

UNIVERSIDADE ESTADUAL PAULISTA

“Júlio de Mesquita Filho”

Pós-Graduação em Ciência da Computação

Jorge Marques Prates

Múltiplas visões coordenadas: uma técnica de
coordenação com o apoio de ontologias

UNESP
Agosto/2014

Jorge Marques Prates

Múltiplas visões coordenadas: uma técnica de
coordenação com o apoio de ontologias

Orientador: Prof. Dr. Rogério Eduardo Garcia

Dissertação apresentada ao Instituto de Biociências, Letras e Ciências Exatas (IBILCE – Unesp – São José do Rio Preto) como parte dos requisitos para a obtenção do título de mestre em Ciência da Computação.

UNESP
Agosto/2014

Jorge Marques Prates

Múltiplas visões coordenadas: uma técnica de coordenação com o apoio de ontologias

Dissertação apresentada ao Instituto de Biociências, Letras e Ciências Exatas (IBILCE – Unesp – São José do Rio Preto) como parte dos requisitos para a obtenção do título de mestre em Ciência da Computação.

Membros da banca:

Prof. Dr. Rogério Eduardo Garcia (Orientador)
FCT – UNESP – Presidente Prudente

Profa. Dra. Maria Cristina Ferreira de Oliveira
ICMC – USP – São Carlos

Prof. Dr. Celso Olivete Júnior
FCT – UNESP – Presidente Prudente

Dedico este trabalho à Elza.

Prates, Jorge Marques

Múltiplas visões coordenadas: uma técnica de coordenação com o apoio de ontologias / Jorge Marques Prates – São José do Rio Preto, 2014.

73 p. :il

Orientador: Rogério Eduardo Garcia

Dissertação (Mestrado) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Instituto de Biociências, Letras e Ciências Exatas.

1. Computação – Matemática. 2. Web semântica. 3. Ontologias (Recuperação da informação) 4. Semântica - Processamento de dados. 5. Visualização de software. I. Garcia, Rogério Eduardo.

II. Universidade Estadual Paulista “Júlio de Mesquita Filho”. Instituto de Biociências, Letras e Ciências Exatas. III. Múltiplas visões coordenadas: uma técnica de coordenação com o apoio de ontologias.

CDU – 518.721

Agradecimentos

Ao Professor Rogério pelo sofrimento, ensinamento e orientação.

Ao Professor Danilo pela paciência, auxílio e coorientação.

À Lilian pela grande ajuda durante as etapas.

À Ana pelo apoio psicológico.

À Unesp, como instituição, pela formação ao longo desta jornada de 12 anos.

À CAPES pelo apoio financeiro.

“Avada Kedavra!”

Resumo

A exploração de múltiplas visões de conjuntos de dados pode auxiliar no processo de descoberta de associações ocultas entre os elementos desses conjuntos. O emprego de uma técnica de coordenação permite associar esses elementos em visões distintas. Normalmente, técnicas de coordenação que mapeiam os elementos apenas considerando os atributos dos dados limitam a descoberta de associações entre os elementos visuais. Neste trabalho é apresentada a técnica *Semantic Coordination*, um mecanismo de coordenação que utiliza ontologias para associar elementos de dados com base nos mapeamentos fornecidos pela semântica. Com isso, a visualização exploratória pode se beneficiar da representação semântica para criar os mapeamentos, pois são estabelecidas associações entre os elementos visuais além das existentes pela relação dos dados. Também, são apresentados e analisados os resultados de dois estudos de caso, aplicando a *Semantic Coordination* e técnicas tradicionais na exploração de coleções textuais. Após a análise, os resultados obtidos mostram que o mecanismo de coordenação proposto estende o processo de exploração de múltiplas visões de conjuntos de dados, em comparação a técnicas de coordenação tradicionais.

Palavras-chave: múltiplas visões coordenadas, técnicas de coordenação, ontologias.

