desta forma a leitura.

O Capítulo 3 (*POESIA: An Ontological Workflow Approach for Composing Web Services in Agriculture*) [83], salvo por pequenas correções efetuadas nesta versão revisada para a tese,

corresponde a um artigo aceito para publicação no *VLDB Journal*, volume 12, número 4, de 2003. Este artigo descreve os fundamentos da abordagem POESIA. Ele mostra como uma ontologia de domínio pode ser utilizada para organizar vastos repertórios de padrões de atividades, que descrevem a composição de dados e serviços na Web para o processamento de dados científicos. Esta proposta de POESIA define sua arquitetura e aborda a questão de ontologia de forma teórica. Os capítulos subsequentes abordam aspectos específicos do desenvolvimento e manipulação de ontologias na implementação de aplicações.

Uma ontologia de domínio em POESIA é organizada em múltiplas dimensões (por exemplo, espaço, tempo, instituição, produto agrícola). *Coberturas ontológicas* – tuplas de termos tomados da ontologia – descrevem o escopo de utilização de dados e serviços, isto é, o contexto específico em que versões distintas dos serviços podem ser utilizadas. Correlações semânticas entre escopos de aplicação definem meios para recuperar e compor recursos, bem como verificar a consistência semântica das composições. O artigo transcrito no Capítulo 3 define operações – agregação, especialização e instanciação – para apoiar a composição de serviços. Essas operações, aplicadas a padrões de atividades, permitem definir frameworks de processos cooperativos e adaptá-los de acordo com necessidades específicas. Um framework de processo é constituído de um conjunto de padrões de atividades, implementadas por serviços Web, que se comunicam para atingir algum objetivo comum (por exemplo, determinar a aptidão agrícola). Cada padrão de atividade está associado a uma cobertura ontológica, que define o seu escopo de utilização, de acordo com conceitos específicos do domínio. A adaptação de um framework consiste em selecionar versões de padrões de atividade, de uma hierarquia de atividades e sub-atividades para realizar uma dada tarefa, referentes a um escopo de utilização específico (por exemplo, determinar a aptidão agrícola para café no Centro-Sul do Brasil).

O Capítulo 4 (*Using Domain Ontologies to Help Track Data Provenance*) [84], foi publicado no SBBD 2003. Ele apresenta um método baseado em ontologia de domínio, estruturada da maneira prescrita na abordagem POESIA, para estimar a proveniência de dados, i.e., a descrição das origens de um dado e do processo utilizado para produzi-lo. O método apresentado deriva a proveniência de dados e captura a semântica operacional de processos de integração de dados, usando a ontologia para descrever e correlacionar escopos e granularidades de dados.

Os estudos de caso utilizados nesse artigo referem-se a duas data warehouses: (i) atributos climatológicos e (ii) produção de frutas no Brasil. Ambas organizam seus dados segundo as dimensões tempo e território, sendo que a primeira também inclui uma dimensão para especificar as organizações responsáveis pela coleta dos dados, e a segunda utiliza uma categorização de produtos agrícolas, para classificar os tipos de frutas produzidos. O artigo mostra como essas dimensões podem ser descritas por uma ontologia multidimensional, tal como prescrito pela abordagem POESIA. O processo de carga da warehouse (i), por exemplo, envolve diversos repositórios intermediários, que provêm serviços de acesso a dados climatológicos providos por diferentes instituições. Esses serviços têm diferentes coberturas espaciais. O artigo propõe a

utilização de coberturas ontológicas para delimitar o escopo dos diferentes serviços e estimar as fontes de dados que contribuem para um dado fornecido por um serviço. O artigo também sugere como coberturas ontológicas podem auxiliar na integração de dados, nos casos em que a falta de um identificador comum pode ser sanada pela descrição do escopo dos dados, utilizando coberturas ontológicas.

O Capítulo 5 (*Applying Semantic Web Technology in Agricultural Sciences*), submetido ao *Information Systems Journal – Special Issue on Semantic Web and Web Services*, reporta uma experiência na construção e manipulação de uma ontologia para o domínio agrícola. Ele analisa as limitações dos padrões e ferramentas atuais da Web semântica para lidar com grandes volumes de dados de ontologias reais. O artigo apresenta uma solução escalável, baseada na criação de visões da ontologia, para a carga, apresentação e manipulação dessa ontologia. Esta solução é implementada em uma biblioteca, denominada *OntoCover*, que conjuga a tecnologia da Web semântica com técnicas tradicionais de manipulação de dados.

A especificação da ontologia manipulada pelo *OntoCover* pode ser produzida com uma ferramenta de edição de ontologias e exportada via RDF. A estrutura geral da ontologia (análoga a um diagrama de classes) é sempre carregada de uma especificação em RDF Schema, contida em um arquivo texto. As instâncias podem ser carregadas de um arquivo texto contendo suas especificações em RDF ou de um banco de dados relacional, mantendo triplas RDF ou instâncias de entidades de um modelo de dados convencional (por exemplo Estado, Cidade, etc.). O sistema de banco de dados relacional provê acesso eficiente aos dados. O *OntoCover* cria a visão da ontologia, conforme especificado pelo desenvolvedor da aplicação, e permite visualizá-la, navegar sobre sua estrutura, selecionar termos para compor coberturas ontológicas e comparar essas coberturas ontológicas. O *OntoCover* foi desenvolvido em Java e pode ser acoplado a aplicações onde essas facilidades básicas sejam necessárias. O artigo apresenta o resultado de experimentos mostrando que a carga e a criação de visões de uma ontologia podem ser realizadas muito mais eficientemente utilizando bancos de dados relacionais com modelagem convencional, do que manipulando triplas RDF representando as propriedades das instâncias de classes da ontologia.

Finalmente, o Capítulo 6 conclui a tese, evidenciando suas contribuições e extensões.

O Anexo I inclui um conjunto de definições e demonstrações, descrevendo formalmente as propriedades fundamentais das ontologias de domínio e do modelo de atividades propostos na abordagem POESIA.

O Anexo II apresenta a arquitetura geral de sistemas para POESIA, e descreve como a implementação realizada, particularmente o *OntoCover*, se insere nessa arquitetura.

Os outros trabalhos publicados durante o doutorado são brevemente descritos a seguir.

1. *The Design of Decision Support Systems for Effective Use of Spatio-Temporal Data* [85] foi apresentado no *EDBT Ph.D. Workshop* de 2000, e constitui um esboço do projeto de tese naquele momento.

2. *An XML-Centered Warehouse to Manage Information of the Fruit Supply Chain* [86], publicado na *World Conference on Computers in Agriculture and Natural Resources (WCCA)* de 2001, descreve uma data warehouse sobre a produção de frutas no Brasil.
3. *Issues on Interoperability of Heterogeneous and Geographical Data* [82], publicado no Simpósio Brasileiro de Geoinformática (GeoInfo) de 2001, é uma resenha sobre integração de dados, sob o enfoque de geoprocessamento.
4. *Querying Multiple Bioinformatics Information Sources: Can Semantic Web Research Help?* [34], cujo autor principal é David Buttler, colega do Georgia Tech durante a estadia naquela instituição, foi publicado na revista *SIGMOD Record* 31(4) 2002. Esse artigo discute as potenciais contribuições da Web semântica à bioinformática.
5. *Aplicando Ontologias de Objetos Geográficos para Facilitar a Navegação em GIS* [236], cujo autor principal é o aluno de iniciação científica Lauro R. Venâncio, foi aceito para publicação no GeoInfo 2003. Esse artigo descreve o *OntoCarta*, uma ferramenta que utiliza uma ontologia de domínio para facilitar a navegação em mapas e possibilitar a integração de objetos geográficos na Web. O *OntoCarta* executa sobre navegadores Web, é aderente aos padrões atuais da Web semântica e utiliza ferramentas de domínio público (inclusive o *OntoCover*) na sua implementação. A ontologia de domínio empregada para navegação em mapas dirigida por conhecimento é aquela desenvolvida para apoiar a abordagem POESIA na área de agricultura.

Chapter 2

A Survey on Information Systems Interoperability

2.1 Introduction

The traditional paradigm for information systems development is based on the cycle modeling-design-implementation, and considers a single database framework, with one schema using one data model. The advent of heterogeneous systems and, more recently, the Web, is changing this picture. Large amounts of data are available in distinct formats and platforms. Data repositories varies from structured database management systems to unstructured files. The lack of agreement on data representation and semantics across heterogeneous systems makes the interoperability problem very complex.

Web systems are in permanent evolution, with new devices, new data sources and new requirements. The possibility of dynamic connections among systems components on the Web adds complexity to the situation. The demand for interoperability has boosted the development of standards and tools to facilitate data transformation and integration. Nevertheless, there are still many challenges to be met, especially those concerned with data semantics and behavior of cooperative systems.

This work surveys some results from the literature related with interoperability and, more specifically, data integration. Our goal is the construction of data warehouses (or materialized views) integrating several kinds of data sources, particularly for scientific applications in agriculture. Data warehouses are a suitable starting point for research and experiments on data integration. The maintenance of consolidated data at the warehouse confers greater versatility to data representation and manipulation. The unidirectional flow of data from the sources to the warehouse, as well as the warehouse update policy which does not require on-line access to data sources, simplifies data processing. The problem can be decomposed into two steps (i) extracting data from the sources to feed the warehouse, and (ii) integrating these multiple

source data into the warehouse. The emphasis of this work is on the second step. The focus is on representational and semantic issues, and the fundamental data integration problems.

Distinct data sources may be maintained independently. In fact, autonomous management of databases is frequently a prerequisite for information systems. However, valuable information may be extracted when collections of data obtained from different data sources are analyzed as a whole. The integrated analysis of data from different sources triggers a wide variety of data heterogeneity problems. Furthermore, connection of autonomous heterogeneous databases complicates classical database problems such as consistency maintenance, concurrency control, transactions and distributed query processing, and optimization. Our research is not concerned with any of these problems. Only consistency maintenance is considered in some degree. The core of our research is semantic data heterogeneity, especially when scientific data are involved.

Instead of trying to coerce all data into a single unified view in one step, we consider integration of small collections of data, in several points of distributed and cooperative processes. Integrated views of selected data sets, materialized or not, define the inputs of data processing activities of distributed processes. The outputs of such an activity, regarded as a data set or service, can be the input of another one. Thus, complex processes involving data integration can be built by composing data sets and services in an open environment like the Web.

The remainder of this paper is organized in the following way. Section 2.2 presents basic concepts related with information systems interoperability. Section 2.3 analyzes interoperability in the context of database systems. Section 2.4 focuses on data representation, data heterogeneity conflicts, and data integration, establishing a framework to analyze related problems and proposed solutions. Section 2.5 presents the most typical apparatus for data integration. Section 2.6 describes the the major standards and technologies of the semantic Web. Section 2.7 outlines the Web services technology and how it can be used to build cooperative distributed systems. Section 2.8 refer to applications demanding technology to support interoperability, particularly in scientific realms. Finally, Section 2.9 presents the conclusions.

2.2 Information Systems Interoperability

Interoperability is the ability of two systems to exchange information, and correct interpret and process this information [144, 118, 105, 9]. It requires some degree of compatibility between systems, to enable data exchange and correct interpretation. Ideally, cooperative systems should be compliant with computational and application domain standards. However, this level of standardization may be impossible to attain in practice, due to the rate of technological changes, the lack of universally accepted standards, the existence of legacy systems, or just for reasons of autonomy of each information system. Thus, in many cases, the only way to reach interoperability is by publishing the interfaces, schemas and formats used for information exchange, making their semantics as explicit as possible, so that they can be properly handled by the cooperative

systems.

2.2.1 Viewpoints of Systems Interoperability

Hasselbring [120] shows that information systems' interoperability must be considered from three viewpoints: application domain, conceptual design and software systems technology. Figure 2.1 illustrates the structure of a set of information systems and their interoperability in each one of these viewpoints.

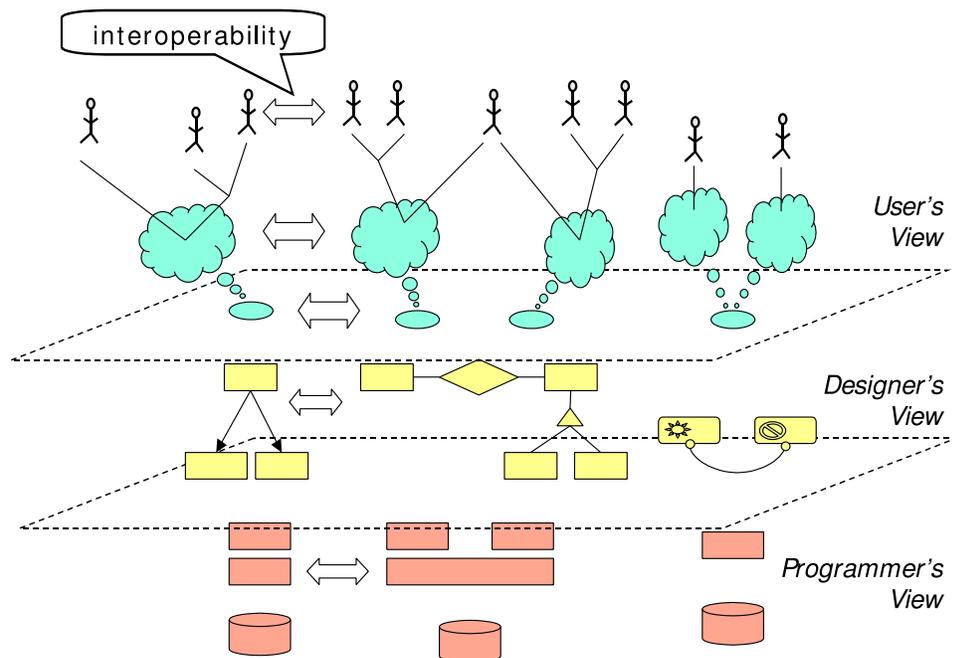


Figure 2.1: The viewpoints of information systems interoperability

The *user's viewpoint* concerns the distinct views and specializations of domain experts. The *designer's viewpoint* refers to requirements modeling and systems design. The *programmer's viewpoint* refers to the systems implementation.

Conflicts may appear in each of those three viewpoints. On the other hand, interoperability must be achieved in all these viewpoints, i.e., users of a system must understand information coming from another system, the system design must accommodate the “foreign” data, and the computer programs must automate information exchange (i.e., the data transfers and transformations). The hardest problems of data interoperability occur at the application and conceptual viewpoints [2].

Furthermore, each viewpoint has the instance level (solutions, projects, application programs), the meta-level (with approaches and models used to describe the characteristics of the

instances), and, maybe, the meta-meta level, where the models are defined. Hence, heterogeneity can also be considered at successive levels of abstraction.

2.2.2 Technologies addressing Interoperability

The growth of computer networks has pushed the development of systems communication technologies beyond protocols for message passing. Several paradigms related with distributed heterogeneous systems interoperability can be singled out in the literature. Some of the most prominent of these paradigms in the Internet era are described in the following.

Distributed objects is the paradigm on the core of technologies like CORBA and DCOM [194].

Each object has an object id, the code to implement its behavior, and a state determined by the value associated with a number of internal variables. An object encapsulates its internal state and code and provides an interface based on methods to externally access and modify its state. Distributed objects communicate with each other through remote method invocation. *CORBA* (Common Object Request Broker Architecture) [52, 194] is the architecture of OMG (Object Management Group) for distributed objects. CORBA objects can be anywhere in a network and are accessed by remote clients, via method invocations, without having to know where each server object resides, what operating system it executes on and how the object is implemented. The language and the compiler used to create CORBA server objects are transparent to clients.

Infopipes [207] are building blocks to implement stream data processing. An infopipe is a language and platform independent abstraction for a data flow from a producer to a consumer. It includes data processing, buffering and filtering. The infopipe model includes facilities for managing quality of service properties (e.g., performance, availability, security), composing and restructuring data flows during execution. This model has inherent parallelism and embraces content semantics and user requirements, allowing information flow control and resource use optimization.

Peer-to-Peer [179] refer to a class of systems that employ resources distributed across a network to perform some function in a decentralized fashion. The resources encompass processing power, data, storage means and network bandwidth. The function can be distributed computing, contents sharing, communication or collaboration. The key characteristic of a peer-to-peer system is that, in opposition to the client-server architecture, each peer can provide some service to other peers, at the same time that it benefits from the services provided by other peers of its community. Peer-to-peer systems, such as Napster, and Kazaa, became popular for allowing people to share audio and video files on the Web.

Composite Web Services [234, 250] use Web services – i.e., self-describing and independent software modules accessible through the Internet – as the building blocks to construct inter-institutional cooperative processes. Web services communicate via messages, using standard Web protocols. These services encapsulate autonomous systems components with Web-based interfaces, taking advantage of the ubiquity of the Web to provide wide access to those components. The fundamental problems of this paradigm are the discovery of the services available on the Web to fulfill a particular need; and the coordination of services in distributed processes to achieve specific goals. Web services technology has been developed and applied in areas like electronic commerce and finance. Our research combines Web services, workflows, and semantic Web technology, to solve problems of scientific applications involving data integration and cooperative work on the Web.

XML and Java are also expected to play an important role in the implementation of interoperable distributed information systems [45, 193]: the former as a syntactic standard for data representation (Section 2.6.1), and the latter as a portable language, allowing the transference of source coded objects' behavior from one platform to another.

2.3 Database Systems Interoperability

Information systems are characterized by the flow consisting of “data input, processing and output”. The uncoordinated creation of heterogeneous files to store data of autonomous systems leads to problems when different applications have to access shared data. Database systems were proposed to solve these problems in centralized environments [152].

2.3.1 Centralized Database Systems

Database and database management systems (DBMS) [72, 73, 5] are among the most common means of managing data. A *centralized database system* accommodates all the data of an organization in a unique internal schema. *Views* [24, 243, 92, 225], or external schemas, are distinct logical database images, allowing (groups of) users to access a central database according to their specific needs. A view is usually built by using a database query language to write a query defining an image of a limited amount of data.

Database views are assigned to particular applications according to users' requirements and privacy concerns. A view can be materialized or non-materialized. *Materialized views* are copies of data to support different database images. *Non-materialized views*, on the other hand, are just abstractions, produced by translating requests to the abstract views into requests to actual database or lower level views.

The user of a database (or view) must know the data model employed and the (external) schema, in order to access the database directly through the DBMS. An alternative approach is

the construction of application programs atop the DBMS to help users in their daily activities. The development of systems integrating different databases demands considerable coordination of the teams responsible for the distinct databases, views and application programs. This coordination is very difficult to be achieved, even when the integration involves only a few departments within the same organization.

2.3.2 Heterogeneous Database Systems

Heterogeneous database systems (HDBS) [72, 219, 152, 126, 5] are software packages that integrate various preexisting database systems (DBSs) or HDBSs called components. The same component can participate in various HDBSs. Components can be developed independently and without any concern about subsequent integration.

Sheth and Larson [219] characterize HDBSs using three orthogonal axes: heterogeneity, distribution, and autonomy. The *heterogeneity* of a HDBS depends on the number and severity of discrepancies among its constituent DBSs, with respect to their schemas, data models, query languages, transaction management capabilities, DBMS, hardware, operating systems and communication protocols. Discrepancies can appear at any abstraction level (data instances, schema, data model). The heterogeneity can be reflected in the data representation or be just a matter of interpretation. *Distribution* refers to the location of the HDBS' components. In principle, distribution is orthogonal to heterogeneity. A distributed system can involve different hardware, software and communication platforms. *Autonomy* refers to the freedom of the HDBS' components to define and manage their databases. The need for maintaining autonomy and the demand for sharing data are often conflicting requirements. The integration of different databases cannot completely block the capacity of each component DBS to manage its data without interference of the HDBS general manager [5]. Autonomy can be classified in four categories [219, 5]:

1. *Design autonomy* refers to the independence of each component DBS to design its database.
2. *Communication autonomy* refers to the ability of a component DBS to decide whether to communicate with other component DBSs. A component DBS with communication autonomy is able to decide when and how it responds to a request from another component DBS.
3. *Execution autonomy* means that a component DBS is independent to execute operations (requested both locally and externally), with full control of transaction processing.
4. *Association autonomy* asserts that component DBSs can independently decide what information they want to share with the HDBS, to which requests they reply, when to start and when to finish their participation in the HDBS.

2.3.3 Integrated Access to Multiple Databases

The approaches to enable integrated access to multiple physical databases can be roughly classified in two categories: schema integration [18, 72] and the federated approach [151, 219, 152]. The former consists in providing some unified schema through which the users access the integrated data. The latter, on the other hand, can just supply some means for accessing exported views of the heterogeneous databases, leaving much of the data integration onus to the users. Figure 2.2 illustrates the differences between these approaches. In the distributed approach (on the left), the schema of each distributed database is a view of the unified schema. In the federated approach, on the other hand, the export/import schemas of the federated databases are externally handled. The schema integration approach makes data heterogeneity transparent to the users, while the federated approach concede more autonomy to the component databases.

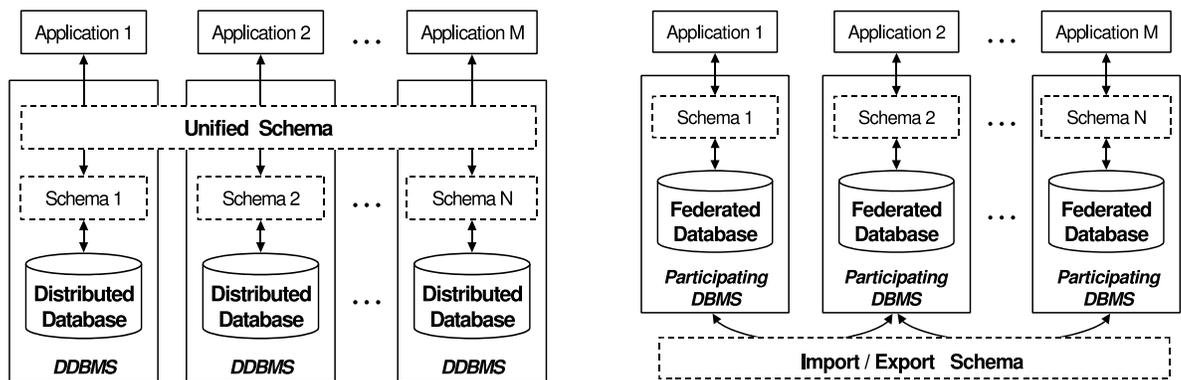


Figure 2.2: Distributed and federated database systems

There are several options for implementing HDBSs, with varying coupling degrees among the component DBSs, and offering different trade-offs between cooperation and autonomy. Elmagarmid and Pu [72] give an introduction to such systems, classifying them as follows.

- *Distributed database system (DDBS)* [72, 5, 73, 197] consists of a single logical database that is physically distributed. Despite the physical fragmentation of data, a DDBS supports a single data model and query language, with one schema integrating all its contents.
- *Federated database system (FDBS)* [219] (also called *heterogeneous database system – HDBS*) is a distributed database system allowing heterogeneous components with different data models, query languages or schemas.
- *Multidatabase system (MDBS)* [151, 152] is a collection of loosely coupled databases. The key properties of a MDBS are the autonomy of the participant databases and the

absence of a globally integrated schema. MDBSs are employed when users want to preserve their autonomy, even to the point of refusing to participate in a globally integrated schema.

All these database systems architectures rely on some integrated or export/export schema. However, they do not address the resolution of data heterogeneity conflicts to build such an schema. They either consider that this problem has been solved or leave it to the user.

2.3.4 Web Databases

Web Databases [60, 240, 106, 187, 36] make data stored in local databases accessible through the Web, enabling applications like on-line stores and digital libraries. The most common interfaces for querying Web databases are forms and navigation menus on Web browsers. The query specification resulting from a user interaction with such an interface is encoded and sent to a Web Server, which submits the query to the DBMS. The result is converted into HTML format to be returned via the Internet and showed in the browser. Options for implementing the interaction between the Web Server and the DBMS are described in [143, 71].

The challenge of the querying Web databases research is the construction of a unified and simple interface. The most common approach to solve this problem is the generation of wrappers and mediators to integrate data from Web pages provided by Web databases [240, 36, 35, 158]. These solutions tend to be complex, inefficient and unsuitable in many cases, due to the dynamics of the sources interface and availability. Other solutions available in the literature include [187, 106, 60]. Neiling *et al.* [187] present automated means to recover and integrate the contents of related Web databases (e.g., movie databases). Gravano *et al.* [106] describe a system to organize Web databases in hierarchies of classes, according their contents. Silva *et al.* [60] use keywords specified by the user to derive structured queries to be submitted to one or more DBMSs.

2.4 Data Integration

Heterogeneous data are those data presenting differences in their representation or interpretation, although referring to the same reality [151]. *Data heterogeneity conflicts* are the incompatibilities that may occur among distinct data sets. The interoperability problem considered in this section is *data integration* [69, 200], i.e., providing a single view for a set of heterogeneous data, with unified syntax, structure and semantics. *Data integration* involves the resolution of heterogeneity conflicts and transformations of source data to accommodate them in the integrated view.

In order to make data integration possible, it is necessary, at first, to categorize the kinds of data to be integrated and the heterogeneity conflicts. Then, conflicts can be solved in a sequence

determined by their categories. The rest of this section discusses the proposals available in the literature and defines a framework to analyze and handle data integration problems.

2.4.1 Data Structuring

Structured Data

Conventional database systems take advantage of rather strict data structuring, expressed via a database schema using a data model, to provide data management facilities, with efficient data access and consistency maintenance. That is the case of the classical relational database management systems and even the object-oriented systems.

Data structuring presents virtues and drawbacks with respect to data integration. On the one hand, structure grants uniformity for data processing and helps maintaining consistency. On the other hand, an structured integrated view from two or more heterogeneous data sets is sometimes very difficult to obtain.

Semantic data models [18], such as the entity-relationship data model, allow data to be described in an abstract and intelligible manner, at the conceptual level. Thus, these models can facilitate data integration. However, semantic data models are not versatile enough and information can be lost on converting data among heterogeneous database schemas using these data models. The automation of the data conversion process is also difficult, because of the gap between the implementation and the conceptual viewpoints.

Semi-structured Data

Semi-structured data [2, 1, 32, 117, 199] are those data whose structure is irregular and partially known. In order to allow the identification of the data elements in the irregular structure, semi-structured data have to be self-describing. Thus, the data and basic descriptions of their structure and meaning (metadata) are assembled together. Differently from structured data, where structure (type and schema) are defined prior to the creation of data instances, semi-structured data instances can be created at the same time their structure is defined.

Semi-structured schemas and data models are usually formalized as graphs, whose nodes represent data elements and whose edges represent nesting and reference relationships between data elements [2, 199]. This data structuring is suitable for data integration and Web systems. Current research in databases includes how to model, query, restructure, store and manage semi-structured data [2, 66, 1]. Other research themes include extracting some structure from data in formats such as those prevalent in the Web [2, 89, 36, 35, 158, 189], text documents [4] and spreadsheets [145], in order to integrate these data.

2.4.2 Characterizing Data Heterogeneity

The most widespread way to characterize data heterogeneity is to separate representation from interpretation concerns [219]. *Representational conflicts* refer to syntactic or structural discrepancies in the portrayal of heterogeneous data. *Semantic conflicts* refer to disagreement about the meaning, interpretation or intended use of the same or related data.

The solution of representational conflicts usually requires the analysis of their semantic counterpart, i.e., establishing correspondences (perfect or not) between the meanings of data items from heterogeneous sources. Semantic matches are often achieved only for specific domains.

Both representational and semantic conflicts may occur in any level of abstraction: instance, schema, data model. Thus data heterogeneity conflicts can also be classified according to the following categories [118, 178, 137, 136]:

- *Data conflicts* are discrepancies in the representation or interpretation of instantiated data values, which can differ in their measurement unit, precision and spelling.
- *Schema conflicts* are differences in schemas due to alternatives to depict the same reality, such as using distinct names for the same entities or modeling attributes as entities and vice-versa.
- *Data versus schema conflicts* are disagreements about what is data and metadata; e.g., a data value under one schema can be the label of an entity or attribute in another schema.
- *Data model conflicts* result from the use of different data models.

2.4.3 Solving Syntactic and Structural Conflicts

The earlier solutions for representational heterogeneity [144, 178, 142] are restricted to the relational data model. They extend SQL to allow the conversion of table and attribute labels into data values and vice-versa. Other works explore languages with logical foundations, aggregation and restructuring capabilities [99, 100].

Proposals for integrating semi-structured and other diverse data sources are surveyed in [210, 89]. Several proposals concern the establishment of a standard syntax and data model. Some of them are centered in object models [118, 209], while others use semi-structured data to represent heterogeneous data at a more abstract level [47, 48, 199, 117]. The use of semi-structured data confers versatility to data representation, enabling data transformations and mappings among irregular structures. On the other hand, as data modeling constructs from typical data models often carry semantics, information can be lost on converting data from such a data model into semi-structured data. The information loss problem can be handled by maintaining proper metadata associated with the transformed semi-structured data.

2.4.4 Reconciling Semantics

The solution of semantic conflicts relies on the standardization of the meaning of the concepts, terminology, and structuring constructs found in source data [218, 195]. It involves metadata enrichment to support the investigation of semantic matching among data items from distinct data sets.

The first step is to semantically describe data, by associating consensual descriptions to published and exchanged data [134]. At this stage, the establishment of an accord is usually possible only for small communities [105]. Common semantics can be expanded to wider communities, as information is better understood and appropriate levels of abstraction are devised to make possible data exchange with minimal loss of meaning.

2.4.5 The Data Integration Steps

Data integration can be regarded as a sequence of steps, involving transformations and investigation of correspondences among data elements, in order to produce a unified view of heterogeneous data. Figure 2.3, adapted from [200], illustrates the information flow along the data integration steps.

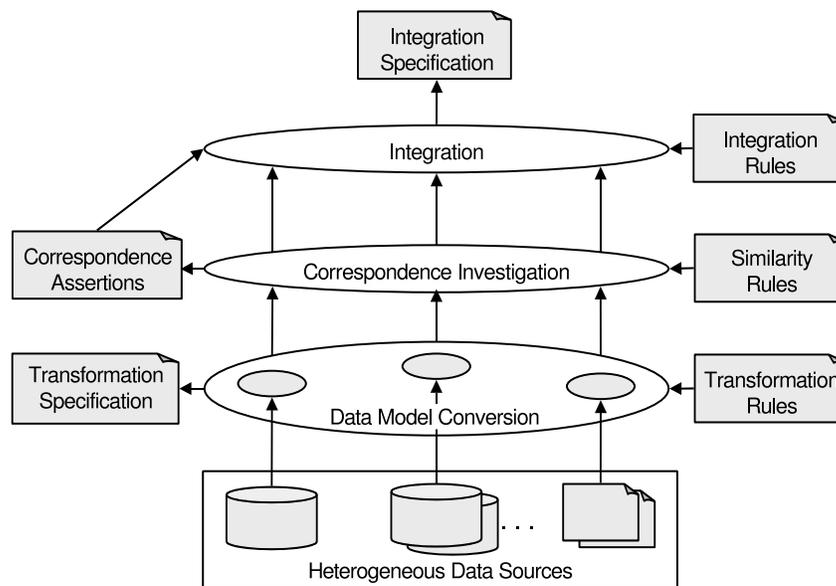


Figure 2.3: The data integration steps

Heterogeneous data are first converted to a homogeneous format (e.g. XML), using transformation rules that explain how to transform data from the source data model to the target data model. The translated data and schemas are semantically poor for integration purposes. Thus they must be enriched with semantic information (e.g., measurement units, meaning of the terms

appearing in tags and data values). Then, the correspondences between elements from heterogeneous sources are investigated, using the semantic descriptions and similarity rules, to produce a collection of correspondence assertions. Finally, the correspondence assertions and integration rules are used to produce an integration specification, which describes how data elements from heterogeneous sources must be transformed and mixed to produce a unified view.

Even though data integration ultimately requires human intervention, it is crucial to automate or at least assist some laborious tasks, in order to make data integration practicable. The goal of automated facilities is to make data integration easier and repeatable, while allowing users to make decisions along the integration process.

2.5 Building Blocks to Integrate Data in Cooperative Systems

This section describes some categories of software apparatus that have been proposed to support integrated data views. Such apparatus allow the interconnection of heterogeneous data repositories, programs, materialized and non-materialized views, in such a way that the output of one software module can supply the input to another module.

2.5.1 Gateways

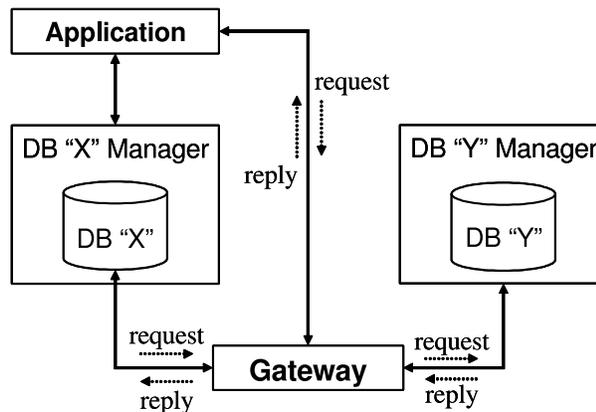


Figure 2.4: A database gateway

A *Gateway* is a software component that allows a DBMS and/or an application program directly connected to this DBMS to access data maintained by another DBMS, using the data model and data manipulation language of the former. It is necessary to develop one specific gateway for each DBMS pair. Gateways do not provide transparency for heterogeneous

database schema and instances. Hence, gateways do not offer support to establish a unifying view of heterogeneous data. Figure 2.4 presents a gateway providing access to database “Y” for an application program and its directly connected database “X”.

2.5.2 Wrappers and Mediators

Wrappers and mediators [244, 97] provide data manipulation services over a reconciled view of heterogeneous data. Wrappers encapsulate details of each data source, allowing data access under a homogeneous data representation and manipulation style (common data model and, sometimes, standardized schema). Mediators offer an integrated view of the data sets of several data sources that can include wrappers and other mediators. Some systems adopt multiple levels of mediators in order to modularize the data transformation and integration along successive levels of abstraction.

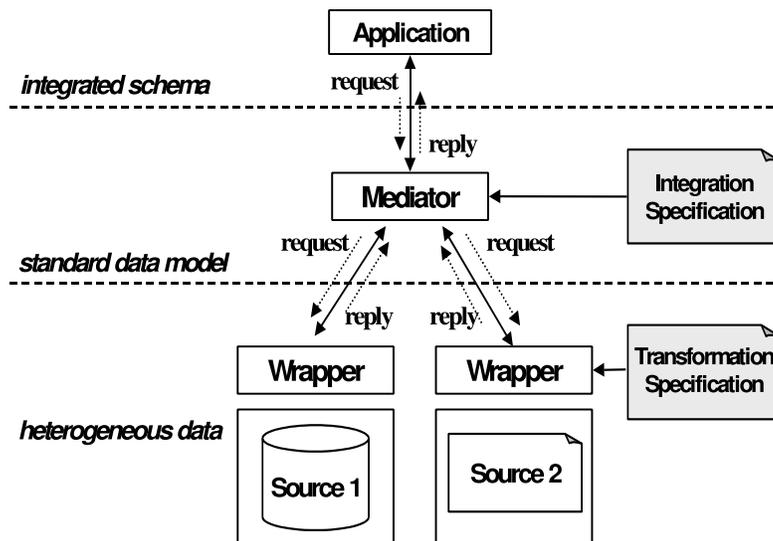


Figure 2.5: Wrappers and mediators

Figure 2.5 shows two wrappers and one mediator providing integrated access to two different data sources. The mediator brokers the requests from the applications into requests to the wrappers of the particular sources involved in the request. On receiving the replies from the source wrappers, the mediator composes the results to return an integrated result to the application. Data transformation and mapping specifications drive the functioning of wrappers and mediators. Wrapper generators and data mapping specification languages [97] enable the specification of data integration in a more intelligible manner than using conventional programming languages to hard code wrappers and mediators.

2.5.3 Data Warehouses

A *data warehouse* [205, 127, 163, 46, 159] is a separated database built specifically for decision support. It provides the basis for analysis of large amounts of data, collected from a variety of possibly heterogeneous data sources. A data warehouse replicates and integrates data from sources such as relational databases maintained by on-line transaction processing systems (OLTP), spreadsheets and textual data. These sources typically run in the operational level of organizations, while data warehouses are intended for the strategic level.

Data warehousing is the activity of collecting, transforming and integrating data for consolidated analysis. This can be performed off-line with periodical updates, perhaps overnight. The separation between the data warehouse and the data sources prevents the warehouse from interfering in the functioning of the systems at the operational level and confers flexibility for data organization and processing in the warehouse. Data from the sources is first processed before being stored at the warehouse.

There are specific methods for modeling and organizing data in a warehouse – e.g. multi-dimensional, star, and snowflake style schemas [125] – and also for data processing and user interaction – e.g., on-line analytical processing (OLAP) [98, 107, 46, 119, 64]. Figure 2.6 shows the loading of data from the sources into a warehouse and their use for data analysis purposes.

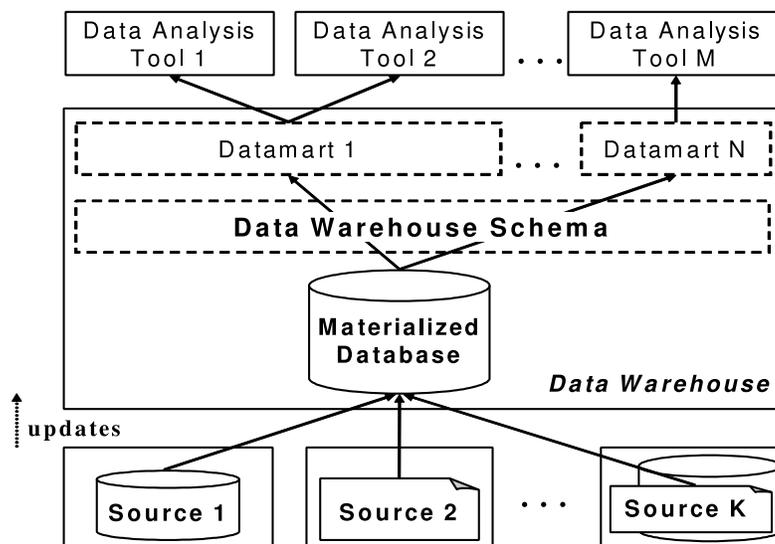


Figure 2.6: A data warehouse

2.5.4 The View Approach

Wrappers and mediators support non-materialized (i.e., abstract) integrated views for heterogeneous data, while data warehouses provide materialized views (i.e., concrete sets of copied,

transformed and integrated data). In data warehouses, the unidirectional data flow, from the data sources to the warehouse repository simplifies the view update problem [113, 208, 261]. The data warehouse cannot be updated by end users. Updates done to the sources have to be periodically loaded in the warehouse to reflect them in the unified view. Figure 2.7 illustrates a general view-based data integration system. In this case, updates posed on the exporting views are difficult to be performed in the lower levels, especially the original data sources. The transformations applied for data analysis purposes (e.g., data aggregation) can lead to complex problems of data lineage and view updating [55, 56].

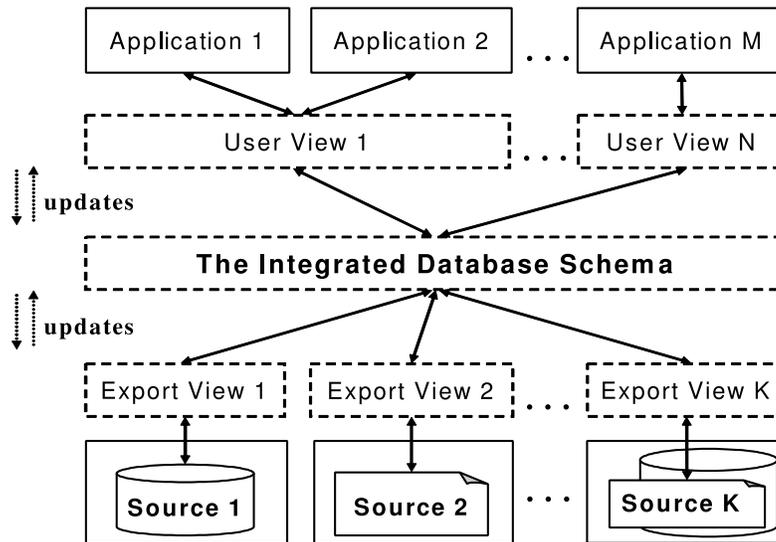


Figure 2.7: The view approach

Many of the techniques developed for views in heterogeneous database systems can be employed for the construction of wrappers, mediators and data warehouses. Unfortunately, integrating highly heterogeneous data and exporting them to specific data analysis tools are harder problems. They demand data transformation and management facilities beyond those provided by the current DBMSs. Views stored in warehouses also involve historical information that may not remain in the original sources. Nevertheless, several works take the view approach for the integration of heterogeneous data [18, 112, 231] and data warehouses [159].

2.6 The Semantic Web

The *semantic Web* [215, 80, 63, 260, 68, 22] is an emerging research area whose goal is to achieve information systems interoperability and enable a variety of sophisticated applications, by taking advantage of semantic descriptions of Web resources (data and services). It is an infrastructure on which different applications can be developed [76]. It intends to enrich the

current Web with formalized knowledge and data, that different human beings and/or computers can exchange and process.

The key requirement for the semantic Web is interoperability. Data and metadata must comply to consensual formats and conceptualizations, in order to enable their exchange and proper processing. Therefore, standards for expressing data and metadata are crucial for the semantic Web. Figure 2.8, adapted from [140], illustrates the semantic Web layers of standards and technologies.

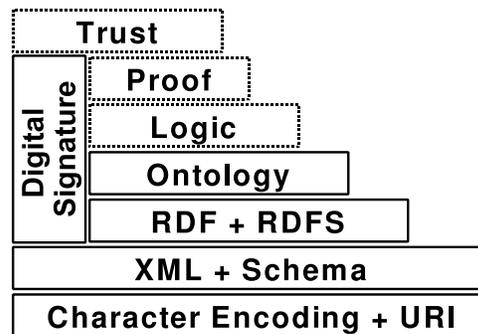


Figure 2.8: Layers of semantic Web standards and technologies

The lowest layer, *character encoding + URI*, provides an international standard for coding character sets (Unicode) and a means to uniquely identifying resources in the semantic Web (the URI specification [232]). The *XML* [256] layer, which includes namespaces [185] and schema definitions [256, 257], constitutes a standard syntax, with an underlying data model, to express interchangeable data and schemas. In the *RDF + RDFS* layer, *RDF* [147] allows statements associating resources with their properties. *RDFS* (RDF Schema) [29] enable the definition of vocabularies that can be referred to by the URIs in which they are published. These vocabularies can be used to associate types to resources and properties. The *Ontology* layer enriches vocabularies and supports their evolution, by extending the repertory of concepts and semantic relationships among them. Several languages for describing ontologies in the Web have been proposed to fulfill the needs of this layer [80, 101, 109, 165, 196, 61, 191].

The top layers: *Logic*, *Proof* and *Trust* are still under development. The *Logic* layer expresses knowledge by rules, while the *Proof* layer uses these rules to infer other knowledge. The *Trust* layer provides mechanisms to determine the degree of trust on inferred knowledge. *Digital Signature* permeates several layers to ensure security, by using means like encryption and digital signatures.