Abstract

Exploring multiple views of data sets can provide assistance on discovering unforeseen associations among their elements. Employing a coordination technique creates associations among elements across related views. Usually, the coordination process uses a mapping among data elements, that are constrained by using data attributes, which is a constraint to user exploration process (only linked data can be usually associated). In this work it is presented *Semantic Coordination*, a mechanism that applies ontologies to link data elements based on mappings provided by semantic. Representing the underlying data into ontology, the exploratory visualization can be benefited from a semantic representation to create the mappings, what might be helpful to establish associations on multiple views. Also, it is shown the results of two case studies applying the *Semantic Coordination* and traditional techniques on exploring text collections. The analysis of the obtained results, have shown the proposed coordination mechanism improves the exploration using multiple views, in comparison with traditional techniques.

Keywords: coordinated and multiple views, coordination technique, ontologies.

Lista de Figuras

2.1	Processo de mapeamento visual, adaptado de Card et al. (1999).	16
2.2	Visualização pela técnica de coordenadas paralelas, extraído de Garcia (2006).	18
2.3	Sobreposição de linhas em técnicas de projeções paralelas, extraído de Shimabukuro (2004).	18
2.4	Exemplo de duas representações orientadas a <i>pixel</i> . Adaptado de Ball e Eick (1996).	19
2.5	Exemplo de representações utilizando técnicas iconográficas; (a) ícone que representa cinco variáveis; (b) imagem com ícones de cinco dimensões. Adaptado de Pickett e Grinstein (1988)	20
2.6	Exemplo de <i>Treemap</i> , extraído de North (2000)	21
2.7	Exemplo de técnicas baseadas em grafos: os agrupamentos dos vértices representam similaridade entre <i>malware</i> , extraído de Santos (2014)	21
2.8	Exemplo de técnicas híbridas: (a) seleção dos documentos; (b) projeção das palavras-chave. Adaptado de Paulovich et al. (2012)	22
2.9	Modelo conceitual do <i>Snap</i> . Adaptado de North (2000).	23
2.10	Coordenação entre chaves. Adaptado de North (2000).	24
2.11	Visão da arquitetura da coordenação. Adaptado de North (2000).	25
2.12	Arquitetura de uma visão proposta por Pattison e Phillips (2001)	25
2.13	Arquitetura geral proposta por Pattison e Phillips (2001)	26
2.14	Coordenação no sistema <i>Cviews</i> , adaptado de Boukhelifa e Rodgers (2003).	27
2.15	Coordenação direta no sistema <i>Improvise</i> . Extraído de Weaver (2004).	28
2.16	Coordenação indireta no sistema <i>Improvise</i> . Extraído de Weaver (2004).	28
2.17	Projeção de um conjunto de documentos pela PEx	29
2.18	Classes do modelo de coordenação	30
2.19	<i>Coordenação entre dois mapas de similaridade. Seleção de documentos em (a) e destaque de documentos em (b), extraído de Eler (2011)</i>	31
2.20	<i>Keyword Network View: mostra relacionamentos entre palavras-chave. Extraído de Chang et al. (2007)</i>	32
2.21	<i>Visão String and Bead: clicando duas vezes sobre um bead específico mostra as transações relacionadas em uma visão separada. Extraído de Chang et al. (2007)</i>	32