The remainder of this section describes the XML, RDF and ontology layers of the semantic Web in more detail, analyzing the major standards and technologies and how they interrelate.

2.6.1 XML

XML (eXtensible Markup Language) [256, 2] is a syntax standard, with a graph-based data model, to represent and exchange semi-structured data. XML derives from the ISO standard SGML (Standard Generalized Markup Language) [128]. These languages are known as meta-markup languages because they allow the definition of specific markup languages. Like HTML, XML employs tags and attributes of tags to structure data. However, the structure and tags of an XML document are user defined. In XML, tags and structure are intended to describe data meaning, not data presentation as in HTML. Web servers, browsers and certain applications are able to process XML-encoded data.

Figure 2.9 presents a fragment of a XML document containing climate data, specifically water balance data (measurements of climate data, soil moisture and evaporation of this moisture). These data refer to a particular point in the earth surface, denoted by its geographic coordinates and the name of the city where that point is located. The major data element contained in this XML document, `WaterBal`, expresses the geographic position by means of the XML attributes `location`, `latitude` and `longitude`, attached to its opening tag. This data element includes several climate measures for each month. Each measure is represented by an atomic data element. The value of each measure appears between the element's opening and closing tags. For example, the value of the average temperature in January is enclosed by the tags `<Temperature>` and `</Temperature>`. This atomic data element is nested in the composite element congregating all the measures for January, delimited by the `<Jan>` and `</Jan>` tags. The default namespace associated with this XML document points to the description of its schema (presented in Figure 2.10), via a http address.

```

<?xml version="1.0"?>
<WaterBal xmlns="http://www.agric.gov.br/WaterBalBrotas.xml"
  location="Brotas" latitude=-22.1500 longitude=-47.5800>
  <Jan>
    <Temperature> 22.0 </Temperature>
    <AvgRainFall> 201.3 </AvgRainFall>
    <PotET> 115.4 </PotET>
    <RealET> 115.4 </RealET>
    <Stored> 125.0 </Stored>
    <WaterDeficit> 0.0 </WaterDeficit>
    <WaterExcess> 86.0 </WaterExcess>
  </Jan>
  :
</WaterBal>

```

Figure 2.9: An XML document for climate data (water balance)

The emergence of XML poses many challenges to academia and industry [67, 44, 242,

150]. Leading software vendors are moving toward adopting XML, either as an internal data representation model for their software or just for data exchange among different applications and platforms. The publication of data in XML format can make the Web a huge XML data source for all sorts of information.

There are many technologies being developed to explore the potential of XML (e.g., XML query languages [2, 258, 25]). The use of XML as a data representation standard can bring many benefits for data integration [2, 150]. Furthermore, since XML is a semi-structured data model, it can lend versatility and openness to data representation and integration.

However, XML alone does not solve all the data heterogeneity conflicts. XML data sets from independent sources can present schema and semantic conflicts, even if these sources provide data about the same domain for the same application. The resolution of these conflicts requires consensual semantics to be associated with XML contents and tags. This cannot be done in one step. Interoperability requires multiple agreements on XML data modeling and terminology.

Common Schemas and Metadata Standards

DTD and *XML Schema* are schema languages for XML [148]. Schema specifications can be stored with XML data, or in a separate document, that can be referenced to by several XML documents. *DTD* (Document Type Description) [256] is part of the XML specification itself. It defines the structure of XML documents using a list of element declarations. These declarations, in the style of regular expressions, define the types of atomic XML components and the nested structure of composite elements.

XML Schema [257] offers an XML-based syntax to describe the structure and constraining the contents of XML documents. XML Schema reconstructs and extends DTD capabilities. Figure 2.10 presents an XML Schema description for the climate data document presented in Figure 2.10. The first line of this description declares the namespace for the XML Schema vocabulary. The second one states that a document conforming to this schema must have an element called `WaterBal` (the string used in its tags) of the type `WaterBalType`. An element of type `WaterBalType` includes twelve nested elements of the type `AggregValues`, to hold the climate measurements for each month of the year. `WaterBalType` also includes attributes to specify the geographic location to which the climate data refer.

Note that the schema description is not enough to ensure the correct interpretation of the XML data and support data integration. Much semantic information is missing. For example, there is no indication of the measurement units and the geographic coordinate system used in the XML and XML Schema fragments of climate data. In addition, the meaning of the data elements is not clearly specified by their tags. For example, `Temperature` probably refers to the average temperature in the month, while `AvgRainfall` refers to the average accumulated rainfall during the particular month (these averages are derived from temporal series of weather

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="WaterBal" type="WaterBalType"/>
  <xsd:complexType name="WaterBalType">
    </xsd:sequence>
    <xsd:element name="Jan" type="AggregValues"/>
    :
    <xsd:element name="Dec" type="AggregValues"/>
  </xsd:sequence>
  <xsd:attribute name="location" type="xsd:string"/>
  <xsd:attribute name="latitude" type="xsd:Latitude"/>
  <xsd:attribute name="longitude" type="xsd:Longitude"/>
</xsd:complexType>
<xsd:complexType name="AggregValues">
  <xsd:sequence>
    <xsd:element name="Temperature" type="decimal"/>
    <xsd:element name="AvgRainfall" type="decimal"/>
    <xsd:element name="PotET" type="decimal"/>
    <xsd:element name="RealET" type="decimal"/>
    <xsd:element name="Stored" type="decimal"/>
    <xsd:element name="WaterDeficit" type="decimal"/>
    <xsd:element name="WaterExcess" type="decimal"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

Figure 2.10: An XML schema for climate data (climate data)

data). The meaning of certain attributes like `PotET` and `RealET` (potential evapotranspiration and real evapotranspiration, respectively) are even harder to infer, and require expert knowledge to be fully understood.

This example illustrates the need to associate consensual semantics with XML data and their markup. The use of standard schemas and metadata standards, with well documented and widely agreed meaning, can decrease this problem. General metadata standards such as Dublin Core [65] define vocabularies and the precise meaning of terms for general use, while metadata standards and standard schemas developed for specific fields help to establish some consensus inside these fields [237]. However, these standards and formats are not enough because: (i) they hinder the autonomy of information systems, (ii) they do not contemplate the evolution of these systems, (iii) they do not cover all types of data, and (iv) they are unsuitable to provide different views of the same data.

2.6.2 RDF

RDF [147, 80] is the major format for machine-processable metadata in the semantic Web. *RDF* is based on knowledge representation formalisms such as frames [180] and description logics [15]. The basic construct of the *RDF* model is the *statement* – a triple of the form *subject-predicate-object*, where *subject* refers to a resource (anything that can be denoted by a URI), *predicate* is a property of that resource, and *object* is the value of that property. The object can be a literal (e.g., a string) or another resource. An *RDF* statement declares a property of a

resource and can also be regarded as a *resource-property-value* triple, where *resource* is used as a synonym for *subject*, *property* for *predicate* and *value* for *object*. Thus, one can stipulate an RDF triple (<http://www.Embrapa.br>, PART_OF, <http://www.Brazil.gov.br>) to indicate that the organization whose home page is accessible by the URI <http://www.Embrapa.br> is part of the Brazilian government.

RDFS (RDF Schema) extends RDF with classes of resources, values, and properties. An RDFS specification defines a structure of classes, properties and subclasses for a particular domain or application, similar to an object-oriented class diagram.

Figure 2.11, adapted from [133], illustrates the use of RDF and RDFS to describe Web resources. Two different RDF schemas, on the top of the figure, describe resources for gathering weather data (e.g. weather stations). The RDF schema on the left describes these resources from the point of view of scientists who are interested in analyzing weather data. These scientists connect their applications to data collecting devices available on the Web (e.g. via Web services) to obtain such data. Their applications are concerned with the geographic location of the data collecting devices and how different land parcels (e.g., states, counties) are interrelated. A company responsible for the maintenance of the data collecting devices, on the other hand, has a different view of the same resources. For such a company, each device is an equipment, with category and model. Each equipment is associated with one client.

Each resource in the unified RDF specification on the bottom of Figure 2.11 is an instance of some class (i.e., another resource describing its type) of one or both RDF schemas on the top. For example, the weather station `&ws1` is an instance of `WeatherStation` in the RDF schema on the left and of `Equipment` in the RDF schema on the right. `&ws1` is a shorthand for the URI `http://www.embrapa.br/WeatherStationX`. Statements involving resource instances must match statements defined at the RDFS level. For example, `&ws1` belongs to `&Embrapa` and is located in `&Rio`, a county of `&RJ` State. The URIs of land parcels and clients are omitted for simplicity.

In addition to their use in providing different views of the same resources, RDF/RDFS also help to define unified views of heterogeneous resources. For example, the weather stations of Figure 2.11, having different technical characteristics and belonging to different institutions, can be originally described and handled in different ways. Furthermore, their positions can be defined in distinct systems of geographic coordinates, and the arrangement of land parcels can differ across institutions (e.g., water supply companies divide land in hydrological basins). The data provided by different weather stations can also differ in their structuring and representation (e.g., measurement units). Several layers of RDF/RDFS descriptions provide the solution for these conflicts.

The RDF/RDFS standards play the following fundamental roles in the semantic Web:

- denote relationships involving resources and resource descriptions;

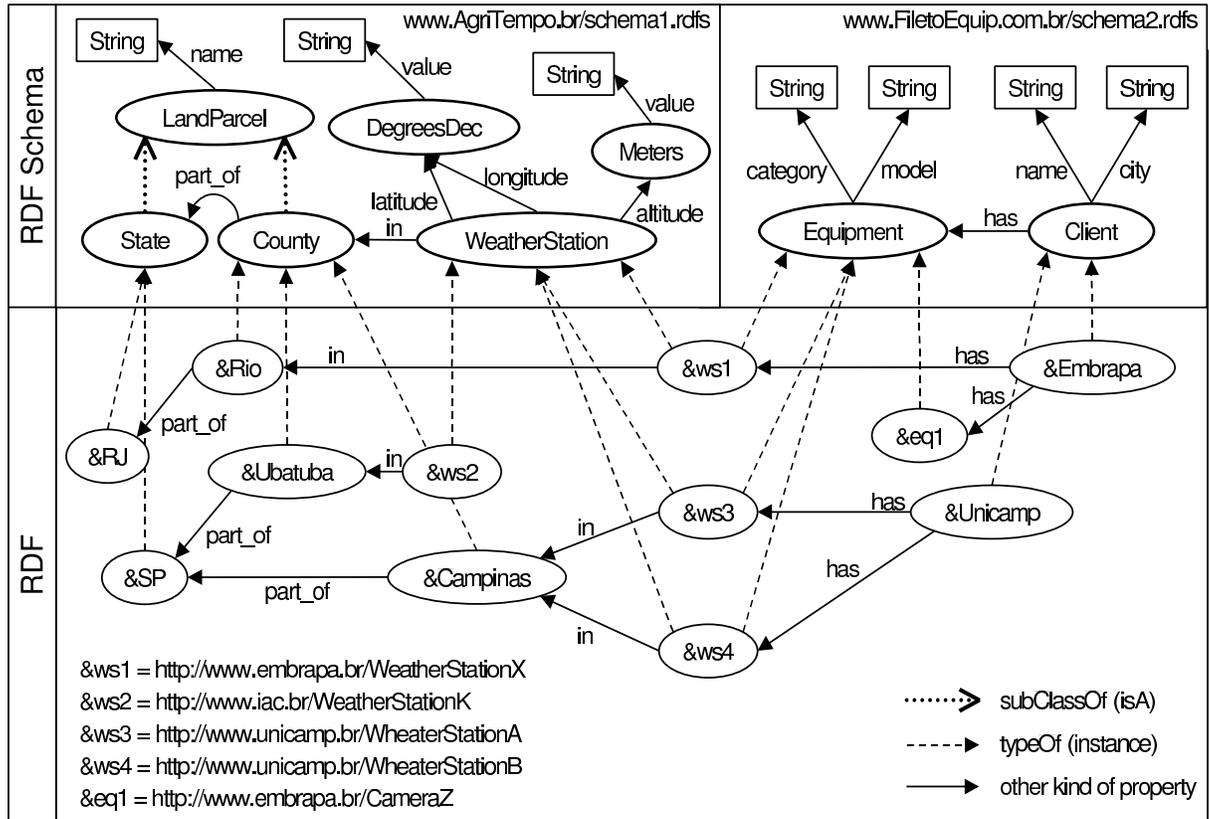


Figure 2.11: RDF descriptions of resources for collecting scientific data

- provide distinct views of the same resources, tailored for different domain or applications;
- build unified views for collections of heterogeneous resources;
- describe knowledge in terms of vocabularies of concepts and the semantic relationships among these concepts.

The XML/RDF Mismatch

RDF/RDFS can be expressed using XML syntax. However, many XML handling facilities are not appropriate for handling RDF. XML and RDF/RDFS are both based on directed graphs, but have different models. The RDF/RDFS model is a directed graph in which labeled nodes represent resources or literals and labeled directed edges represent properties linking resources to the values of their properties. The edges of the RDF graph-based model are unordered and their labels define properties. The XML semi-structured data model, on the other hand, is more hierarchical. The labeled nodes of the XML model represent data elements or attributes, and its directed edges represent nesting and reference relationships between data elements. In the XML

model edges are unlabeled and the outgoing edges of a node have a total order. Patel-Schneider and Siméon [202, 201] point out problems resulting from this mismatch between the XML and RDF/RDFS models. They propose a semantic foundation for the Web, based on model theory, to reconcile XML and RDF information sources.

Handling RDF/RDFS

XML query languages, such as XQuery [258, 2], are not suitable for RDF, due to the models' mismatch. Thus, several languages and tools have been developed specifically for querying RDF metadata. Jena [132, 248] is a popular toolkit for handling RDF triples. It allows navigation in RDF triples through an application program interface (API) or the RDQL query language, an implementation of SquishQL [177]. Nevertheless, procedural languages for handling RDF triples and their components are cumbersome. For many applications, a template-based declarative language would be more appropriate. RQL (RDF Query Language) [133] is a declarative language for querying RDF according to its graph model. RQL adapts functionality of query languages for semi-structured and XML data [2], to provide functional constructs, in the style of OQL [40], for uniformly querying RDF/RDFS. Sesame [30] is a server-based architecture for storing and querying large quantities of metadata in RDF/RDFS, with support for RQL and concurrency control. Most of the current facilities for handling large RDF repositories, including Jena and Sesame, rely on relational or object-oriented database management systems to provide persistence and scalability [132, 248, 74, 162, 133, 30]

2.6.3 Ontologies

Ontologies [233, 109, 110, 172] are shared conceptualizations of knowledge about delimited domains. An ontology organizes definitions and interrelationships involving a set of concepts (e.g., entities, attributes, processes). It captures the meaning of classes and instances from a universe of discourse, by arranging the symbols (e.g., words, expressions, signs) referring to them, according to semantic relationships [247].

An ontology entails or embodies a particular viewpoint of a given domain. This viewpoint must be *shared* by a group of individuals, formed according to factors like geographic proximity, cultural background, profession, interests or involvement in particular enterprises. These people establish agreements with respect to their views of the world and the symbols used to communicate their views. Ontologies can be explicit or implicit, formal or informal. However, they must be *explicit* and *formal*, to be represented and processed by computers.

There is no convention with respect to the form of a machine-processable ontology. A simple type hierarchy, specifying classes and their subsumption relationships, like a taxonomy, is an ontology. Even a relational schema can serve as an ontology, by specifying the possible relationships and integrity constraints in a database.

Ontologies constitute a means to structure knowledge to support information retrieval and interoperability [109]. The shared knowledge carried in ontologies enable precise stipulation and resolution of queries [111, 216, 121, 13, 7, 184, 134] and information brokering [135, 173] in open environments. Ontologies also help data integration, particularly the investigation of correspondences between elements of heterogeneous data sources [13, 171, 21, 172]. Related research proposes the development of information systems components by translating ontologies into object-oriented hierarchies to implement these systems, giving rise to the concept of Ontology-Driven Information Systems [110, 90].

The following paragraphs describe the currently proposed means to describe, develop and manage ontologies in the semantic Web. Sections 2.7 and 2.8 include more specific discussions of the use of ontologies in semantic Web applications.

Ontology Specification Languages

Several languages and formalisms have been proposed to express knowledge in ontologies [101, 109]. DAML+OIL and OWL are some of the most prominent ontology languages for the semantic Web. They extend the RDF/RDFS vocabulary and enrich expressiveness for delineating ontologies (e.g., to express disjunction of classes and other constraints). DAML+OIL [165] combines the basic constructs and syntax of DAML-ONT (DARPA Agent Markup Language) [61, 80] with OIL's (Ontology Inference Layer) [191] frame-based modeling primitives [180] and formal semantics and reasoning services, based on description logics [15].

OWL (Web Ontology Language) [196] is a W3C candidate standard recommendation. It is intended to describe classes and relations that are inherent in Web documents and applications. OWL carries influences of DAML+OIL, among other languages and formalisms. Like OIL, OWL comes in three different flavors, with increasing expressiveness and complexity.

Descriptions of other ontology languages appear in [80, 101, 201]. The relationship and integration of XML with ontology representation languages and formalisms is addressed in [13, 7, 202, 201, 6, 139].

Ontologies Development and Management

The development of ontologies is a laborious and error prone task, especially if it is done by hand. Ontology engineering tools [190, 227, 103] can automate parts of this task and hide the idiosyncrasies of the ontology specification languages and formalisms. These tools can offer graphical interfaces, facilities for knowledge acquisition (e.g., legacy data set conversion and incorporation in the ontology), remote access to knowledge repositories and means to check the quality and consistency of the specifications produced.

Protégé [190, 227, 103, 78] is an example of an open-source graphic tool for ontology editing and knowledge acquisition. It can be extended with plugins to incorporate new functionality.

Available plugins allow, for example, the development and exchange of ontology specifications in a variety of formats, including DAML+OIL and OWL.

Methodologies and guidelines for developing ontologies appear in [109, 226, 19]. They help to enhance productivity and to improve the quality of the ontologies developed. Methods and tools for automatically extracting ontologies from text documents and semi-structured data are proposed in [94, 62, 181, 182, 184].

The spreading of ontologies for different domain and applications leads to interoperability problems among diverse ontologies. Proposed solutions for this problem involve ontology composition algebras and graph-based models for ontologies articulation [79, 183, 131, 247, 245, 246].

Finally, Jess [91] and Algernon [122] are examples of inference engines for the semantic Web. These engines handle RDF/RDFS specifications and related formats as rules formalizing declarative knowledge. They apply inference to derive other knowledge from the base knowledge present in ontology specifications. These engines can be plugged to an ontology editor such as Protégé or simply process RDF/RDFS exported by such a tool.

2.7 Web Services

A *Web service* [81, 222, 39, 253] is a software module accessible through the Internet. Web services are usually self-describing and independent. They communicate with clients and other services via messages, over standard Web protocols. Each Web service can be identified by a URI and exposes a XML interface to allow its discovery and invocation across the Web.

The Web services technology is based on the notion of building new applications by combining network-available services. The services participating in distributed processes cooperate to achieve some goal, by exchanging messages and coordinating their executions. It enables interoperability of information systems, while allowing decoupling and just-in-time applications integration. The resulting cooperative systems are potentially self-configuring, adaptive and robust, because they can allow the dynamic incorporation of alternative services and avoid single points of failure. Furthermore, implementing systems components as Web services reduces complexity, as application designers do not have to worry about platform and implementation details, which are encapsulated by the Web services interfaces.

2.7.1 Architecture and Basic Standards

A service oriented architecture postulates cooperation of software components with three distinct roles: service providers, service requesters and service brokers. A *service provider* holds the implementation of one or more services and manages the public interfaces that make these services available on the Web. A *service requester* is the party that has a need to be fulfilled

by some published service. It can be a human user accessing services through a console or Web browser, an application program or another Web service. The *service broker* provides a searchable repository of service descriptions, where service providers publish their services and service requesters find descriptions and binding information to access services contemplating their particular needs.

Service providers, requesters and brokers communicate using standard technologies. There are many standards currently under development to allow language and platform independent implementation of Web services [141, 229]. Figure 2.12 outlines the layers of standards and technologies supporting Web services-based applications.

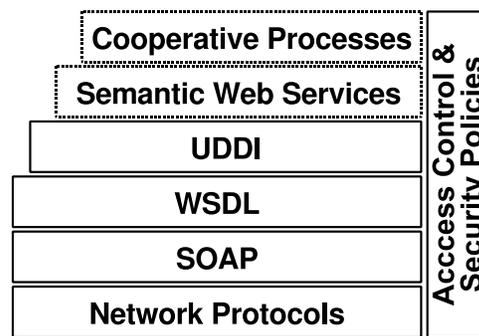


Figure 2.12: Layers of Web services standards and technologies

The *Network Protocols* layer provides the basic communication facilities and protocols (e.g., HTTP). *SOAP* [28] is a lightweight protocol for services to exchange XML-encoded messages and make procedure calls over the Internet. Messages can be routed along a message path. SOAP provides enveloping facilities to describe the intent of a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. Though SOAP was originally designed to use HTTP as the transport protocol, it can run on other network protocols such as FTP, SMTP or even raw TCP/IP sockets. SOAP is extensible, allowing different communication models such as one-way, request-response and multicast. In addition, SOAP is not tied to any language or component technology.

WSDL (Web Services Definition Language) [254] is a XML-based format for describing Web services. WSDL specifies what a Web service does, where it is located and how it is invoked. In WSDL, a service is regarded as a set of related endpoints called ports. The ports of a service can communicate with ports of other services via messages, that can contain either document-oriented or procedure-oriented information. The abstract definitions of ports and messages are separated from their network deployment and data format bindings. This allows the reuse of abstract definitions: port types that define sets of operations supported by ports, and data types that define the data being exchanged. A concrete data format and protocol

specification for a port type constitutes a reusable binding. WSDL can work in conjunction with SOAP, HTTP GET/POST or MIME.

UDDI (Universal Description, Discovery and Integration) [230] is a set of standard XML schemas, SOAP messages and API specifications to build catalogs for finding specific Web services. UDDI provides information about business (e.g., name, description, contact), services offered and particular standards used to bind with these services. It also provides identifiers and various taxonomies to describe business (e.g., related industry, products and services, geographical region). A UDDI registry is itself a Web service, providing facilities to create, modify, delete and query service descriptions. These registries can be public or private. IBM and Microsoft provide public UDDI registries. Service providers only have to register to one of these public registries, since updates to any of them are replicated in the others on a daily basis.

The two top layers of Figure 2.12 refer to the semantic and functional aspects of Web services integration. These layers are still under development with many proposals from industry and academia. The *semantic Web services* layer employs semantic Web technologies, such as ontologies, to support Web services discovery, selection and composition, according to the needs of specific domains or applications. The *Cooperative Processes* layer concerns the coordinated execution of Web services in cooperative processes across organizational boundaries. Finally, *Access Control and Security Policies* can be enforced in any Web services implementation layer.

2.7.2 Cooperative Distributed Processes enabled by Web Services

Semantic Web Services

Semantic Web services [166] are associated with well-defined semantics to express their functional properties, capabilities, applicability and ontological relationships, in order to enable their utilization in cooperative processes over an open and distributed environment. Research in this area rely on semantic Web ideas and technologies [124, 260, 108, 220, 164, 213, 221, 14, 198, 38, 166, 37].

The capabilities of registries such as UDDI and languages like WSDL are not enough to support services discovery [198]. DAML-S (or DAML-services) [14] is an extension of the DAML ontology specification language for Web services. It includes mechanisms to describing, discover, select, activate, compose, and monitor Web resources. The work of [198] employs DAML-S for services discovery, presenting an algorithm to match service requests with the profile of advertised services, based on the minimum distance between concepts in a taxonomy tree. Cardoso and Sheth [38] present metrics to select Web services for composing processes. These metrics take into account functional and operational features such as the purpose of the services, quality of service (QoS) attributes, and the resolution of structural and semantic conflicts. McIlraith *et al.* [166] use agent programming to define generic procedures involving the

interoperation of Web services. These procedures, expressed in terms of concepts defined with DAML-S, do not specify concrete services to perform the tasks or the exact way to use available services. Such procedures are instantiated by applying deduction in the context of a knowledge base, which includes properties of the agent, its user, and the Web services.

Topology and models have been proposed to enable cooperation and composition of services. Schlosser *et al.* [213] propose a graph topology, determined by a globally known ontology, to speed up communication of Web services in a peer-to-peer system. Maximilien and Singh [164] present a model for gathering and assessing information relative to the use of Web services to determine their trustfulness. Sirin *et al.* [220] presents a prototype to guide a user in the dynamic composition of Web services. Finally, Grüninguer [108] show how an ontology for process specification languages can serve as a semantic foundation for the composition of Web services.

Web Services Coordination

Nowadays, there is a myriad of proposals concerning the interoperability and synchronization of Web services [234, 116, 20, 87, 204, 250, 175]. Examples of Web services composition languages include BPEL4WS (BEA, IBM, Microsoft) [250], WSFL (IBM) [255], BPML (BPMI), XLANG (Microsoft), WSCI (BEA, Intalio, SAP, Sun), XPD (WfMC), EDOC (OMG) and UML 2.0 (OMG). Some challenges of these technologies are: (i) reducing the amount of low-level programming necessary for the interconnection of Web services (e.g., through declarative languages), (ii) providing flexibility to establish interactions among growing numbers of continuously changing Web services during run time, and (iii) devising mechanisms for the decentralized and scalable transaction control for cooperative processes running on the Web. Much of the current technology for synchronizing processes are based on centralized control, even if the execution is distributed. This centralization is inappropriate for Web systems, for reasons of autonomy and scalability. Thus, in opposition to techniques to orchestrate services, Web-based workflows require technology to allow service to choreograph their executions, based on agreed upon protocols.

Van der Aalst [234] compares the major candidate standards for Web services composition and synchronization. He points out problems related with the lack of formal semantics, expressiveness, complexity and adequacy of these proposals. [234] suggests the incorporation of well-established process modeling techniques in a single standard for Web services composition. The use of Petri-nets for this purpose is considered in [116, 235, 186]. Activity models appear in [93, 157, 156, 155, 154].

2.8 Applications and Supporting Environments

Semantic Web applications take advantage of knowledge, represented in proposed standards like RDF, to leverage automated means to describe, organize, discover, select and compose Web resources for the solution of a variety of problems. The most usual approach is to define semantic markup based on some ontology, and use them to integrate and provide unified access to data and services, typically via Web portals. There are many examples of this approach in the literature [121, 216, 111, 13].

Some experimental systems possess distinctive features. Edutella [188] is a Peer-to-Peer infrastructure using RDF metadata to facilitate access to educational resources. In Edutella, each peer holds a set of resources and has an RDF repository of resource descriptions, to allow querying its contents at the storage layer (e.g., SQL) or user layer (e.g., RQL). Peers can be heterogeneous in their internal organization and the query language they provide. The common data model and the exchange language of Edutella enables a standard interface for posing queries to specific peers or communities and find resources across the network.

Piazza [115] is an infrastructure to provide interoperability of data sources in the Web, by mapping their contents at the domain level (RDF) and the document structure level (XML), and addressing the interoperation between these levels. The mappings are specified declaratively for small sets of nodes. A query answering algorithm chains these mappings together to obtain relevant data from across the network.

Papers focusing specifically scientific applications of the semantic Web and Web services include [224, 160, 174, 102, 43]. Some scientific applications refer to particular fields such as bioinformatics [34, 153, 41, 223, 114, 104], earth sciences [17, 241] and the environment [16, 42, 161]. The *grid* – a platform for coordinated resource sharing through the Internet, increasingly used for scientific data processing – and the semantic Web have mutual characteristics and goals [102]. Both operate in a global, distributed and dynamic environment, and both need computationally accessible and sharable metadata to support automated information discovery, integration and aggregation.

POESIA (Chapter 3) introduces the concept of ontological coverages – tuples of terms taken from a multidimensional ontology – which are used to describe the utilization scope of data and processing resources, particularly in agricultural sciences. The partial ordering among these descriptors enable the organization, discovery, and reuse of resources. POESIA also includes mechanisms, based on ontologies, workflows and activity models, to semantically orient the composition of Web services in cooperative distributed processes (Chapter 3) and help to trace the information flow across these processes (Chapter 4).

Web services development and execution platforms are described in [88, 53, 138, 249, 176]. Bandholtz [16] propose the use of Web services to share ontologies and describes the implementation of a service network for this purpose.

2.8.1 Scientific Workflows

Scientific work is typically based in experiments [43]. Sometimes scientists rely on simplified models of real world phenomena to found their investigation, and use vast amounts of data to corroborate their results. The technological development has generated a great availability of data, from a variety of heterogeneous sources, that scientists can use to enhance their experiments. Moreover, scientists can exchange models and computer programs implementing these models. Although scientific work can vary among diverse people, disciplines and organizations, it can benefit a lot from data and systems interoperability.

Scientific Workflows [41, 12, 168, 239, 8] use workflow technology [130, 123, 59] to manage scientific work. They regard scientific experiments as complex processes with intricate data transformations and information flow. These processes may encompass automatic and manual activities. The data and execution dependencies among these activities can be very complex, yielding interoperability and synchronization problems. Many scientific processes are distributed, in order to enable cooperation of different groups and foster reuse of partial results. Therefore, semantic Web service technologies are fundamental to implement these processes in an open environment encompassing different platforms.

Scientific processes differ from business processes in several aspects. Scientific work demands freedom to try alternative ways of doing things. The sequence of steps (and even the goal, sometimes) is not totally known in advance. The scientist perform some task and decides on the further steps only after evaluating the previous ones. Specific subjects in scientific processes management include documentation [238] and reorganization [156] of these processes.

The exploitation of the workflows paradigm for managing scientific processes has been exploited in specific domains such as bioinformatics [34, 41, 223, 170] and geoinformatics [214, 241, 169, 11, 129, 259, 10]. For instance, Cavalcanti *et al.* [41] combines metadata support with Web services in a framework to support scientific workflows and apply this framework to structural genomics. Seffino *et al.* [214], on the other hand, use scientific workflows to describe and reuse patterns of geographic data processing in agricultural and environmental applications.

2.8.2 Geographic Information Systems Interoperability

Geographic information systems (GIS) [3, 167, 49] manage data referring to geographic entities or phenomena. These data are geo-referenced, i.e., they carry some indication of the geographic location. A GIS provides specialized basic facilities to process geographic data, being useful for information extraction, planning and decision support, among other kinds of applications.

The GIS market is characterized by proprietary formats that make interoperability hard to achieve. Many formats have been proposed for exchanging geographic data [192, 217, 9]. However, scientists have progressively found out that standard formats are not enough to strengthen GIS interoperability [105]. The conversion of data through these formats often results in in-

formation loss, incorrect interpretation of data and poor information quality [51]. It happens because formats for geographic data exchange are mainly concerned with syntax, structure and the geometry of geographic objects. Even GML (Geography Markup Language) [192] do not ensure the correct interpretation of data, because it does not take into account the semantics and the behavior of geographic objects.

The importance of establishing a semantic basis for geographic data representation and management has been recognized in several papers [70, 203, 54, 90, 161, 252]. Córcoles *et al.* [54] describes an approach for integrating geographic data, based on mappings between ontologies and XML schemas. They present an ontology to support the creation and exchange of semantic descriptors for geographic resources (XML documents containing geographic data). The descriptors and the links among them and the resources themselves are both expressed in RDF. It enables a unique language for querying GML documents, without knowledge of their structure.

Ontologies for the integration of geographic data appear in [90, 161, 252]. Fonseca *et al.* [90] employs ontologies to define classes for developing geographic applications. Their applications rely on ontology servers and mediators to access their data sources. It allows, for example, loading data instances from heterogeneous data sources, using a schema defined by one ontology.

GIS interoperability also requires additional levels of integration such as commonality of systems behavior and system-user interaction. The adoption of a common geographic data model [228, 26] or at least a framework to unify heterogeneous models [50] constitutes one ingredient to achieve this goal.

2.9 Conclusions

Integration of heterogeneous data has been one of the greatest challenges in database research. The advent of the Web is pushing the demand for solutions, and reformulating this problem into a more complex setting – the discovery, selection and composition of data and services. Solutions for all these problems involve versatile standards and enriching the Web with semantics, in order to allow interoperability while embracing diversity.

The Web is becoming the common platform for implementing cooperative distributed systems. The semantic Web and workflows based on the collaboration of services across the Web, are expected to expand the role of computers to support human activities in a variety of fields. In this open distributed environment, data processing and semantics cannot be dissociated, because the meaning of data depends on the whole process employed to produce them. Technology to support the idealized systems is under fast development, in areas ranging from knowledge management to Web services development and composition. Concrete applications must be developed in the near future to fulfill end users' expectations.

This survey has outlined the research on information systems interoperability, from work

on interconnection of relational databases, to the most recent developments in semantic Web services. The major contributions are: (1) describing and comparing proposed standards and architectures; (2) categorizing heterogeneity and proposed solutions; (3) discussing specific needs related with data and services integration, particularly for scientific applications.

Acknowledgments

This work was partially supported by Embrapa, CAPES, CNPq and the MCT/PRONEX-SAI and the CNPq WebMaps projects. Thanks to professors Ana Carolina Salgado, Caetano Traina Júnior, Célio Cardoso Guimarães and Edmundo Madeira, who provided several suggestions for the improvement of this work.

Chapter 3

POESIA: An Ontological Workflow Approach for Composing Web Services in Agriculture

3.1 Introduction

Web services [253] are components for constructing next-generation Web applications. These composite Web applications are built by establishing meaningful data and control flows among individual Web services. These data and control flows form *workflows* connecting components distributed over the Internet. However, there has been very limited research on the composition of Web services using workflow concepts and techniques. This is partially due to the limitations of centralized control in traditional workflow management systems, which are inadequate for the scalability and versatility requirements of Web applications (e.g., dynamic restructuring of processes [168] and activities [157]).

This paper bridges this gap by applying advanced workflow and activity concepts in the composition of Web services toward the construction of sophisticated Semantic Web applications. Our approach is called POESIA (Processes for Open-Ended Systems for Information Analysis), an open environment for developing Web applications using metadata and ontologies to describe data processing patterns developed by domain experts. These patterns specify the collection, analysis, and processing of data from a variety of Internet sources, thus providing building blocks for next-generation Semantic Web applications.

The main contribution of the paper is POESIA's support of Web service composition using domain ontologies with multiple dimensions (e.g., space, time, and object description). Tuples of terms taken from these ontologies, called *ontological coverages*, formally describe and organize the utilization scopes of Web services. A *utilization scope* is a context in which different data sets and specific versions of a repertoire of services can be used. In POESIA, Web services

are composed under these scopes through well-defined operations such as specialization and aggregation. Rules based on the correlation of utilization scopes and their ontological relationships enable systematic means to verify the semantic and structural consistency of Web services compositions. In addition, POESIA ontologies are used in the determination of the granularities for selecting and integrating data and processes as well as helping to describe their semantics.

The second main contribution of this paper consists in showing how POESIA resolves some open issues in Web services composition. This is done through the modeling of a substantial application of practical impact using POESIA. Our application is in the area of environmental information systems, specifically, agricultural zoning – the determination of land suitability for important crops. Agricultural zoning is a challenging application for several reasons. First, several kinds of heterogeneous scientific data streams, such as meteorological measurements, are gathered continuously in large volumes and correlated for specific temporal and spatial conditions. Second, these data sources are distributed over the Web, increasingly through Web services. Third, agricultural zoning is a cooperative (distributed) decision-making process involving experts from several fields. Finally, it requires continuous processing since the situation is frequently reevaluated depending on temporal (seasonal) changes.

POESIA is a contribution toward the realization of the vision of the Semantic Web for scientific applications. It allows the partial automation of some expert reasoning for organizing, reusing, and composing not only data but also the Web services that provide access to and process these data.

The remainder of this paper is organized as follows. Section 3.2 describes our example application. Section 3.3 defines the domain ontologies and ontological coverages that are the basis of our approach. Section 3.4 presents the POESIA approach to specify and reuse Web services. Section 3.5 outlines the main technical issues in the implementation of the POESIA environment. Section 3.6 discusses related work, and Section 3.7 concludes the paper.

3.2 Application scenario

3.2.1 Agricultural zoning

Agricultural zoning is a scientific process to determine land suitability in a geographic region for a collection of crops. This process classifies the land into parcels according to their suitability for a particular crop and the best time of year for key cultivation tasks (such as planting, harvesting, pruning, etc). The goal of agricultural zoning is to determine the best choices for a productive and sustainable use of the land while minimizing the risks of failure. However, some constraints may impose inevitable trade-offs that lead to compromises (e.g., short-term productivity vs. long-term sustainability). Typically, agricultural zoning requires looking at many factors such as regional topography, climate, soil properties, and crop requirements. Ad-

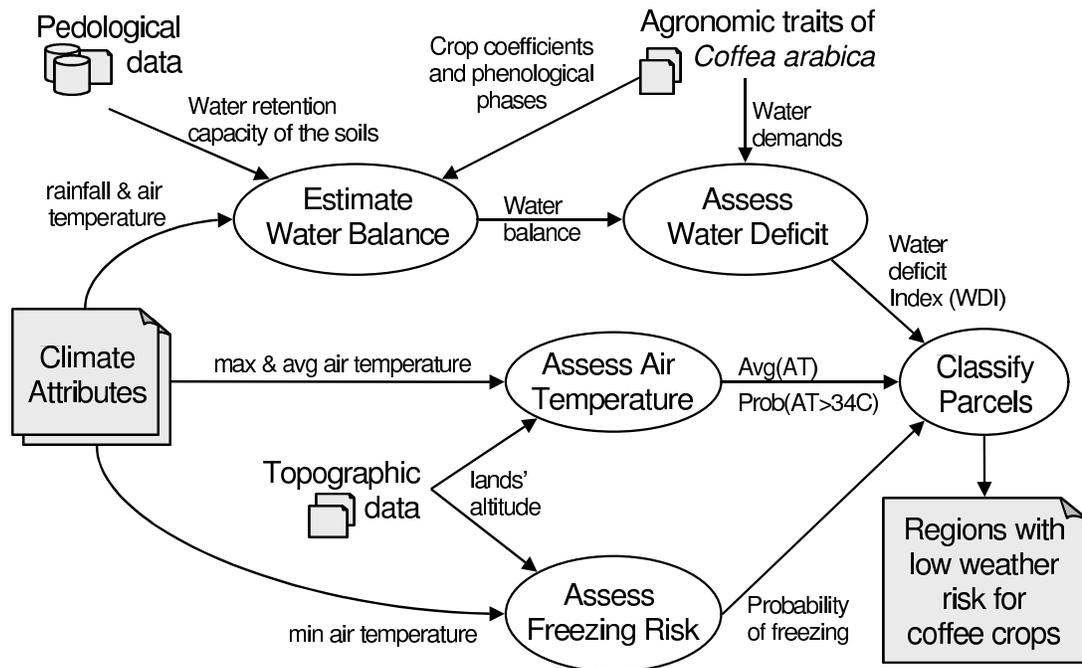


Figure 3.1: Determining land suitability for *Coffea arabica* in Brazil's Center-South

ditional concerns include interactions with wildlife, environmental preserves, and social and market impact.

As illustrated in Section 3.2.2, agricultural zoning is a complex process consisting of intricate interactions among a variety of data sources. The process is built by cooperation of experts from many scientific and engineering disciplines. For example, agronomists contribute with planting techniques and crop management models. Biologists provide crop growth and nutrient requirements. Statisticians provide risk management analysis for potential crop failures (e.g., due to severe weather). Environmental scientists analyze the impact of crop selection over the environment for both the short and long term. These and other scientists and engineers bring together their expertise and a variety of computational and data analysis tools to build an agricultural zoning model.

At run time, an agricultural zoning process obtains relevant data from a variety of heterogeneous sources, primarily sensors that collect data on physical and biological phenomena (e.g., weather stations, satellites, laboratory automation equipment). Since gathering and processing real-time data can be costly, database systems and existing documents in different formats are frequently used as alternative sources. In any case, large amounts of fine-grained data are usually required for extracting the needed information. Both data and data processing tools can be encapsulated and provided through Web services. In summary, agricultural zoning combines tools and services developed by a diverse set of scientists and integrates data from many

heterogeneous sources through coordinated activities, as described by POESIA.

Agricultural zoning has been a labor-intensive process that is both expensive and slow to develop due to the complexities mentioned above. This is a serious issue since it is an extremely important problem for a country with many commercial crops such as Brazil. Suppose we want to produce an agricultural zoning model for the top 20 crops for each region. Let us consider the 10 major varieties of each crop (these varieties usually have different weather and soil requirements). Simply dividing Brazil according to state boundaries (27 states) will result in more than 5000 models. It is clear that we need a systematic way to develop and maintain these models since manual processes will be too expensive and error prone.

3.2.2 Case study

Figure 3.1 illustrates a specific agricultural zoning process, namely, land suitability for *Coffea arabica* in the Center-South region of Brazil. *Coffea arabica* is the main species of coffee produced by Brazil. Although coffee is no longer the country's number one export product, it remains one of the major farm export products due to the high commercial value of good coffee. The zoning process for *Coffea arabica* is composed of several distributed and cooperating activities, represented by ellipses. Data from several sources are processed by these activities, and the results generated by each activity are transferred to other activities or data repositories.

According to domain experts [75, 262], the most influential environmental factors for *Coffea arabica* are: (1) soil water availability, (2) air temperature, and (3) the risk of freezing. These factors are reflected in the structure of the land suitability process in Figure 3.1, which relies on a data warehouse of climate attributes to obtain aggregated values of measurements, such as maximum, minimum, and average temperature, and total rainfall, in appropriate time granularities. This warehouse is a composite Web service encompassing resources for collecting and maintaining climate data from several regions and institutions. It serves as input to three activities that can be executed in parallel – *Estimate Water Balance*, *Assess Air Temperature*, and *Assess Freezing Risk*. The activity *Estimate Water Balance* takes the expected rainfall and the average air temperature for each month of the year, the water retention capacity of the soils, and some phenological coefficients of coffee plants (collected from legacy database systems and scientific publications in agronomy) to estimate the water balance – a measurement of the expected amount of moisture available in the ground through the year. *Estimate Water Balance* is followed by *Assess Water Deficit*, which compares the data from water balance with the water demands of the plants during their successive phenological stages, producing the water deficit index (WDI) – a measurement of the expected deficit of water for the crop throughout the year.