2.22	Exploração de uma coleção de documentos por meio do <i>framework VisTools</i> , extraído de Sabol et al. (2009)	33
2.23	Interface do sistema SWAPit, adaptado de Seeling e Becks (2004)	35
2.24	Interface do sistema IVEA, extraído de Thai et al. (2008)	35
3.1	<i>Dois modelos conceituais UML</i> , extraído de Euzenat e Shvaiko (2007)	43
3.2	<i>Fragments de ontologias</i> , extraído de Euzenat e Shvaiko (2007)	44
3.3	Processo de <i>matching</i> , adaptado de Euzenat e Shvaiko (2007)	44
3.4	Exemplo de alinhamento entre ontologias, extraído de Euzenat e Shvaiko (2007)	46
3.5	Técnicas de <i>Matching</i> , adaptado de Euzenat e Shvaiko (2007)	47
3.6	Comparação de dois conjuntos de propriedades, extraído de Euzenat e Shvaiko (2007)	50
3.7	<i>Matching</i> entre duas ontologias usando um <i>background</i> de conhecimento de ontologias. Adaptado de Aleksovski et al. (2006)	52
3.8	<i>Matching</i> entre duas ontologias usando um <i>background</i> de conhecimento. Adaptado de Aleksovski et al. (2006)	53
4.1	Processo de geração de tópicos	55
4.2	Visão geral do esquema de coordenação	55
4.3	Visão geral da obtenção dos termos frequentes	56
4.4	Visão geral do processo de <i>matching</i>	57
4.5	Mapeamento e destacamento dos itens na visão destino	57
4.6	Apresentação dos termos frequentes a partir de uma seleção	58
4.7	Apresentação dos termos relacionados por meio de uma ontologia	58
4.8	Exemplo da aplicação da <i>Semantic Coordination</i>	59
4.9	Visão parcial da vHOG	60
4.10	Visão geral da base de dados explorada	60
4.11	Exploração de uma coleção de documentos cobertos pelo termo “bone”	61
4.12	Termos relacionados ao termo “bone”	61
4.13	Destacamento dos documentos abrangidos pelo termo “vertebra”	62
4.14	Exploração de uma coleção de documentos cobertos pelo termo “aorta”	62
4.15	Destacamento dos documentos cobertos pelo termo “artery”	63
4.16	Exploração de uma coleção de documentos cobertos pelo termo “ligament”	63
4.17	Destacamento dos documentos cobertos pelo termo “joint”	64
4.18	Estrutura da <i>OntoTest</i> , extraído de (Barbosa et al., 2006)	64
4.19	Exploração de um conjunto de documentos cobertos pelo termo “training”	65
4.20	Exploração de um conjunto de documentos cobertos pelo termo “organizational activity”	65

Sumário

1	Introdução	12
1.1	Contexto	12
1.2	Motivação e Justificativa	12
1.3	Formulação do problema	13
1.4	Objetivos	14
1.5	Organização	14
2	Visualização de Informação: Múltiplas Visões Coordenadas	15
2.1	Considerações iniciais	15
2.2	Técnicas de visualização de informação	16
2.2.1	Técnicas geométricas	17
2.2.2	Técnicas orientadas a <i>pixels</i>	17
2.2.3	Técnicas iconográficas	19
2.2.4	Técnicas hierárquicas	20
2.2.5	Técnicas baseadas em grafos	20
2.2.6	Técnicas híbridas	21
2.3	Modelos de coordenação	22
2.3.1	<i>Snap Together Visualization</i>	22
2.3.2	Modelo proposto por Pattison e Phillips	25
2.3.3	Modelo proposto por Boukhelifa e Rodgers	26
2.3.4	Modelo Proposto por Weaver	27
2.4	Técnicas de coordenação entre mapas de similaridade	29
2.5	Aplicações de múltiplas visões coordenadas	31
2.5.1	Aplicações em dados textuais	31
2.5.2	Aplicações com o uso de ontologias	33
2.6	Considerações Finais	36
3	Ontologia	37
3.1	Considerações iniciais	37
3.2	Linguagens para Definir Ontologias	38

3.2.1	OWL (Web Ontology Language)	39
3.3	Estrutura de uma ontologia	42
3.4	Processo de <i>matching</i> de ontologias	44
3.5	Técnicas de <i>matching</i> de ontologias	46
3.5.1	Técnicas baseadas em <i>strings</i>	48
3.5.2	Técnicas baseadas em linguagens	49
3.5.3	Técnicas baseadas na estrutura	49
3.5.4	Técnicas extensionais	51
3.5.5	Técnicas baseadas em semântica	51
3.6	Considerações Finais	52
4	Técnica de Coordenação com o Apoio de Ontologias	54
4.1	Considerações iniciais	54
4.2	Esquema de coordenação com o apoio de ontologias	55
4.2.1	Obtenção dos termos frequentes	55
4.2.2	Etapa de <i>matching</i>	56
4.2.3	Mapeamento na visão destino	57
4.3	Avaliação da técnica de coordenação proposta	59
4.3.1	Estudo de caso: descrições de imagens médicas	59
4.3.2	Estudo de caso: artigos de teste de software	63
4.4	Considerações finais	66
5	Conclusões	67
5.1	Contribuições e limitações	67
5.2	Trabalhos futuros	68
5.3	Produção bibliográfica	69
	Referências	70

Introdução

1.1 Contexto

O uso de **visualização** está cada vez mais presente na vida cotidiana, seja na produção de novos conhecimentos ou no compartilhamento de informações a partir de dados armazenados. Com esse propósito, recursos computacionais podem ser usados para gerar representações gráficas e interativas de conjuntos de dados, explorando a capacidade visual e cognitiva humana. Quando a visualização é utilizada para a representação de dados oriundos de fenômenos que não possuem uma representação espacial própria, é caracterizada a **visualização de informação** (Card et al., 1999, Ware, 2004).

Para a representação desses conjuntos de dados são usadas **técnicas de visualização**, responsáveis por apresentar ao usuário representações visuais a partir das características presentes nesses conjuntos. Um recurso utilizado para apoiar a utilização de visualização de informação é o de **múltiplas visões coordenadas**, que proporciona a visualização de conjuntos de dados sob perspectivas variadas (Keim, 2002, Chen, 2004). Para isso, é necessária a aplicação de uma **técnica de coordenação** responsável por realizar um mapeamento entre os elementos das visões ¹ coordenadas. Esse mapeamento é gerado seguindo algum critério e varia entre uma técnica e outra (Eler, 2011).

Na literatura há diferentes técnicas de coordenação presentes em **modelos de coordenação** (North, 2000, Weaver, 2004, Pattison e Phillips, 2001, Eler, 2011), porém essas técnicas apresentam limitações. Por exemplo, técnicas que utilizam identificadores no processo de coordenação, realizam a associação entre os elementos presentes em visões distintas por meio do *matching* léxico dos seus identificadores.

1.2 Motivação e Justificativa

Múltiplas visões de conjuntos de dados permitem analisá-los sob perspectivas distintas (Ware, 2004). Assim, propiciam ao usuário a ampliação do seu poder de exploração, o que pode resultar na

¹Neste texto, o termo visão se refere a uma representação visual obtida por meio de uma projeção (ou outra técnica) de um conjunto de dados

melhor compreensão e assimilação da informação transmitida (Keim e Kriege, 1996, Ware, 2004). Um ponto importante é a coordenação das visões, responsável por identificar o relacionamento entre as instâncias dos dados entre visões distintas (North, 2000, Eler, 2011).

Dentre as aplicações de múltiplas visões coordenadas, destaca-se a exploração de dados textuais. Na literatura há ferramentas que realizam a criação e a exploração de visões a partir de coleções de documentos, como a ferramenta *Projection Explorer* (PEX) (Paulovich et al., 2007). Na PEX, cada documento é representado em uma visão por um círculo projetado em um plano. Se dois círculos são projetados próximos um do outro, o conteúdo dos documentos por eles representados é semelhante.

Usando essa representação gráfica é possível identificar grupos de documentos similares e verificar a correlação entre áreas do conhecimento. Além disso, é possível selecionar grupos de documentos e criar rótulos para tais grupos, identificando os tópicos a serem discutidos (Paulovich et al., 2007). Outro recurso disponibilizado pela PEX é a projeção simultânea de duas ou mais visões, podendo ser projeções de uma mesma base de dados ou de bases distintas. Ao selecionar um conjunto de documentos projetados em uma visão, são identificados os documentos relacionados por meio de uma técnica de coordenação (Paulovich et al., 2007).

O uso de semântica nas técnicas de coordenação presentes em ferramentas como a PEX possibilita a execução de um tipo de mapeamento diferente dos tradicionais. Esse mapeamento detecta elementos que apresentam outros relacionamentos que não são baseados nos atributos dos dados, proporcionando ao usuário um aumento no poder de exploração dos dados.

A disponibilização de novos relacionamentos entre os elementos presentes nas visões enriquece o processo de coordenação, pois proporciona um aumento da possibilidade da vinculação entre esses elementos. Assim, relacionamentos que permaneciam ocultos em técnicas que empregam o mapeamento léxico, são descobertos com o emprego de semântica no processo de coordenação.