In a similar way, the activities *Assess Air Temperature* and *Assess Freezing Risk* use other climate data and topographic data to produce the average air temperature, the probability of air temperature exceeding 34°C, and the probability of freezing. These partial results (indices and

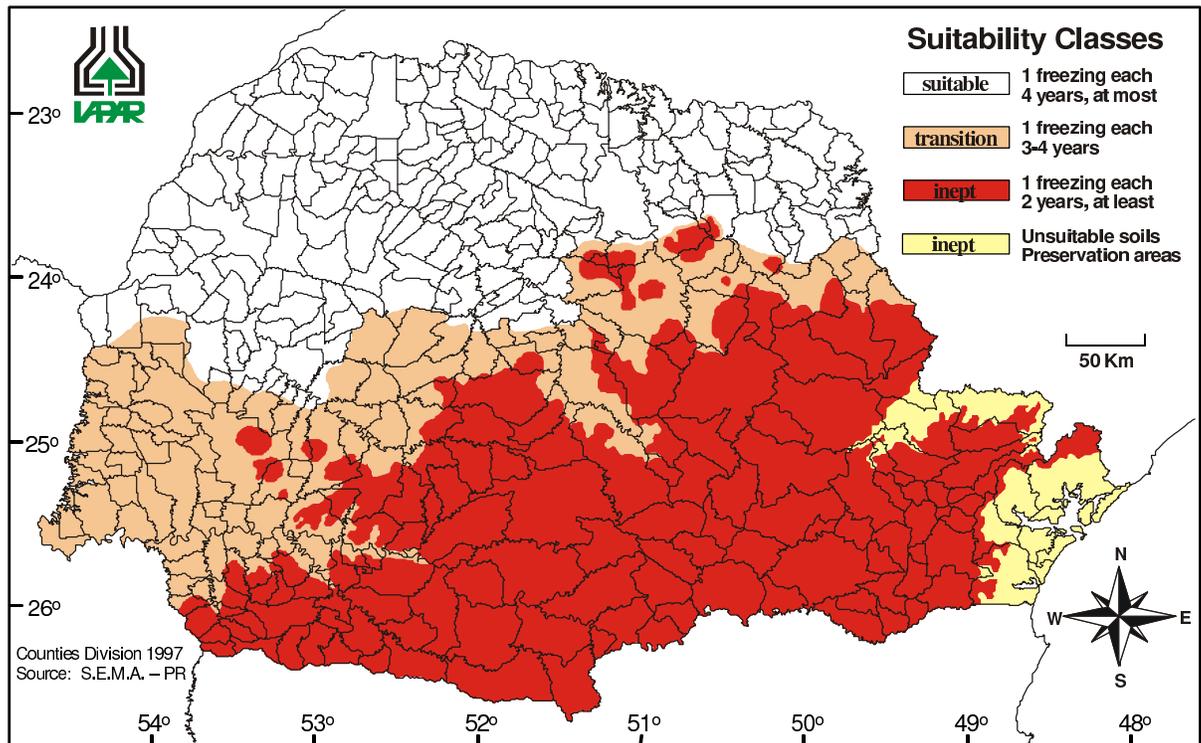


Figure 3.2: Land suitability map for *Coffea arabica* in Paraná State

probabilities) are visualized as maps, showing the distribution of the relevant measurements or estimations across the region. When all these activities finish and deliver their results, the activity *Classify Parcels* fuses these partial results to determine the suitability of the expected environmental conditions across the lands for the crop.

The data sources and activities for agricultural zoning may be dispersed across different sites over the Internet. Furthermore, these processes are sensitive to crop, location, and time, i.e., they depend on the species and variety of the crop, the environmental characteristics of the region, and the opinion of the experts involved. The granularities for which these processes are defined are usually not uniform. Indeed, for some crops it is possible to devise a generic zoning process, while other crops require specific processes for each plant variety. Similarly, certain zoning processes are defined for vast regions and others for specific land parcels.

The map of Figure 3.2, borrowed from [75], shows the land suitability results for *Coffea arabica* in the state of Paraná. It shows, for instance, that in the southern area of the state, one freezing event happens on average every 2 years. Freezings can impair the productivity and even kill coffee trees, rendering that area unsuitable for coffee cultivation. Governments and financial institutions rely on this kind of information, for instance, to define and enforce adequate loan granting policies. These policies direct farmers to choices and practices that

contribute to lessen risks and increase the productivity of their enterprises. Experiences in sectors of Brazilian agriculture [212] in the last few years corroborate the economic advantages of adopting this scientific approach to agricultural zoning.

3.2.3 Technical challenges

In our application example, the semantics of data are interrelated with the processes that manipulate them, so that data and processes cannot be completely decoupled. Interconnected activities cooperate with each other to process data collected from several heterogeneous distributed sources, giving rise to distributed processes whose complexity requires their organization in several abstraction levels. The outputs of a process can contribute to the inputs of other processes. The data sources to be taken into account and the resulting information, for each specific application, are dynamically defined by user requirements and contingent on climatic conditions. The analysis of the results gives feedback to improve the process or devise new ones. However, despite the numerous variants of these processes, some patterns can be recognized.

These scientific processes are in fact vast and distributed efforts for data integration and fusion. By *data integration* we mean the transformations applied to heterogeneous data so that they can be analyzed together for some specific purpose. It does not imply that data must be coerced and congealed into a global schema. What matters is the correct interpretation and use of the data. *Data fusion* consists in applying some function to a collection of data values to produce other meaningful values (e.g., fuse the expected environmental conditions of a land parcel to determine its suitability for a crop). Our experience with scientific applications shows that data integration and fusion are scattered across the constituent activities of complex processes at distinct abstraction levels. Experts in this kind of context face many challenges, some of which are described below.

Identifying Resources Lack of catalogs and inspection mechanisms to find and reuse available Web resources to solve each particular problem.

Systems Interoperability Domain experts and technicians waste time converting data among formats of different tools. This effort should be spent on application-specific issues.

Data Traceability There is no means to track data provenance, i.e., their original source and the way they were obtained and processed. This hampers the evaluation of whether the quality of a data item satisfies the requirements of a particular application.

Process Documentation and Execution Processes are rarely documented. When this is done, the specifications produced are either not broad enough for giving a general view of the processes or not formal enough to allow the automatic repetition of the process with different data sets.

Process Versatility There should be schematic means to reformulate processes on the fly. This kind of decision support system relies on continuous feedback to improve the processes – as data keep arriving and results are produced, the processes may evolve.

Adaptation and Reuse Mechanisms for adaptation and reuse of Web services could boost productivity and enhance the quality of the results.

These issues are common to several kinds of applications involving distributed processes over the Web. The following sections describe the POESIA approach for handling some of these issues.

3.3 Ontological delineation of utilization scopes

Ontologies [110] describe the meaning of terms used in a particular domain, based on semantic relationships observed among these terms. In the POESIA approach, they play a crucial role in composing Web services. Concretely, ontologies delineate the utilization scopes of data sets and processes and orient the refinement and composition of Web services. A *utilization scope*, or *scope* for short, is a context in which different data sets and specific versions of a repertoire of services can be used. In this section, we describe the structure of our multidimensional ontologies and how they delineate and correlate utilization scopes. These are the foundations of our scheme to catalog and reuse components and ensure the semantic consistency of the resulting Web services compositions.

3.3.1 Semantic relationships between words

Let Ω be a set of simple and/or composite words referring to objects or concepts from a universe of discourse U . *Objects* are specific instances (e.g., Brazil). *Concepts* are classes that abstractly define and characterize a set of instances (e.g., Country) or classes. The *universe of discourse* gives a context where the meaning of each word $w \in \Omega$ is stable and consistent.

The field of linguistics defines several semantic relationships between words. We consider the following subset in this work:

Synonym Two words are *synonyms* of each other if they refer to exactly the same concepts or objects in U .

Hypernym/hyponym A word w is a *hypernym* of another word w' (conversely w' is a *hyponym* of w) if w refers to a concept that is a generalization of the concept referred to by w' in U . Hyponym is the inverse of hypernym.

Holonym/meronym A word w is a *holonym* of w' (conversely w' is a *meronym* of w) if w' refers to a concept or object that is part of the one referred to by w in U . Meronym is the inverse of holonym.

Roughly speaking, synonym stands for equivalence of meaning, hypernym for generalization (IS_A), and holonym for aggregation (PART_OF). For example, in the agriculture realm, *Cultivar* is a *synonym* of *Variety of Plant* and *Crop* is a *hypernym* of *Cultivar*.

A set of words Ω is said to be *semantically consistent* for the universe of discourse U and a set of semantic relationships Υ if at most one semantic relationship of Υ holds between any pair of words in Ω . This ensures some coherence for the meanings of the words in Ω for U .

The semantic relationships defined above preserve certain properties. Let w , w' , and w'' be any three words and θ denote one of the semantic relationships considered. Then, for a given universe of discourse U , the following conditions hold:

- w *synonym* w (reflexivity)
- $w \theta w' \wedge w' \theta w'' \Rightarrow w \theta w''$ (transitivity)
- w *synonym* $w' \wedge w' \theta w'' \Rightarrow w \theta w''$ (transitivity wrt synonyms)

These properties enable the organization of a set of semantically consistent words Ω according to their semantic relationships in a given universe of discourse U . The *synonym* relationship partitions Ω into a collection of subsets such that the words of each subset are all synonyms. The transitivity of the *hypernym* and *holonym* relationships correlates the semantics of words from different subsets of synonyms, inducing a partial order among the words of Ω . The resulting *arrangement of semantically consistent words* is a directed graph G_Ω that expresses the relative semantics of the words of Ω for the universe of discourse U (see proof in Annex I). The nodes of G_Ω are the subsets of synonyms of Ω . The directed edges of G_Ω represent the semantic relationships among the words of different subsets. There is a directed edge from vertex \mathfrak{R} to vertex \mathfrak{R}' of G_Ω if and only if each word of \mathfrak{R} is the *hypernym* of all the words of \mathfrak{R}' or each word of \mathfrak{R} is the *holonym* of all the words of \mathfrak{R}' .

Consider the case where all the words of Ω represent concepts. Then an arrangement of semantically consistent words is called an arrangement of semantically consistent concepts. Figure 3.3 illustrates an arrangement of concepts for territorial subdivisions. It is an extract from a very large set of ontological concepts used by experts for developing agricultural applications.

The concepts appear in the rectangles. The edges representing hypernym relationships are denoted by a diamond close to the specific concept, and the edges representing holonym relationships are denoted by a black circle close to the component concept. This graph denotes that a *Country* is composed of a set of *States* or, alternatively, a set of *Country Regions*. A *Country Region* may be a *Macro Region*, an *Official Region*, or another kind

of region. Macro and Official Regions are composed of States, but a region of type Metro Area is composed of Counties. Eco Region and Macro Basin define other partitions of space based on ecological and hydrological issues, respectively. There is no constraint on the geometry of the land parcels modeled according to these concepts, except for the containment relationships implied by the *hypernym* and *holonym* relationships (e.g., each state must be inside one country).

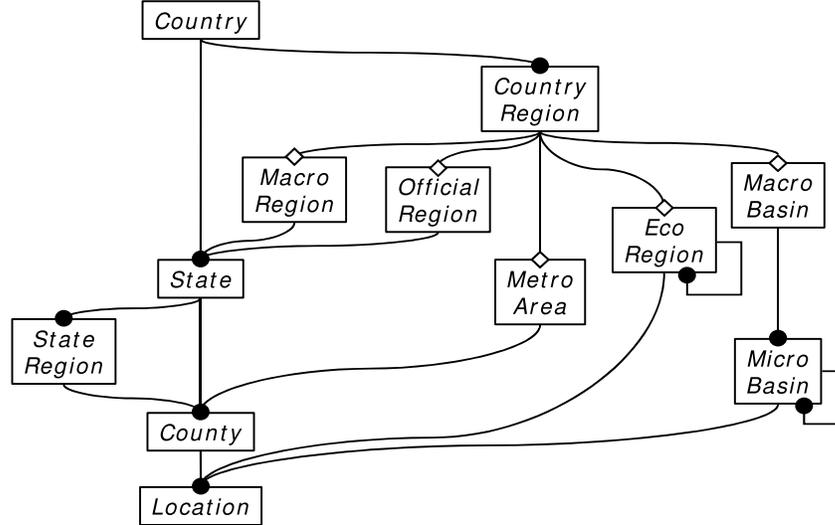


Figure 3.3: An arrangement of concepts relative to territorial subdivisions

Given an arrangement G_Ω for a semantically consistent set of words Ω , we say that a word $w \in \Omega$ *encompasses* another word $w' \in \Omega$, denoted by $w \models w'$, if and only if w and w' are in the same vertex of G_Ω (i.e., $w = w'$ or w *synonym* w') or there is a path in G_Ω leading from the vertex containing w to the vertex containing w' (i.e., there is a sequence of *hypernym* and/or *holonym* relationships relating the meaning of w to the more restricted meaning of w'). The encompass relationship is transitive (see proof in Annex I). According to Figure 3.3, $\text{Country} \models \text{State}$, $\text{Country} \models \text{County}$, and so on.

Now consider the instantiation of the concepts from Figure 3.3. For example, the concept Country can be instantiated to Brazil, State to its states, and so on. Let us call the instances of concepts *terms*. If there is a semantic relationship between two concepts of an arrangement of concepts, the same relationship holds between terms instantiated from these concepts. Therefore, the arrangement of semantically consistent concepts plays a role like that of a schema for the corresponding set of terms, inducing a similar structure (direct graph) to arrange the semantically consistent terms. Figure 3.4a illustrates a subgraph of the arrangement of concepts from Figure 3.3 and one corresponding arrangement of terms referring to Brazilian regions, states, and so on.

Terms are not restricted to instances of objects. Figure 3.4b illustrates an arrangement of

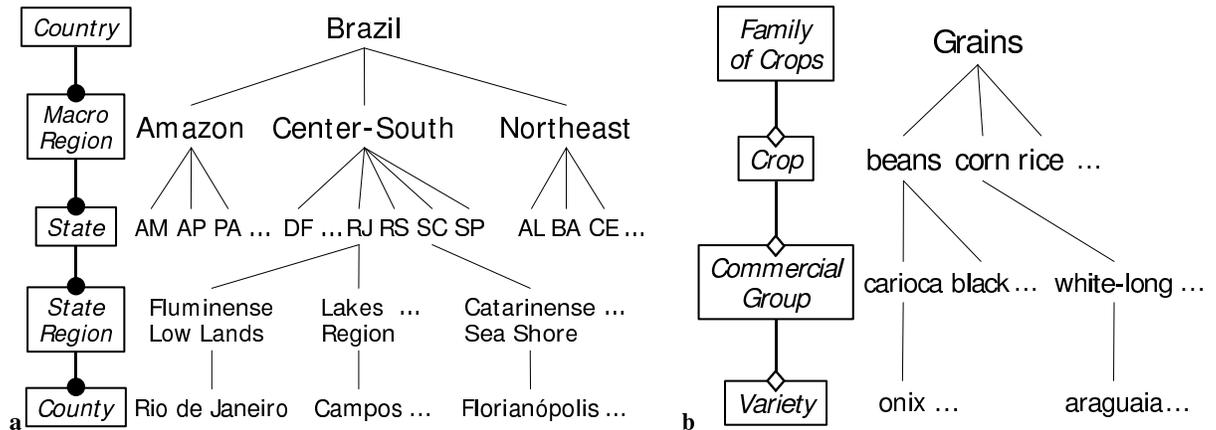


Figure 3.4: Arrangements of semantically consistent terms

concepts and one corresponding arrangement of terms referring to crops and their varieties. Grains, beans, rice, corn, etc. do not refer to specific objects but to concepts (or classes). This is an example of a specialization relationship between the terms and their respective concepts. Further formalization of these notions is outside the scope of this paper and appears in (Annex I).

3.3.2 POESIA ontologies and ontological coverages

A *POESIA ontology* is a collection of arrangements of semantically consistent terms. Each arrangement describes a particular dimension of the domain. For instance, Figure 3.4 presents fragments of arrangements of terms for the (a) space and (b) product dimensions, with the respective arrangement of concepts on the left of each hierarchy. On referring to a term of such a hierarchy, one must qualify the term with the corresponding concept of the respective arrangement of concepts by using the expression *concept(term)* in order to avoid ambiguity. Thus *State(RJ)* refers to the Brazilian state called Rio de Janeiro (RJ is an acronym), while *County(RJ)* refers to the county of the same name.

An entire path in the hierarchy may be required to precisely indicate a term (e.g., if the same county name appears in different states). An *unambiguous reference to a term* of an ontology Σ is a path in one of the arrangements of terms of Σ . This path is expressed by the concatenated sequence of *concept(term)* vertices visited within it. This sequence, when taken as a string, must be unique across all the dimensions of the ontology. For instance, *State(RJ).County(Campos)* is an unambiguous reference to the county called Campos in the state called Rio de Janeiro. The term *Crop(bean)* is an unambiguous

reference, too, because there is only one crop called beans.

Finally, we are ready to define ontological coverages and their properties. An *ontological coverage* is a tuple of unambiguous references to terms of a POESIA ontology. Some examples of ontological coverages are:

[Country(Brazil)],
 [Crop(beans)],
 [Country(Brazil),Crop(beans)], and
 [Country(Brazil),Crop(beans),Crop(rice)].

Each of these ontological coverages expresses one *utilization scope*, or *scope* for short, i.e., a context in which a data set or service can be used.

An individual term of an ontological coverage expresses a utilization scope in a particular dimension. For instance, the term Country(Brazil), defined in the space dimension, expresses the utilization scope “the whole country called Brazil”. The universal coverage (denoted by ∞) is the empty tuple. It does not restrict the utilization scope in any dimension. The scope expressed by terms referring to the same dimension is a restriction of the universal scope to the union of the scopes expressed by the individual terms. For instance, the ontological coverage [State(RJ),State(SP)] expresses a scope obtained by the union of the scopes individually expressed by the terms State(RJ) and State(SP). The scope expressed by terms referring to different dimensions restricts the universal scope to the intersection of the scopes expressed by the individual terms. For example, [State(RJ),Crop(orange)] restricts the scope to the intersection of the scopes defined by the spatial dimension term State(RJ) and the agricultural product dimension term Crop(orange). To narrow the scope in a particular dimension, one has to choose a more specific term in the ontology (e.g., go from State(RJ) to County(Campos)). The absence of terms for a particular dimension means that the scope is not restricted to that dimension.

The semantic relationships among the terms of a POESIA ontology induce semantic relationships among ontological coverages. Given two ontological coverages, C and C' , defined with respect to the same ontology Σ , C *encompasses* C' , denoted by $C \models C'$, if and only if for each term $w \in C$ there is another term $w' \in C'$ such that $w \models w'$ (where w and w' are in the same dimension of Σ). For example, [Country(BR)] \models [Country(BR).Region(CS)], i.e., the whole country encompasses its Center-South region.

The encompass relationship between ontological coverages is transitive, inducing a partial order among coverages referring to the same ontology (see proof in Annex I). The universal coverage encompasses any other. Thus, $\infty \models$ [Country(BR)], [Country(BR)] \models

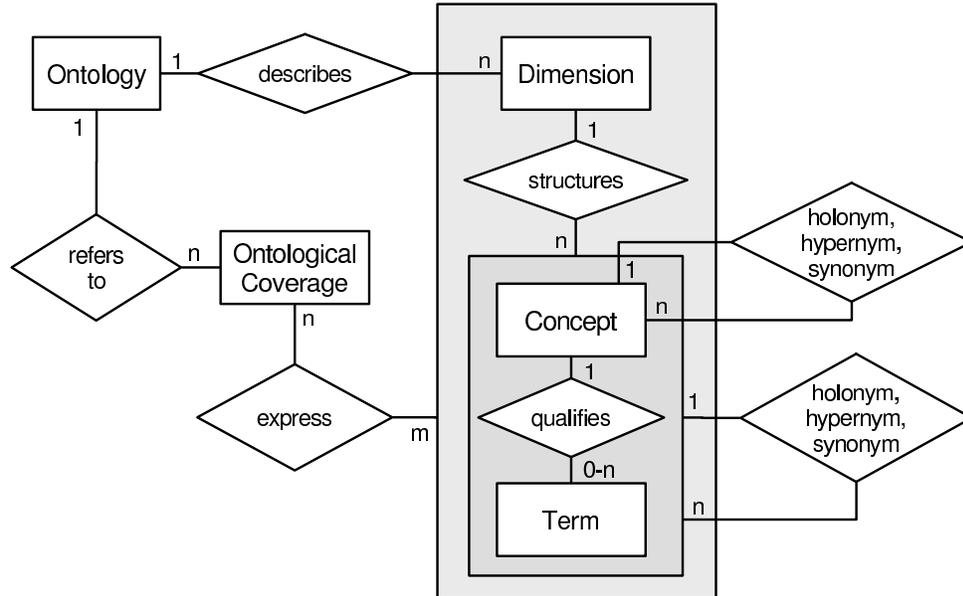


Figure 3.5: A schema for POESIA ontologies and ontological coverages

[Country(BR), Crop(beans)], and so on. One can also evaluate the *equivalence of ontological coverages*. Two ontological coverages C and C' are equivalent (denoted by $C \equiv C'$) if and only if they encompass each other (i.e., $C \models C'$ and $C' \models C$). This occurs if each term in C has a synonym in C' and vice versa. For example, [Country(Brazil)] \equiv [Country(BR)] because BR can be used as a synonym of Brazil.

Figure 3.5 presents an entity-relationship diagram for POESIA ontologies and the ontological coverages defined according to such ontologies. It shows that a POESIA ontology has one or more dimensions. The domain-specific terms for each dimension are organized in an arrangement of semantically consistent terms. The qualifiers of these terms, i.e., the concepts defining the classes of terms, are organized in an arrangement of semantically consistent concepts for each dimension. An ontological coverage is a tuple of terms taken from one or more dimensions of an ontology.

3.4 The POESIA activity model

3.4.1 Overview

The basic construct of the model is the *activity pattern*. It may refer to any kind of data processing task – computational and/or manual. These tasks are performed in an open environment, comprising several platforms. In POESIA, activity patterns are implemented as Web services.

An activity pattern has a set of communication ports, called *parameters*, to exchange data

with other activity patterns and data repositories. Each parameter of an activity pattern refers to a Web service encapsulating a data source or sink for that particular pattern. Each input parameter is associated with outputs of another activity pattern or with a data repository. Conversely, each output parameter is associated with inputs of another activity pattern or with a data repository.

POESIA employs aggregation, specialization, and instantiation of activity patterns to organize and reuse the components of processes as proposed in [157, 154]. These mechanisms determine how processes can be composed and adapted. Activity pattern composition is depicted by a hierarchical graph, where intermediate nodes are composite patterns and leaves are atomic or simple patterns. The latter must be specialized before they are decomposed.

A hierarchy of activity patterns, i.e., of Web services, is called a *process framework*. Each activity pattern of a process framework is associated with an ontological coverage that expresses its utilization scope in order to drive the selection and reuse of components. A process framework must be refined, adapted to a particular situation, and instantiated before execution. POESIA provides some rules to check the semantic consistency of process frameworks and instantiated processes based on correlations of the ontological coverages of their constituents. For example, the ontological coverages of all the components of a process framework must be compatible with (encompass or be encompassed by) the ontological coverage of the highest activity in the hierarchy.

Let us illustrate these notions with a simple example. Figure 3.6 presents a simplified framework for agricultural zoning. It shows that the major components of *Agricultural Zoning* are *Calculate Climate Attributes* and *Determine Land Suitability*. The former, which is composed of *Collect Weather Indicators* and *Consolidate Climate Data*, collects weather data from a variety of Web services and consolidates them into the Web services of land climate attributes. The activity pattern *Determine Land Suitability* takes the climate attributes, along with other data relevant for one specific crop, to determine the most appropriate lands for that crop.

This framework applies to the zoning of any crop. To obtain instantiated processes for specific crops, one must adapt the constituent activities to the peculiarities of that crop. For example, the relevant environmental conditions for zoning coffee (discussed in Section 3.2.1) are different from those for zoning rice. Thus *Determine Land Suitability* and its two constituents must be specialized for each crop. In addition, a specific activity must be defined to assess each relevant environmental condition for each crop. On the other hand, the activities that calculate climate attributes do not require adaptation, as one general Web service can supply climate data to several specific services for determining land suitability for different crops. The ontological coverages associated with the Web services enable automated means to check their compatibility for composition with respect to their utilization scopes. This helps domain experts to organize and compose the services necessary for their applications and factor their solutions to reduce costs according to domain-specific concepts and reasoning.

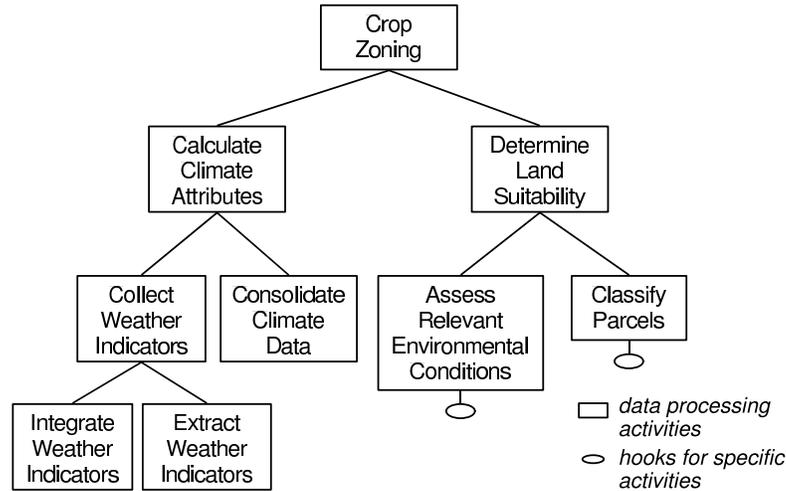


Figure 3.6: Process framework for agricultural zoning

3.4.2 Activity pattern

An *activity pattern* is an abstraction that defines the structure and behavior of a collection of instances of data processing activities implemented as Web services, much like a class does for instances of objects [154]. Activity patterns also resemble software design patterns [96] in the sense that each activity pattern is designed to solve a well-defined category of problems in a particular utilization scope. Definition 3.4.1 depicts the structure of an activity pattern.

Definition 3.4.1 An **activity pattern** α is a five-tuple:

$$(NAME, COVER, IN, OUT, TASK)$$

where:

NAME is the string used as the name of α

COVER is the ontological coverage of α
i.e., expresses its utilization scope

IN is the list of input parameters of α

OUT is the list of output parameters of α

TASK describes the processing chores that α does

NAME, *COVER*, *IN*, and *OUT* represent the *external interface* or *signature* of the pattern. *TASK* specifies the behavioral semantics of the activity pattern including the composition semantics and the execution dependencies between component patterns.

Figure 3.7 presents the textual specification of an activity pattern to determine land suitability for an arbitrary crop whose *NAME* is *DetLandSuitability*, ontological coverage,

```

#DEFINE RNA "http://www.agric.gov.br/rna/pub_docs"

ACTIVITY_PATTERN DetLandSuitability [Country(BR), Cons(RNA)]

  INPUTS
    ClimAttr:  "RNA/clim_info.wsd";
    LandsInfo: "RNA/lands_info.wsd";
    CropInfo:  "RNA/crops_info.wsd";
  OUTPUTS
    Zoning:    "RNA/agric_zoning.wsd";
  LOCAL
    EnvCond:  "RNA/env_cond.wsd";

  BEGIN TASK
    COMPOSITION
      AssessEnvCond (IN: ClimAttr, LandInfo, CropInfo;
                    OUT: EnvCond);
      ClassifyParcels(IN: EnvCond; OUT: Zoning);
    EXECUTION DEPENDENCIES
      AssessEnvCond PRECEDES ClassifyParcels;
  END TASK;

END ACTIVITY_PATTERN;

```

Figure 3.7: Activity pattern to *Determine Land Suitability* for an unspecified crop

COVER, is [Country(BR), Cons(RNA)], i.e., Brazil, according to the methodology of RNA,¹ the *IN* and *OUT* parameters are specified as *INPUTS* and *OUTPUTS*, and *TASK* is composed of two activity patterns – *AssessEnvCond* and *ClassifyParcels* – invoked within *DetLandSuitability*. These component patterns are assumed to be declared elsewhere. Figure 3.7 also shows a few special keywords. The `#DEFINE` clause specifies an alias for a URI that is frequently used in the pattern specification. `LOCAL` declares the internal variables of the pattern. The delimiters `BEGIN TASK` and `END TASK` enclose the specification of the *TASK*. `COMPOSITION` enumerates the constituent patterns of a composite pattern. `EXECUTION DEPENDENCIES` establishes the relative order of execution of the constituent patterns. `EXECUTION DEPENDENCIES` and `TASK DESCRIPTION` are optional. Another example of task description is provided in Section 3.4.4.

An activity pattern implemented as a Web service is uniquely identified by the URI of the site holding it, its name, and its ontological coverage. All the data exchanged by activity patterns can be viewed in XML. Each parameter is associated with some description of the capabilities of the corresponding Web service – like the `.wsd` (Web Service Description) files referenced in

¹RNA stands for *Rede Nacional de Agrometeorologia* (National Agro-meteorological Network), a consortium of Brazilian institutions linked to agricultural research.

Figure 3.7. The service descriptions must provide links to DTD or XML-schema specifications that define the types of all data elements that can be exchanged via the respective parameters. Links are defined as URIs.

The description of each activity pattern parameter includes the description of the interface of the services that can be bound to that parameter to support more sophisticated communication than just transferring packets of semistructured data. For example, the service that supplies climate data to `DetLandSuitability`, denoted by the parameter `ClimAttr`, allows the target to pose queries (e.g., OLAP operators) specifying filters and granularities for the data to be transferred (e.g., to get the average temperature in a certain region for each month). Note that data filters and granularities can also be expressed by ontological coverages. This makes POESIA ontologies central not only as a means of organizing data and services but also for defining the communication interfaces for Web services. The designer of a process can refer to published Web service and schema descriptions or develop his own descriptions to fulfill specific demands. This encourages standardization and at the same time confers flexibility to Web services and data representation.

The following subsections present the operations for composing activity patterns (implemented as Web services) and some rules to check the semantic consistency of these compositions. The specifications of activity patterns and their compositions (Figures 3.7, 3.10 and 3.12) are written in a language that we are developing for this purpose. This language takes advantage of ontological coverages to describe, organize and ensure semantic correctness of Web service compositions. Some aspects of our workflow specification language, such as synchronizing mechanisms, are outside the scope of this work. In the future, we can substitute our language for some standard for Web services composition (e.g., WSFL [255], BPEL4WS [250]). We plan to extend such a standard with ontological coverages and associated rules to express the composition of Web services, by aggregation and specialization of the respective activity patterns, emphasizing the correlations of the services' utilization scopes.

3.4.3 Activity pattern aggregation

In POESIA, a complex activity pattern is defined as an aggregation of a set of component activity patterns. A component activity pattern can itself be a complex activity pattern or an elementary activity pattern. Figure 3.8 shows the activity pattern *Determine Land Suitability*, which is an aggregation of the activity patterns *Assess Environmental Conditions* and *Classify Parcels*.

When decomposing an activity pattern into its constituents (or, conversely, composing an activity pattern from the components), we have to make sure that there is no conflict among names and ontological coverages of the activity patterns involved and that all parameters are connected.

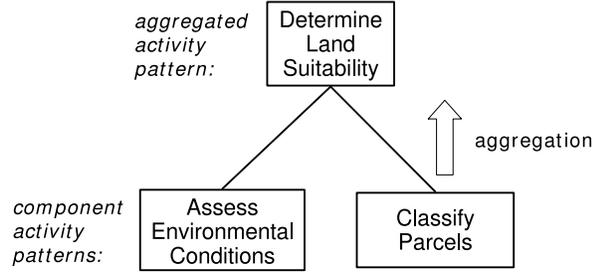


Figure 3.8: An aggregation of two activity patterns

Definition 3.4.2 Activity pattern α is an **aggregation** of the activity patterns β_1, \dots, β_n ($n \geq 1$) if the following conditions are verified (let $1 \leq i, j \leq n; i \neq j$ for each condition):

1. $\forall \beta_i : NAME(\alpha) \neq NAME(\beta_i) \vee COVER(\alpha) \neq COVER(\beta_i)$
2. $\forall \beta_i, \beta_j : NAME(\beta_i) \neq NAME(\beta_j) \vee COVER(\beta_i) \neq COVER(\beta_j)$
3. $\forall \beta_i : COVER(\alpha) \models COVER(\beta_i) \vee COVER(\beta_i) \models COVER(\alpha)$
4. $\forall p \in IN(\alpha) : \exists \beta_i$ such that $p \in IN(\beta_i)$
5. $\forall p \in OUT(\alpha) : \exists \beta_i$ such that $p \in OUT(\beta_i)$
6. $\forall \beta_i, p' \in IN(\beta_i) : p' \in IN(\alpha) \vee (\exists \beta_j$ such that $p' \in OUT(\beta_j))$
7. $\forall \beta_i, p' \in OUT(\beta_i) : p' \in OUT(\alpha) \vee (\exists \beta_j$ such that $p' \in IN(\beta_j))$

We call α an *aggregated (or composite) activity pattern* and each β_i a *constituent (or component) activity pattern*.

Definition 3.4.2 states that an activity pattern α is defined as an aggregation of n component activity patterns β_1, \dots, β_n if they satisfy the above-mentioned seven conditions. Condition 1 says that the name and the ontological coverage of each constituent pattern β_i must be different from the name and coverage of the aggregated activity pattern. Condition 2 specifies that the name and coverage of a constituent activity pattern can uniquely distinguish itself from other constituent patterns of α . Condition 3 states that the ontological coverage of the composite pattern α must encompass the coverage of each constituent pattern β_i or vice versa, i.e., the intersection of their utilization scopes is not null. Condition 4 ensures that every input parameter of α is connected to an input parameter of some constituent β_i . Similarly, condition 5 ensures that each output parameter of α is connected to an output parameter of some β_i . Finally, conditions 6 and 7 state that all parameters of constituent patterns must be connected to a parameter of other constituent or the aggregated pattern.

3.4.4 Activity pattern specialization

The descriptors of an activity pattern can be refined when specializing that activity pattern for a particular situation. Figure 3.9 illustrates a specialization of the activity pattern *Classify Parcels* for the crop *C. arabica*.

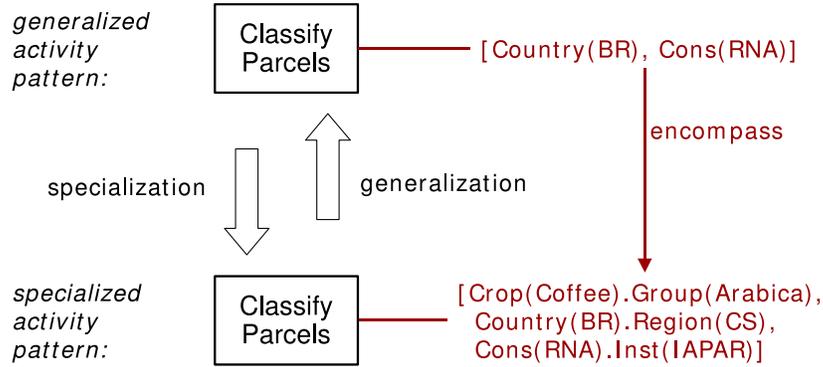


Figure 3.9: A specialization of *Classify Parcels*

The specialization of an activity pattern can be formally defined by relationships similar to those used to define the aggregation abstraction.

Definition 3.4.3 Activity pattern β is a **specialization** of the activity pattern α (conversely α is a generalization of β) if the following conditions are verified:

1. $NAME(\alpha) \neq NAME(\beta) \vee COVER(\alpha) \neq COVER(\beta)$
2. $COVER(\alpha) \models COVER(\beta)$
3. $\forall p \in IN(\alpha) : \exists p' \in IN(\beta)$ such that $p \vdash p'$
4. $\forall p \in OUT(\alpha) : \exists p' \in OUT(\beta)$ such that $p \vdash p'$

We call α the generalized activity pattern of β and β a specialized activity pattern (version) of α .

Condition 1 of definition 3.4.3 states that the name and/or ontological coverage of the generalized activity pattern α must be different from those of its specialized version β . Condition 2 states that the ontological coverage of α must encompass that of β . The notation $p \vdash p'$ in conditions 3 and 4 means that each parameter p' of β must refer to a Web service that is a refinement of the Web service referred to by the corresponding parameter p of α . This refinement of Web services can refer to their capabilities or data contents. The exact relationship between the

generic and the refined parameters is defined in the description of the corresponding Web services. Ontological coverages can be associated with these Web services to express and correlate their utilization scopes.

```
#DEFINE IAPAR "http://www.pr.gov.br/iapar/pub_docs"

ACTIVITY_PATTERN
ClassifyParcels [Crop(Coffee).Group(arabica),
                 Country(BR).Region(CS).State(PR),
                 Cons(RNA).Inst(IAPAR)]

REFINES ClassifyParcels [Country(BR), Cons(RNA)]

INPUTS
  EnvCond->WDI:           "IAPAR/wdi.wsd";
  EnvCond->AvgAT:         "IAPAR/avg_at.wsd";
  EnvCond->ProbHeat:      "IAPAR/prob_heat.wsd";
  EnvCond->ProbFreeze:    "IAPAR/prob_freeze.wsd";
OUTPUTS
  Zoning->Zon_Coffee:    "IAPAR/zoning_coffee.wsd";

BEGIN TASK
  DESCRIPTION
    OVERLAY
      IF WDI <= 150 THEN "OK" ELSE "Water restriction";
      IF ProbHeat <= 30 THEN "OK"
        ELSE "Thermal restriction";
      IF AvgAT <= 24 THEN "OK" ELSE
        IF WDI <= 100 THEN "OK"
          ELSE "Thermal restriction";
      IF ProbFreeze <= 25 THEN "Low risk of freeze" ELSE
        IF ProbFreeze <= 50 THEN "Medium risk of freeze";
        ELSE "High risk of freeze";
    END TASK;
END ACTIVITY_PATTERN;
```

Figure 3.10: *Classify Parcels* for *Coffea arabica* in Paraná

Figure 3.10 shows the specialized version of the activity pattern *Classify Parcels* for *Coffea arabica*, according to the methodology of Paraná Agricultural Institute (IAPAR) [75], a member of RNA. The clause `REFINES` indicates that this pattern is one specialization of the pattern `ClassifyParcels` with a wider scope expressed by `[Country(BR), Cons(RNA)]`. Each parameter declared in the specialized version is explicitly related to the corresponding one of the generalized pattern. The notation `EnvCond->WDI` indicates that the parameter `WDI` of the specialized version is derived from the parameter `EnvCond` (the expected environmental conditions) of the generalized version of `ClassifyParcels`. The other input paramete-

ters of the specific version of `ClassifyParcels` also derives from the generic parameter `EnvCond`. The output parameter `ZonCoffee` of the specialized version is a refinement of the parameter `Zoning` of the generalized activity pattern. The `TASK DESCRIPTION` clause overlays logical conditions involving the measurements of the relevant environmental conditions for the crop.

3.4.5 The combined refinement mechanism

The aggregation and specialization of activity patterns can be combined to define a complex activity pattern whose constituents depend on the utilization scope to which the complex pattern is specialized. The definition of such a complex activity pattern must conform to both the conditions of aggregation and the conditions of specialization. Figure 3.11 illustrates a refinement of the activity pattern *Assess Environmental Conditions* for *C. arabica*.

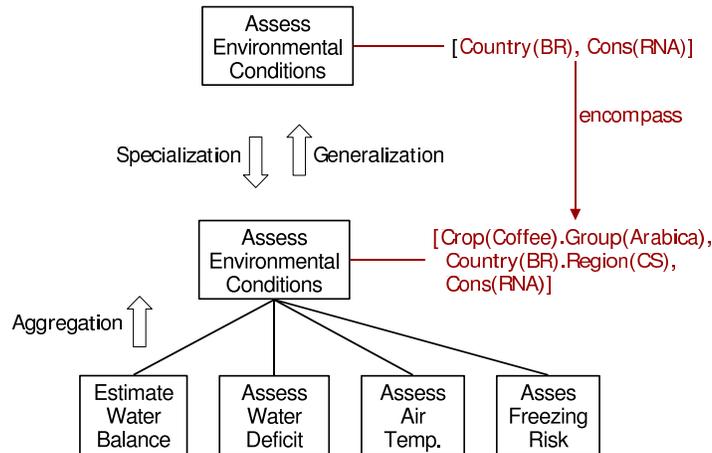


Figure 3.11: Combining specialization and aggregation

Specialization and aggregation of activity patterns are intertwined. The specialization details the parameters and constituents of a pattern for a particular utilization scope, establishing a flat view at a particular abstraction level to express the cooperation of the constituent patterns. Problems related to parameter passing – type checking, parameter uniqueness, and disambiguation – are solved by defining parameter scopes just as in programming languages: a parameter’s scope is local to the specification of activity pattern where it is defined.

Figure 3.12 shows the specialized version of `AssessEnvCond` (*Assess Environmental Conditions*). The input parameter `ClimAttr` appears in both the generalized and the specialized version. The `LandsInfo` parameter of the generalized version unfolds in `Relief` and `WaterRetSoil` in the specialization. `CropInfo` unfolds in `CropCoef` and `WaterDemands`. The output `EnvCond` of the generalized version unfolds in `WDI`, `AvgAT`, `ProbHeat`, and `ProbFreeze`. The `LOCAL` parameter `WaterBal` is used to transfer data between

```

ACTIVITY_PATTERN
AssessEnvCond [Crop(Coffee).Group(Coffea arabica),
               Country(BR).Region(CS), Cons(RNA)]

REFINES AssessEnvCond [Country(BR), Cons(RNA)]
INPUTS
  ClimAttr:           "RNA/clim_info.wsd";
  LandsInfo->Relief:  "RNA/relief.wsd";
  LandsInfo->WaterRetSoil: "RNA/water_ret_soil.wsd";
  CropInfo->CropCoef:   "RNA/coffee_water_coef.wsd";
  CropInfo->WaterDemands: "RNA/coffee_water_dem.wsd";
OUTPUTS
  EnvCond->WDI:       "RNA/wdi.wsd";
  EnvCond->AvgAT:     "RNA/avg_at.wsd";
  EnvCond->ProbHeat:  "RNA/prob_heat.wsd";
  EnvCond->ProbFreeze: "RNA/prob_freeze.wsd";
LOCAL
  WaterBal: "RNA/water_bal.wsd";

BEGIN TASK
COMPOSITION
  EstWaterBal (IN: ClimAttr,WaterRetSoil,CropCoef;
              OUT: WaterBal);
  AssessWaterDeficit (IN: WaterBal,WaterDemands;
                    OUT: WDI);
  AssessAirTemp(IN: ClimAttr; OUT: AvgAT,ProbHeat);
  AssessFreezeRisk (IN: ClimAttr,Relief; OUT: ProbFreeze);
EXECUTION DEPENDENCIES
  EstWaterBal PRECEDES AssessWaterDeficit;
  (AssessWaterDeficit AND AssessAirTemp
   AND AssessFreezeRisk)
   PRECEDES ClassifyParcels;

END TASK;
END ACTIVITY_PATTERN;

```

Figure 3.12: *Assess Environmental Conditions for Coffea arabica* in Brazil's Center-South

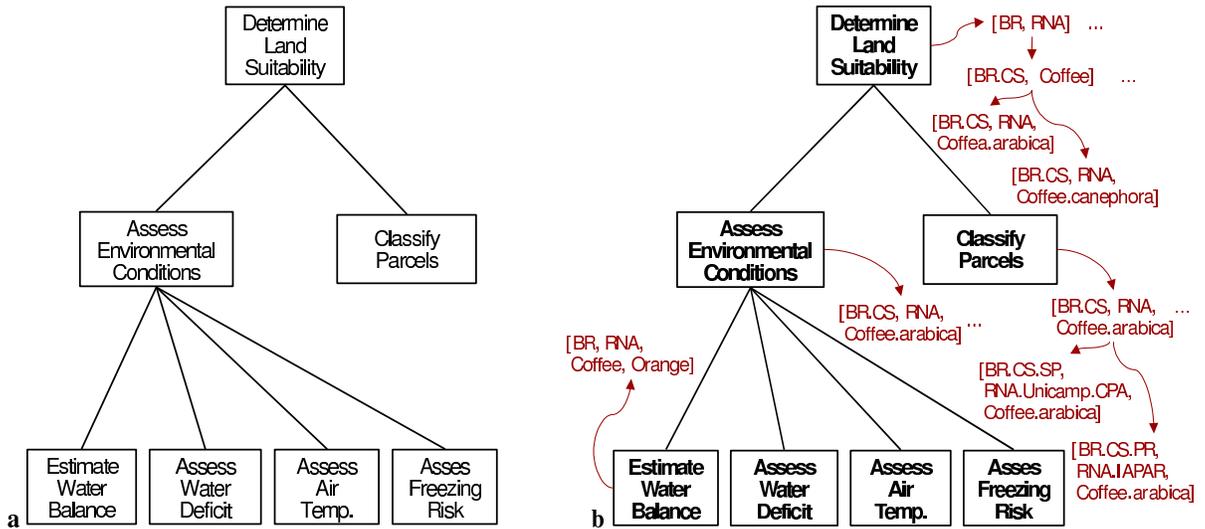


Figure 3.13: Hierarchies of activity patterns for determining land suitability for *Coffea arabica*: (a) decomposition hierarchy; (b) multi-fold hierarchy or process framework

EstWaterBal and AssessWaterDeficit. The binding of these parameters expresses the data flow illustrated in Figure 3.1. The clause EXECUTION DEPENDENCIES states that EstWaterBal precedes AssessWaterDeficit, and ClassifyParcels initiates after all the other constituents have finished.

3.4.6 Process framework

In POESIA, activity patterns can be defined in terms of other activity patterns through aggregation and specialization of activity patterns. As a result, a hierarchy of activity patterns can be formed. We call such a hierarchy a process framework of the root activity pattern. Figure 3.13a shows a process framework to determine land suitability for *Coffea arabica*, presenting only compositions of activity patterns. Figure 3.13b extends Figure 3.13a by adding the hierarchies of specializations of some activity patterns in the hierarchy. We say that a hierarchy like that shown in Figure 3.13b is multifold because each of its activity patterns (nodes) can have two kinds of immediate subordinates: its constituent patterns and its specialized versions.

Definition 3.4.4 A process framework is a directed graph $\Phi(V_\Phi, E_\Phi)$ satisfying the following conditions:

1. V_Φ is the set of vertices of Φ
2. E_Φ is the set of edges of Φ

3. $\forall v \in V_{\Phi} : v$ is an activity pattern
4. $(v, v') \in E_{\Phi} \Leftrightarrow v'$ constituent $v \vee v'$ specialization v
5. Φ is acyclic
6. Φ is connected

Definition 3.4.4 establishes the structural properties of a process framework – a directed graph $\Phi(V_{\Phi}, E_{\Phi})$ whose nodes represent the activity patterns and whose directed edges correspond to the aggregation and specialization relationships among these patterns. Condition 4 states that there is a directed edge (v, v') from vertex v to vertex v' in Φ if and only if v' is a constituent of v or v' is a specialization of v . Condition 5 states that no sequence of aggregations and/or specializations of patterns in Φ can lead from one pattern to itself. This restriction is necessary because aggregation and specialization can intermingle. In such a case, an aggregation may break the gradual narrowing of the utilization scopes achieved by specialization. Condition 6 guarantees the connectivity of the activity patterns participating in the process framework Φ .

Adaptation of a process framework

A process framework captures the possibilities for reusing and composing Web services to build consistent processes for different situations in terms of utilization scopes, data dependencies, and execution dependencies among components. The adaptation of a process framework for a particular scope consists in choosing (and developing if necessary) components to compose a process tailored for that scope.

Definition 3.4.5 A process specification $\Pi(V_{\Pi}, E_{\Pi})$ associated with a utilization scope expressed by an ontological coverage C is a subgraph of a process framework satisfying the properties:

1. $\forall (v, v') \in E_{\Pi} : v'$ constituent v
2. $\forall v \in V_{\Pi} :$
 $(\nexists v' \in V_{\Pi} \text{ such that } (v, v') \in E_{\Pi}) \Rightarrow v$ is atomic
3. $\forall v \in V_{\Pi} : COVER(v) \models C$

Definition 3.4.5 states that a process specification Π is a subgraph of a process framework. Condition 1 states that Π is a decomposition hierarchy, i.e., all its edges refer to aggregations of activity patterns. Condition 2 states that all the leaves of Π are atomic patterns, otherwise Π

would be missing some constituents for its execution. Condition 3 ensures that the ontological coverage of each pattern participating in Π encompasses the coverage C associated with Π , i.e., the intersection of the utilization scopes of all the constituents of Π are equivalent or contain the utilization scope of Π .

Refinement and adaptation of process frameworks can alternate in practice. Frameworks, specific processes, or individual activity patterns can always be reused to produce new or extended frameworks. Additionally, when adapting a framework, the development of activity patterns to contemplate specific needs also contributes to enrich the repertoire of specialized patterns of a framework.

Process instantiation

Note that all the elements of the POESIA model presented above are at the conceptual level. Thus, after adapting a process framework to produce a process specification for a particular situation, this process has to be instantiated for execution. Instantiating a process specification Π consists in assigning concrete Web services to handle the inputs and outputs of each activity pattern of Π , allocating sites to execute the corresponding tasks and designating agents (humans or programs with the appropriate abilities and roles) to perform them.

The location of the concrete resources assigned to execute a process is independent of the locations of their descriptions. The selection of the concrete resources to perform the process during its instantiation confers an extra level of execution independence to POESIA. Once particular resources have been assigned, the specific formats and protocols used to connect them can be defined. This may be done by using the binding mechanisms of Web services specification languages like WSDL [254].

POESIA metamodel

Figure 3.14 shows the POESIA metamodel, which is an extension of the workflow reference model of the WfMC [123]. It summarizes, in bold, our extensions: (1) associate an ontological coverage with each activity pattern; and (2) associate a resource description with each port (parameter) of each activity pattern. A resource description also includes an ontological coverage to describe its utilization scope. This allows the organization of a repertoire of activity patterns according to their utilization scopes and helps to determine the services for reuse in specific situations and the rules to connect them.

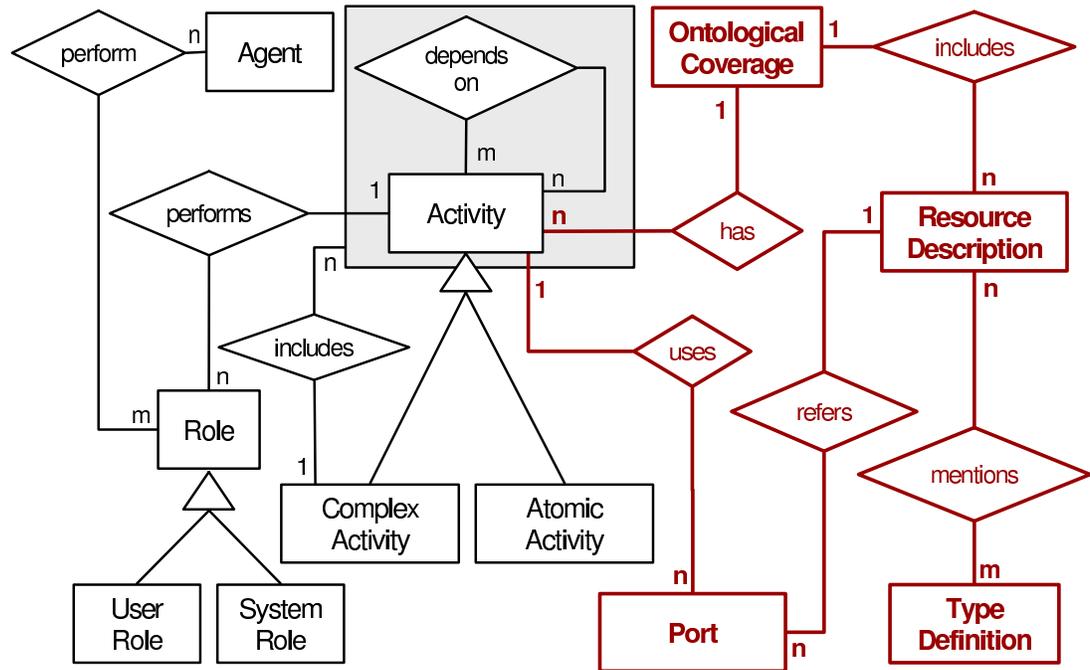


Figure 3.14: The POESIA process definition meta model

3.5 Implementation issues

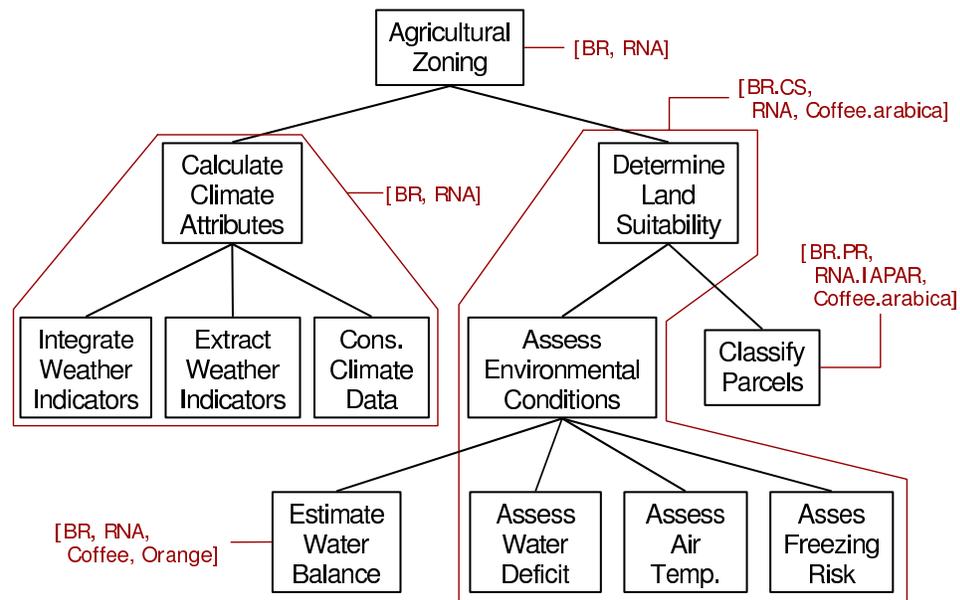
A number of issues are important in the implementation of the POESIA approach to Web services composition: (1) correctness of the composition semantics, (2) mechanisms for composing Web services through ontology construction and ontology reasoning, and (3) an efficient and scalable implementation architecture. In this section, we discuss how POESIA handles these issues.

3.5.1 Checking specifications

Hierarchy of activity patterns

The aggregations and specializations of activity patterns must be checked for the properties expressed in definitions 3.4.2 and 3.4.3. The direct graphs corresponding to process frameworks must be acyclic and connected as stated in definition 3.4.4. Furthermore, the conditions expressed in definition 3.4.5 must be checked when adapting a framework for a particular utilization scope.

Figure 3.15 illustrates a process for zoning *C. arabica* in Paraná State. All the activity patterns in this structure, starting with its root, have compatible ontological coverages. The ontological coverage of *Agricultural Zoning* encompasses that of *Calculate Climate Attributes*,

Figure 3.15: Zoning *Coffea arabica* in Paraná State

Determine Land Suitability, and so on. The activity pattern *Estimate Water Balance* has a wider coverage including coffee and orange, i.e., the same pattern for calculating the water balance is used for both crops.

Execution and data dependencies

The collection of execution dependencies among activity patterns can be represented in a dependency graph. Figure 3.16 presents the dependency graph for the process framework for zoning *C. arabica*. It shows that the execution of the activity pattern *Consolidate Climate Attributes* can be initiated only after successfully finishing the execution of *Integrate Weather Indicators* or *Extract Weather Indicators*, which provide data (from weather stations or remote sensing, respectively) for updating the climate attributes. When *Consolidate Climate Data* has done its work, *Estimate Water Balance*, *Assess Air Temperature*, and *Assess Freezing Risk* can execute in parallel. The conclusion of *Estimate Water Balance* triggers the execution of *Assess Water Deficit*. *Classify Parcels* can only start executing after a successful execution of all the previous activities.

A similar dependency graph for the data dependencies is inferred from the connection of parameters amid process frameworks. These two graphs must be compatible. Individually, these graphs must be acyclic and connected. Properties relative to the structure and the dynamics of the execution and data dependencies among activity patterns can be evaluated with algorithms based on Petri Net formalisms. For example, [235] proposes an algorithm to translate workflow

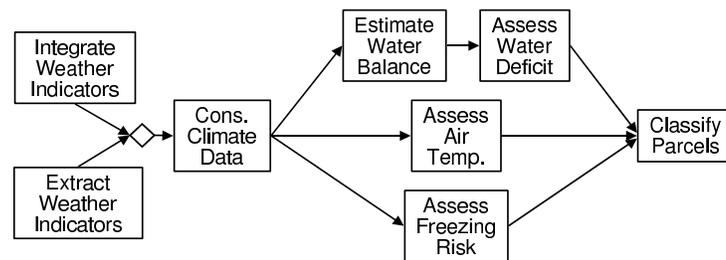


Figure 3.16: Execution dependencies among activity patterns for zoning *Coffea arabica*

graphs into WF-Nets, a class of Petri Nets tailored to workflow analysis. The verification of the properties of WF-Nets allows the automatic detection of design errors in the corresponding workflow specifications. The absence of deadlocks in a workflow, for instance, is associated with the soundness property of the corresponding Petri Net. Roughly speaking, the soundness property states that for every reachable state of the Petri Net there must be a sequence of steps leading to the final state.

3.5.2 Composing Web services: an implementation perspective

A POESIA Web service can access a collection of existing Web service functioning as data sources for its processes and publish its own processes and data sets as Web services. Each POESIA-enabled Web site organizes its service description, composition, and interconnection apparatus according to the representation layers of the Semantic Web [80, 215]. In the bottom layer, XML wrapping, source data are converted into XML, thus providing a syntax standard for semistructured data in the extensional level. The XML-related standards confer versatility and expression power for representing and interrelating documents on the Internet. The second layer is the schemas and processes layer. It uses DTDs or XML schema to represent data sets at the intentional level to factor the problems related to data heterogeneity. POESIA frameworks appear at the top of the second layer and provide specific criteria based on utilization scopes to select services and check the semantic consistency of their connections. The third layer is the semantic description layer, which describes the services, at a higher abstraction level, using RDF statements and process description standards like DAML-S [61, 14]. These resource descriptions must conform to metadata standards and vocabularies, including domain-specific ones. The vocabulary used in the first, second, and third layers is defined in the fourth layer, which maintains a dictionary. The top layers of the Semantic Web infrastructure – namely, logic, proof, and trust – are not contemplated at this moment.

POESIA services in different sites can be logically arranged in successive abstraction levels. Figure 3.17 illustrates such a situation. The process specification stored in server A is composed

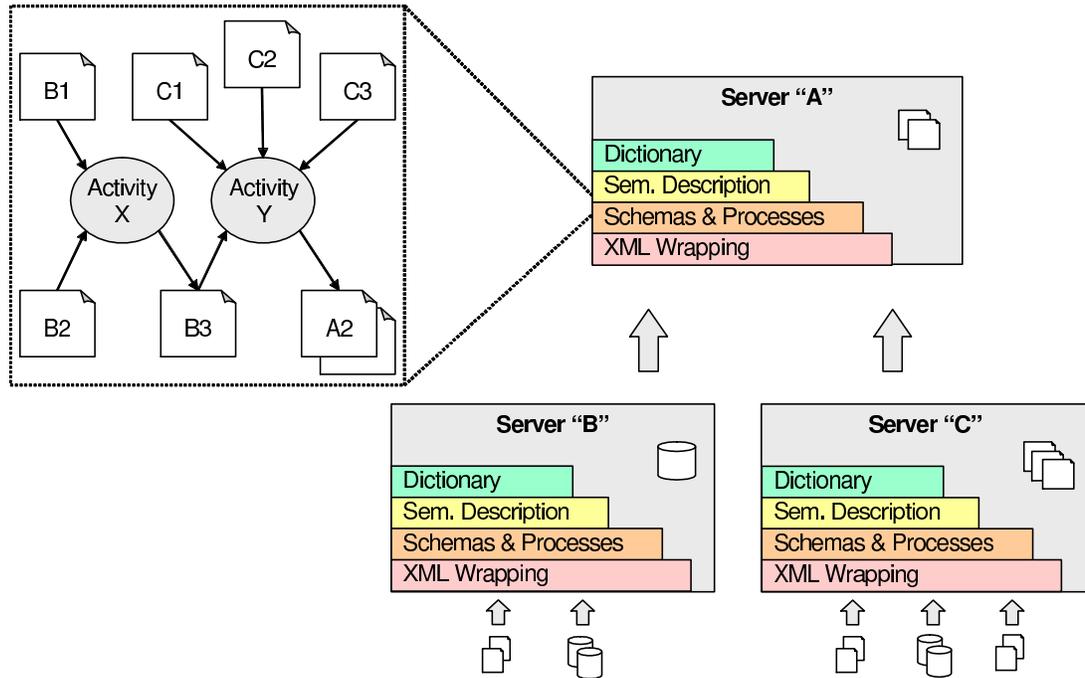


Figure 3.17: The multi-tier distributed infrastructure for composition of Web services

of two cooperating activity patterns, X and Y . Activity pattern X accesses the Web services described by $B1$ and $B2$ to take its inputs, process them, and push its outputs into the Web service described by $B3$ (consider that $B1$, $B2$, and $B3$ are published in server B). Then Y takes its data inputs from the Web services described by $C1$, $C2$, $C3$ (all published in C), and $B3$ to generate the outputs pushed in the Web service described by $A2$ (maintained and published by A itself).

3.5.3 Architecture

Figure 3.18 presents the architecture of a peer-to-peer site supporting POESIA services, outlining the communication with external sites and service brokers. The *Services Specification Tool* allows the domain expert to build solutions for particular needs. This tool supports browsing the resources available locally or remotely in order to discover components to reuse. The descriptions and formal specifications of the local services are stored in the *Local Services* repository. One service may encapsulate one or more data sets. The *Local Data* repository maintains the data and metadata associated with local services. All the constituents of a service specification stored in the site are indexed by one ontology of the *Local Ontologies* repository. The *External Resources Locator* provides access to the descriptions of external resources. The *Catalog of External Resources* functions as a cache for the descriptions of external resources frequently

accessed. Each local service and ontology can be published and used by external Web services.

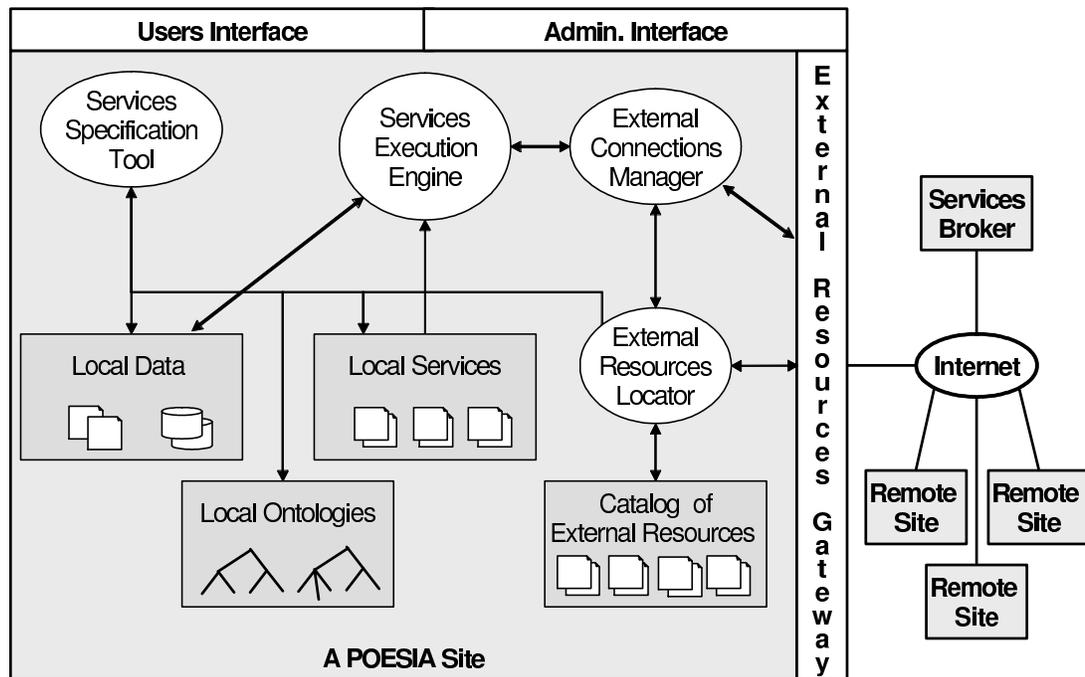


Figure 3.18: The architecture of a POESIA-enabled peer to peer Web site

The *Services Execution Engine* interprets the service specifications to properly manage the corresponding fragments of distributed processes. A service can be activated locally or by some external connection. A locally running service can also activate remote services to obtain its inputs or send its outputs. The *External Connections Manager* controls the communication with remote components and users at run time. It relies on the *External Resources Locator* to retrieve the descriptions of external resources whenever necessary. The thicker double arrows connecting the *Local Data* repository with the *Services Execution Engine*, and the latter with the *External Connections Manager*, which is linked to the *External Resources Gateway*, represent the data exchange between a local service and remote resources during the execution of the distributed processes. A POESIA site also has two kinds of human-computer interfaces. The *User Interface* allows the domain experts to specify and activate services; the *Administration Interface* serves configuration purposes.

The architecture of a POESIA-enabled Web site contemplates two types of external resources: *Remote Sites* and *Service Brokers*, though it does not rule out connections with other kinds of resources. A *Remote Site* has the internal structure described for our POESIA site. *Service Brokers* are special sites that catalog the descriptions of the resources available across the Web to support the discovery and selection of resources.

3.6 Related work

The Semantic Web [80, 215] intends to extend the capabilities of the current Web to cope with problems such as finding precise information in the vast amount of resources available and supporting interinstitutional applications like electronic commerce. The means for achieving this are: standards for expressing machine-processable meta-information (e.g., RDF, DAML+OIL), development and dissemination of terminologies using these standards (e.g., domain ontologies), and new tools and architectures based on this apparatus to build applications empowered with semantics and automated reasoning capabilities. POESIA relies on the infrastructure of the Semantic Web to implement certain techniques, based on domain expertise, to organize, select, and reuse data and services in the Web.

The POESIA approach to compose Web services through activity aggregation and specialization was inspired by the needs of our application domain and is founded by earlier work done in transactional activity modeling by Liu [157, 154], where a set of mechanisms are proposed and formalized for specification and reuse of activities. Other research areas directly related to POESIA are the use of metadata and ontologies for Web services description, discovery and composition [14, 61, 198, 38, 166, 37], and workflow techniques for scientific processes and Web service composition [123, 234, 235]. Descriptions of the meaning, properties, capabilities, and ontological relationships among Web services, expressed in languages like DAML services [14, 61], support mechanisms to discover, select, activate, compose, and monitor Web resources. Related work covers various aspects, ranging from theoretical studies to implementation efforts, from architecture issues to conceptual models [124, 260].

Concretely, Paolucci et al. [198] show that the capabilities of registries such as UDDI and languages like WSDL are not enough to support services discovery. They employ DAML-S for this purpose and present an algorithm to match service requests with the profile of advertised services based on the minimum distance between concepts in a taxonomy tree. Cardoso and Sheth [38], on the other hand, present metrics to select Web services for composing processes. These metrics take into account functional and operational features such as the purpose of the services, quality of service (QoS) attributes, and the resolution of structural and semantic conflicts. McIlraith et al. [166] use agent programming to define generic procedures involving the interoperation of Web services. These procedures, expressed in terms of concepts defined with DAML-S, do not specify concrete services to perform the tasks or the exact way to use available services. Such procedures are instantiated by applying deduction in the context of a knowledge base, which includes properties of the agent, its user, and the Web services. Finally, Bussler et al. [37] sketch an architecture for Web services attaining Semantic Web aspirations.

The grounding of Web services involves several abstraction layers between the semantic specification and the implementation [221]. Currently there is a myriad of proposals for speci-

ifying Web services composition in intermediate layers, such as WSFL (IBM), BPML (BPMI), XLANG (Microsoft), BPEL4WS (BEA, IBM, Microsoft), WSCI (BEL, Intalio, SAP, Sun), XPDL (WfMC), EDOC (OMG), and UML 2.0 (OMG). These proposals concern the synchronization of the execution of Web services in processes running across enterprise boundaries [234, 20]. They build on top of standards like XML, SOAP, WSDL, and UDDI, providing facilities to interoperate and synchronize the execution of Web services that can use different data formats (e.g., heterogeneous XML schemas) and communication protocols (HTTP, XMTP, etc.). Some challenges for these technologies are to (i) reduce the amount of low-level programming necessary for the interconnection of Web services (e.g., through declarative languages), (ii) provide flexibility to establish interactions among growing numbers of continuously changing Web services during run time, and (iii) devise mechanisms for the decentralized and scalable control of cooperative processes running on the Web.

To illustrate the differences between our approach and Web service synchronizing languages, let us consider two of them: WSFL and BPML. The Web Services Flow Language (WSFL) [255] is an XML language for the description of Web services compositions. WSFL considers two types of Web services compositions. *Flow models* specify the appropriate usage pattern of a collection of Web services and how to choreograph the functionality provided by a collection of Web services to achieve a particular business need. *Global models* specify the interaction pattern of a collection of Web services, describing how components of a set of Web services interact with each other. POESIA can be seen as a value-added method with an emphasis on using domain-specific ontologies to guide and facilitate the interaction among a set of Web services in terms of service utilization scopes.

The Business Process Modeling Language (BPML) is specialized in supporting control flows of business process patterns. BPML and POESIA share the same objectives of supporting Web service composition. The main differences, however, lie in the mechanisms and methodology used in the underlying framework. BPML promotes the use of control constructs such as merge, split, multimerge, exclusive choice, and so forth to facilitate the composition of services, whereas POESIA combines the control logic with domain-specific ontologies, with an emphasis on complex composition semantics at both the data level and workflow activity level.

In summary, to the best of our knowledge, current proposals focus mainly on business processes; there is a lack of research on supporting semantic consistency for Web services refinement and reuse. The POESIA approach contemplates the demands of some scientific applications. Furthermore, it addresses the semantic consistency issue by using domain ontologies. POESIA complements the current technologies for Web services description, discovery, and composition (including approaches based on ontologies for describing services, like DAML-S) in two ways. First, it provides mechanisms to select Web services according to their utilization scopes (e.g., services intended for particular regions and classes of products). Second, it enables automated means to check if compositions of Web services are semantically correct with respect

to these scopes (e.g., to determine if a Web service for estimating the water balance of lands covered with bushes can be properly incorporated in a process to determine land suitability for coffee).

3.7 Conclusions

Many scientific applications, including agroenvironmental applications such as agricultural zoning, are built by composing heterogeneous data sources and services. Large data sets are organized according to time and space dimensions, e.g., climate data rely on time series of weather data and expected water content in soil is measured in spatial terms. Well-defined metadata precisely describing the meaning of these data sets are required for their correct composition. Agricultural zoning is an application built on scientific models (e.g., the matching of weather data with the plant model of growth and water requirements over time) and has very high economic impact. For example, government agencies and financial institutions use agricultural zoning to make decisions on policies and loan approvals for farmers that want to plant specific crops.

In this paper, we introduced the POESIA approach to support the systematic composition of Web services. It is founded on domain ontologies in which the properties of the semantic relationships between terms induce a partial order among the terms for each dimension of a reality (e.g., space, time, product). Current ontology engineering tools, such as Protégé and OntoEdit, can help to develop such ontologies. Using tuples of terms from these ontologies to express and correlate the utilization scopes of data and services, the POESIA activity model defines activity patterns that specify the Web service composition and communication channels that link these services together.

POESIA complements current proposals for Web services description, selection, and composition by using domain ontologies to (i) conceptually organize vast collections of services, (ii) uncover and select data and services according to their utilization scopes, and (iii) check semantic and structural consistency properties of compositions of Web services. We illustrated the POESIA approach through a real application scenario: the agricultural zoning of *Coffea arabica* in the Center-South region of Brazil.

On top of this foundation, we are investigating further extensions of POESIA. Knowledge management and keeping track of data provenance in distributed processes can be more easily supported when Web services are built from well-defined ontologies and through well-defined operations based on activity pattern composition. Precise documentation of data provenance will be useful in the evaluation of the quality and suitability of results for many applications. A richer set of semantic relationships can also be considered to enhance POESIA capabilities for expressing and managing the utilization scopes of data and services. Another concern is aspects of the synchronization of Web services. These issues are being considered by several

Web services synchronization languages (e.g., WSFL, BPEL4WS, XPDL). POESIA's strength is in handling semantic aspects of Web services composition using domain ontologies. We are investigating extensions to its activity model to incorporate synchronization mechanisms using an existing proposal. On the one hand, our research will continue to be guided by real-world applications such as agricultural zoning. On the other hand, the generality and abstraction of POESIA makes it useful to many next-generation Web service-based applications.

Acknowledgments

The first author is partially supported by Embrapa, CAPES, and the Finep/Pronex/IC/SAI95/97 project. The authors from Georgia Tech are partially supported by two grants from the Operating Systems and ITR programs (CISE/CCR division) of NSF, by a contract from the SciDAC program of DoE, and by a contract from the PCES program (IXO) of DARPA. All agriculture data used in this paper were provided by Brazilian experts. Thanks to the anonymous reviewers who contributed to improve this work.

Chapter 4

Using Domain Ontologies to Help Track Data Provenance

4.1 Introduction

Data provenance (also called data *genealogy* or *pedigree*) is the description of the origins of a piece of data and the process by which it was produced [33]. This problem has been studied in a variety of settings, ranging from cooperative processes with data exchange in several formats, to chains of views over relational databases for loading data warehouses. The solutions proposed in the literature usually involve some kind of annotation or the “inversion” of the functions/queries used to transform data.

The Internet poses new challenges for provenance tracking. The autonomy of the components and the multi-institutional nature of Web applications results in a profusion of data contents, demanding self-describing data sets. Traditional approaches for tracking data provenance, relying on detailed descriptions and tight control of the data transformation flow, cannot be easily adapted to the Web. Detailed information about distributed data processing on the Web, such as the queries/functions used to transform and move data across sites, are often unavailable. A better solution in this context is to build a general framework for provenance tracking, including detailed analysis of specific portions when necessary and empathizing the semantics of data and processes.

POESIA (Chapter 3) (Processes for Open-Ended Systems for Information Analysis) is an approach for multi-step integration of semi structured data in an open and distributed environment. Inspired by the needs of scientific applications such as agricultural planning, POESIA combines ontologies, workflows and activity models to provide novel facilities for data integration using cooperative services. This approach pursues the vision of the Semantic Web [22, 215] and offers some concrete solutions for data integration, service composition and provenance tracking on the Web.

This paper focuses on the POESIA ontological approach for estimating data provenance. Domain ontologies depict the semantic relationships among terms, grouped according to different dimensions of one reality (e.g., space, time and product). Tuples of terms, called *ontological coverages*, express the *scopes* of data sets and *granularities* of data values in several dimensions (e.g., the spatial extents, periods of time and products that a data set or value refers to). The semantic relationships between terms induces a partial order among ontological coverages. This order is used to correlate scopes and granularities of data, enabling an estimation of data provenance. The major contribution of this paper is a framework for tracking data provenance, using ontologies to express data contents and the effect of chains of data integration operations on data sets. This framework can achieve efficient and fine grain provenance tracking with negligible maintenance cost.

The remainder of this paper is organized as follows. Section 4.2 presents an agricultural application used as a running example throughout the paper. Section 4.3 outlines the fundamentals of POESIA ontologies needed for provenance tracking. Section 4.4 describes the ontological method for tracking the provenance of aggregated values. Section 4.5 analyzes typical operators for data integration and the use of ontologies for data integration and provenance tracking, from a general perspective. Section 4.6 discusses related work. Finally, section 4.7 summarizes contributions and extensions.

4.2 Motivating Example

The problem investigated here is the following: given a data item, what were the original data items and the chain of data processing steps that produced it? Let us examine a real life scenario concerning data integration in agricultural applications. Figure 4.1(a) illustrates the consolidation of weather data through a hierarchy of intra and inter-institutional repositories. Each institution has a set of weather stations (data collecting devices), scattered across its operational area, to collect measurements such as maximum, minimum and average temperature and total rainfall per hour. These data are maintained in the repositories of the institutions that collect them. The spatial and temporal *scopes* of the institutional data sets (i.e., the land parcels and periods of time they cover) can overlap. For example, institution I1 operates in a limited region, while institution I2 has a wider spatial scope. Institution I3 encompasses units I3a and I3b. The data warehouse of consortium C1 consolidates data from I1 and I2, C2 from I2 and I3, and C3 from C1 and C2. This processing scheme produces data sets with successively broader scopes and denser sampling. The data granularity in the upper levels can be either the same or coarser than the granularity of the source data (e.g., from an hourly to a daily basis). The data at the lower levels tend to be more detailed and precise (but not necessarily accurate), while the data at the higher levels usually convey more abstraction, since they refer to increasingly broader scopes. Typical operations to produce such aggregations of the source data can be seen

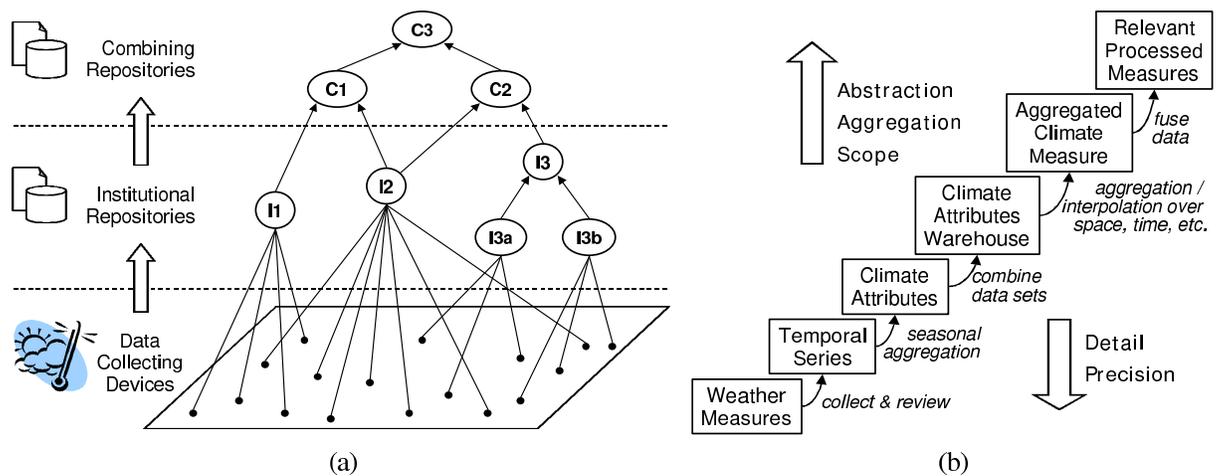


Figure 4.1: Integrating data sets in many steps

as variations of the basic data cube operations such as slide, dice, roll-up, and drill down.

Figure 4.1(b) gives a general view of the step-wise integration of weather data. First, the raw data collected by the weather stations of each institution are gathered, reviewed and stored as temporal series. Then, aggregation of historical data from each weather station generates the climate attributes for that particular point on the earth surface (e.g., average temperature and rainfall per month). Data warehouses (such as those in C1, C2 and C3) offer unified access to climate attributes originated from several sources, with aggregation and interpolation facilities for recovering consolidated data – typically OLAP to select and aggregate data over time and space, and interpolations to produce maps with estimations of the distribution of climate measurements across the lands. Finally, applications such as agricultural zoning (Chapter 3) integrate and fuse data taken from these warehouses, among other sources, to derive other relevant information. Most of these applications need to understand not only the semantics of the data used, but also their provenance.

Figure 4.2 shows the star schema of the data warehouses used in case studies throughout this paper. The Climate data warehouse has a data table with the values of maximum, minimum and average temperature and total rainfall, organized by the dimensions of territorial divisions, time, products and organizations. The Crops production warehouse maintains the planted area, production, unit and monetary value, for each county, month and crop produced. Notice the similarities between the respective dimensions of these warehouses. The following sections show how to represent these dimensions in an ontology and the use of such an ontology to help track data provenance.

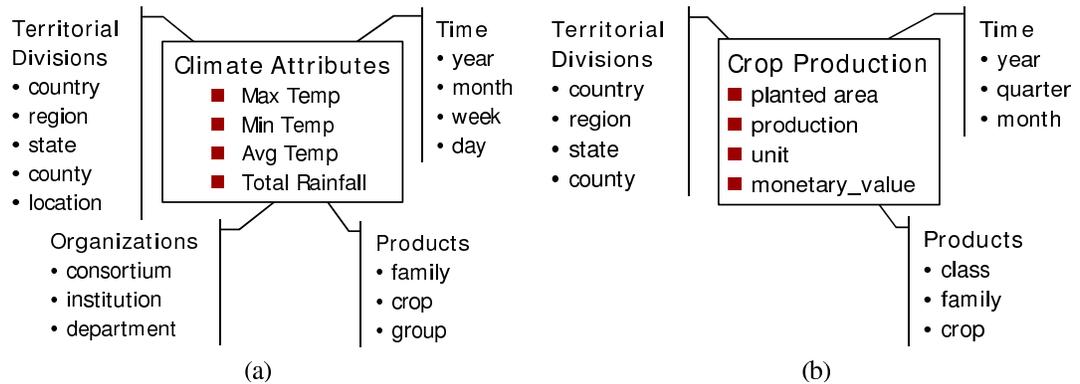


Figure 4.2: Agricultural data warehouses: (a) climate attributes; (b) crops production

4.3 POESIA Ontologies and Ontological Coverages

Figure 4.3 shows the space dimension described in a POESIA ontology. The directed acyclic graph in the left, called an *arrangement of concepts*, formalizes the semantic relationships among the territorial subdivision concepts. The edges representing PART_OF relationships have a black circle close to the specific concept, and the edges representing IS_A relationships have a diamond close to the component concept. This graph denotes that a Country is composed of a set of States or, alternatively, a set of Country Regions. A Country Region may be a Macro Region, an Official Region or another kind of region. Macro and Official Regions are composed of States, but a region of type Metro Area is composed of Counties. Eco Region and Macro Basin define other partitions of space, based on ecological and hydrological issues, respectively. The arrangement of concepts provides a general framework, being instantiated by arrangements of terms. The middle part of figure 4.3 illustrates a subgraph of the arrangement of territorial subdivision concepts. An *arrangement of terms* instantiated from these concepts is represented by the directed acyclic graph (in this case a hierarchy) on the right side. There are also SYNONYM relationships not represented in the figure due to space limitations (e.g., BR can be used as a synonym to Brazil). An instantiated term need to be qualified with the corresponding concept, in order to avoid ambiguity. Thus, State(RJ) refers to the state Rio de Janeiro, while County(RJ) refers to the county of the same name.

Similar structures describe concepts and instantiated or instancialized terms for other dimensions (such as time and products). The arrangements of concepts and terms for all the relevant dimensions constitutes a *POESIA ontology*. A tuple of terms from a POESIA ontology, called an *ontological coverage*, can describe the *scope* of a data set or the *granularity* of an aggregated value. For example, [State(RJ), Crop(orange), Year(2002)] restricts the scope to the intersection of the spatial, crop and temporal scopes defined by the terms State(RJ),

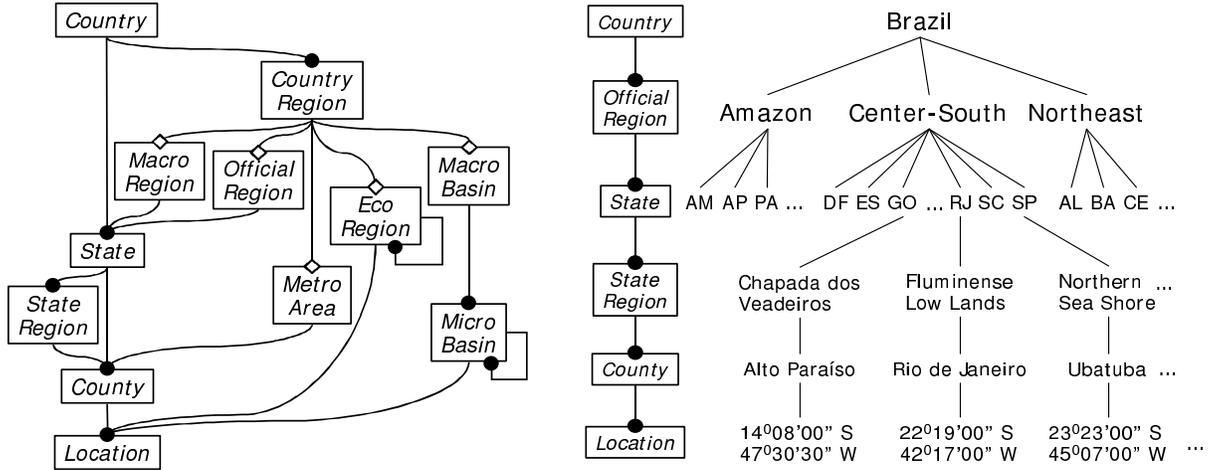


Figure 4.3: The space dimension of a POESIA ontology: (left) arrangement of concepts; (middle and right) a compatible arrangement of terms

Crop(orange) and Year(2002) in a multidimensional space. The ontological coverage $[State(RJ), State(SP)]$, on the other hand, denotes the union of the spatial scopes expressed by the two terms, because both refer to the same dimension. To narrow the scope in a particular dimension one has to choose a more specific term (e.g., go from $State(SP)$ to $County(Ubatuba)$).

The semantic relationships represented in POESIA ontologies induce a partial order among ontological coverages that we call *semantic encompassing*: e.g., country Brazil encompasses state Rio de Janeiro, denoted by $[Country(Brazil)] \models [State(RJ)]$. Furthermore, $[State(RJ)] \models [State(RJ), Year(2002)]$ and $[State(RJ), State(SP)] \models [State(SP)]$. Two ontological coverages are *equivalent* if they refer to the same scope (e.g., $[Country(BR)] \equiv [Country(Brazil)]$). A data set or item can be associated with an ontological coverage expressing its scope and another one expressing the minimum among the granularities of its components. The scope of a data set or item must encompass the scopes of its components and its minimal granularity. The scope of a data value is equivalent to its granularity. We can show, for a limited set of semantic relationships between terms, that the encompassing relationship is reflexive and transitive. A more formal treatment of POESIA ontologies, with demonstrations of their properties, can be found in Annex I.