1.3 Formulação do problema

Após a realização de um levantamento bibliográfico foi constatado que as principais técnicas de coordenação empregadas na análise de conjuntos de dados utilizam mapeamentos léxicos. Técnicas como a proposta por North e Shneiderman (2000) fazem um mapeamento léxico utilizando os atributos dos dados. Assim, para realizar a associação entre as instâncias contidas em diferentes visões são utilizadas apenas as estruturas dos termos extraídos dessas instâncias.

Com a aplicação do mapeamento léxico no processo de coordenação, instâncias identificadas por palavras distintas com o mesmo significado, ou ainda que estejam relacionadas a um mesmo conceito não são relacionadas a coordenação. Isso limita a exploração, pois instâncias semanticamente relacionadas não são mapeadas, comprometendo a descoberta de relacionamentos.

Técnicas de coordenação, que durante o processo de coordenação se utilizam apenas da estrutura dos termos utilizados no mapeamento em cada visão, apresentam um poder de exploração limitado. Isso acontece porque apenas são mapeados na visão destino, os elementos com identificadores de estruturas idênticas às dos itens selecionados na visão origem.

1.4 Objetivos

Mapeamentos semânticos podem ser realizados com o apoio de ontologias. Uma ontologia, em conjunto com as instâncias individuais de suas classes é capaz de relacionar conceitos distintos, por meio dos cruzamentos das suas relações. Assim, palavras que possuem algum relacionamento podem ser ligadas por meio do mapeamento oriundo das ontologias.

Este trabalho tem como objetivo geral a proposta de uma nova técnica de coordenação para gerar um mapeamento semântico entre múltiplas visões. Desta maneira, as instâncias relacionadas não se restringem àquelas mapeadas de maneira léxica.

Para a implementação e adição da *Semantic Coordination* às demais técnicas de coordenação é necessário o uso de ontologias. Além disso, o conjunto de dados explorado deve pertencer ao mesmo domínio da ontologia escolhida para realizar a coordenação.

Como objetivos específicos, tem-se:

- Incorporar o uso de semântica em mecanismos de coordenação entre múltiplas visões;
- Analisar a efetividade da técnica proposta em diferentes domínios de conhecimento.

Como hipótese de pesquisa para este trabalho tem-se: o emprego de semântica no processo de coordenação entre múltiplas visões **não** resulta em mapeamentos diferentes do que técnicas que não a utilizam. Como hipótese alternativa, tem-se que o emprego de semântica gera novos mapeamentos.

1.5 Organização

Com o propósito de contextualizar e apresentar a proposta de uma coordenação semântica entre múltiplas visões, esta dissertação está organizada da seguinte maneira:

- No Capítulo 2 são brevemente apresentados conceitos e técnicas de visualização. Em seguida, são apresentados conceitos de múltiplas visões coordenadas, com enfoque no processo de coordenação, assim como trabalhos relacionados que aplicam esses conceitos.
- No Capítulo 3 são apresentados conceitos de ontologias, fundamentais para a proposta do mapeamento semântico entre as visões. Em especial, o processo e as técnicas de *matching*, utilizadas para comparar elementos externos com os elementos da ontologia.
- No Capítulo 4 é apresentada a proposta de uma técnica de coordenação com o apoio de ontologias, implementada como uma extensão das técnicas de coordenação já existentes na *Projection Explorer*. Além disso, é apresentada a avaliação da proposta, realizada por meio da comparação dos resultados obtidos com os resultados provenientes das técnicas de coordenação contidas na *Projection Explorer*.
- No Capítulo 5 são apresentadas as conclusões do trabalho desenvolvido, suas contribuições, limitações e trabalhos futuros.

Visualização de Informação: Múltiplas Visões Coordenadas

2.1 Considerações iniciais

Card et al. (1999) definem visualização como o uso interativo de representações visuais de dados para amplificar a cognição de quem os analisa. Visualização pode ser empregada na apresentação, na análise e na exploração de um conjunto de dados. Também é utilizada em áreas em que a recuperação da informação a partir desses dados é essencial em atividades laborais e cotidianas. Porém, a grande quantidade e a dificuldade de manipulá-los são barreiras que impedem o usuário de discernir o que é informação útil em meio a esses dados (Oliveira e Levkowitz, 2003, Ware, 2004).