4.4 Ontological Estimation of Data Provenance

Let us consider the union of data sets in data warehouses. The ontological coverages described in the previous section can express the scope of the data sources and of the resulting data sets. Figure 4.4(a) illustrates the data flow for the consolidation of crop production data, involving

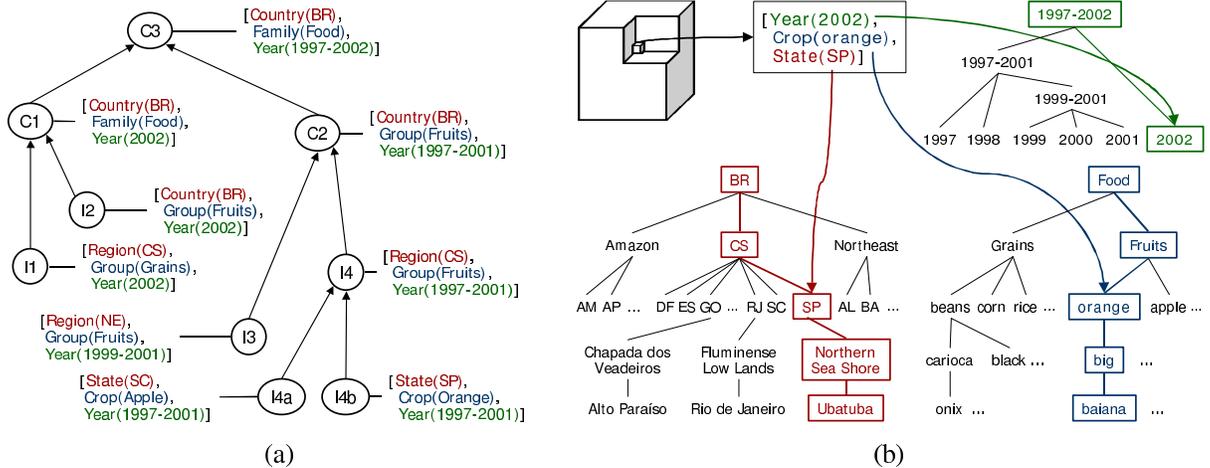


Figure 4.4: The use of POESIA ontologies: (a) scopes of cooperating services; (b) the level of granularity of an aggregated value

cooperating institutions and consortia. The scopes of the data repositories are described by the ontological coverages attached to the nodes. For instance, institution I1 maintains data about the production of grains in the center-south region of Brazil during the year 2002, while I2 is concerned with the production of fruits in the whole of Brazil during the same year. The information flow, indicated by the arrows, shows for example that the data set of consortium C1 consolidates data from I1 and I2, in a scope encompassing those of its sources: the production of food in Brazil during 2002.

The provenance of an aggregated value in a node can be estimated by analyzing the scopes of the data sources of the node. The potential sources, for each dimension, are those whose ontological coverage overlaps (encompasses or is encompassed by) the coverage of the aggregated value in that dimension. For example, consider the average production of orange in São Paulo State during 2002. Figure 4.4(b) shows how the ontological coverage expresses the granularity of the aggregated value, by indicating specific terms in different dimensions of a POESIA ontology. Each term whose semantics overlaps the ontological coverage of the aggregated value is surrounded by a rectangle.

Figure 4.5 illustrates the identification of the potential data sources for different dimensions. It shows the arrangements of concepts for the space and product dimensions, with pointers associating the data sources to the terms used to express their scopes (e.g., C3 is associated with BR because its ontological coverage refers to `Country(BR)`). Then, provenance tracking in one dimension reduces to collecting the sources associated with all the ancestors and descendants of the terms expressing the coverage of the aggregated value in that dimension. Figure 4.5(a) highlights the potential sources in the space dimension. For instance, sources C3, C1, C2 and I2 are candidates because their ontological coverages refer to `Country(BR)` and `Country(BR) ⊇ State(SP)`. I4b is also a potential source because its ontological coverage

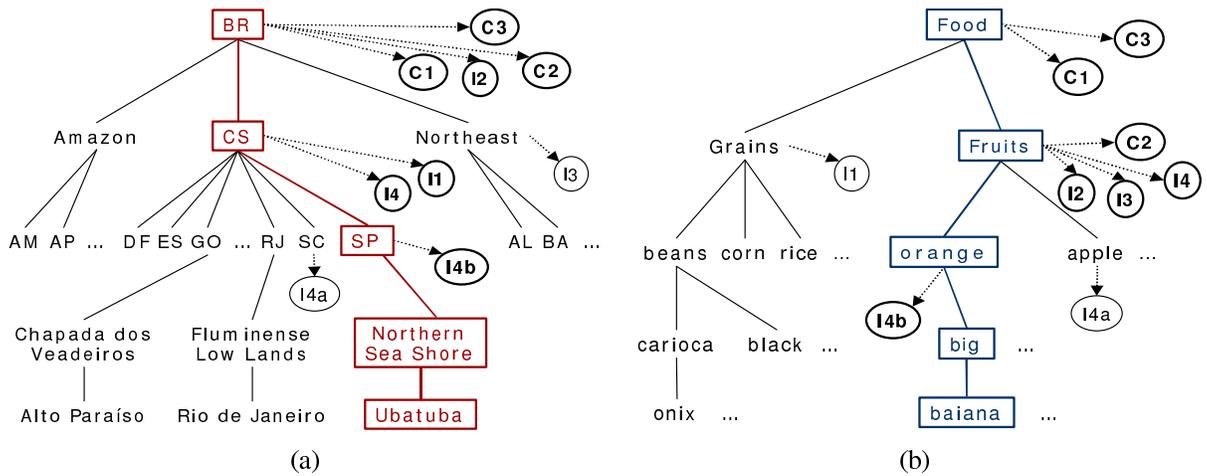


Figure 4.5: Potential data sources in different dimensions: (a) space; (b) product

refers directly to $State(SP)$. If there were other sources associated with the descendants of $State(SP)$ they should also be taken into account. $I3$ and $I4a$ are not potential sources because they refer to nodes outside of the closure of ancestors and descendants of $State(SP)$. Figure 4.5(b) shows the same method applied to the product dimension. A similar analysis can be done for the time dimension.

The potential sources for an aggregated value are those figuring as candidates in all dimensions contributing to its ontological coverage. Figure 4.6(a) illustrates the conclusion of the ontological estimation of the data provenance. The table on the left side shows that only $C1$, $C3$ and $I2$ figure as potential sources in all dimensions. Figure 4.6(b) highlights the relevant flow for the aggregated value considered. The granularity of that value, expressed by $[State(SP), Crop(orange), Year(2002)]$, can be used to select the specific data items which may have been used to calculate the aggregation. This method gives only an estimation of the data provenance because the overlapping of the scopes of the data sources can lead to alternative paths for supplying a particular data value.

4.5 Ontological Nets for Data Integration

An *ontological net for data integration* is an infra-structure for consolidating and fusing data through distributed cooperative processes, where the description, discovery and composition of data sets and services are based on domain ontologies. In order to better explain this concept, let us analyze the basic operators for data integration in cooperative geographical applications and the role of domain ontologies in this context, from a higher level perspective.

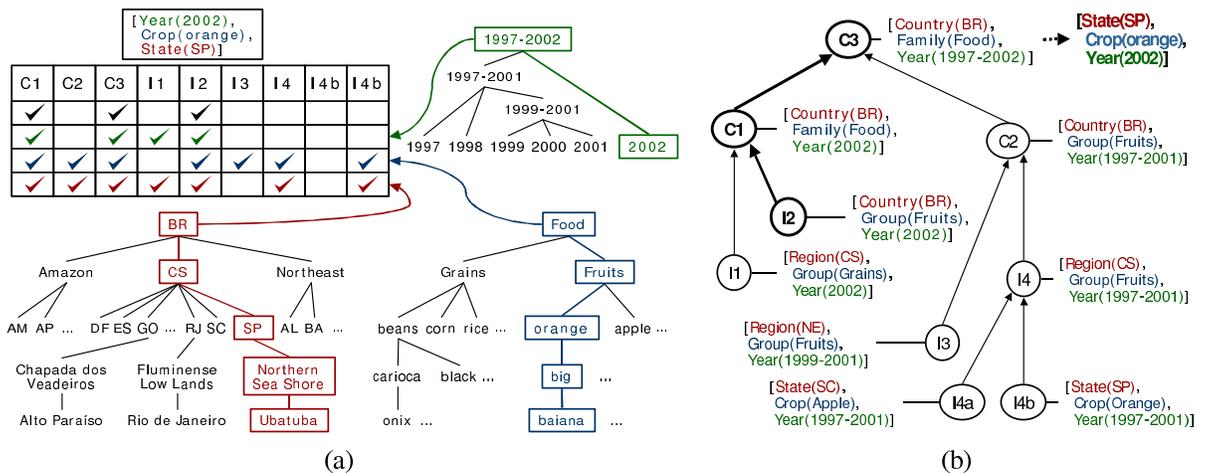


Figure 4.6: Appraising data provenance: (a) contrasting dimensions; (b) estimated data flow

4.5.1 Data Integration Operators

The POESIA approach classifies the operators typically used for integrating data in cooperative geographical applications in three categories: combination of data sets, filtering data and transforming data values. Figure 4.7 presents some examples of the operators for combining data sets. The *union* operator collects data items from two data sources into a composite data set, whose schema matches those of the sources. In figure 4.7(a), data about the production of fruits in Brazil during 2002 is united with another data set about the production of fruits in the Center-South region of the country between 1997 and 2001, generating a data set which covers the production of fruits in Brazil from 1997 to 2002. The *merge* operator relaxes the semantics of the union operator by allowing slightly different semi-structured data sources and user intervention to solve conflicts. Figure 4.7(b) shows an example of merging two heterogeneous data sets, into a semi-structured data set, whose schema is a composition of the source schemas. The *union* and *merge* operators produce data sets whose scope encompasses those of the data sources. The result may contain data with the granularities present in both sources. Additionally, POESIA ontologies help to identify conflicts on merging data sets in the absence of a common key. Data items from different sources, but with equivalent utilization scopes are called *semantically identifiable matches*. These matches are converted into one item in the target, using heuristics and, if necessary, user intervention to solve conflicts. For example, one can detect discrepant values between data items (from different sources) referring to the same product, at the same, place and time, by looking for equivalence of their ontological coverages in all these dimensions. The heuristics to choose the most accurate value among the matches can be, for example, using the value coming from the data source with better reputation or the value that fits better in the typical distribution for that value.

The *intersection* operator employs heuristics to produce data items in the target for

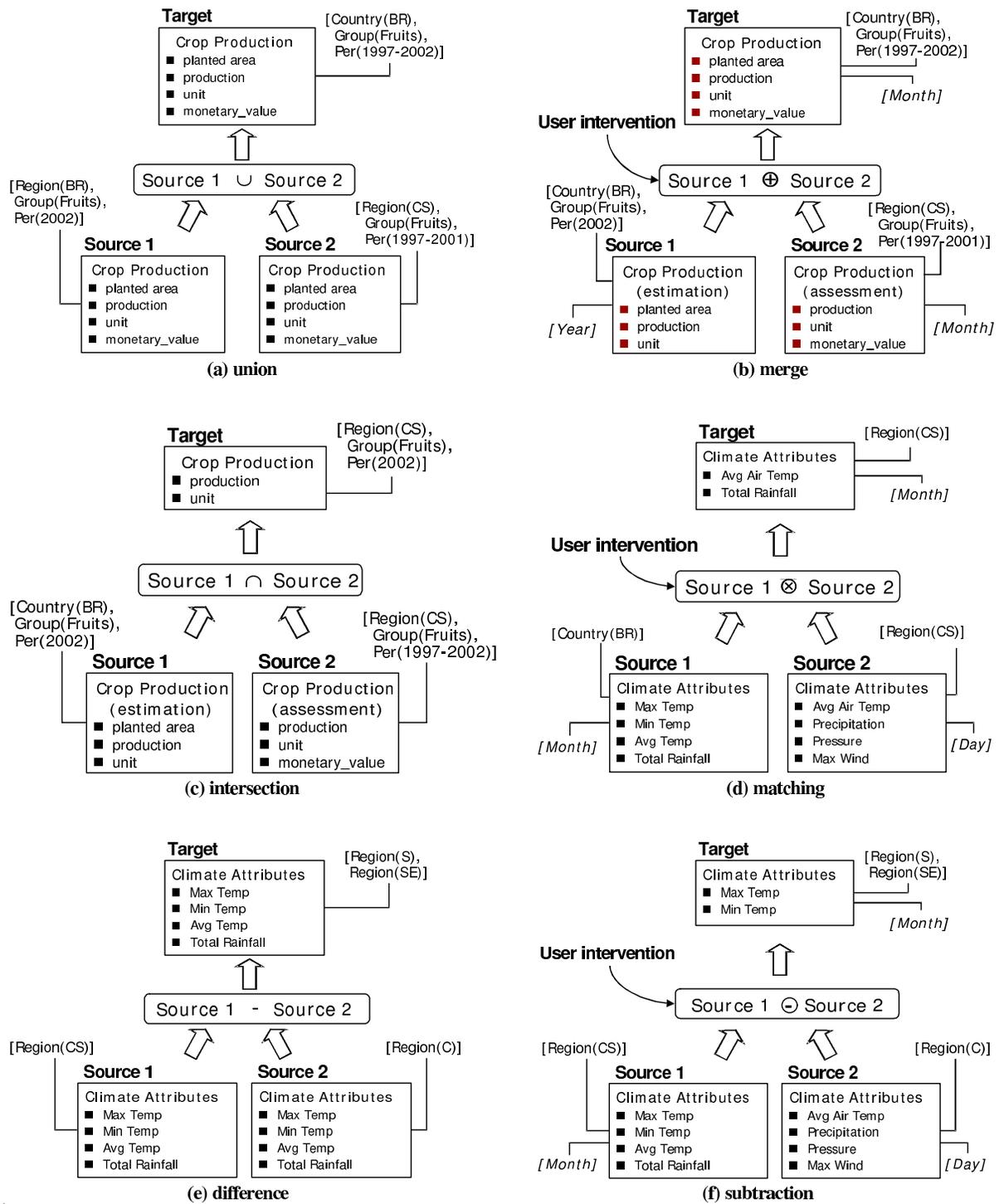


Figure 4.7: Combining data sets

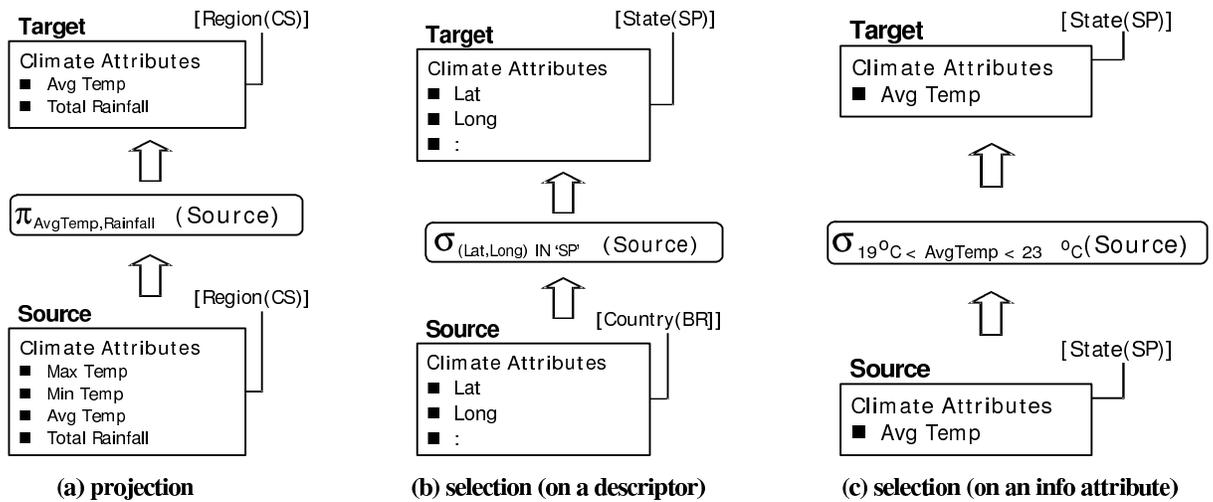


Figure 4.8: Data Filtering

each pair of matching items from the two data sources. The schema of the target can be the union or the intersection of the source schemas, depending on the matching data items. Figure 4.7(c) shows the intersection of two heterogeneous data sets about crop production. The matching operator is similar to the intersection, but allows user intervention to analyze matches and define the target schema. For example, one can identify that `Total rainfall` in `Source 1` matches `Precipitation` in `Source 2`, define the corresponding target attribute and choose the data values to put in the target. Figure 4.7(d) shows the matching of two heterogeneous sources of weather data. For intersection and matching, the scope of the target data set is the intersection of those of the data sources, and the minimum granularity provided by the target is the maximum among the minimum granularities of the sources.

The `difference` and the `subtraction` operators return the data items of the first data source which do not have a match in the second source. The resulting schema derives from the schema of the first data source. The difference between these operators is that `subtraction` allows heterogeneous schemas and user intervention. Figures 4.7(e) and 4.7(f) illustrate the application of these operators to climate data sets. For both operators the scope and the minimum granularity of the target is given by subtracting the scope and minimum granularity of the second source from those of the first one.

Figure 4.8 illustrates the operators for filtering data sets: `projection` and `selection`. These operators keep the semantics of the corresponding relational operators, i.e., projecting attributes or selecting data items according to some predicate, respectively. Projection preserves the scope of the source in the target (figure 4.8(a)), while selection may not. If the selecting predicate stipulates filtering on a term defined in a POESIA ontology, the restricted scope of the target can be determined by that term (figure 4.8(b)). However, it is not straightforward for filtering on values of the data table (figure 4.8(c)).

Figure 4.9 presents the operators that transform data values. The `aggregation` calculates

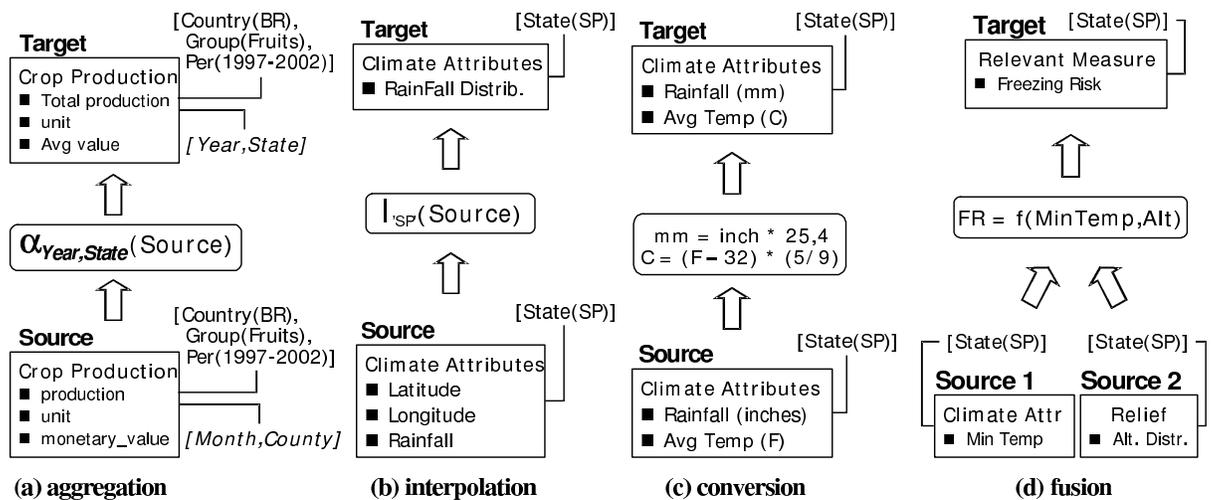


Figure 4.9: Transforming values

coarse grain measurements from data in finer granularities. Figure 4.9(a) illustrates the aggregation of crop production data for each month and county into the respective values for each year and state. The interpolation estimates the continuous distribution of measurements from discrete samples. Figure 4.9(b) illustrates the interpolation of average rainfall samples to produce a map expressing the distribution of this measurement across the lands. The conversion employs user defined functions to convert data (e.g., from one measurement unit into another). Figure 4.9(c) illustrates the conversion of rainfall measurements from inches to millimeters and measurements of average temperature, for the same scope, from Fahrenheit to Celsius degrees. Finally, the fusion operator combines values from different data sources, whose respective scopes match each other, into another meaningful measurement, according to user defined functions. Figure 4.9(d) illustrates the synthesis of the freezing risk from the minimum temperature and altitude. All these operators preserve the scopes of the data sets, though only aggregation and interpolation impact the data granularity.

4.5.2 Data Reconciling through Articulation of Ontologies

POESIA ontologies help the integration process with respect to data scopes and granularities as discussed in section 4.5.1. General and application ontologies help to investigate the semantic correspondences among heterogeneous data items and index libraries of data conversion functions. Some decisions made when integrating data must be annotated, in order to explain the relevant details of data provenance that cannot be captured by ontological coverages alone.

Let us consider the integration of two heterogeneous data sets of weather measurements from distinct institutions, in a particular portion of a cooperative process. The schema for the semi-structured data of each data set can be represented as a directed graph (e.g. XML). The POESIA approach enriches these graphs with metadata describing the data elements, and uses ontologies

to express the properties of these elements and interrelate them. These enriched schemas are themselves specific ontologies. Thus, ontologies articulation [183] can be used as a basis to integrate data sources. Figure 4.10 illustrates this approach. The two graphs at the bottom of the figure describe the data sources, the graph at the top represents the target data set and the dotted and dashed links between nodes of these graphs represent the articulation rules, i.e., the data flows from the sources to the target. These articulations show, for example, that the values of latitude and longitude from the source in the left-bottom corner of the figure, represented in degrees, minutes and seconds must be converted into degrees and decimals of degrees to be inserted into the target.

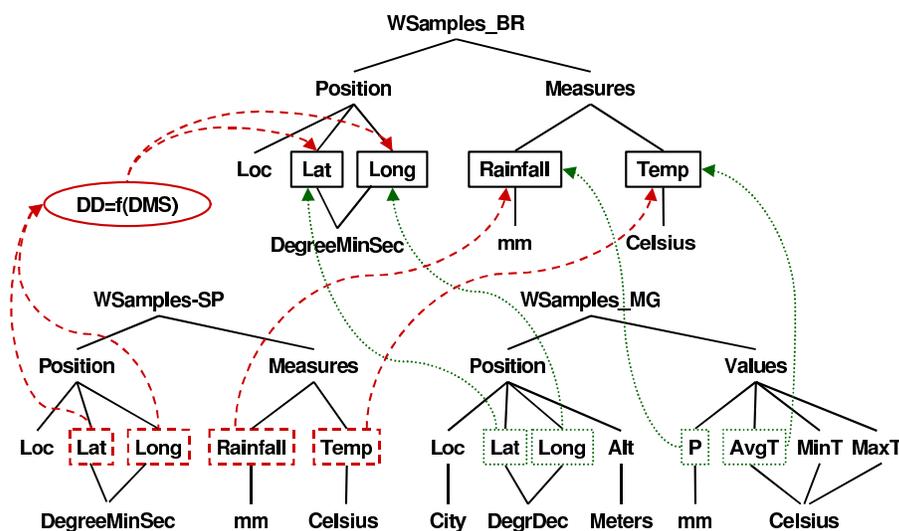


Figure 4.10: Reconciling heterogeneous data sets by ontologies articulation

4.5.3 Semantic Workflows

Semantic workflows are cooperative process running on ontological nets. These processes employ data integration operators, according to ontologies articulations. POESIA ontologies contribute to render a general view of what is going on in these workflows, by expressing the scopes and granularities of the data involved. Figure 4.11(a) illustrates the integration of weather data from different institutions. Each service is characterized by its scope and the minimum granularity it supports for data recovery. For example, the INMET (National Institute of Meteorology) collected weather data samples across Brazil in the period between 1931 and 2002. The minimum time granularity for the data supplied by INMET is month. The ultimate recipient of data in this cooperative process is the RNA Warehouse (National Agrometeorology Network), which can provide weather and climate data about virtually any place in Brazil. The temporal scope of the weather data supplied by the RNA Warehouse is 1892 to 2002 and the minimum

granularity supported is day. The granularity for each data item depends on the sources of that item. Figure 4.11(b) illustrates the role of the RNA Warehouse on supplying climate data to determine land suitability for different crops. The scope of the sub-processes for determining land suitability for coffee and rice must be compatible with the coverages of the respective subsets of climate attributes recovered from the RNA Warehouse (see (Chapter 3) for details).

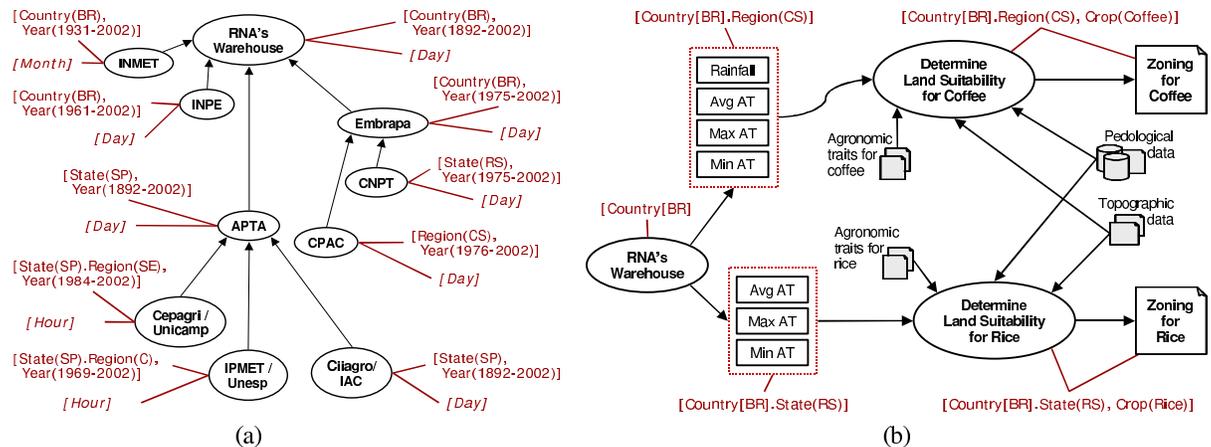


Figure 4.11: Ontologies as a framework for estimating data provenance: (a) scopes and minimum granularities of cooperating services; (b) the use of the integrated data by different processes

4.6 Related Work

The traditional solutions for tracking data provenance, some of which consider general data formats and processing, employ metadata to annotate the processing history [146, 31, 149, 23]. However, these solutions do not scale well to large data sets, long processing flows and fine grained provenance. Many other studies on data provenance are limited to views defined by query operations on databases, calling this restricted problem *lineage tracing*. Woodruff *et al.* [251] introduce the concept of *inverse query*, which maps an output to the data items used to produce that output. They define the class of functions admitting inversion and the concept of *weak inversion* to estimate the lineage for a wider class of functions. However, they do not show how to determine the inverse queries, but expect the data transformation definer to provide them.

Cui *et al.* [55, 57] define the *lineage* of the result of a relational database query as the minimal set of tuples necessary to produce that result. They present an algorithm for tracing lineage over chains of aggregate-select-project-join views. Their approach is based on the inversion of the view definition and requires materializations of original relations and intermediate views. [56] generalizes their previous results for graphs of general transformations used

for loading data warehouses. Nevertheless, their methods are built upon some constraints and specific information about the sources and transformations employed, and require considerable storage for intermediate results.

Buneman [33] distinguishes between *why provenance* and *where provenance*. The former refers to the data items which have some influence on the result (e.g., which determine the logical value of a predicate used to select tuples). The latter refers to the items effectively used to synthesize the result (e.g., multiple values summed up to obtain an aggregated value like average). He provides a framework to track both kinds of data provenance for specific classes of `select-project-join-union` queries in a data model generalizing relational and hierarchical data representations such as XML. Galhardas *et al.* [95] present some data lineage facilities coupled to a data cleansing scheme based in a graph of transformations with exceptions management to support the refinement of the cleaning criteria. Fan [77] provides algorithms to trace data lineage in automatically reversible sequences of schema conversions, employing the hyper-graph based high level data model and the functional query language of the Automated system.

Therefore, current approaches either support just coarse grain provenance tracking or rely on detailed descriptions of the data sources and the data transformations applied (e.g., schemas and query expressions), making them unfit in many situations for cooperative systems over the Web. Furthermore, these approaches lack abstraction mechanisms to enable a general understanding and exploration of the information flow. To the best of our knowledge, domain ontologies [233, 110, 182] has not been yet exploited as a framework for tracking data provenance. This paper has shown that such a solution can eliminate some of these shortcomings.

4.7 Conclusions

Data provenance tracking is becoming increasingly important as more on-line data sources become available. This paper has shown how domain ontologies are used in POESIA as a basis for tracking data provenance in cooperative processes involving data integration. POESIA employs tuples of domain specific terms defined in multidimensional ontologies to correlate the scope and granularities of the target data with those of the data sources, enabling the estimation of the data provenance. Additionally, POESIA ontologies help to semantically identify matches on heterogeneous data sources, i.e., data items from different sources referring to the same scope. It helps to detect and solve conflicts among heterogeneous data sources, and allow tracking the data transformation flow across chains of data integration operators.

The benefits of this ontological method for estimating data provenance are (1) a framework for understanding data provenance based on domain specific concepts; (2) support for fine grain provenance tracking; (3) precision and conciseness for expressing the scopes and granularities; (4) coupling with a general approach for data integration and services composition; (5) the

cost for maintaining the infra-structure for provenance tracking is shared with facilities for cataloging, discovering and integrating data and services.

This research is focused on the conceptual definition and formalization of the ontological approach for multi-step data integration and provenance tracking. Ongoing work includes the implementation of prototypes to validate the POESIA approach for scientific applications in agriculture, and conjugating the ontological scheme with other methods for provenance tracking.

Acknowledgments

The authors from Campinas University are partially supported by Embrapa, CAPES and the Finep/Pronex/IC/SAI95/97 project. The authors from Georgia Tech are partially supported by two grants from the Operating Systems and ITR programs (CISE/CCR division) of NSF, by a contract from the SciDAC program of DoE, and a contract from the PCES program (IXO) of DARPA. The application scenarios and data used in this work were provided by Brazilian experts in agriculture. Special thanks to Daniel Andrade and Flavio Silva from the IC-Unicamp DB-group for some valuable discussions on preliminary versions of this material.

Chapter 5

Applying Semantic Web Technology in Agricultural Sciences

5.1 Introduction

The Semantic Web [22, 215, 80] foresees a new generation of Web based systems, taking advantage of semantic descriptions of data and services to enhance the role of computers on supporting several human activities. Such machine processable descriptions, conforming to metadata standards, are expected to boost interoperability and enable automatic reasoning in cooperative processes inside and across organizational boundaries. Nevertheless, there are many open questions relative to the applicability, adequacy and maturity of the Semantic Web technology for real world applications.

In the Internet era, scientific communities have been creating and accessing a myriad of data sets and computational services, in a diversity of fields such as earth sciences, bio-informatics and medicine. Several applications require the integration of these heterogeneous data sources and the composition of these services. Consequently, there is a growing demand for accurate and efficient means to search, recover and interconnect these resources. The development, adaptation and use of Semantic Web technologies for scientific purposes is a promising route to fulfill these needs.

Much research effort has been directed to Semantic Web issues [80, 124, 63], including those involving scientific applications [224, 160, 174, 102, 43]. However, very few domain-specific studies have been reported to describe the engineering challenges, the domain-specific usages, and the impact of ontology structure and ontology size on system design and performance.

POESIA (Processes for Open-Ended Systems for Information Analysis) (Chapter 3) pursues the vision of the Semantic Web to bring about solutions for resources discovery and composition, interoperability of information systems and traceability of processes. Inspired by the

needs of scientific applications such as agricultural planning, POESIA combines domain ontologies, workflows and activity models to provide novel facilities for multi-step integration and processing of semi-structured data in an open and distributed environment. The foundations of POESIA are (1) Web Services to encapsulate data sets and processes; and (2) domain ontologies to organize, recover and drive the composition of these services, according to their utilization scopes (i.e., the situations in which they can be used). POESIA's mechanisms for organizing and composing Web services using domain ontologies, including rules to assure the semantic consistency of the resulting processes, appear in (Chapter 3). The use of these domain ontologies to track data provenance and support data integration in POESIA is described in (Chapter 4).

This paper focuses on the engineering challenges of developing and using domain ontologies in POESIA. Though the case study refers to a particular scientific application – agricultural zoning – the approach is extensible to other domains, and useful in a wide class of applications, that require data integration and cooperative work on the Web. In particular, the paper points out the obstacles met in loading and utilizing domain ontologies in application programs, and describes the solutions adopted, which were implemented in a prototype. These solutions involve the extraction of *ontology views* – i.e., application relevant parts of an ontology. Rather than forcing applications to deal with large, cumbersome ontologies, the notion of ontology views is adopted to discover and compose Web resources, and managing the resulting cooperative processes. The experiments reported in this paper give an insight on the limitations of the current Semantic Web technology to deal with ontologies, when faced with real world applications using large data sets. These experiments show that the combination of Semantic Web standards and tools with conventional data management techniques provides better scalability than the solutions based only on the Semantic Web.

The remainder of this paper is organized as follows. Section 5.2 describes the needs of scientific applications over the Web, and particularly of agricultural zoning processes. Section 5.3 describes how the POESIA approach addresses these needs. Section 5.4 presents the design and implementation of the ontology for the agriculture realm. Section 5.5 outlines the use of this domain ontology to support services discovery and other facilities in POESIA. Section 5.6 reports some implementation experiences involving the construction of ontology views and the use of these views to support Semantic Web applications. Finally, Section 5.7 discusses related work and Section 5.8 concludes the paper.

5.2 Motivation: Agricultural Zoning

This research has been motivated by the needs for versatile tools to support scientific applications on the Web, and more specifically the development of decision support systems for agriculture. One example of an application in this domain is *agricultural zoning* – a scientific

process that classifies the land in a given geographic region into parcels, according to their suitability for a particular crop, and the best time of the year for key cultivation tasks (such as planting, harvesting, pruning, etc). The goal of agricultural zoning is to determine the best choices for a productive and sustainable use of the land, while minimizing the risks of failure. It requires looking at many factors such as regional topography, soil properties, climate, crop requirements, social and environmental issues.

Typically, this kind of application involves intricate data processing activities across different organizations. Agricultural zoning relies on data from a variety of heterogeneous sources, including sensors that collect data on physical and biological phenomena (e.g., weather stations, satellites, and laboratory automation equipment). These data may be stored in legacy databases or files in several formats.

An agricultural zoning process is built by cooperation of experts from many scientific and engineering disciplines. Agronomists contribute with planting techniques and crop management models. Biologists provide crop growth and nutrient requirements. Statisticians provide risk management analysis for potential crop failures (e.g., due to severe weather). These people, working in inter-institutional teams for particular enterprises, bring together their expertise in several fields to produce cooperative processes using a variety of computational platforms and data analysis tools.

Figure 5.1 presents an example of output of an agricultural zoning process. It shows the suitability map for planting short cycle varieties of soybeans, considering a specific class of soils, in the Brazilian state of Goiás. The map in Figure 5.1(a) classifies the lands of the state according to their suitability for sowing soybeans in the beginning of October, and the map in Figure 5.1(b) for sowing in the beginning of November. These maps result from inter-institutional cooperative work as described previously. In order to produce them, experts had to combine data on the climate, soils and topography of that state, and the environmental needs of the soybean plants along their development cycle.

Experiences in some sectors of the Brazilian agriculture in the last few years corroborate the economic advantages of adopting a scientific approach to agricultural zoning [58]. However, the current agricultural zoning processes are labor-intensive, and consequently expensive and slow to develop and run. This is a serious problem, since it is an extremely important issue for a country with a vast territory and many commercial crops such as Brazil.

The problems of such a data processing apparatus applied to cooperative scientific applications like agricultural zoning become more apparent from the perspective of the Semantic Web:

1. There is a growing demand to publish, browse and interconnect data sets and processes on the Web.
2. Web-based systems lack semantic support for discovering, selecting and interconnecting

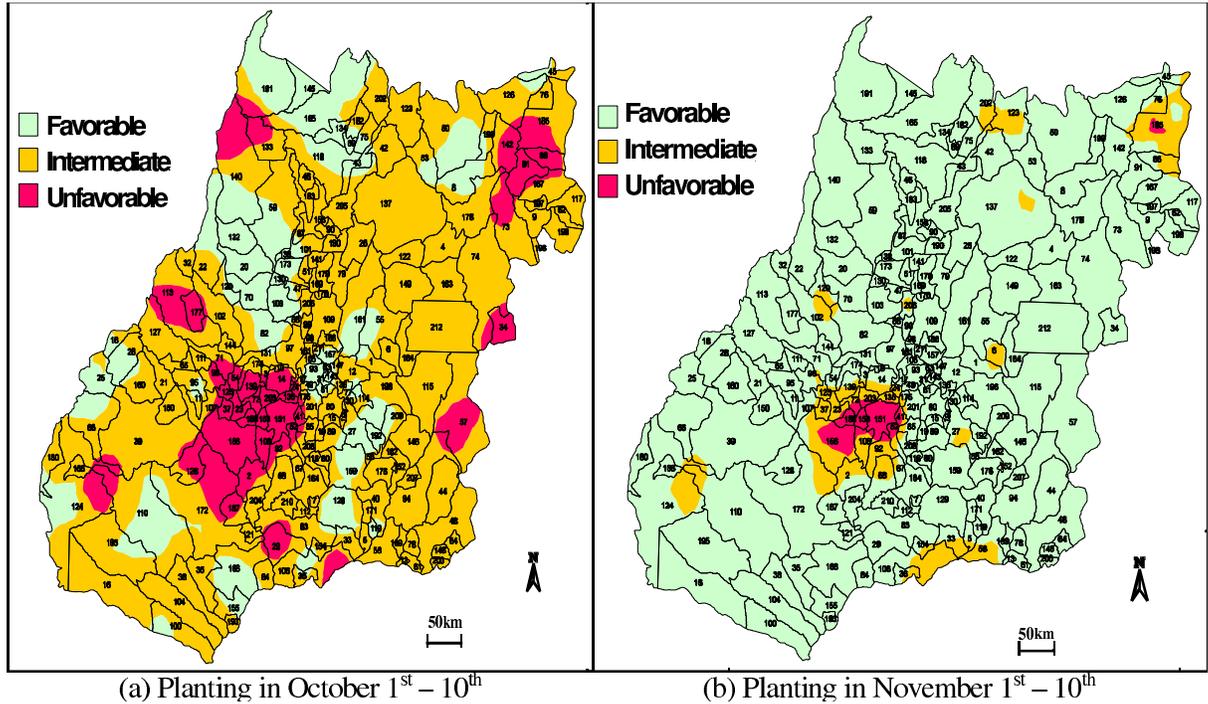


Figure 5.1: Suitability maps for planting soybeans in Goiás (Sources: Embrapa/CNPSo, DNAEE and INMET)

the available resources. In order to facilitate these tasks, the resources should be described according to domain specific knowledge. Such semantic descriptions could also contribute to data cleansing, integration and aggregation, which occur in multiple steps across distributed cooperative processes.

3. The processes through which data pass are rarely documented. Even when documented, the specifications produced are either not generic enough to give a general view of the processes or not formal enough to allow the automatic repetition of these processes with different data.
4. There should be some means to track data provenance across these processes, i.e., determine the original data sources and the way data were obtained and processed.

The following section outlines the POESIA approach for coping with these problems, which is based on combining ontologies with workflows. We point out that these issues are not particular to agricultural zoning. Indeed, they are common to a wide range of domains, as mentioned in the introduction of this paper. Our solution can be generalized to other domains, provided that the appropriate ontologies are used.

5.3 Solution Context

5.3.1 The POESIA Approach

The foundation of the POESIA approach (Chapter 3) is the use of a domain ontology for multiple purposes in inter-enterprise processes that gather, integrate, transform and analyze data. Figure 5.2 illustrates the central role of the domain ontology in such a cooperative process. A domain ontology depicts the semantic relationships among terms of a knowledge domain. In POESIA, terms are grouped according to different dimensions of one reality; in the agriculture domain, geographic space and crops are examples of dimensions. Tuples of terms, called *ontological coverages*, express the utilization scopes of Web Services that encapsulate data sets and data processing activities (e.g., the spatial extent and the crops for which a particular service is intended). Ontological coverages serve as concise descriptors of resources based on domain specific knowledge. The semantic relationships among the terms of the ontology, particularly relationships of the type IS_A and PART_OF, induce a partial order among ontological coverages, thereby ensuring the possibility of:

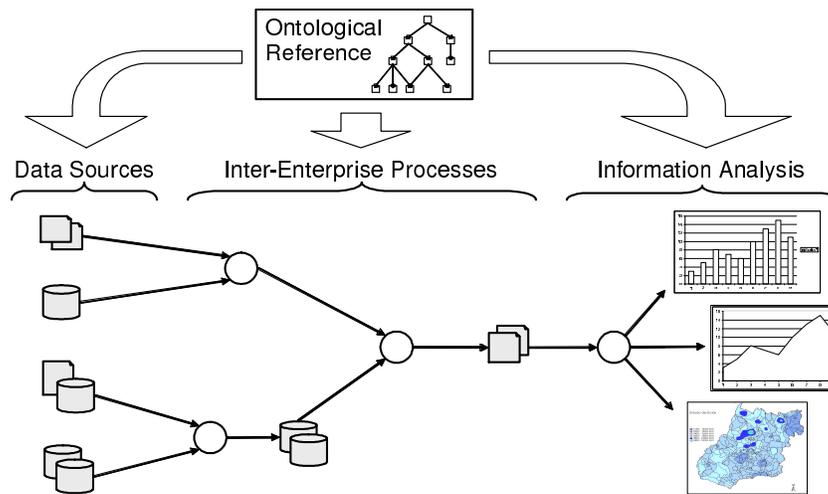


Figure 5.2: The multiple roles of a domain ontology in the POESIA approach

- automation of means to support the discovery and composition of Web Services (Chapter 3);
- estimation of data provenance across distributed cooperative processes (Chapter 4);
- detection of correspondences among heterogeneous data items, for data integration purposes, based on semantic relationships between their ontological coverages (Chapter 4);

5.3.2 POESIA Ontologies as Web Services

POESIA ontologies can be published and looked up through Web Services. An ontology server encapsulates ontologies for different domains (e.g., agriculture, biology, biotechnology), and provides access and adaptation means to allow several applications to use these ontologies. The sharing of ontologies among application programs enable enactment of cooperative workflows that use resources distributed across the Web.

Figure 5.3 illustrates how ontologies may be encapsulated within an ontology server, and how this server can be used to manage data and services in cooperative processes for different application areas. The Supply Chain Ontology is a subset of the Logistics Ontology. These ontologies refer to the production and distribution of goods to satisfy any kind of need (e.g., food, energy, water). The Agriculture Ontology, in turn, has some intersection with the specialization of the former ontologies to the agriculture realm. Each of these three ontologies is referred to by several workflows, for the respective application domains. A given workflow, on the other hand, can only be associated with a given ontology, which will allow it to adequately manage the resources necessary for its execution. The interoperability of ontologies and workflows designed for different domains is beyond the scope of this paper, and left to future work.

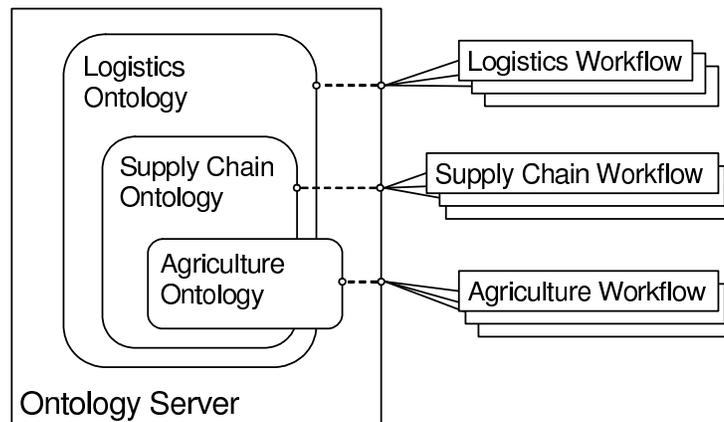


Figure 5.3: Using domain ontologies to handle workflows in POESIA

The rest of this paper describes the design, development and use of an ontology for the agriculture realm, providing a concrete example of the basic facilities to build POESIA applications. It provides an insight of some implementation issues, with respect to the Semantic Web standards and tools available nowadays.

5.4 An Ontology for the Agriculture Realm

5.4.1 The Ontology Design

As part of the effort to implement and validate the POESIA approach in real life applications, we have been developing an ontology to support agricultural zoning. This ontology is divided in several *facets*, congregating, interrelating and providing unified access to a variety of themes relevant to the agriculture realm. Figure 5.4 illustrates the overall structure of this ontology, rooted at `thing`. The three topmost facets are Measurement Units, Agricultural Topic and Geo-Entity. Data instances appear at the bottom level.

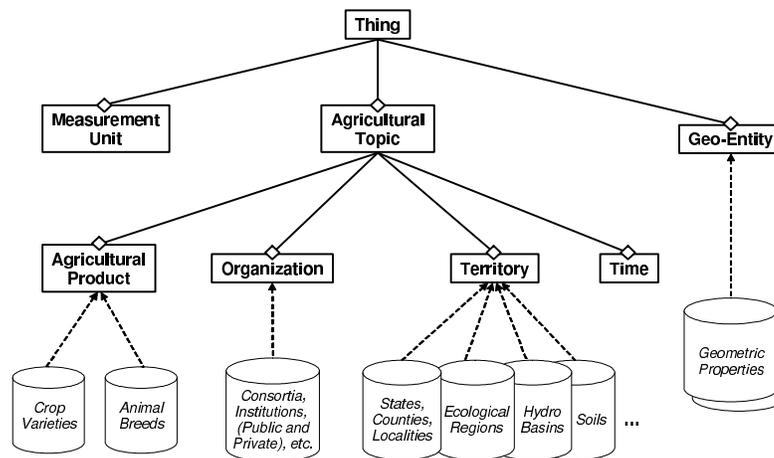


Figure 5.4: General conception of the ontology for the agriculture realm

The `Measurement Unit` facet describes the physical, chemical, biological and other kinds of units appearing in agricultural data. It can be adapted from an existing ontology of measurement units. One particular issue in this facet is the modeling of the relationships between compatible units, to facilitate data integration and conversion among these units.

The `Agricultural Topic` facet is divided in *dimensions* for particular agricultural concerns. These dimensions are used to specify ontological coverages describing the utilization scopes of data sets and processes in the agricultural domain. Let us consider these dimensions in more detail. Figure 5.5 depicts the `Agricultural Product` dimension. The rectangles in this diagram represent *classes of objects*. The edges ending with a diamond represent *specialization relationships* (of type `IS_A`) between classes, i.e., the class at the target of such an edge (indicated by the diamond) is a subclass of the class in the source of that edge. The diagram shows, for example, that an `Agricultural Product` can be `Raw` or `Processed`. A `Raw Product` can be a `Plant` or an `Animal`, both of which have several subclasses. This hierarchy is in fact a directed graph, because of multiple inheritance. The levels are not uniform for each kind of plant or animal. The bottom part of Figure 5.5 details the hierarchy for

commercial types of Coffee (Arabica and Robusta) and categories of Cattle (Dairy or Meat cattle, i.e., for primarily producing milk or meat, respectively).

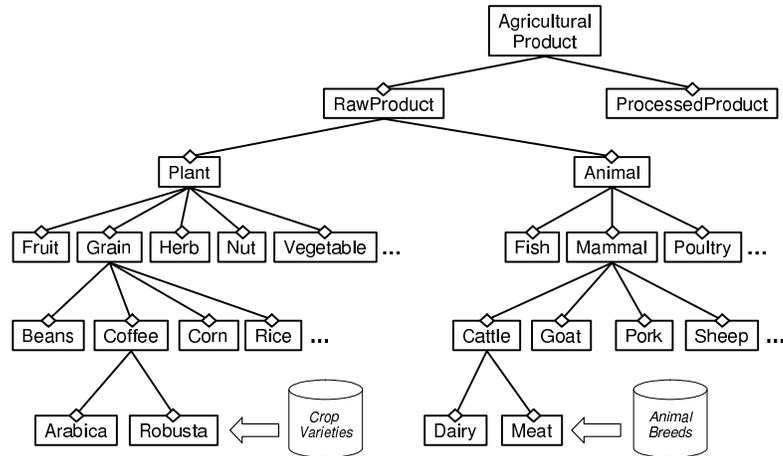


Figure 5.5: The Agricultural Product dimension

Figure 5.6 depicts the Organizations dimension of the ontology for the agriculture realm. Figure 5.6(a) shows that an Organization can be a Consortium, an Institution (e.g., company, association, governmental body) or a specific Unit of a Consortium or Institution. A Consortium is composed of a number of participating Institutions and an Institution is composed of its Units. These *aggregation relationships* (of type PART_OF) are represented by edges with a black circle on the side of the class playing the role of component. Figure 5.6(b) presents a hierarchy of instances of the classes presented in Figure 5.6(a). This hierarchy shows, for example, that the Consortium called RNA (*Rede Nacional de Agrometeorologia – Brazilian Agro-meteorological Network*) has Embrapa (*Empresa Brasileira de Pesquisa Agropecuária – Brazilian Agricultural Research Corporation*) and Unicamp (University of Campinas) as its participants. CPAC, CNPTIA and CEPAGRI are the acronyms of specific research centers within these institutions.

The Territory and Time dimensions are also represented with the basic constructs previously described. The Territory dimension includes several layers of geographic data, such as political division (country, regions, states, etc.), ecological regions, hydrological basins and types of soil. The Geo-Entity facet, based on the GML standard [192], describes how to represent geographic features.

5.4.2 The Ontology on Protégé

The ontology for the agriculture realm has been developed with Protégé [190], an open-source graphic tool for ontology editing and knowledge acquisition. Figure 5.7 presents a snapshot

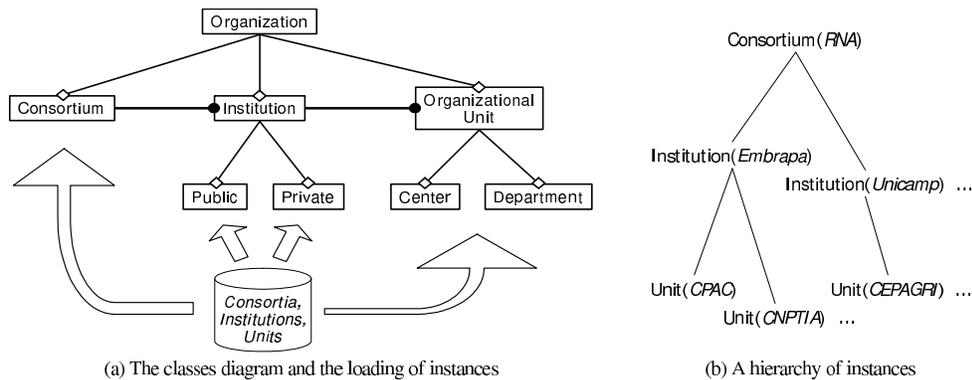


Figure 5.6: The Organization dimension

of this ontology on Protégé, showing its overall structure (on the left) and some details of the interface for the Territory dimension, with the states and different kinds of regional subdivisions in Brazil. Some details of the São Paulo State appear in a pop-up window centered in the bottom.

Protégé can be extended with plugins, enabling the incorporation of new functionalities and the development of ontology specifications in a variety of formats. POESIA's present implementation accepts ontologies in the RDF format [211]. The adoption of DAML+OIL [165] and OWL [196] is also being considered.

5.5 Exploiting Ontological Relationships

5.5.1 Ontological Coverages to Express and Interrelate Scopes

A POESIA ontology can be defined as a directed graph whose nodes represent concepts (e.g., Country) or instances of concepts (e.g., Country(Brazil)) and whose directed edges represent semantic relationships between nodes (instantiation, specialization or aggregation). Edges go from the general to the instantiated, specialized or constituent concepts or instances. These relationships induce a partial order among the terms denoting ontology concepts and their instances (Chapter 3). This order is determined by the relative positions of the terms in the ontology graph. Let t and t' be two terms of an ontology Σ . We say that t encompasses t' , denoted by $t \models t'$, if and only if there is a path in Σ leading from t to t' , i.e., a sequence of instantiation, specialization and aggregation relationships relating t to t' . The encompass relationship is transitive – if Σ has a path from t to t' and another path from t' to t'' then Σ has a path from t to t'' . In the ontology presented in the previous section, $\text{Plant} \models \text{Grain}$, $\text{Consortium}(\text{RNA}) \models \text{Institution}(\text{Embrapa})$ and $\text{Plant} \models \text{Coffee.Arabica.Variety}(\text{Tupi})$. The string Cof-

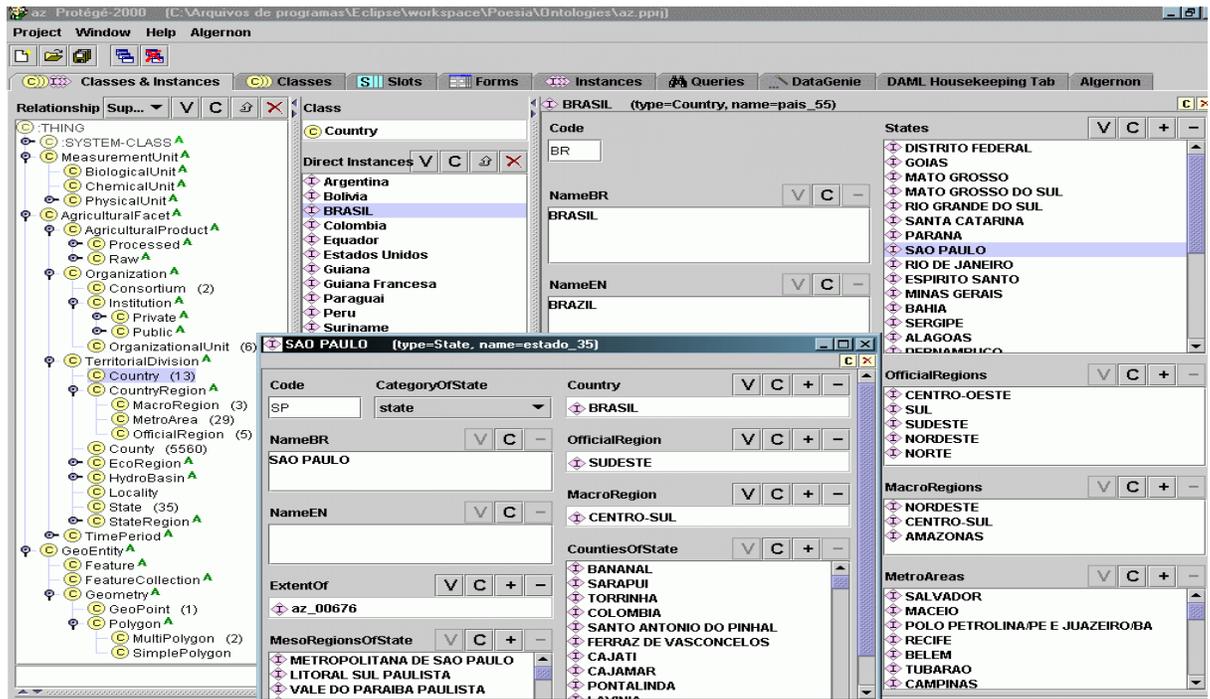


Figure 5.7: The ontology for the agriculture realm on Protégé

fee.Arabica.Variety(Tupi) represents the path to reach the term Variety(Tupi). The path to a term can be omitted if there is no possibility of ambiguity – e.g., there is only one Country called Brazil, but several kinds of crops, such as Soybeans, have a variety called Tupi.

Consider a POESIA ontology with a number of facets describing different aspects of one reality (such as Measurement Units and Agricultural Topics). A facet is a sub-graph of the ontology graph whose nodes have no connection by instantiation, specialization or aggregation with nodes of other facets. The dimensions of a facet are the sub-graphs whose roots are children of the facet's root. An *ontological coverage* is a tuple of terms taken from the dimensions of some facet of a POESIA ontology. For instance, the ontological coverage [Orange, Country(Brazil)] of the Agricultural Topic facet is a tuple with terms from two dimensions – Agricultural Product and Territory. When an ontological coverage is attached to a Web Service it plays the role of metadata, describing the *utilization scope* of the service. The ontological coverage [Orange, Country(Brazil)] when attached to a Web Service of agricultural production data, indicates that data from that service refer to the production of Oranges in Brazil. Figure 5.8 illustrates the specification of an ontological coverage in which the term Institution(Embrapa) expresses the utilization scope in the Organization dimension, Orange in the Agricultural Product

dimension and $\text{Country}(\text{Brazil}) . \text{Region}(\text{SE})$ in the Territory dimension.

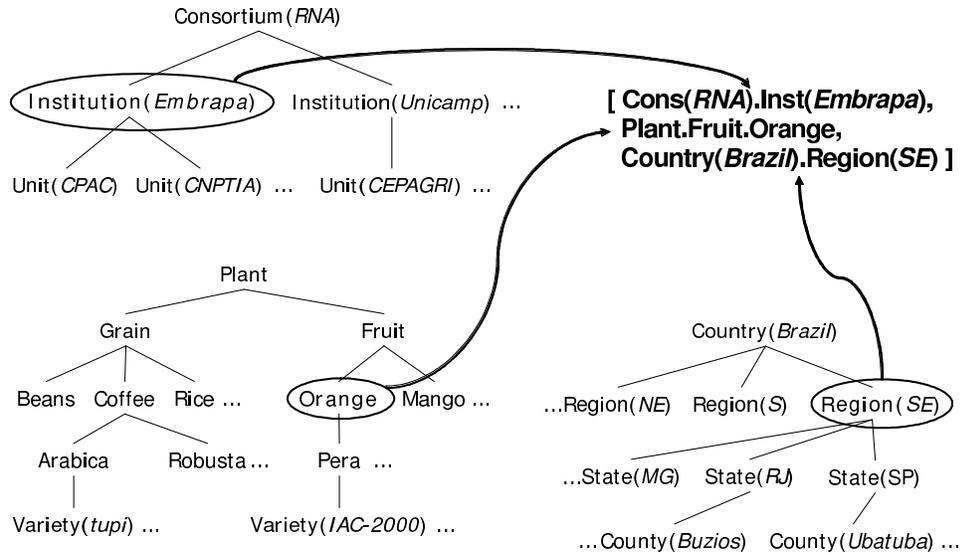


Figure 5.8: An ontological coverage in the ontology for the agriculture realm

Semantic Relationships between Ontological Coverages

The *encompass* relationship between terms gives rise to corresponding relationships between ontological coverages. For simplicity, let us consider that an ontological coverage has exactly one term for each dimension of a facet. Given two ontological coverages, $OC = [t_1, \dots, t_n]$ and $OC' = [t'_1, \dots, t'_n]$ ($n \geq 1$), where $t_i \in OC$ and $t'_j \in OC'$ are terms from the same ontology and facet, OC and OC' may be disjoint or satisfy one of the following relationships.

Overlapping: OC overlaps OC' if and only if the following conditions are satisfied:

1. $\forall t \in OC : \exists t' \in OC'$ such that $t \models t' \vee t' \models t$
2. $\forall t' \in OC' : \exists t \in OC$ such that $t \models t' \vee t' \models t$

Encompassing: $OC \models OC'$ if and only if the following conditions are satisfied:

1. $\forall t \in OC : \exists t' \in OC'$ such that $t \models t'$
2. $\forall t' \in OC' : \exists t \in OC$ such that $t \models t'$

Equivalence: $OC \equiv OC'$ if and only if the following conditions are satisfied:

1. $\forall t \in OC : \exists t' \in OC'$ such that $t \models t'$
2. $\forall t' \in OC' : \exists t \in OC$ such that $t' \models t$

Overlap is bidirectional and the weakest of these relationships. The *encompass* relationship between ontological coverages, on the other hand, only accepts encompassing relationships between terms in one direction. The *equivalence* relationship requires that each pair of terms taken from the two ontological coverages reciprocally encompass each other. Finally, two ontological coverages are *disjoint* if they do not overlap each other in at least one dimension, i.e., there is a term in one of the coverages that does not encompass or is encompassed by any term of the other ontological coverage.

The *encompass*, *overlap* and *equivalence* relationships between ontological coverages are *reflexive* and *transitive*, and the two latter are also *symmetric*. The transitivity of these relationships induces a partial order among ontological coverages referring to the same ontology and same facet. Figure 5.9 illustrates this ordering. In the figure, ontological coverages are used to describe services for accessing agricultural production data. The coverages in the figure are defined with respect to the `Agricultural Topic` dimension of the ontology. The `Organization` dimension was eliminated for simplification purposes. The ontological coverage `[Plant, Country(Brazil)]` encompasses the coverage `[Plant.Grain, Country(Brazil)]`. Only the former encompasses `[Plant.Fruit.Orange, Country(Brazil).State(RJ)]`. The coverage `[Plant.Grain, Country(Brazil)]` does not overlap `[Plant.Fruit, Country(Brazil).State(SP)]`, because these coverages refer to different kinds of crops. The coverages `[Plant.Grain, Country(Brazil)]` and `[Plant, Country(Brazil).Region(NE)]` overlap, though neither encompasses the other.

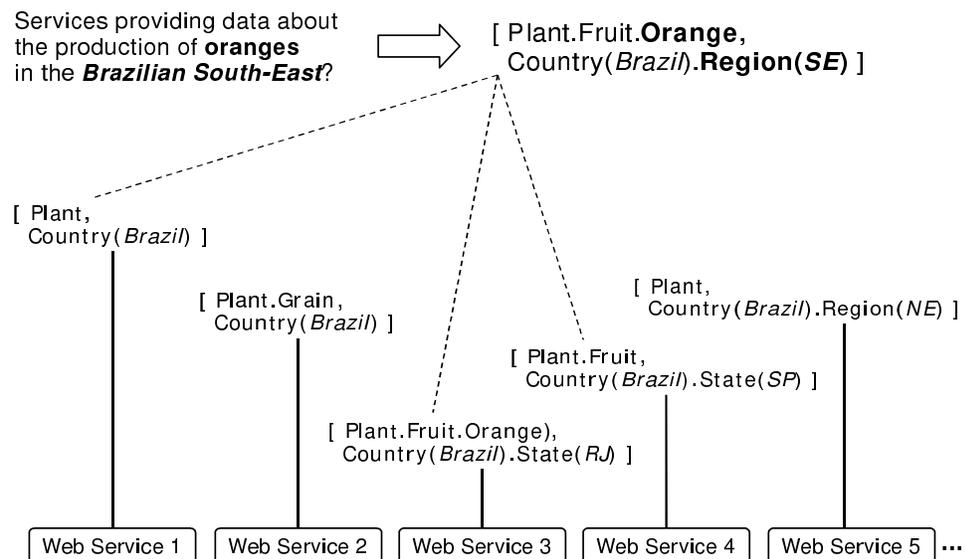


Figure 5.9: Using the relationships among ontological coverages for services discovering

Ontological coverages help to describe scope and goals of a service. Suppose one wants to find the services providing data about the production of oranges in the Brazilian South-East region. Such services are those whose ontological coverage overlaps [`Plant.Fruit.Orange`, `Country(Brazil).Region(SE)`], where SE is the acronym of the South-East region. The dashed lines linking this ontological coverage to those of the Web Services numbered 1, 3 and 4 indicates that those are the services that satisfy the search criteria. For other details about the specification, comparison and use of ontological coverages see Chapter 3.

5.5.2 Representing Ontological Relationships

Given the semantic relationships defined in Section 5.5.1, we now turn to analyzing how they can be expressed using Semantic Web formalisms. Here, we use DAML to represent semantic relationships between terms of the ontology for the agricultural domain. Even though POESIA presently uses RDF, this paper uses DAML just to avoid cumbersome RDF statements that could hinder understanding.



Figure 5.10: The ontology for the agriculture realm in DAML

Figure 5.10(a) shows an extract of the `Agricultural Product` dimension. It corresponds to a hierarchy of classes where each subclass is linked to its parent class by the `IS_A` relationship. Figure 5.10(b) shows the `Organization` dimension, with the hierarchy of classes appearing on the left side. The right side presents the properties used to represent the `PART_OF` relationships between instances of these classes. For example, property `InstitutionsOfConsortium` is used to indicate the institutions that participate in a particular consortium. Figure 5.10(c) presents similar constructs for the `Territory` dimension. Finally, Figure 5.10(d) defines the *encompass* relationship in DAML – a transitive property that has both `IS_A` and `PART_OF` as sub-properties. The `IS_A` property is also a sub-property of the predefined property *subClassOf* of DAML.

5.5.3 Defining Ontology Views

In real life, domain ontologies can become very large, and applications will seldom need to use an entire ontology. Thus, we propose the notion of *view*, which is a subset of an ontology that is needed by an application. Different POESIA applications can require distinct views of the same ontology, characterized by distinct subsets of the ontology concepts and semantic relationships, and respective instances. Such views can facilitate knowledge visualization and manipulation in application programs. Ontology views can be specified with a template based method. Classes and semantic relationships to be included in the view are marked with tags. The possible tags are:

DIM_CLASS is associated with an ontology class referring to a dimension of the ontology, to denote that the dimension must be taken into account in the view (e.g., the dimensions `Agricultural Product`, `Organization` and `Territory` of the agricultural topic facet).

ROOT_CLASS marks the root classes of a dimension (e.g., `Country` and `Ecological Region` as roots for the `Territory` dimension).

SHOW_CLASS indicates an intermediate class to be shown in the view.

SHOW_RELATIONSHIP labels a relationship between instances to be considered in the view.

We developed an algorithm to generate an ontology view, following the hierarchy of classes and the semantic relationships among their instances, and using these tags to decide on the classes, instances and relationships to put in the view. Figure 5.11 presents an ontology view obtained by this method. This view is displayed in a user interface we developed with the Tree-bolic implementation of the hyperbolic tree [27]. This interface allows one to browse the view

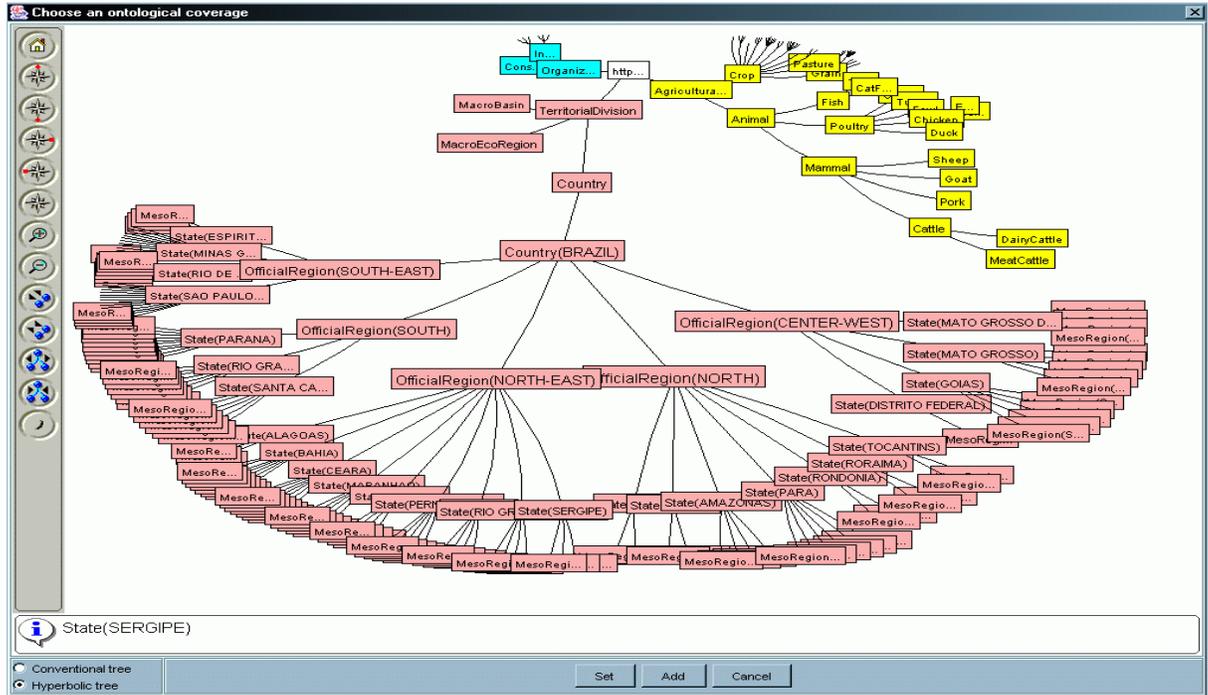


Figure 5.11: A view of the ontology for the agriculture realm embedded in an application program

and choose ontological coverages. The root of the tree shows the URI of the site that provides the ontology. This snapshot details the Territory dimension with the Brazilian regions, states, and finer territorial divisions. The Agricultural Product dimension appears at the right of the root, while the Organization dimension is practically hidden at the left side of the root. One can navigate from the root to the leaves of the tree, to explore the arrangement of concepts and instantiated terms in a view. The use of hyperbolic trees to browse ontology views has proved to be user friendly, despite the high number of nodes to represent all the terms in the ontology (more than 15000 in some experiments).

5.6 Engineering Considerations and Systems Evaluation

5.6.1 Architecture and Design Tradeoffs

The following issues need to be solved in order to implement the ontology-driven facilities in POESIA applications:

1. how to give efficient support to compute semantic relationships between ontological coverages;

2. how to construct ontology views tailored for particular application domains.

These two problems are related: structural restrictions imposed on ontology views enable more efficient algorithms for comparing ontological coverages than using, for example, inference engines like Jess [91] or Algernon [122] to process a full ontology for this purpose. In a tree-like view, determining if a term t encompasses another term t' reduces to determining if the string representing the path from the root o to t is the head of the string representing the path from o to t' . In a DAG-like view, one can use graph search algorithms to determine if there is a path from t to t' . Most of the ontology views used in agricultural zoning applications have the number of edges (semantic relationships) proportional to the number of nodes (terms). This enables computing semantic relationships between ontological coverages with linear complexity.

Given our option for views, our engineering solution to handle ontologies in POESIA involves three aspects: (i) adopt a procedural approach to ontology management, backed by databases to attain persistence and scalability; (ii) project the ontology into views tailored for particular applications, thereby reducing the number of terms and relationships to be handled; (iii) restrict the ontology views used in applications to directed acyclic graphs (DAGs) or trees, in order to facilitate the implementation of visualization and navigation tools and enable efficient algorithms to check relationships between ontological coverages.

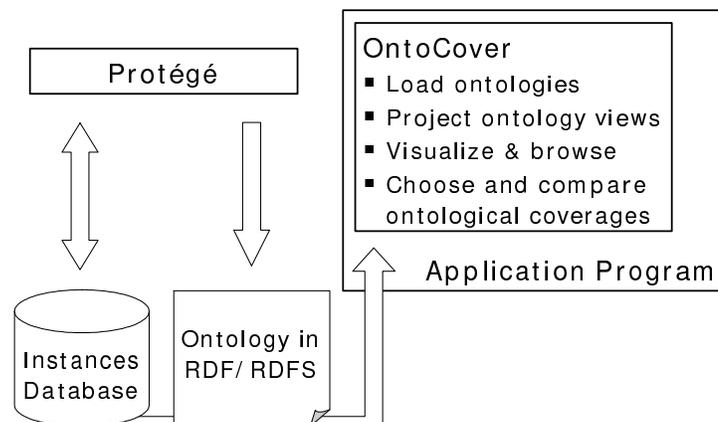


Figure 5.12: Developing and using ontologies in POESIA applications

Figure 5.12 illustrates the major components involved in the development of POESIA ontologies and their use in applications. *Protégé* supports the development of domain ontologies and uploads them into RDF and RDFS files. Instances of ontology concepts can be stored and loaded from databases, in order to speed-up the loading of large ontology data sets. *OntoCover* is a Java library we have developed to load ontologies, build ontology views and handle these views in application programs. *OntoCover* provides the following functionality:

- load ontologies from RDF and RDFS files into databases and vice-versa;
- assemble tree-like views of ontologies in application programs, taking instances of concepts from RDF files or databases;
- graphically browse these ontology views in application programs;
- select ontological coverages (tuples of ontology terms) and check overlap, encompass and equivalence relationships between these ontological coverages in a tree-like view of an ontology.

5.6.2 Constructing Ontology Views

OntoCover uses the Jena toolkit [132] version 2.0 to parse RDF/RDFS specifications of ontologies developed with Protégé and to handle their statements (*resource-property-value* triples). An RDFS (RDF-Schema) file delineates the hierarchies of classes of a domain ontology. An RDF file, on the other hand, specifies instances of those classes and semantic relationships among those instances. Jena loads RDF/RDFS text files in memory or in a database management system (DBMS) and allows navigation in the RDF triples through an application program interface (API) or the RDQL query language, an implementation of SquishQL [177]. The DBMS provides persistence and scalability for large ontology specifications.

We construct an ontology view by using Jena in two steps: (1) load the RDFS of the ontology in RAM; and (2) manipulate RDFS according to the tags described in Section 5.5.3, considering three alternatives for getting instances of the ontology concepts to complete the view:

RAM: use Jena to parse RDF specifications from files into an auxiliary data structure in RAM, which is manipulated via the Jena API to build the tree;

DB RDF: use the Jena API to handle instance data stored as RDF triples in PostgreSQL [206];

DB Conventional: take instances directly from a conventional PostgreSQL database.

The database schema used by Jena to store RDF triples in the DBMS – for the DB RDF strategy – is presented in [248].

Figure 5.13 illustrates the database schema used by the DB Conventional strategy for the instances of territorial divisions in the DBMS. In this figure, rectangles represent tables and the links between rectangles represent 1:N relationships, with a black circle indicating cardinality N. This schema denotes, for example, that a `Country` is politically divided into `OfficialRegions`, each `OfficialRegion` into its constituent `States`, and so on. The territory can also be divided according to ecological issues in `MacroEcoRegions` and their

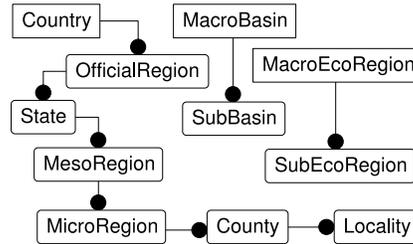


Figure 5.13: The legacy database for territorial divisions

specific `SubEcoRegions`. Alternatively, one can divide the territory according to hydrological, pedological and a number of other criteria.

The RAM strategy to generate an ontology view is expected to give the best performance. However, it has scalability limitations, due to the extensive use of memory. `DB RDF` and `DB Conventional`, on the other hand, combine the flexibility of knowledge management in ontologies with the capabilities of a DBMS for handling large data volumes. They avoid RAM scalability problems, without compromising functionality. Our experiments reported on Section 5.6.3, show that `DB RDF` takes too long, especially for large data sets, probably due to the idiosyncrasies of storing instances in RDF triples and handling them with Jena. `DB Conventional` gives better performance than `DB RDF`. We never keep RDFS specifications of our ontologies in the DBMS, because these specifications are typically too small to be advantageous to do so.

5.6.3 Experimental Evaluation

We have conducted several experiments for implementing `OntoCover` and handling ontologies in `POESIA`. The goal of these experiments was to compare implementation alternatives in terms of ontology view management, from an application point of view. Basically, we investigated the performance of different alternatives in terms of response time, given a user's request concerning relationships between ontological coverages. The results of preliminary experiments clearly showed the advantages of using ontology views as opposed to inference engines. Therefore, we focused further experiments on comparing the alternatives described in Section 5.6.2 to build ontology views. In the following, we report all the experiments, and details the results relative to the construction of ontology views.

Our experiments used the ontology described in Section 5.4. Instances for the `Territory` dimension of this ontology were provided `IBGE` (*Instituto Brasileiro de Geografia e Estatística* – Brazilian Institute of Geography and Statistics). `IBGE`'s data set includes instances for all `Official Regions`, `States`, `Meso-Regions` and `Micro-Regions` (of the states), `Counties` and `Districts` of Brazil. This data set has around 5000 counties and 10000 districts, that we used to generate an ontology graph with more than 15000 nodes, to allow

experiments with large volumes of data.

Views versus Inference Engines

A query on this ontology using Algernon [122] inference engine to determine if a given `State` encompasses a given `District` took several minutes, on Windows 98, in a 2.0 GHz Pentium IV machine, with 512 megabytes of RAM. This is just one example of the scalability problems of the currently proposed Semantic Web technology, in particular of rule-based inference engines. These problems are hard to circumvent for ontologies with arbitrary semantic relationships and complex structures. The whole ontology for the agricultural domain, for example, has multiple applications and includes inverse relationships that give rise to cycles in the ontology's graph-like structure.

When the ontology is reduced to a view in the form of a DAG or tree, the algorithms for comparing ontological coverages run fast (linear time in the input size). Thus, all subsequent experiment were based on views.

View Construction

Given the engineering option for views, the bottleneck has been the memory and time necessary for loading ontology specifications and extracting the views. Therefore, we focused our experiments on this part of the solution. We conducted a series of experiments with Jena version 1.6.1 and Jena 2. We found out that Jena 2.0 outperforms version 1.6.1 by 40% in average and reduces the memory use by almost 2/3 for keeping RDF/RDFS in RAM. For this reason, we only report here the results of the experiments with Jena 2.

Figure 5.14 presents the results of some experiments on constructing tree-like views of chunks of the ontology for the agricultural realm, with increments of 1000 nodes. The Y-axis represents the time to build the view (Figure 5.14(a)) or the memory use (Figure 5.14(b)), for each ontology chunk whose number of nodes appear in the X-axis. We compare the strategies described in Section 5.6.2; namely `RAM`, `DB RDF` and `DB Conventional`. For the `RAM` strategy we consider the time to parse RDFS and RDF, plus the time to build the tree by handling these RDF specifications loaded in memory. `DB RDF` and `DB Conventional`, on the other hand, take advantage of the efficiency of a DBMS to manage large data sets in persistent memory. These strategies only load RDFS as a whole in memory, and query individual instances of the ontology chunk in a PostgreSQL database modeled as RDF triples (`DB RDF`) or in a conventional way (`DB Conventional`). The memory use is the peak of memory allocation for loading the necessary RDF/RDFS triples and build the view. These experiments run on Linux (Red Hat 8), in a 1.6 GHz Pentium IV machine, with 512 megabytes of RAM.

The running time measurements presented in Figure 5.14(a) show that `DB Conventional` is the fastest strategy. `RAM` is slightly slower than `DB Conventional` for large data sets, be-

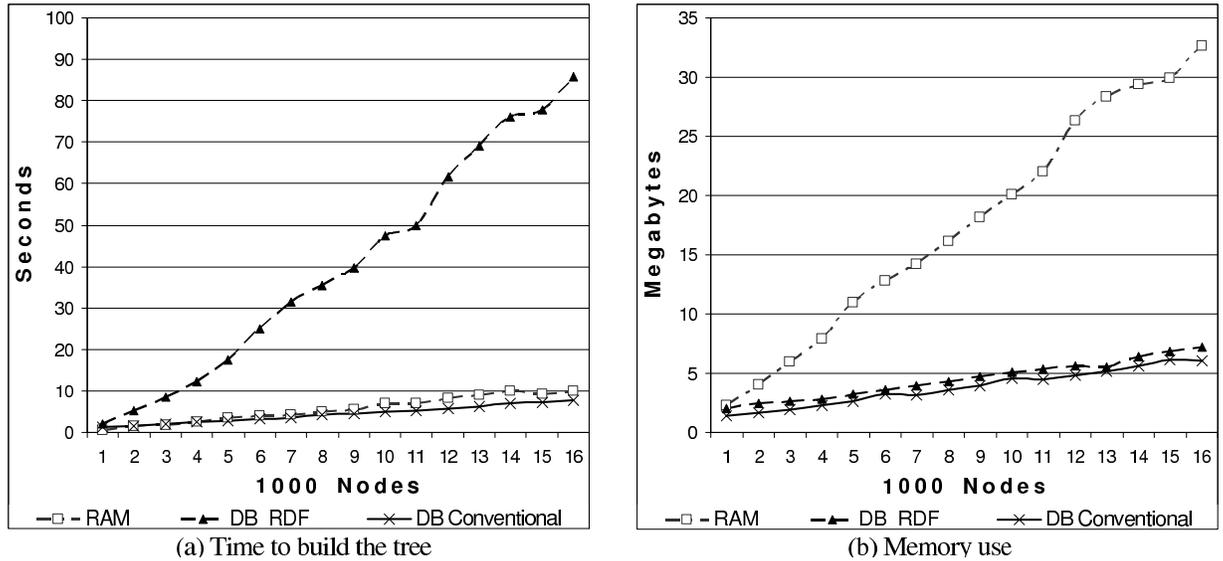


Figure 5.14: Comparing alternative schemes for generating ontology views

cause of the burden of parsing RDF files, as opposed to efficiently taking instances from a database via queries that use its indexes. DB RDF is by far the slowest alternative. This bad performance is probably due to the way RDF breaks the data about each instance – one RDF triple for each field value – leading to additional levels of indirection. Another advantage of DB Conventional over the other two strategies is that it orders the sibling nodes in the ontology view by their labels, using a secondary index for the field supplying the label values. This ordering facilitates browsing and location of specific items of the ontology view in the user interface.

When comparing the memory consumption of the three strategies, Figure 5.14(b) shows that indeed the RAM strategy consumes the largest amount of memory. In contrast, both DB Conventional and DB RDF strategies are more economical, because they do not require the construction of intermediate data structures in memory and take advantage of a database to load large sets of instances. DB Conventional is slightly more economical than DB RDF, perhaps due to Jena's memory management strategies for housekeeping.

From the experimental results in Figure 5.14, we observe two clear trends in the three implementations of ontology view support in OntoCover using Jena 2. First, DB RDF is expensive to build the ontology view (Figure 5.14(a)). Second, RAM consumes significant amount of main memory (Figure 5.14(b)). Fortunately, the DB conventional approach is both fast and economical with memory consumption. Other implementations may solve the instances loading problem and the main memory consumption problem using some alternative technique, but the problems we found seem to be inherent to RDF and main memory management. For currently available software tools, using conventional databases to store and access large ontologies seems to be a good choice.

5.7 Related Work

There is plenty of research nowadays intended to apply Semantic Web ideas in a variety of domains. On the other hand, few studies use the experience acquired in real world applications to evaluate the viability of Semantic Web proposals, and devise methods to provide the scalability and efficiency required in practice. Usually, solutions to handle large repositories of metadata conjugate knowledge representation and manipulation techniques with conventional Web technology and database management systems [132, 248, 74, 162, 133, 30].

RDF [147, 80] is the major format for machine-processable metadata in the Semantic Web. The basic construct of the RDF model is the *statement* – a triple of the form *subject-predicate-object*, where *subject* refers to a resource (anything that can be denoted by a URI), *predicate* is a property of that resource, and *object* is the value of that property. The object can be a literal (e.g., a string) or another resource. RDFS (RDF-Schema) extends RDF with classes and properties to specify domain vocabulary and object structures, i.e., define specific classes and properties for a particular domain or application. Other languages such as DAML+OIL [165] and OWL [196] extend the RDF/RDFS vocabulary to enrich ontologies' expressiveness (e.g., express disjunction of classes and other constraints). Thus, for metadata analysis purposes, one can consider just RDF triples.

RDF can be expressed using XML syntax. However, XML query languages such as XQuery [258, 2] are not suitable for RDF, because they are based on the XML tree structure and ignore the RDF model. Hence, several languages and tools have been developed specifically to query RDF. Jena [132, 248] and its improved version Jena 2 is a popular toolkit for handling RDF triples. [74] and [162] propose other improvements to accelerate queries on RDF triples.

Nevertheless, procedural languages for handling RDF triples and their components are cumbersome. For many applications, such as building ontology views in POESIA (Section 5.6.2), a template-based declarative language would be more appropriate. RQL (RDF Query Language) [133] is a declarative language for querying *RDF directed graphs*, in which resources and objects are represented by labeled nodes and properties by labeled edges. RQL adapts functionality of query languages for semi-structured and XML data [2], to provide functional constructs, in the style of OQL [40], for uniformly querying RDF and RDFS. Sesame [30] is a server-based architecture for storing and querying large quantities of metadata in RDF/RDFS, with support for RQL and concurrency control. Sesame can be deployed on top of a variety of storage devices, such as triple stores, relational and object-oriented databases.

However, it is possible to handle RDF/RDFS in an even higher abstraction level. Jess [91] and Algernon [122] are examples of inference engines able to handle metadata in RDF/RDFS and related formats. These tools can be plugged to an ontology editor such as Protégé [190] or simply process RDF/RDFS exported by such an editor. They regard RDF/RDFS statements as rules formalizing declarative knowledge, and apply inference to derive other knowledge. Our

experiments showed that the performance of Algernon is insufficient for our applications. Thus, as discussed in Section 5.6, we decided to combine tools for handling RDF at the statements level. The theoretical results enabling our implementation appear in Chapter 3.

From the perspective of Semantic Web applications, the key point is to take advantage of knowledge, represented in standards like RDF, to leverage automated means to describe, organize, discover, select and compose Web resources for the solution of a variety of problems. The most usual approach is to define semantic markup based on some ontology, and use them to integrate and provide unified access to data and services, typically via Web portals. There are many examples of this approach in the literature [121, 216, 111, 13].

Some experimental systems possess distinctive features. Edutella [188] is a Peer-to-Peer infrastructure using RDF metadata to facilitate access to educational resources. In Edutella, each peer holds a set of resources and has an RDF repository of resource descriptions, to allow querying its contents at the storage layer (e.g., SQL) or user layer (e.g., RQL). Peers can be heterogeneous in their internal organization and the query language they provide. The common data model and the exchange language of Edutella enables a standard interface for posing queries to specific peers or communities and find resources across the network. Piazza [115] is an infrastructure to provide interoperability of data sources in the Web, by mapping their contents at the domain level (RDF) and the document structure level (XML), and addressing the interoperation between these levels. The mappings are specified declaratively for small sets of nodes. A query answering algorithm chains these mappings together to obtain relevant data from across the network. Other works focus on the interoperability of scientific data repositories on the Web [160, 224]. Finally, the *grid* – a platform for coordinated resource sharing through the Internet, increasingly used for scientific data processing – and the Semantic Web have mutual characteristics and goals [102]. Both operate in a global, distributed and dynamic environment, and both need computationally accessible and sharable metadata to support automated information discovery, integration and aggregation.