Representações visuais utilizam recursos gráficos para representar um conjunto de dados brutos, em que cada unidade de dado descreve múltiplos atributos relacionados (Oliveira e Levkowitz, 2003). Entretanto, os dados não podem ser projetados diretamente em uma representação gráfica, sendo necessária a “transformação” desses dados em representações visuais por meio de um **processo de mapeamento visual**. Segundo Card et al. (1999), esse processo pode ser dividido em quatro estágios, ilustrados na Figura 2.1.

Inicialmente há a coleta de dados sobre os quais se deseja obter informações. Esse conjunto de dados brutos pode ser heterogêneo, descrevendo, quantidades, códigos, descrições, medidas, imagens ou vídeos, por exemplo, . Depois da coleta, esses dados são transformados e organizados em tabelas (Card et al., 1999). O próximo passo é a transformação das tabelas de dados em estruturas visuais. O mapeamento visual é dependente das características dos dados representados, para analisar as propriedades visuais a serem utilizadas na sua representação. Nessa etapa é estabelecido como os atributos devem ser representados visualmente, ou seja, quais propriedades gráficas são utilizadas. O último passo é a geração da representação gráfica usando técnicas de visualização.

Entretanto, essas representações quando projetadas única e isoladamente podem não ser eficientes na exploração dos dados. Dependendo da natureza desses dados e da técnica utilizada podem

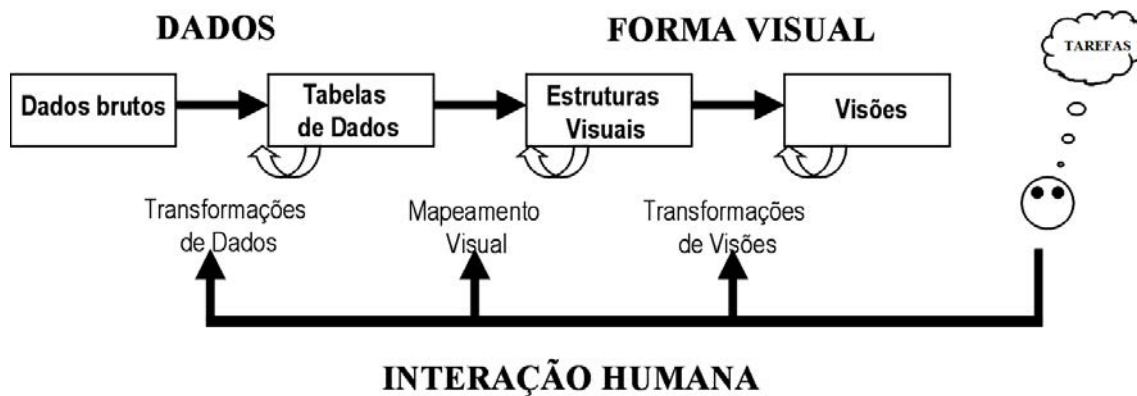


Figura 2.1: Processo de mapeamento visual, adaptado de Card et al. (1999).

haver problemas, como sobreposição ou baixo valor semântico da representação visual gerada. Uma alternativa é a combinação de diferentes técnicas, aumentando o poder de exploração do usuário.

Uma área importante na visualização exploratória é conhecida como *Múltiplas Visões Coordenadas* (CMV – do inglês *Coordinated and Multiple Views*). Diferentes visões de um mesmo conjunto de dados, ou de conjuntos relacionados, permitem observá-los sob diferentes perspectivas (Keim e Krieger, 1996). O mecanismo para coordenar essas visões deve prover a interação ao usuário, enquanto realiza tarefas de exploração. Com isso, pode-se revelar relações como distribuição, agrupamento ou subordinação, que podem ser utilizadas para facilitar a interpretação dos dados (Boukhelifa e Rodgers, 2003, Chen, 2004, Eler, 2011).

As relações entre as visões são estabelecidas por meio de uma **técnica de coordenação**. Na literatura há várias propostas de técnicas de coordenação, sendo a mais utilizada a baseada em identificadores, que utiliza o identificador das instâncias dos dados para gerar o mapeamento entre as visões (North, 2000, Pattison e Phillips, 2001, Weaver, 2004, Eler et al., 2009). Também são encontradas técnicas para auxiliar a exploração de conjuntos de documentos, que geralmente utilizam tópicos extraídos dos documentos para gerar o mapeamento entre as visões.