POESIA is similar to some of these initiatives, in the sense that they favor cooperation of peers, using Semantic Web apparatus to boost interoperability, instead of trying to coerce the peers to a unique integration schema. Yet, to the best of our knowledge, POESIA is the only approach that employs the partial ordering of resource descriptors – namely ontological coverages and their semantic relationships – to organize, discover, and reuse resources in a particular domain. POESIA also includes mechanisms, based on ontological coverages, workflows and activity models, to semantically orient the composition of Web Services in cooperative distributed processes (Chapter 3) and help to trace the information flow across these processes (Chapter 4).

5.8 Conclusions

The Semantic Web technology has potential to support scientific applications that gather and integrate data from several sources and use a variety of data processing resources. It can improve the functionalities of current syntax-based data processing, and provide enhanced facilities in semantic aware open-ended information systems.

This paper has outlined the POESIA approach for data integration, cooperative data processing and information analysis. It considered particular implementation issues for a new generation of information systems based on the Semantic Web – the loading, adaptation and use of domain ontologies in applications involving data and services discovery and composition on the Web. The main contributions are (1) carrying out facilities adhering to the Semantic Web in a scientific application for the agricultural domain; (2) pointing out some shortcomings of currently proposed standards and tools, when faced with real life systems and large data volumes; (3) the design and implementation of some solutions to overcome these limitations. Though these results were presented in the context of a case study in agriculture, they apply to several domains and a wide class of ontology-based systems. In order to apply POESIA to other domains, two basic requirements must be met: the availability of domain ontologies; and the cooperation of domain experts to specify their workflows and define the appropriate ontology views.

The OntoCover package for generating ontology views, browsing these views and coping with ontological coverages has been completely implemented and incorporated in WOODSS (Workflow-based Decision Support System), a tool that applies scientific workflows to process geographic data for decision making purposes [214]. The association of ontological coverages with workflow activities and data in WOODSS provides a testbed for the use of POESIA semantic descriptions to organize the resources required by cooperative processes involving geographic data – e.g., in environmental planning or biodiversity studies. This approach has been developed in conjunction with experts in agriculture. Complete implementation and validation involve many other issues (e.g., Web services implementation, choreographing services in cooperative processes on the Web), and are left to future work.

The POESIA approach could be applied to the agriculture realm because domain experts in this area were able to establish the ontological agreements necessary to describe and interrelate data and processing activities of cooperative processes. In cases where this is not possible, it is necessary to establish semantic connections between the ontologies used to describe the resources employed in different parts of a cooperative process. This requires further research on ontologies integration, and articulation of processes frameworks using different ontologies.

Another extension for the Semantic Web research is to develop an algebra for handling ontologies, with facilities for declaratively expressing and generating ontology views, as well as merging and integrating ontologies. A richer set of semantic relationships could also be con-

sidered to extend the POESIA approach. RQL and other languages for querying RDF in the semantic level must also be examined to express the ontology views and/or determine term encompassing in the POESIA approach. Still, other research themes include evaluating various standards and tools arising from the Semantic Web research (e.g., DAML+OIL, OWL) to implement POESIA; developing catalogs to support services discovery and composition founded by domain ontologies; and applying the POESIA approach in other domains, such as ecology, biotechnology, sociology, economy and business.

Acknowledgments

The authors from Campinas University are partially supported by Embrapa, CAPES, CNPq and the MCT/PRONEX-SAI and CNPq WebMaps projects. The authors from Georgia Tech are partially supported by two grants from the Operating Systems and ITR programs (CISE/CCR division) of NSF, by a contract from the SciDAC program of DoE, a contract from the PCES program (IXO) of DARPA, a faculty award and a SUR grant from IBM. The application scenarios used in this work were provided by Brazilian experts in agriculture. The usual thanks to Daniel Andrade, who is always ready to help in several issues in our lab. Special thanks to professors Ana Carolina Salgado, Caetano Traina Júnior, Célio Cardoso Guimarães and Edmundo Madeira, who provided several suggestions for the improvement of this work.

Capítulo 6

Conclusões

*“So let us not be blind to our differences –
but let us also direct attention to our common interests
and to the means by which those differences can be resolved.
And if we cannot now end our differences,
at least we can make this world safe for diversity.”*

John F. Kennedy, 1963

A Web semântica visa estender o papel dos computadores no suporte a diversas atividades humanas, através de descritores semânticos dos recursos disponíveis em rede. Esta tese apresentou resultados aderentes à Web semântica para auxiliar a localização de recursos, a integração de dados e a determinação de sua proveniência, em processos obtidos mediante a composição, semanticamente consistente, de serviços Web. A abordagem POESIA, centrada em uma ontologia de domínio, modelos de atividades e workflows, fornece facilidades complementares a outros resultados para a integração de dados e serviços em aplicações Web, particularmente no campo científico.

6.1 Contribuições

A principais contribuições deste trabalho são:

1. descrição dos requisitos estruturais e funcionais de uma aplicação científica – zoneamento agrícola – em que grandes volumes de dados heterogêneos são correlacionados sob condições espaciais e temporais, em processos complexos na Web;
2. um arcabouço teórico, baseado em ontologias de domínio, modelos de atividades e workflows, para a descrição, organização, recuperação e composição de dados e serviços;

3. regras para verificar a consistência semântica de composições de recursos, com base em conceitos específicos do domínio de aplicação;
4. combinação de uma ontologia de domínio e descrições de fluxos de dados para avaliar a proveniência de dados em processos distribuídos na Web;
5. critérios para auxiliar a integração de dados, fundamentados no uso de coberturas ontológicas para expressar o escopo e a granularidade dos dados;
6. validação parcial do arcabouço teórico, através da implementação de alternativas para lidar com grandes volumes de dados em um domínio específico. Em particular, estes experimentos de implementação, descritos no Capítulo 5, indicaram que as técnicas e ferramentas atualmente disponíveis para a Web Semântica não conseguem gerenciar grandes volumes de dados de maneira satisfatória.

Estas contribuições foram publicadas ou submetidas para publicação resultando em um artigo em revista internacional indexada [83] e quatro artigos em conferências [86, 82, 84, 236], além de um artigo para conferência internacional e um relatório técnico recentemente submetidos.

6.2 Extensões

Os trabalhos futuros na abordagem POESIA incluem:

Generalização das ontologias: A abordagem POESIA é baseada em propriedades de relações semânticas entre os termos de uma ontologia de domínio, especificamente equivalência, agregação e especialização. Tais propriedades definem uma ordem parcial entre os termos e a estruturação da ontologia e dos frameworks de processos sob a forma de grafos acíclicos direcionados. Essas características, por sua vez, permitem a implementação eficiente das facilidades propostas para POESIA. A inclusão de outras relações semânticas pode enriquecer a abordagem. Por exemplo, a relação de disjunção pode expressar que duas regiões geográficas (tais como dois estados) não se sobrepõem. Extensões ao arcabouço teórico da abordagem POESIA precisam ser analisadas com cuidado, para garantir a manutenção da consistência da abordagem.

Implementação com diferentes tecnologias: Os experimentos realizados neste doutorado limitaram-se à implementação dos mecanismos necessários à manipulação de ontologias e coberturas ontológicas. Essas implementações utilizam RDF para representar a ontologia de domínio e ferramentas procedurais para a carga e utilização das ontologias em aplicações. Linguagens mais expressivas para a representação de ontologias (como

OWL), linguagens declarativas para manipulação de conhecimento (como RQL) e padrões de metadados para áreas específicas (como GML) devem ser considerados em implementações futuras. Além disso, pacotes para o desenvolvimento de serviços Web têm evoluído rapidamente, e precisam ser avaliados para a implementação completa de POESIA.

Validação em diversas áreas de aplicação: Este trabalho limitou-se à definição dos requisitos de aplicações em agricultura, particularmente do zoneamento agrícola. O próximo passo é a validação da abordagem POESIA junto a especialistas de outros domínios, utilizando e aperfeiçoando os protótipos desenvolvidos nesse trabalho. POESIA tem potencial para aplicação em domínios como ecologia, bioinformática, sociologia, economia e negócios.

Outras extensões transcendem a abordagem POESIA e constituem desafios para pesquisa:

Geração de ontologias: A construção de ontologias é uma tarefa laboriosa e sujeita a erros, omissões e imprecisões. Desta forma, métodos e ferramentas para automatizar a construção de ontologias a partir de textos, dados semi-estruturados e estruturados podem contribuir para baixar os custos e elevar a qualidade das ontologias [94, 62, 181, 182, 184].

Interoperabilidade de ontologias: O desenvolvimento de ontologias para diferentes domínios e aplicações leva a problemas de interoperabilidade entre ontologias. A abordagem POESIA só pôde ser aplicada à agricultura porque os especialistas desse domínio foram capazes de estabelecer acordos para a definição de um referencial ontológico comum. Nos casos em que isso não for possível, deve-se definir conexões entre ontologias distintas. Propostas de solução para esse problema incluem álgebras e modelos baseados em grafos para a composição e articulação de ontologias [79, 183, 131, 247, 245, 246].

Sincronização de processos cooperativos na Web: A tecnologia atual de sincronização de processos baseia-se principalmente em “orquestração” de tarefas, i.e., controle centralizado, mesmo que a execução seja distribuída. Processos cooperativos na Web requerem “co-geografia” de atividades autônomas, baseada na integração de protocolos para garantir a execução harmônica de workflows interorganizacionais. Algumas linguagens de composição de serviços Web [234, 116, 20, 255, 87, 204, 250, 175] e técnicas de modelagem de processos [116, 235, 186, 93, 157] visam contemplar esses requisitos.

Referências

- [1] S. Abiteboul. Querying semi-structured data. Em *Proc. ICDT Conf.*, volume 1186 of *LNCS*, pp. 1–18. Springer-Verlag, 1997.
- [2] S. Abiteboul, P. Buneman e D. Suciu. *Data on the Web – from relations to semistructured data and XML*. Morgan Kaufmann, San Francisco, CA, 2000.
- [3] T. Abraham e J. F. Roddick. Survey of spatio-temporal databases. *GeoInformatica*, 3(1):61–995, 1999.
- [4] B. Adelberg. NoDoSE - a tool for semi-automatically extracting semi-structured data from text documents. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 283–294. ACM Press, 1998.
- [5] C. D. Aguiar. Heterogeneous database integration into urban planning applications. Dissertação de Mestrado, Department of Computer Science, State University of Campinas, Brazil, 1995. (in portuguese).
- [6] L. Ahmedi, P. J. Marrón e G. Lausen. Ontology-based access to heterogeneous XML data, 2001.
- [7] L. Ahmedi, P. J. Marrón e G. Lausen. Ontology-based querying of linked XML documents, 2002.
- [8] A. Ailamaki, Y. E. Ioannidis e M. Livny. Scientific workflow management by database management. Em *Proc. Conf. on Statistical and Scientific Database Management*, pp. 190–199. IEEE Computer Society, 1998.
- [9] J. Albrecht. Geospatial information standards a comparative study of approaches in the standardisation of geospatial information. *Computers & Geosciences*, 25:9–24, 1999.
- [10] G. Alonso e A. E. Abbadi. Cooperative modeling in applied geographic research. Em *CoopIS*, pp. 227–234, 1994.

- [11] G. Alonso e C. Hagen. Geo-opera: Workflow concepts for spatial processes. Em *Advances in Spatial Databases - 5th Intl. Symp. on Large Spatial Databases (SSD)*, volume 1262 of *LNCS*, pp. 238–258. Springer-Verlag, 1997.
- [12] I. Altintas, S. Bhagwanani, D. Buttler, S. Chandra, Z. Cheng, M. Coleman, T. Critchlow, A. Gupta, W. Han, L. Liu, B. Ludäscher, C. Pu, R. Moore, A. Shoshani e M. A. Vouk. A modeling and execution environment for distributed scientific workflows. Em *Proc. Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*, pp. 247–250. IEEE Computer Society, 2003.
- [13] B. Amann, C. Beeri, I. Fundulaki e M. Scholl. Ontology-based integration of xml web resources. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 117–131. Springer-Verlag, 2002.
- [14] A. Ankolekar, M. H. Burstein, J. R. Hobbs, O. Lassila, D. Martin, D. V. McDermott, S. A. McIlraith, S. Narayanan, M. Paolucci, T. R. Payne e K. P. Sycara. DAML-S: Web service description for the semantic web. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 348–363. Springer-Verlag, 2002.
- [15] F. Baader, D. McGuinness, D. Nardi e P. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [16] T. Bandholtz. Sharing ontology by web services: Implementation of a semantic network service (SNS) in the context of the german environmental information network (gein). Em *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pp. 189–201, 2003.
- [17] T. Barclay, D. R. Slutz e J. Gray. TerraServer: A spatial data warehouse. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 307–318. ACM Press, 2000.
- [18] C. Batini, M. L. e S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323–364, 1986.
- [19] C. Behrens e V. Kashyap. The "emergent" semantic web: An approach for derivation of semantic agreements on the web. Em *Proc. Semantic Web Working Symposium (SWWS)*, 2001.
- [20] B. Benatallah, Q. Z. Sheng e M. Dumas. The self-serv environment for web services composition. *IEEE Internet Computing*, 7(1):40–48, 2003.
- [21] S. Bergamaschi, S. Castano e M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

- [22] T. Berners-Lee, J. Hendler e O. Lassila. The semantic web. *Scientific American*, May 2001.
- [23] P. A. Bernstein e T. Bergstraesser. Meta-data support for data transformations using microsoft repository. *IEEE Data Engineering Bulletin*, 22(1):9–14, 1999.
- [24] E. Bertino. A view mechanism for object-oriented databases. Em *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 580 of *LNCS*, pp. 136–151. Springer-Verlag, 1992.
- [25] A. Bonifati e S. Ceri. Comparative analysis of five XML query languages. *SIGMOD Record*, 29(1):68–79, 2000.
- [26] K. A. V. Borges. Geographical data modeling: Extension of OMT for spatial applications. Dissertação de Mestrado, Department of Computer Science, Federal University of Minas Gerais, Brazil, 1997. (in portuguese).
- [27] B. Bou. Treebolic a java applet for hyperbolic hendering of hierarchical data. <http://treebolic.sortilege.net/en> as of September 2003.
- [28] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte e D. Winer. W3C's Simple Object Access Protocol (SOAP). <http://www.w3.org/TR/SOAP/> (as of October 2003).
- [29] D. Brickley e R. V. Guha. RDF vocabulary description language 1.0: RDF schema, 2003. <http://www.w3.org/TR/rdf-schema/> (as of October 2003).
- [30] J. Broekstra, A. Kampman e F. van Harmelen. Sesame: A generic architecture for storing and querying RDF and RDF schema. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 54–68. Springer-Verlag, 2002.
- [31] P. Brown e M. Stonebraker. BigSur: A system for the management of earth science data. Em *Proc. VLDB Conf.*, pp. 720–728. Morgan Kaufmann, 1995.
- [32] P. Buneman. Semistructured data. Em *16th ACM Symposium on Principles of Database Systems (PODS'97)*, pp. 117–121, 1997.
- [33] P. Buneman, S. Khanna e W. C. Tan. Why and where: A characterization of data provenance. Em *Proc. Intl. Conf. on Data Theory (ICDT)*, volume 1973 of *LNCS*, pp. 316–330. Springer-Verlag, 2001.
- [34] D. Buttler, M. Coleman, T. Critchlow, R. Fileto, W. Han, C. Pu, D. Rocco e L. Xiong. Querying multiple bioinformatics information sources: Can semantic web research help? *SIGMOD Record*, 31(4):59–64, 2002.

- [35] D. Buttler, L. Liu e C. Pu. A fully automated object extract system for the web. Em *Proc. Intl. Conf. on distributed Computing Systems (ICDCS)*. IEEE Press, 2001.
- [36] D. Buttler, L. Liu, C. Pu, H. Paques, W. Han e W. Tang. OminiSearch: A method for searching dynamic content on the web. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 604. ACM Press, 2001.
- [37] A. M. C. Bussler, D. Fensel. A conceptual architecture for semantic web enabled web services. *SIGMOD Record*, 31(4):24–29, 2003.
- [38] J. Cardoso e A. Sheth. Semantic e-workflow composition. Report, LSDIS Lab, Computer Science Dep., Univ. of Georgia, 2002.
- [39] F. Casati e U. D. (editors). Special issue on web services. *IEEE Data Engineering Bulletin*, 25(4), 2002.
- [40] R. G. G. Cattell, D. Barry, M. Berler, D. Jordan, C. Russel, O. Schadow, T. Stanienda e F. Velez. *The Object Data Standard - ODMG 3.0*. Morgan Kaufmann, 2000.
- [41] M. C. Cavalcanti, F. A. Baião, S. C. Rössle, P. M. Bisch, R. Targino, P. F. Pires, M. L. Campos e M. Mattoso. Structural genomic workflows supported by web services. Em *Proc. Intl. Conf. on Database and Expert Systems Applications (DEXA)*, pp. 45–49. IEEE Computer Society, 2003.
- [42] M. C. Cavalcanti, M. Mattoso, M. L. Campos, F. Llirbat e E. Simon. Sharing scientific models in environmental applications. Em *Proc. ACM Symposium on Applied computing (SAC)*, pp. 453–457. ACM Press, 2002.
- [43] M. C. Cavalcanti, M. Mattoso, M. L. Campos, E. Simon e F. Llirbat. An architecture for managing distributed scientific resources. Em *Proc. Intl. Conf. on Scientific and Statistical Database Management (SSDBM)*, pp. 47–55. IEEE Computer Society, 2002.
- [44] S. Ceri, P. Fraternali e S. Paraboschi. XML: Current developments and future challenges for the database community. Em *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 1777 of LNCS, pp. 3–17. Springer-Verlag, 2000.
- [45] D. Chang e D. Harkey. *Client/Server Data Access with Java and XML*. John Wiley & Sons, 1998.
- [46] S. Chaudhuri e U. Dayal. An overview of data warehousing and olap technology. *SIGMOD Record*, 26(1):65–74, 1997.

- [47] V. Christophides, S. Cluet e J. Siméon. On wrapping query languages and efficient XML integration. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 141–152. ACM Press, 2000.
- [48] S. Cluet, C. Delobel, J. Siméon e K. Smaga. Your mediators need data conversion! Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 177–188. ACM Press, 1998.
- [49] G. Câmara, M. A. Casanova, A. Hemerly, G. C. Magalhães e C. B. Medeiros. *Anatomy of Geographical Information systems*. State University of Campinas, Brazil, 1996. (in portuguese).
- [50] G. Câmara, A. M. V. Monteiro, J. A. Paiva, J. Gomes e L. Velho. Towards a unified framework for geographical data models. *Journal of the Brazilian Computing Society*, 7(1), 2000.
- [51] G. Câmara, R. Thomé, U. Freitas e A. Monteiro. Interoperability in practice: Problems in semantic conversion from current technology to open GIS. Em *Intl. Conf. on Interoperable GIS*, 1999.
- [52] OMG's Common Object Ranguageesquest Broker Architecture (CORBA). <http://www.omg.org/CORBA> (as of October 2003).
- [53] O. Corcho, A. Gómez-Pérez, M. Fernández-López e M. Lama. ODE-SWS: A semantic web service development environment. Em *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pp. 203–216, 2003. <http://www.cs.uic.edu/~ifc/SWDB/> (as of November 2003).
- [54] J. E. Córcoles, P. González e V. L. Jaquero. Integration of spatial XML documents with RDF. Em *Proc. Ibero American Conference on Web Engineering (ICWE)*, volume 2722 of *LNCS*, pp. 407–410. Springer-Verlag, 2003.
- [55] Y. Cui e J. Widom. Practical lineage tracing in data warehouses. Em *Proc. Intl. Conf. on Data Engineering (ICDE)*, pp. 367–378. IEEE, 2000.
- [56] Y. Cui e J. Widom. Lineage tracing for general data warehouse transformations. Em *Proc. VLDB Conf.*, pp. 471–480. Morgan Kaufmann, 2001.
- [57] Y. Cui, J. Widom e J. L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM TODS*, 25(2):179–227, 2000.
- [58] G. R. Cunha e E. D. A. (eds.). An overview of the special issue on crop zoning in Brazil. *Brazilian Journal of Agrometeorology*, 9(3), 2001. (in Portuguese).

- [59] B. Curtis, M. Kellner e J. Over. Process Modeling. *Communications of the ACM*, 35(9):75–90, 1992.
- [60] A. S. da Silva, P. Calado, R. Vieira, A. H. F. Laender e B. A. Ribeiro-Neto. *Keyword-Based Queries over Web Databases*, pp. 74–92. IRM Press, 2003.
- [61] The DARPA Agent Markup Language (DAML). <http://www.daml.org/> (as of August 2003).
- [62] H. Davulcu, S. Vadrevu e S. Nagarajan. Ontominer: Bootstrapping and populating ontologies from domain specific web sites. Em *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pp. 259–276, 2003. <http://www.cs.uic.edu/~ifc/SWDB/> (as of November 2003).
- [63] Y. Ding, D. Fensel, M. Klein e B. Omelayenko. The semantic web: yet another hip? *Data & Knowledge Engineering*, 41(2/3):205–227, junho 2002.
- [64] B. Dinter, C. Sapia, G. Hofling e M. Blaschka. The OLAP market: state of the art and research issues. Em *ACM 1st Intl. Workshop on Data Warehousing and OLAP (DOLAP'98)*, pp. 22–27, 1998.
- [65] Dublin Core Metadata Initiative. <http://www.dublincore.org/> (as of October 2003).
- [66] D. S. (ed.). Special issue on management of semistructured data. *SIGMOD Record*, 26(4), 1997.
- [67] A. Y. H. (editor). Special issue on XML data management. *IEEE Data Engineering Bulletin*, 24(2), 2002.
- [68] G. W. (editor). Special issue on organizing and discovering the semantic web. *IEEE Data Engineering Bulletin*, 25(1), 2002.
- [69] R. M. (editor). Special issue on integration management. *IEEE Data Engineering Bulletin*, 25(3), 2002.
- [70] M. Egenhofer. Toward the semantic geospatial web. Em *Proc. ACM GIS*, 2002.
- [71] G. Ehmayr, G. Kappel e S. Reich. Connecting databases to the web: A taxonomy of gateways. Em *Proc. Intl. Conf. on Database and Expert Systems Applications (DEXA)*, pp. 1–15. IEEE Computer Society, 1997.
- [72] A. K. Elmagarmid e C. Pu. Introduction: Special issue on heterogeneous databases. *ACM Computing Surveys*, 22(3):175–178, 1990.

- [73] R. Elmasri e S. B. Navathe. *Fundamentals of Database Systems*. Addison-Wesley, Menlo Park, CA, 1994.
- [74] F. Esposito, L. Iannone, I. Palmisano e G. Semeraro. RDF core: A component for effective management of RDF models. Em *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pp. 169–187. VLDB endowment, 2003. <http://www.cs.uic.edu/~ifc/SWDB/> (as of November 2003).
- [75] P. H. C. et al. Climatic risk for coffee in Paraná state. *Brazilian Journal of Agrometeorology*, 9(3):486–494, 2001. (in Portuguese).
- [76] J. Euzenat. Research challenges and perspectives of the semantic web. *IEEE Intelligent Systems*, 17(5):86–88, 2002.
- [77] H. Fan e A. Poulouvasilis. Tracing data lineage using schema transformation pathways. Em *Workshop on Knowledge Transformation for the Semantic Web KTSW/ECAI*, volume 95, pp. 64–79. IOS Press, 2003.
- [78] A. Farquhar, R. Fikes e J. Rice. The ontolingua server: a tool for collaborative ontology construction. *Intl. Journal of Human Computer Studies*, 46(6):707–727, 1997.
- [79] D. Fensel. Ontology-based Knowledge Management. *IEEE Computer*, 35(11):56–59, 2002.
- [80] D. Fensel, J. Hendler, H. Lieberman e W. W. (editors). *Spinning the Semantic Web*. MIT Press, 2003.
- [81] C. Ferris e J. Farrel. What are web services? *Communications of the ACM*, 46(6):31, 2003.
- [82] R. Fileto. Issues on interoperability of heterogeneous and geographical data. Em *Simpósio Brasileiro de Geoinformática (GEOINFO)*, pp. 133–140, 2001.
- [83] R. Fileto, L. Liu, C. Pu, E. D. Assad e C. B. Medeiros. POESIA: An ontological workflow approach for composing web services in agriculture. *The VLDB Journal*, 12(4):352–367, 2003.
- [84] R. Fileto, L. Liu, C. Pu, E. D. Assad e C. B. Medeiros. Using domain ontologies to help track data provenance. Em *Proc. Brazilian Symposium on Databases*, pp. 84–98, 2003.
- [85] R. Fileto e C. B. Medeiros. The design of decision support systems for effective use of spatio-temporal data. Em *Ph.D. Students Workshop at EDBT Conf.*, Konstanz, Germany, 2000.

- [86] R. Fileto, C. A. A. Meira, A. S. Neto, J. Naka e C. B. Medeiros. An XML-centered warehouse to manage information of the fruit supply chain. Em *The World Conf. on Computers in Agriculture and Natural Resources (WCCA)*, 2001.
- [87] D. Florescu, A. Grünhagen e D. Kossmann. XL: An XML programming language for web service specification and composition. Em *Proc. WWW Conf.*, pp. 65–76. ACM Press, 2002.
- [88] D. Florescu, A. Grünhagen e D. Kossmann. XL: A platform for web services. Em *Proc. Conf. on Innovative Data Systems Research (CIDR)*, 2003.
- [89] D. Florescu, A. Y. Levy e A. O. Mendelzon. Database techniques for the world-wide web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.
- [90] F. T. Fonseca, M. Egenhofer, P. Agouris e G. Câmara. Using ontologies for integrated geographic information systems. *Trans. in GIS*, 6(3):13–19, 2002.
- [91] E. Friedman-Hill. JESS – the rule engine for the java platform. <http://herzberg.ca.sandia.gov/jess> (as of August 2003).
- [92] A. L. Furtado, K. C. Sevcik e C. S. dos Santos. Permitting updates through views of data bases. *Information Systems*, 4(4):269–283, 1979.
- [93] A. Gal. Semantic interoperability in information services: Experiencing with CoopWARE. *SIGMOD Record*, 28(1):68–75, 1999.
- [94] A. Gal, G. Modica e H. Jamil. OntoBuilder: Fully automatic extraction and consolidation of ontologies from web sources. Em *Intl. Conf. on Conceptual Modeling*, 2003.
- [95] H. Galhardas, D. Florescu, D. Shasha, E. Simon e C. Saita. Improving data cleaning quality using a data lineage facility. Em *Proc. Conf. on Data Management and Data Warehouses (DMDW)*, volume 39 of *CEUR-WS.org*, 2001.
- [96] E. Gamma, R. Helm, R. Johnson e J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, Massachusetts, 1995.
- [97] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos e J. Widom. The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117–132, 1997.
- [98] H. Garcia-Molina, J. D. Ullman e J. Widom. *Database System Implementation*. Prentice Hall, Upper Saddle River, NJ, 2000.

- [99] F. Gingras e L. V. S. Lakshmanan. nd-sql: A multi-dimensional language for interoperability and olap. Em *Proc. VLDB Conf.*, pp. 134–145. Morgan Kaufmann, 1998.
- [100] F. Gingras, L. V. S. Lakshmanan, I. N. Subramanian, D. Papoulis e N. Shiri. Languages for multi-database interoperability. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 536–538. ACM Press, 1997.
- [101] A. Gómez-Pérez e O. Corcho. Ontology specification languages for the semantic web. *IEEE Intelligent Systems*, 17(1):54–60, 2002.
- [102] C. Goble e D. D. Roure. The grid: An application of the semantic web. *SIGMOD Record*, 31(4):65–70, 2002.
- [103] C. A. Goble, D. L. McGuinness, R. Möller e P. F. Patel-Schneider. OilEd a reasonable ontology editor for the semantic web. Em *Intl. Description Logics Workshop*, volume 49 of *CEUR Workshop Proceedings*, 2001.
- [104] C. A. Goble, R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim e A. Brass. Transparent access to multiple bioinformatics information sources. *IBM Systems Journal*, 40(2):532–551, 2001.
- [105] M. F. Goodchild, M. J. Egenhofer, R. Fegeas e C. Kottman. *Interoperating Geographical Information Systems*. Kluwer, 1997.
- [106] L. Gravano, P. G. Ipeirotis e M. Sahami. QProber: A system for automatic classification of hidden-web databases. *ACM Transactions on Information Systems (TOIS)*, 21(1):1–41, 2003.
- [107] J. Gray, A. Bosworth, A. Layman e H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. Em *Proc. Intl. Conf. on Data Engineering (ICDE)*, pp. 152–159. IEEE, 1996.
- [108] M. Grüninger. Applications of PSL to semantic web services. Em *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pp. 217–230, 2003. <http://www.cs.uic.edu/~ifc/SWDB/> (as of November 2003).
- [109] M. Gruninger e J. L. (eds.). Special issue on ontologies applications and design. *Communications of the ACM*, 45(2):39–65, 2002.
- [110] N. Guarino. Formal ontology and information systems. Em *Proc. Intl. Conf. on Formal Ontologies in Information Systems (FOIS)*, pp. 3–15. IOS Press, 1998.

- [111] R. Guha, R. McCool e E. Miller. Semantic search. Em *Proc. WWW Conf.*, pp. 700–709. ACM Press, 2003.
- [112] A. Gupta, H. V. Jagadish e I. S. Mumick. Data integration using self-maintainable views. Em *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 1057 of *LNCS*, pp. 140–144. Springer-Verlag, 1996.
- [113] A. Gupta e I. S. Mumick. Maintenance of materialized views: Problems, techniques, and applications. *IEEE Data Engineering Bulletin*, 18(2):3–18, 1995.
- [114] L. M. Haas, P. M. Schwarz, P. Kodali, E. Kotlar, J. E. Rice e W. C. Swope. Discoverylink: A system for integrated access to life sciences data sources. *IBM Systems Journal*, 40(2):489–511, 2001.
- [115] A. Y. Halevy, Z. G. Ives, P. Mork e I. Tatarinov. Piazza: Data management infrastructure for semantic web applications. Em *Proc. WWW Conf.*, pp. 556–567. ACM Press, 2003.
- [116] R. Hamadi e B. Benatallah. A petri net-based model for web service composition. Em *Proc. Australasian Database Conf. (ADC)*, pp. 191–200. Australasian Computer Society, 2003.
- [117] J. Hammer, H. Garcia-Molina, K. Ireland, Y. Papakonstantinou, J. D. Ullman e J. Widom. Information translation, mediation, and mosaic-based browsing in the TSIMMIS system. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 483. ACM Press, 1995.
- [118] T. Härder, G. Sauter e J. Thomas. The intrinsic problems of structural heterogeneity and an approach to their solution. *The VLDB Journal*, 8(1):25–43, 1999.
- [119] V. Harinarayan, A. Rajaraman e J. D. Ullman. Implementing data cubes efficiently. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 205–216. ACM Press, 1996.
- [120] W. Hasselbring. Information system integration. *Communications of the ACM*, 43(6):33–38, 2000.
- [121] S. Haustein e J. Pleumann. Is participation in the semantic web too difficult? Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 448–453. Springer-Verlag, 2002.
- [122] M. Hewett. Algernon in java. <http://smi.stanford.edu/people/hewett/research/ai/algernon/> (as of August 2003).
- [123] D. Hollingsworth. *The Workflow Reference Model*. Workflow Management Coalition, January 1995.

- [124] I. Horrocks e J. Hendler, editors. *Intl. Semantic Web Conf.(ISWC)*, volume 2342 of LNCS, Sardinia, Italy, June 2002. Springer-Verlag.
- [125] B. Hüsemann, J. Lechtenböcker e G. Vossen. Conceptual data warehouse modeling. Em *2nd Intl. Workshop on Design and Management of Data Warehouses*, 2000. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/> (as of November 2000).
- [126] D. K. Hsiao e M. N. Kamel. Heterogeneous databases: Proliferation, issues, and solutions. *IEEE Tran. on Knowledge and Data Engineering*, 1(1):45–62, 1989.
- [127] W. H. Inmon. *Building the Data Warehouse*. John Wiley and Sons, New York, 1996.
- [128] International Organization for Standardization. Standard generalized markup language (SGML), 1986. ISO 8879.
- [129] Y. E. Ioannidis, M. Livny, A. Ailamaki, A. Narayanan e A. Therber. ZOO: A desktop emperiment management environment. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 580–583. ACM Press, 1997.
- [130] S. Jablonski e C. Bussler. *Workflow Management. Modeling Concepts, Architecture and Implementation*. International Thomson Computer Press, 1996.
- [131] J. Jannink, P. Mitra, E. Neuhold, S. Pichai, R. Studer e G. Wiederhold. An algebra for semantic interoperation of semistructured data. Em *IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, pp. 86–100, 1999.
- [132] Jena semantic web toolkit. <http://jena.sourceforge.net/> (as of September 2003).
- [133] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis e M. Scholl. RQL: A declarative query language for RDF. Em *Proc. Intl. World Wide Web Conf.*, pp. 592–503. ACM Press, 2002.
- [134] V. Kashyap e A. Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. Em M. P. Papazoglou e G. Schlageter, editors, *Cooperative Information Systems*, pp. 139–178. Academic Press, San Diego, 1998.
- [135] V. Kashyap e A. Sheth. *Information Brokering Across Heterogenous Digital Data*. Kluwer Academic Publishers, 2000.
- [136] W. Kent. The many forms of a single fact. Em *IEEE COMPCON*, pp. 438–443, 1989.

- [137] W. Kim e J. Seo. Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, 24(12):12–18, 1991.
- [138] S. Kleijnen e S. Raju. An open web services architecture. *ACM Queue*, 1(1):39–46, 2003.
- [139] M. Klein, D. Fensel, F. van Harmelen e I. Horrocks. The relation between ontologies and xml schemas, 2001. <http://www.ep.liu.se/ea/cis/2001/004/> (as of November 2003).
- [140] M. R. Koivunen e E. Miller. W3C semantic web activity, 2001. <http://www.w3.org/2001/12/semweb-fin/w3csw> (as of November 2003).
- [141] H. Kreger. Fulfilling the web services promise. *Communications of the ACM*, 46(6):29–34, 2003.
- [142] R. Krishnamurthy, W. Litwin e W. Kent. Language features for interoperability of databases with schematic discrepancies. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 40–49. ACM Press, 1991.
- [143] A. Labrinidis e N. Roussopoulos. Generating dynamic content at database-backed web servers: cgi-bin vs. mod-perl. *SIGMOD Record*, 29(1):26–31, 2000.
- [144] L. V. S. Lakshmanan, F. Sadri e I. N. Subramanian. SchemaSQL - a language for interoperability in relational multi-database systems. Em *Proc. VLDB Conf.*, pp. 239–250. Morgan Kaufmann, 1996.
- [145] L. V. S. Lakshmanan, S. N. Subramanian, N. Goyal e R. Krishnamurthy. On query spreadsheets. Em *Proc. Intl. Conf. on Data Engineering (ICDE)*, pp. 134–141. IEEE, 1998.
- [146] D. P. Lanter. Design of a lineage-based metadata base for GIS. *Cartography and Geography Information Systems*, 18(4):255–261, 1991.
- [147] O. Lassila e R. R. Swick. Resource Description Framework (RDF): Model and syntax specification, 1999. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222> (as of November 2003).
- [148] D. Lee e W. W. Chu. Comparative analysis of six XML schema languages. *SIGMOD Record*, 29(3):76–87, 2000.
- [149] T. Lee, S. Bressan e S. E. Madnick. Source attribution for querying against semi-structured documents. Em *Proc. Workshop on Web Information and Data Management*, pp. 33–39, 1998.

- [150] A. Levy. More on data management for XML, 1999. <http://www.cs.washington.edu/homes/alon/widom-response.html> (as of November 1999).
- [151] W. Litwin e A. Abdellatif. Multidatabase interoperability. *IEEE Computer*, 19(12):10–18, 1986.
- [152] W. Litwin, L. Mark e N. Roussopoulos. Interoperability of multiple autonomous databases. *ACM Computing Surveys*, 22(3):267–293, 1990.
- [153] L. Liu, D. Buttler, T. Critchlow, W. Han, H. Paques, C. Pu e D. Rocco. Bioseek: Exploiting source-capability information for integrated access to multiple bioinformatics data sources. Em *Intl. Symp. on BioInformatics and BioEngineering (BIBE)*, pp. 263–274. IEEE Computer Society, 2003.
- [154] L. Liu e R. Meersman. The building blocks for specifying communication behavior of complex objects: An activity-driven approach. *ACM TODS*, 21(2):157–207, 1996.
- [155] L. Liu e C. Pu. Activityflow: Towards incremental specification and flexible coordination of workflow activities. Em *Intl. Conf. on Conceptual Modeling (ER)*, volume 1331 of *LNCS*, pp. 169–182. Springer-Verlag, 1997.
- [156] L. Liu e C. Pu. Methodical restructuring of complex workflow activities. Em *Proc. Intl. Conf. on Data Engineering (ICDE)*, pp. 342–350. IEEE, 1998.
- [157] L. Liu e C. Pu. A transactional activity model for organizing open-ended cooperative activities. Em *Hawaii Intl. Conf. on System Sciences (HICSS)*, 1998.
- [158] L. Liu, C. Pu e W. Han. XWrap: An XML-enabled wrapper construction system for web information sources. Em *Proc. Intl. Conf. on Data Engineering (ICDE)*, pp. 611–621. IEEE Press, 2000.
- [159] D. B. Lomet e J. W. (eds.). Special issue on materialized views and data warehouses. *IEEE Data Engineering Bulletin*, 18(2), 1995.
- [160] K. S. M. Gertz. Integrating scientific data through external, concept-based annotations. Em *Proc. VLDB Workshop on Efficiency and Effectiveness of XML Tools and Techniques (EEXTT)*, volume 2590 of *LNCS*, pp. 220–240. Springer-Verlag, 2002.
- [161] D. S. Mackay. Semantic integration of environmental models for application to global information systems and decision-making. *SIGMOD Record*, 28(1):13–19, 1999.

- [162] A. Matono, T. Amagasa, M. Yoshikawa e S. Uemura. An indexing scheme for RDF and RDF schema based on suffix arrays. Em *Proc. Intl. Workshop on Semantic Web and Databases (SWDB)*, pp. 169–187. VLDB endowment, 2003. <http://www.cs.uic.edu/~ifc/SWDB/> (as of November 2003).
- [163] R. Mattison. *Data warehousing: strategies, technologies and techniques*. John Wiley and Sons, New York, 1996.
- [164] E. M. Maximilien e M. P. Singh. Conceptual model of web service reputation. *SIGMOD Record*, 31(4):36–41, 2002.
- [165] D. L. McGuinness, R. Fikes, J. Hendler e L. A. Stein. DAML+OIL: An ontology language for the semantic web. *IEEE Intelligent Systems*, 17(5), Sep 2002.
- [166] S. A. McIlraith, T. C. Son e H. Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2):46–53, 2001.
- [167] C. B. Medeiros e F. Pires. Databases for GIS. *SIGMOD Record*, 23(1):107–115, março 1994.
- [168] C. B. Medeiros, G. Vossen e M. Weske. WASA - a workflow-based architecture to support scientific database applications. Em *Proc. Intl. Conf. on Database and Expert Systems Applications (DEXA)*, volume 978 of LNCS, pp. 574–583. Springer-Verlag, 1995.
- [169] C. B. Medeiros, M. Weske e G. Vossen. GEO-WASA - combining GIS technology with workflow management. Em *Israeli Conf. on Computer-Based Systems and Software Engineering*, pp. 129–139, 1996.
- [170] J. Meidanis, G. Vossen e M. Weske. Using workflow management in dna sequencing. Em *CoopIS*, pp. 114–123, 1996.
- [171] E. Mena, A. Illarramendi, V. Kashyap e . P. Sheth. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.
- [172] E. Mena, V. Kashyap, A. Illarramendi e A. P. Sheth. Managing multiple information sources through ontologies: Relationship between vocabulary heterogeneity and loss of information. Em *KRDB*, number 4 inCEUR-WS.org, 1996.
- [173] E. Mena, V. Kashyap, A. Sheth e A. Illarramendi. Domain specific ontologies for semantic information brokering on the global information infrastructure, 1998.

- [174] G. A. Mihaila, L. Raschid e A. Tomasic. Locating and accessing data repositories with web semantics. *VLDB Journal*, 11(1):41–57, 2002.
- [175] T. Mikalsen, S. Tai e I. Rouvellou. Transactional attitudes: Reliable composition of autonomous web services. Em *Proc. Workshop on Dependable Middleware-based Systems (WDMS)*, 2002.
- [176] G. Miller. The web services debate – .NET versus J2EE. *Communications of the ACM*, 46(6):64–67, 2003.
- [177] L. Miller, A. Seaborne e A. Reggiori. Three implementations of SquishQL, a simple RDF query language. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 423–435. Springer-Verlag, 2002.
- [178] R. J. Miller. Using schematically heterogeneous structures. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 189–200. ACM Press, 1998.
- [179] D. S. Milojevic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins e Z. Xu. Peer-to-peer computing. Report HPL-2002-57, HP Labs, Palo Alto, CA, 2002. <http://www.hpl.hp.com/techreports/2002/HPL-2002-57.html> (as of December 2003).
- [180] M. Minsky. A framework for representing knowledge. Em *The Psychology of Computer Vision*, pp. 211–277. McGraw-Hill, 1975.
- [181] M. Missikoff, R. Navigli e P. Velardi. Integrated approach to web ontology learning and engineering. *IEEE Computer*, 35(11):60–63, 2002.
- [182] M. Missikoff, R. Navigli e P. Velardi. The Usable Ontology: An Environment for Building and Assessing a Domain Ontology. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 39–53. Springer-Verlag, 2002.
- [183] P. Mitra, G. Wiederhold e M. L. Kersten. A graph-oriented model for articulation of ontology interdependencies. Em *Proc. Intl. Conf. on Extending Database Technology (EDBT)*, volume 1777 of *LNCS*, pp. 86–100. Springer-Verlag, 2000.
- [184] G. A. Modica, A. Gal e H. M. Jamil. The use of machine-generated ontologies in dynamic information seeking. Em *Proc. Intl. Conf. on Cooperative Information Systems (CoopIS)*, volume 2172 of *LNCS*, pp. 433–448. Springer-Verlag, 2001.
- [185] Namespaces in XML 1.1. <http://www.w3.org/TR/xml-names11/> (as of October 2003).

- [186] S. Narayanan e S. A. McIlraith. Simulation, verification and automated composition of web services. Em *Proc. WWW Conf.*, pp. 77–88. ACM Press, 2002.
- [187] M. Neiling, M. Schaal e M. Schumann. WrapIt: Automated integration of web databases with extensional overlaps. Em *Proc. Intl. Conf. on Web Databases and Web Services*, volume 2593 of *LNCS*, pp. 184–198. Springer-Verlag, 2002.
- [188] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér e T. Risch. EDUTELLA: a P2P networking infrastructure based on RDF. Em *Proc. WWW Conf.*, pp. 604–615. ACM Press, 2002.
- [189] S. Nestorov, S. Abiteboul e R. Motwani. Extracting schema from semistructured data. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 295–306. ACM Press, 1998.
- [190] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Ferguson e M. A. Musen. Creating semantic web contents with protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2002.
- [191] Ontology inference layer (OIL). <http://www.ontoknowledge.org/oil/> (as of November 2003).
- [192] OpenGIS Consortium. Geography markup language (GML). <http://www.opengis.net/gml/02-069/GML2-12.html> (as of October 2003).
- [193] R. Orfali e D. Harkey. *Client/Server Programming with Java and CORBA*. John Wiley & Sons, 2 edition, 1998.
- [194] R. Orfali, D. Harkey e J. Edwards. *The Essential Distributed Objects Survival Guide*. John Wiley & Sons, 1996.
- [195] A. M. Ouksel e A. P. Sheth. Semantic interoperability in global information systems: A brief introduction to the research area and the special section. *SIGMOD Record*, 28(1):5–12, 1999.
- [196] Web ontology language (OWL) version 1.0. <http://www.w3.org/TR/2003/WD-owl-ref-20030221/> (as of November 2003).
- [197] M. T. Ozsú e P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, San Ysidro, CA, 1999.
- [198] M. Paolucci, T. Kawamura e K. P. S. T. R. Payne. Semantic matching of web services capabilities. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 333–347. Springer-Verlag, 2002.

- [199] Y. Papakonstantinou, H. Garcia-Molina e J. Widom. Object exchange across heterogeneous information sources. Em *Proc. ICDT Conf.*, pp. 251–260. IEEE Press, 1995.
- [200] C. Parent e S. Spaccapietra. Issues and approaches of database integration. *CACM*, 41(5):166–178, 1998.
- [201] P. Patel-Schneider e J. Siméon. Building the semantic web on XML. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 147–161. Springer-Verlag, 2002.
- [202] P. Patel-Schneider e J. Siméon. The yin/yang web: Xml syntax and rdf semantics. Em *Proc. Intl. Conf. on World Wide Web*, pp. 443–453. ACM Press, 2002.
- [203] G. R. B. Pinto, S. P. J. Medeiros, J. M. de Souza, J. C. M. Strauch e C. R. F. Marques. Spatial data integration in a collaborative design framework. *Communications of the ACM*, 46(3):86–90, 2003.
- [204] P. F. Pires, M. R. F. Benevides e M. Mattoso. Building reliable web services compositions. Em *Proc. Intl. Conf. on Web Databases and Web Services*, volume 2593 of *LNCS*, pp. 59–72. Springer-Verlag, 2002.
- [205] V. Poe, P. Klauer e S. Brost. *Building a Data warehouse for Decision Support*. Prentice Hall, 1998.
- [206] PostgreSQL. <http://www.postgresql.org/> as of September 2003.
- [207] C. Pu, K. Schwan e J. Walpole. Infosphere project: System support for information flow applications. *SIGMOD Record*, 30(1):25–34, 2001.
- [208] D. Quass e J. Widom. On-line warehouse view maintenance. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 393–404. ACM Press, 1997.
- [209] A. R. P. D. Smedt, W. Du, W. Kent, M. A. Ketabchi, W. Litwin, A. Rafii e M.-C. Shan. The pegasus heterogeneous multidatabase system. *IEEE Computer*, 24(12):19–27, 1991.
- [210] S. Raghavan e H. Garcia-Molina. Integrating diverse information management systems: A brief survey. *IEEE Data Engineering Bulletin*, 24(4):44–52, 2001.
- [211] W3C’s Resource Description Framework (RDF). <http://www.w3.org/RDF/> (as of October 2003).
- [212] L. A. Rossetti. Agricultural zoning: Lessening the risks of agriculture and providing sustainable regional development. Em *Intl. Symp. on Making Sustainable Regional Development Visible*, 2000.

- [213] M. Schlosser, M. Sintek, S. Decker e W. Nejdl. A scalable and ontology-based p2p infrastructure for semantic web services. Em *Proc. Intl. Conf. on Peer-to-Peer Computing (P2P)*, pp. 104–111. Australasian Computer Society, 2002.
- [214] L. A. Seffino, C. B. Medeiros, J. V. Rocha e B. Yi. WOODSS - a spatial decision support system based on workflows. *Decision Support Systems*, 27(1-2):105–123, 1999.
- [215] W3C's Semantic web Activity. <http://www.w3.org/2001/sw/> (as of July 2003).
- [216] U. Shah, T. Finin e J. Mayfield. Information retrieval on the semantic web. Em *Proc. Intl. Conf. on Information and Knowledge Management (CIKM)*, pp. 461–468. ACM Press, 2002.
- [217] S. Shekthar, S. Chawla, S. Ravada, A. Fetterer, X. Liu e C. tien Lu. Spatial databases - accomplishments and reseach needs. *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 1999.
- [218] A. P. Sheth. Semantic issues in multidatabase systems - preface by the special issue editor. *SIGMOD Record*, 20(4):5–9, 1991.
- [219] A. P. Sheth e J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.
- [220] E. Sirin, J. A. Hendler e B. Parsia. Semi-automatic composition of web services using semantic descriptions. Em *Proc. Workshop on Web Services: Modeling, Architecture and Infrastructure (WSMAI)*, pp. 17–24. ICEIS Press, 2003.
- [221] T. Sollazzo, S. Handschuh, S. Staab e M. Frank. Semantic web service architecture – evolving web service standards toward the semantic web. Em *FLAIRS Conf. – Special Track on Semantic Web*, pp. 425–429, 2002.
- [222] M. Stal. Web services: beyond component-based computing. *Communications of the ACM*, 45(10):71–76, 2002.
- [223] L. Stein. Creating a bioinformatics nation. *Nature*, 417(6885):119–120, 2002.
- [224] E. Stolte, C. von Praun, G. Alonso e T. Gross. Scientific data repositories – designing for a moving target. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 349–360. ACM Press, 2003.
- [225] M. Stonebraker. Implementation of integrity constraints and views by query modification. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 65–78. ACM Press, 1975.

- [226] Y. Sure, J. Angele e S. Staab. OntoEdit: Guiding ontology development by methodology and inferencing. Em *Intl. Conf. on Cooperative Information Systems (CoopIS)*, volume 2519 of *LNCS*, pp. 1205–1222. Springer-Verlag, 2002.
- [227] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer e D. Wenke. Ontoedit: Collaborative ontology development for the semantic web. Em *Proc. Intl. Semantic Web Conf. (ISWC)*, volume 2342 of *LNCS*, pp. 348–363. Springer-Verlag, 2002.
- [228] N. Tryfona e C. S. Jensen. Conceptual data modeling for spatiotemporal applications. *GeoInformatica*, 3(3):245–268, 1999.
- [229] A. Tsalgaidou e T. Pilioura. An overview of standards and related technologies in web services. *Distributed and Parallel Databases*, 12:135–162, 2002.
- [230] Universal description, discovery and integration of web services (UDDI). <http://www.uddi.org/> as of October 2003.
- [231] J. D. Ullman. Information integration using logical views. Em *Proc. ICDT Conf.*, volume 1186 of *LNCS*, pp. 19–40. Springer-Verlag, 1997.
- [232] Naming and addressing (URIs, URLs, ... <http://www.w3.org/Addressing/> (as of October 2003).
- [233] M. Uschold e M. Gruninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.
- [234] W. M. P. van der Aalst. Don't go with the flow: Web services composition standards exposed. *IEEE Intelligent Systems*, 18(1):72–85, 2003.
- [235] W. M. P. van der Aalst, A. Hirsenschall e H. M. W. Verbeek. An alternative way to analyze workflow graphs. Em *Advanced Information Systems Engineering (CAiSE)*, volume 2348 of *LNCS*, pp. 535–552. Springer-Verlag, 2002.
- [236] L. R. Venâncio, R. Fileto e C. B. Medeiros. Aplicando ontologias de objetos geográficos para facilitar a navegação em GIS. Em *Simpósio Brasileiro de Geoinformática (GEOINFO)*, 2003.
- [237] V. Ventrone e S. Heiler. Semantic heterogeneity as a result of domain evolution. *SIGMOD Record*, 20(4):16–20, 1991.
- [238] A. Voisard, C. B. Medeiros e G. Jomier. Database support for cooperative work documentation. Em *COOP*, 2000.

- [239] J. Wainer, G. Vossen, M. Weske e C. B. Medeiros. Scientific workflow systems. Em *NSF Workshop on Workflow and Process Automation: State of the Art and Future Directions*, 1995.
- [240] J. Wang e F. H. Lochovsky. Data extraction and label assignment for web databases. Em *Proc. Intl. Conf. on World Wide Web (WWW)*, pp. 187–196. ACM Press, 2003.
- [241] M. Weske, G. Vossen, C. B. Medeiros e F. Pires. Workflow management in geoprocessing applications. Em *ACM-GIS*, pp. 88–93, 1998.
- [242] J. Widom. Data management for XML: Research directions. *IEEE Data Engineering Bulletin*, 22(3):44–52, 1999.
- [243] G. Wiederhold. Views, objects, and databases. *IEEE Computer*, 19(12):37–44, 1986.
- [244] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(1):38–49, 1992.
- [245] G. Wiederhold. An algebra for ontology composition. Em *Monterey Workshop on Formal Methods*, pp. 56–61, 1994.
- [246] G. Wiederhold. Interoperation, mediation, and ontologies. Em *Intl. Symp. on Fifth Generation Computer Systems (FGCS) – Workshop on Heterogeneous Cooperative Knowledge-Bases*, pp. 33–48, 1994.
- [247] G. Wiederhold e J. Jannik. Composing diverse ontologies. Report, Stanford University, 1998.
- [248] K. Wilkinson, C. Sayers e H. Kuno. Efficient RDF storage and retrieval in jena2. Em *Proc. Intl. Workshop on Semantic Web and Databases*, pp. 131–150. Humboldt-Universität, 2003.
- [249] J. Williams. The web services debate – J2EE versus .NET. *Communications of the ACM*, 46(6):59–63, 2003.
- [250] P. Wohed, W. M. P. van der Aalst, M. Dumas e A. H. M. ter Hofstede. Pattern based analysis of BPEL4WS. Report FIT-TR-2002-04, Queensland University of Technology, Queensland, Australia, 2002.
- [251] A. G. Woodruff e M. Stonebraker. Supporting fine-grained data lineage in a database visualization environment. Em *Proc. Intl. Conf. on Data Engineering (ICDE)*, pp. 91–102. IEEE Computer Society, 1997.

- [252] M. F. Worboys e S. M. Deen. Semantic heterogeneity in distributed geographic databases. *SIGMOD Record*, 20(4):30–34, 1991.
- [253] The W3C web services activity. <http://www.w3.org/2002/ws/> (as of October 2003).
- [254] Web services description language (WSDL) 1.1. <http://www.w3.org/TR/wsdl> (as of October 2003).
- [255] Web Services Flow Language (WSFL 1.0). <http://www.ibm.com/software/solutions/webservices/pdf/WSFL.pdf> (as of October 2003).
- [256] W3C's Extensible Markup Language (XML) 1.0 (second edition). <http://www.w3.org/TR/REC-xml> (as of October 2003).
- [257] XML Schema part 0: Primer. <http://www.w3.org/XML> (as of October 2003).
- [258] W3C's XML Query Activity. <http://www.w3.org/XML/Query> (as of October 2003).
- [259] E. I. Y, M. Livny, A. Ailamaki, A. Ranganathan, A. Therber, M. Yuin, M. Anderson e J. Norman. Managing soil science experiments using ZOO. Em *Proc. Conf. on Statistical and Scientific Database Management*, pp. 121–124. IEEE, 1997.
- [260] N. Zhong, J. Liu e Y. Y. (eds.). Special Issue – In Search of the Wisdom Web. *IEEE Computer*, 35(11):27–76, 2002.
- [261] Y. Zhuge, H. Garcia-Molina, J. Hammer e J. Widom. View maintenance in a warehousing environment. Em *Proc. Intl. Conf. on Management of Data (SIGMOD)*, pp. 316–327. ACM Press, 1995.
- [262] Zoning coffee in Brazil. <http://orion.cpa.unicamp.br/cafe> (in Portuguese – as of October 2003).

Annex I

Formal Definitions and Properties for POESIA

This annex presents all the formal definitions and proofs of theorems that enable the POESIA ontology-based approach for resources discovery and composition. It is organized as follows:

- Section I.1 formally describes the structure of a POESIA ontology and the basic concepts related with such an ontology, such as paths and semantic encompassing between ontology terms.
- Section I.2 defines ontological coverages and the semantic relationships of encompassing and equivalence between coverages.
- Finally, Section I.3 presents the concepts of activity pattern (which can refer to simple or composite services), specialization and aggregation of activity patterns, process framework and specific processes for particular needs.

The concepts of Section I.3 are based on the notion of ontological coverages and semantic relationships between these coverages. The rules used to define process frameworks and specific processes ensure the semantic consistency of the compositions of services, according to their associated ontological coverages.

I.1 POESIA Ontologies

Definition I.1 states that for a set of words Ω to be considered consistent, at most one semantic relationship of the set Υ (which includes *synonym*, *IS_A*, and *PART_OF*, with their inverse relationships) occurs between any pair of words.

The following definitions and theorems show that any set of semantically consistent words Ω can be represented as a graph G_Ω , whose nodes correspond to maximal sets of words that are synonym of each other, and whose edges are semantic relationships between sets of synonyms. The graph G_Ω is called an arrangement of semantically consistent words (Definition I.3) if it is acyclic and connected.

Definition I.1 (Set of Semantically Consistent Words) *Let Ω be a finite set of words for a universe of discourse U . Ω is a set of semantically consistent words with respect to the set of semantic relationships*

$$\Upsilon = \{synonym, hypernym, hyponym, holonym, meronym\}$$

iff:

$$\forall \theta \in \Upsilon : w \theta w' \Rightarrow \nexists \varphi \in \Upsilon \text{ such that } \varphi \neq \theta \wedge w \varphi w'$$

i.e., for any pair of words $w, w' \in \Omega$ at most one semantic relationship $\theta \in \Upsilon$ leads from w to w' .

Definition I.2 (Maximal Sets of Synonymous) *Given a set of semantically consistent words Ω with respect to the set of semantic relationships*

$$\Upsilon = \{synonym, hypernym, hyponym, holonym, meronym\}$$

MaxSyn $\subseteq \Omega$ is a maximal set of synonymous from Ω iff the following conditions are satisfied:

1. *MaxSyn $\neq \emptyset$*
2. *$\forall w, w' \in MaxSyn : w \text{ synonym } w'$*
3. *$\forall w, w' \in \Omega : (w \in MaxSyn \wedge w' \notin MaxSyn) \Rightarrow \neg(w \text{ synonym } w')$*

Algorithm 1 – Generate the collection of maximal sets

Given a set of semantically consistent words Ω for the universe of discourse U , apply the following sequence of steps to partition Ω in a collection of maximal sets of synonymous.

1. *Let $\Omega = \{w_1, \dots, w_n\}$.*

Build the list of unitary sets of words $Parts = (\{w_1\}, \dots, \{w_n\})$.

2. *If $\exists \mathfrak{R}, \mathfrak{R}' \in Parts$ such that $\forall w \in \mathfrak{R}, w' \in \mathfrak{R}' : w$ synonym w' then remove \mathfrak{R} and \mathfrak{R}' from $Parts$ and insert $(\mathfrak{R} \cup \mathfrak{R}')$ in $Parts$.*

3. *Repeat step 2 until*

$$\neg(\exists \mathfrak{R}, \mathfrak{R}' \in Parts, w \in \mathfrak{R}, w' \in \mathfrak{R}' \text{ such that } w \text{ synonym } w')$$

When the execution of algorithm 1 stops, the partition of Ω in the collection of maximal sets of synonymous is available in the list $Parts$.

Theorem I.1 *Let Ω be an arbitrary set of semantically consistent words with respect to the set of semantic relationships*

$$\Upsilon = \{synonym, hypernym, hyponym, holonym, meronym\}$$

The set Ω can be partitioned in a collection of maximal sets of synonymous.

Proof:

Algorithm 1 generates the partition of Ω in a list of maximal sets of synonymous denominated Parts. The correctness of algorithm 1 can be verified in two phases.

Phase 1: *Algorithm 1 partitions Ω in a collection Parts of sets of synonymous.*

We prove it by induction on the number of repetitions of step 2.

Base: *After executing step 1 (and before the first execution of step 2 of algorithm 1, $Parts = (\{w_1\}, \dots, \{w_n\})$. Therefore:*

1. $\bigcup_{\mathfrak{R} \in Parts} \mathfrak{R} = \Omega$
2. $\forall \mathfrak{R} \in Parts : \mathfrak{R} \neq \emptyset$
3. $\forall \mathfrak{R} \in Parts; w, w' \in \mathfrak{R} : w \text{ synonym } w'$

Step: *Each execution of step 2 of algorithm 1 preserves conditions 1, 2 and 3 of the base, because step 2 can only replace a pair of sets of synonyms \mathfrak{R} and \mathfrak{R}' from Parts with the union $\mathfrak{R} \cup \mathfrak{R}'$ when $\forall w \in \mathfrak{R}, w' \in \mathfrak{R}' : w \text{ synonym } w'$.*

Phase 2: *Each set of synonymous present in Parts by the end of the execution of algorithm 1 is maximal.*

It is guaranteed by the condition to stop the loop in step 3:

$$\neg(\exists \mathfrak{R}, \mathfrak{R}' \in Parts, w \in \mathfrak{R}, w' \in \mathfrak{R}' \text{ such that } w \text{ synonym } w')$$

Theorem I.2 *Given a set of semantically consistent words Ω with respect to the set of semantic relationships*

$$\Upsilon = \{\text{synonym, hypernym, hyponym, holonym, meronym}\}$$

There exists a directed graph $G_\Omega(V_\Omega, E_\Omega)$ that organizes the words of Ω according to the semantic relationships in Υ .

Proof: *Let $Parts$ be the collection of sets of synonymous obtained by partitioning Ω with algorithm 1. $G_\Omega(V_\Omega, E_\Omega)$ has the following constitution:*

- V_Ω is the set of vertices of G_Ω
- $\mathfrak{R} \in V_\Omega \Leftrightarrow \mathfrak{R} \in Parts$
- E_Ω is the set of directed edges of G_Ω
- $(\mathfrak{R}, \mathfrak{R}') \in E_\Omega \Leftrightarrow (\forall w \in \mathfrak{R}, w' \in \mathfrak{R}' : w \text{ hypernym } w' \vee w \text{ holonym } w')$

Definition I.3 (Arrangement of Semantically Consistent Words) *Let Ω be a set of semantically consistent words with respect to the set of semantic relationships*

$$\Upsilon = \{\text{synonym, hypernym, hyponym, holonym, meronym}\}$$

The arrangement of semantically consistent words of Ω (or simply the arrangement of words from Ω) is a graph $G_\Omega(V_\Omega, E_\Omega)$, with the set of vertices V_Ω and set of edges E_Ω , such that the following conditions are verified:

1. $\forall w \in \Omega : \exists \mathfrak{R} \in V_\Omega$ such that $w \in \mathfrak{R}$
2. $\forall \mathfrak{R} \in V_\Omega; w, w' \in \mathfrak{R} : w \text{ synonym } w'$
3. $\forall \mathfrak{R} \in V_\Omega, w \in \mathfrak{R}, w' \in \Omega : w' \notin \mathfrak{R} \Rightarrow \neg(w \text{ synonym } w')$
4. $(\mathfrak{R}, \mathfrak{R}') \in E_\Omega \Leftrightarrow (\forall w \in \mathfrak{R}, w' \in \mathfrak{R}' : w \text{ hypernym } w' \vee w \text{ holonym } w')$
5. G_Ω is acyclic
6. G_Ω is connected

Definitions I.4 to I.6 describe the encompass relationship between terms of a POESIA ontology and Theorem I.3 shows that this relationship is transitive.

Definition I.4 (Path) *Let Ω be a set of semantically consistent words with respect to the set of semantic relationships*

$$\Upsilon = \{synonym, hypernym, hyponym, holonym, meronym\}$$

Let $G_\Omega(V_\Omega, E_\Omega)$ be an arrangement of semantically consistent words for the words of Ω , where V_Ω is the set of vertices of G_Ω and E_Ω the set of edges of G_Ω .

A path from vertex $\mathfrak{R}_1 \in V_\Omega$ to vertex $\mathfrak{R}_n \in V_\Omega$ is a sequence of directed edges of E_Ω leading from \mathfrak{R}_1 to \mathfrak{R}_n , with the form:

$$(\mathfrak{R}_1, \mathfrak{R}_2), \dots, (\mathfrak{R}_{n-1}, \mathfrak{R}_n)$$

where $(\mathfrak{R}_i, \mathfrak{R}_{i+1}) \in E_\Omega (1 \leq i < n)$.

Definition I.5 (Vertices Reachability) *If there is a path from vertex \mathfrak{R}_1 to vertex \mathfrak{R}_n in the arrangement of semantically consistent words G_Ω , then we say that \mathfrak{R}_n is reachable from \mathfrak{R}_1 in G_Ω , denoted by*

$$\mathfrak{R}_1 \rightsquigarrow \mathfrak{R}_n$$

Definition I.6 (Semantic Encompassing) *Let Ω be a set of semantically consistent words with respect to the set of semantic relationships*

$$\Upsilon = \{synonym, hypernym, hyponym, holonym, meronym\}$$

Let $G_\Omega(V_\Omega, E_\Omega)$ be the arrangement of semantically consistent words for the words of Ω , and $w, w' \in \Omega$ be two arbitrary words such that:

$$\mathfrak{R}(w) \in V_\Omega \wedge w \in \mathfrak{R}(w) \quad \wedge \quad \mathfrak{R}(w') \in V_\Omega \wedge w' \in \mathfrak{R}(w')$$

The word w encompasses the word w' , denoted by $w \prec w'$, iff:

$$\mathfrak{R}(w) = \mathfrak{R}(w') \wedge \mathfrak{R}(w) \rightsquigarrow \mathfrak{R}(w')$$

Theorem I.3 *Let Ω be a set of semantically consistent words with respect to the set of semantic relationships*

$$\Upsilon = \{\text{synonym, hypernym, hyponym, holonym, meronym}\}$$

The encompass relationship among the words of Ω is transitive.

Proof:

Let $w, w', w'' \in \Omega$ such that

$$w \prec w' \wedge w' \prec w'' \tag{1}$$

Let $G_\Omega(V_\Omega, E_\Omega)$ be the arrangement of semantically consistent words for Ω .

Let $\mathfrak{R}(w), \mathfrak{R}(w'), \mathfrak{R}(w'') \in V_\Omega$ such that:

$$w \in \mathfrak{R}(w) \wedge w' \in \mathfrak{R}(w') \wedge w'' \in \mathfrak{R}(w'')$$

Then, from (1) and definition I.6:

$$\begin{aligned} w \prec w' \wedge w' \prec w'' &\equiv \\ &\equiv (\mathfrak{R}(w) = \mathfrak{R}(w') \vee \mathfrak{R}(w) \rightsquigarrow \mathfrak{R}(w')) \wedge \\ &\quad (\mathfrak{R}(w') = \mathfrak{R}(w'') \vee \mathfrak{R}(w') \rightsquigarrow \mathfrak{R}(w'')) \equiv \\ &\equiv (\mathfrak{R}(w) = \mathfrak{R}(w') \wedge \mathfrak{R}(w') = \mathfrak{R}(w'')) \vee \\ &\quad (\mathfrak{R}(w) = \mathfrak{R}(w') \wedge \mathfrak{R}(w') \rightsquigarrow \mathfrak{R}(w'')) \vee \\ &\quad (\mathfrak{R}(w) \rightsquigarrow \mathfrak{R}(w') \wedge \mathfrak{R}(w') = \mathfrak{R}(w'')) \vee \\ &\quad (\mathfrak{R}(w) \rightsquigarrow \mathfrak{R}(w') \wedge \mathfrak{R}(w') \rightsquigarrow \mathfrak{R}(w'')) \Rightarrow \\ &\Rightarrow \mathfrak{R}(w) = \mathfrak{R}(w'') \vee \mathfrak{R}(w) \rightsquigarrow \mathfrak{R}(w'') \equiv w \prec w'' \end{aligned}$$

Definition I.7 describes a term as a word that is an specialization or an instance of another word, called a concept. Terms are organized in arrangements of semantically consistent words (as described in Definition I.3), called arrangements of semantically consistent terms. Each arrangement of terms has an associated arrangement of semantically consistent concepts, to qualify its terms. Definition I.8 describes a domain specific ontology as a collection of arrangements of semantically consistent terms, with the respective arrangements of semantically consistent concepts. Each pair of arrangements of words refer to a dimension of the ontology.

Definition I.7 (Term) *Let Γ and Λ be two sets of semantically consistent words with respect to the set of semantic relationships*

$$\Upsilon = \{synonym, hypernym, hyponym, holonym, meronym\}$$

Consider that Γ and Λ satisfy the following conditions:

1. $\forall u \in \Gamma : u$ refers to a concept
2. $\forall w \in \Lambda : \exists u \in \Gamma$ such that w hyponym $u \vee w$ instance u
3. $\forall w, w' \in \Lambda : \exists u, u' \in \Gamma$ such that :
 - (a) w hyponym $u \vee w$ instance u
 - (b) w' hyponym $u' \vee w'$ instance u'
 - (c) $\forall \varphi \in \Upsilon : w \varphi w' \Leftrightarrow u \varphi u'$

A term is a word $t \in \Lambda$. Term t can be denoted by $q(t)$, where q , called the qualifier of t , satisfy the condition:

$$q \in \Gamma \wedge (t \text{ hyponym } q \vee t \text{ instance } q)$$

Definition I.8 (Domain Specific Ontology) *A domain specific ontology Σ is a collection of arrangements of semantically consistent terms $\{G_{\Lambda}^1, \dots, G_{\Lambda}^n\}$ ($n \geq 1$), where each arrangement G_{Λ}^i ($1 \leq i \leq n$) characterizes dimension i of the application domain.*

Definition I.9 (Specification of a Path in an Arrangement of Terms) Let $G_\Lambda(V_\Lambda, E_\Lambda)$ be an arrangement of semantically consistent terms related to some arbitrary dimension of a domain specific ontology Σ , where V_Λ is the set of vertices of G_Λ and E_Λ is the set of edges of G_Λ .

Let $G_\Gamma(V_\Gamma, E_\Gamma)$ be the arrangement of the semantically consistent concepts used as the qualifiers of the terms organized in G_Λ , where V_Γ is the set of vertices of G_Γ and E_Γ the set of edges of G_Γ .

An specification of a path in G_Λ is a sequence of terms of the form:

$$T = q_1(t_1).q_2(t_2). \cdots .q_n(t_n)$$

satisfying the following conditions:

1. $\exists \mathfrak{R}_\Gamma \in V_\Gamma$ such that $q_j \in \mathfrak{R}_\Gamma$ ($n \geq 1; 1 \leq j \leq n$)
2. $\exists \mathfrak{R}_\Lambda \in V_\Lambda$ such that $q_j \in \mathfrak{R}_\Lambda$ ($n \geq 1; 1 \leq j < n$)
3. $(t_j, t_{j+1}) \in E_\Lambda$ ($n \geq 1; 1 \leq j < n$)

Definition I.10 (Unambiguous Reference to a Term) Let Σ be a domain specific ontology and

$$Str(T) = "q_1(t_1).q_2(t_2). \cdots .q_n(t_n)"$$

be the string correspondent to the specification of path

$$T = q_1(t_1).q_2(t_2). \cdots .q_n(t_n)$$

leading to the term $q_n(t_n)$ in the arrangement of semantically consistent terms for some dimension of the ontology Σ .

The path $Str(T)$ is an unambiguous reference to the term $q_n(t_n)$ iff $Str(T)$ is unique among all the strings $Str(T')$ produced from any path

$$T' = q'_1(t'_1).q'_2(t'_2). \cdots .q'_n(t'_n)$$

in any arrangement of semantically consistent terms in Σ , by using the same method as that used to produce $Str(T)$ from T .

I.2 Ontological coverages and their relationships

The definitions and theorems in this section can be summarized as follows. Definition I.11 says that an ontological coverage is a tuple of terms from a POESIA ontology. Definition I.12 states that the universal coverage is the empty tuple. Definition I.13 defines ontological coverages encompassing. Theorem I.4 shows that the universal coverage encompass any other coverage, and Theorem I.5 shows that the encompass relationship among ontological coverages is transitive. Finally, equivalent ontological coverages, as stated by Definition I.14, reciprocally encompass each other.

Definition I.11 (Ontological Coverage) *Let*

$$\Sigma = \{G_{\Lambda}^1(E_{\Lambda}^1, V_{\Lambda}^1), \dots, G_{\Lambda}^n(V_{\Lambda}^n, E_{\Lambda}^n)\} \quad (n \geq 1)$$

be a domain specific ontology, where $G_{\Lambda}^j(V_{\Lambda}^j, E_{\Lambda}^j)$ ($1 \leq j \leq n$) is the arrangement of semantically consistent terms for dimension j of Σ , V_{Λ}^j is the set of vertices of G_{Λ}^j and E_{Λ}^j is the set of edges of G_{Λ}^j .

An ontological coverage taken from Σ is an m -tuple

$$[t_1, \dots, t_m] \quad (m \geq 0)$$

satisfying the condition:

$$\forall t_i \in [t_1, \dots, t_m] : \exists G_{\Lambda}^j(V_{\Lambda}^j, E_{\Lambda}^j) \in \Sigma; \mathfrak{R} \in V_{\Lambda}^j \text{ such that } t_i \in \mathfrak{R}$$

Definition I.12 (Universal Coverage) *The universal coverage (denoted by ∞) is the empty tuple. ∞ doesn't restrict the application scope in any dimension of the ontology for the application domain it refers to.*

Definition I.13 (Ontological Coverages Encompassing) *Let*

$$C = [t_1, \dots, t_n] \text{ and } C' = [t'_1, \dots, t'_m] \quad (n, m \geq 1)$$

be two ontological coverages taken from the same domain specific ontology Σ .

Let Λ be the set of semantically consistent terms organized in the arrangement of terms G_Λ for an arbitrary dimension of Σ .

We say that C encompasses C' (denoted by $C \prec C'$) iff:

$$\forall t_i \in C : \exists t'_j \in C' \text{ such that } t_i \in \Lambda \wedge t'_j \in \Lambda \wedge t_i \prec t'_j$$

Theorem I.4 *The universal coverage encompasses any ontological coverage.*

Proof:

It follows directly from definitions I.12 and I.13.

From definition I.12:

$$\exists t \in \infty$$

Then, for any ontological coverage C' taken from an arbitrary domain specific ontology Σ .

$$\exists t_i \in \infty \text{ such that } \exists t'_j \in C' \text{ such that } t_i \prec t'_j \quad (2)$$

Therefore, according to (2) and definition I.13 the universal coverage (∞) encompasses any ontological coverage, including ∞ itself.

Theorem I.5 *The encompass relationship among ontological coverages defined with respect to a given domain specific ontology is transitive.*

Proof:

Let C_1 , C_2 and C_3 be arbitrary coverages such that

$$C_1 \prec C_2 \wedge C_2 \prec C_3 \quad (3)$$

Let $t_i \in C_i$ ($1 \leq i \leq 3$) be an arbitrary term of the coverage C_i .

From equation (3) we have the following deductions:

$$C_1 \prec C_2 \Rightarrow \forall t_1 \in C_1 : \exists t_2 \in C_2 \text{ such that } t_1 \prec t_2 \quad (4)$$

$$C_2 \prec C_3 \Rightarrow \forall t_2 \in C_2 : \exists t_3 \in C_3 \text{ such that } t_2 \prec t_3 \quad (5)$$

Then, from equations (4) and (5), and the transitiveness of the encompass relationship between words or terms (theorem I.3), we can deduce that

$$\forall t_1 \in C_1 : \exists t_3 \in C_3 \text{ such that } t_1 \prec t_3 \quad (6)$$

Now, suppose that

$$\neg(C_1 \prec C_3) \quad (7)$$

We can deduce from (7) that

$$\exists t_1 \in C_1 \text{ such that } \nexists t_3 \in C_3 \text{ satisfying } t_1 \prec t_3 \quad (8)$$

Which is a contradiction with (6) derived from (3).

Definition I.14 (Ontological Coverages Equivalence) Given two ontological coverages, C and C' , we say that C is equivalent to C' , denoted by $C \equiv C'$, iff:

$$C \prec C' \wedge C' \prec C$$

I.3 Services Composition in POESIA

This section presents the formal definitions related with the composition of services, that also appear in Chapter 3, for summarization purposes. Definition I.15 describes a simple or composite service as an activity pattern, with a name, an associated ontological coverage, input and output ports, and a task definition. Definitions I.16 to I.19 describe the rules for the semantically consistent composition of activity patterns, in terms of the interconnection of their input and output ports, and the semantic relationships among their associated ontological coverages. Aggregation and specialization are the basic operations for composing activity patterns in processes frameworks and to adapt these frameworks (by taking appropriate specialized versions of their activity patterns), when building processes for specific needs. For extensive descriptions of these definitions, see Section 3.4.

Definition I.15 An activity pattern α is a five-tuple:

$$(NAME, COVER, IN, OUT, TASK)$$

where:

- NAME* is the string used as the name of α
- COVER* is the ontological coverage of α
i.e., expresses its utilization scope
- IN* is the list of input parameters of α
- OUT* is the list of output parameters of α
- TASK* describes the processing chores that α does

Definition I.16 Activity pattern α is an **aggregation** of the activity patterns β_1, \dots, β_n ($n \geq 1$) if the following conditions are verified (let $1 \leq i, j \leq n; i \neq j$ for each condition):

1. $\forall \beta_i : NAME(\alpha) \neq NAME(\beta_i) \vee COVER(\alpha) \neq COVER(\beta_i)$
2. $\forall \beta_i, \beta_j : NAME(\beta_i) \neq NAME(\beta_j) \vee COVER(\beta_i) \neq COVER(\beta_j)$
3. $\forall \beta_i : COVER(\alpha) \models COVER(\beta_i) \vee COVER(\beta_i) \models COVER(\alpha)$
4. $\forall p \in IN(\alpha) : \exists \beta_i$ such that $p \in IN(\beta_i)$
5. $\forall p \in OUT(\alpha) : \exists \beta_i$ such that $p \in OUT(\beta_i)$
6. $\forall \beta_i, p' \in IN(\beta_i) : p' \in IN(\alpha) \vee (\exists \beta_j$ such that $p' \in OUT(\beta_j))$
7. $\forall \beta_i, p' \in OUT(\beta_i) : p' \in OUT(\alpha) \vee (\exists \beta_j$ such that $p' \in IN(\beta_j))$

Definition I.17 Activity pattern β is a **specialization** of the activity pattern α (conversely α is a generalization of β) if the following conditions are verified:

1. $NAME(\alpha) \neq NAME(\beta) \vee COVER(\alpha) \neq COVER(\beta)$
2. $COVER(\alpha) \models COVER(\beta)$
3. $\forall p \in IN(\alpha) : \exists p' \in IN(\beta)$ such that $p \vdash p'$
4. $\forall p \in OUT(\alpha) : \exists p' \in OUT(\beta)$ such that $p \vdash p'$

Definition I.18 A **process framework** is a directed graph $\Phi(V_\Phi, E_\Phi)$ satisfying the following conditions:

1. V_Φ is the set of vertices of Φ
2. E_Φ is the set of edges of Φ
3. $\forall v \in V_\Phi : v$ is an activity pattern
4. $(v, v') \in E_\Phi \Leftrightarrow v' \text{ constituent } v \vee v' \text{ specialization } v$
5. Φ is acyclic
6. Φ is connected

Definition I.19 A **process specification** $\Pi(V_\Pi, E_\Pi)$ associated with a utilization scope expressed by an ontological coverage C is a subgraph of a process framework satisfying the properties:

1. $\forall (v, v') \in E_\Pi : v' \text{ constituent } v$
2. $\forall v \in V_\Pi :$
 $(\nexists v' \in V_\Pi \text{ such that } (v, v') \in E_\Pi) \Rightarrow v \text{ is atomic}$
3. $\forall v \in V_\Pi : COVER(v) \models C$

Annex II

POESIA Architecture and Implementation Issues

An information system supporting the POESIA approach has three categories of modules, communicating through the Internet:

Ontology services encapsulate ontologies and allow several applications to use these ontologies. An ontology server provides access and adaptation means for several ontologies in different domains. The sharing of ontologies among applications enables cooperative processes, using resources distributed across the Web.

Application services support the definition, composition and execution of services, using domain ontologies provided by ontology services. Composite services (i.e., cooperative processes) are handled as workflows running on the Web. A workflow is associated with a unique ontology. An ontology, on the other hand, can be associated with several workflows. A workflow and each one of its component services and data flows are associated with ontological coverages that refer to terms of the same ontology. The composite services of a given workflow are also handled as workflows, and are associated with the same ontology as their parent.

Service brokers service brokers provide facilities to search for services available on the Web to fulfill specific needs, which are expressed by service descriptions (e.g., denoted in DAML-Services [14]) and ontological coverages. An ontology broker has the capability to adapt a process framework to a particular need, by choosing the versions of the component services compatible with the intended ontological coverage of the desired process.

Figure 1 illustrates the role of ontology services and application services on supporting a POESIA application for the agriculture domain. Ontology Server 1 provides three ontologies for different but overlapping domains. The Supply Chain Ontology is a subset

of the Logistics Ontology. These ontologies refer to the production and distribution of goods to satisfy any kind of need (e.g., food, energy, water). The Agriculture Ontology, in turn, has some intersection with the specialization of the former ontologies to the agriculture realm. Each of these three ontologies is referred to by several workflows, for the respective application domains. A given workflow, on the other hand, can only be associated with one ontology. The association of the workflow with the ontology is fundamental to enable the POESIA facilities for managing the resources necessary to execute the workflow.

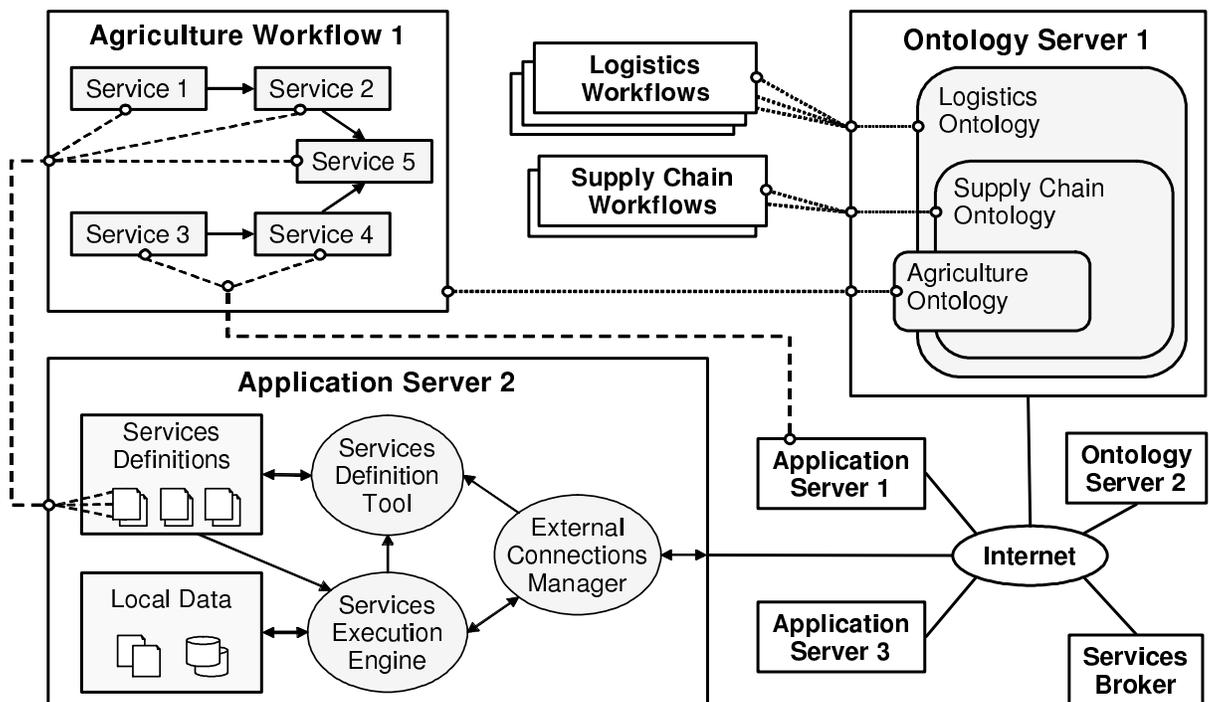


Figure 1: POESIA System Architecture

The Agriculture Workflow 1 is associated with the Agriculture Ontology. Figure 1 presents some details of this workflow on the top left corner. This workflow involves the cooperation of 5 services. The data connections among these services are indicated by arrows. Thus, according to the figure, the output of Service 1 is inputted in Service 2, the output of Service 2 is inputted in Service 5, and so on. The execution of these services are supported by different application servers. Application Server 2 is responsible for the definition and execution of Service 1, Service 2, Service 5 and the coordination of the execution of all component services of Agriculture Workflow 1. On the other hand, Service 4 and Service 5 are individually defined and executed in Application Server 1.

The internal architecture of each application server includes: (i) a *Service Definition Tool*, for building the definition of the services that are provided by that application server; (ii) a *Services Execution Engine*, for executing the local services and managing the local data sets, according with the definitions of the services; and (iii) an *External Connections Manager*, to manage the connections of local services with external services (i.e., supported by other application servers) and ontology servers.

The *OntoCover* Java library, implemented in this thesis (see Chapter 5), enables the construction and utilization of ontology views to manage data and services in POESIA applications. *OntoCover* will be useful to implement ontology servers, application servers and/or service brokers. The exact points where the ontology views will be built (the ontology servers, or the application servers and service brokers), depends on more detailed systems design, and further experiments to determine the most appropriate solutions in practice. *OntoCover*'s facilities for browsing ontology views and managing ontological coverages defined over these views are certainly useful for implementing application servers. In this thesis, we incorporated *OntoCover* in *WOODSS* (Workflow-based Decision Support System), a tool that applies scientific workflows to process geographic data for decision making purposes [214], in order to evaluate the facilities provided by *OntoCover* in a concrete workflow system. The association of ontological coverages with workflow activities and data in *WOODSS* provides a testbed for the use of POESIA semantic descriptions to organize the resources required by cooperative processes involving geographic data. However, *WOODSS* only supports the definition and execution of workflows in a centralized environment, i.e., a unique application server.

The design and implementation of service brokers, the full implementation of ontology servers and application servers, and the complete validation of the POESIA approach, in agriculture and other domains, are all left as future work. Other challenges include the interoperability of workflows associated with different ontologies and the development or incorporation of mechanisms for synchronizing cooperative Web services in a POESIA system.