Neste capítulo são apresentados elementos de Múltiplas Visões Coordenadas. Na próxima seção são apresentadas técnicas para representar dados visualmente, que ao serem combinadas podem auxiliar no processo de exploração de CMV. Na Seção 2.3 são apresentados alguns modelos de coordenação encontrados na literatura. Em seguida, na Seção 2.4 são apresentadas técnicas de coordenação entre visões. Na Seção 2.5 são exibidas aplicações de sistemas baseadas no paradigma de CMV, em especial aplicações que exploram coleções de textos. Por fim, na Seção 2.6 são expostas as considerações finais.

2.2 Técnicas de visualização de informação

Técnicas de visualização de informação proporcionam ao usuário o auxílio na busca por informação. Segundo Keim (2002), essas técnicas podem ser classificadas com base em três critérios: os dados a serem visualizados, a técnica de visualização empregada e a técnica de interação utilizada para manipular as representações visuais. Ainda segundo o autor, elas podem ser agrupadas nas seguintes categorias:

1. Técnicas geométricas: identificam projeções relevantes no conjunto de dados multidimensional;
2. Técnicas orientadas a *pixels*: o *pixel* representa um atributo;
3. Técnicas iconográficas: utilizam-se de ícones para mapear os valores dos atributos de um item de cada elemento multidimensional;
4. Técnicas hierárquicas: realizam uma subdivisão do espaço n -dimensional dos dados, que são organizados e exibidos de maneira hierárquica;
5. Técnicas baseada em grafos: essa técnica se utiliza de grafos para mapear as características de um conjunto de dados para depois gerar uma visualização;
6. Técnicas híbridas: integram múltiplas estratégias de visualização.

2.2.1 Técnicas geométricas

Técnicas geométricas projetam dados multidimensionais no espaço bidimensional. Um exemplo desta técnica é a **projeção em coordenadas paralelas**, que mapeia um espaço de dimensão k para um espaço visual bidimensional, usando k eixos equidistantes e paralelos. Cada eixo representa um atributo cujo intervalo de valores é mapeado linearmente com os respectivos valores sobre o eixo. Um exemplo é apresentado na Figura 2.2.

Os itens de dados são apresentados como segmentos de reta conectados que interpretam os n -eixos paralelos, permitindo a identificação de dependências funcionais e distribuição dos dados. Porém, apresenta como desvantagem uma limitação do número de itens de dados projetados. Como ilustrado na Figura 2.3, o aumento de itens de dados projetados faz com que a representação visual obtida seja de difícil interpretação devido à sobreposição. Alternativas para contornar esses problemas têm sido propostas, mas fogem do escopo deste trabalho.

2.2.2 Técnicas orientadas a *pixels*

Nas técnicas orientadas a *pixels* existe uma associação entre os itens de dados e a projeção dos *pixels*, em que a coloração de cada *pixel* depende do valor de um atributo dos dados. Outro aspecto relevante é o arranjo espacial dos *pixels* (ou mais precisamente, dos elementos de dados) no espaço de projeção, que deve apresentar boas propriedades ao agrupar elementos semelhantes, e manter algum significado semântico (Keim e Kriege, 1996).

No processo de visualização, essas representações são ajustadas em janelas individuais. Por exemplo, se um conjunto de dados possuir m atributos, então o espaço de projeção é dividido em m janelas. Na Figura 2.4 são mostrados dois exemplos de projeções orientadas a *pixel*. Na parte esquerda das Figuras 2.4(a) e (b) é mostrada a representação em linha e na parte direita são mostradas suas representações orientadas a *pixel* de um trecho de código. Nessas representações, cada linha de código corresponde a um, ou a um pequeno conjunto de *pixels* arranjados na tela pela ordem das linhas (indicadas por *) ou pela cor (indicadas por **) (Ball e Eick, 1996).

Essa técnica apresenta como principal vantagem a possibilidade de exibição de milhares de valores simultaneamente dependendo da resolução. Porém, para usufruir dos benefícios dessa técnica, os

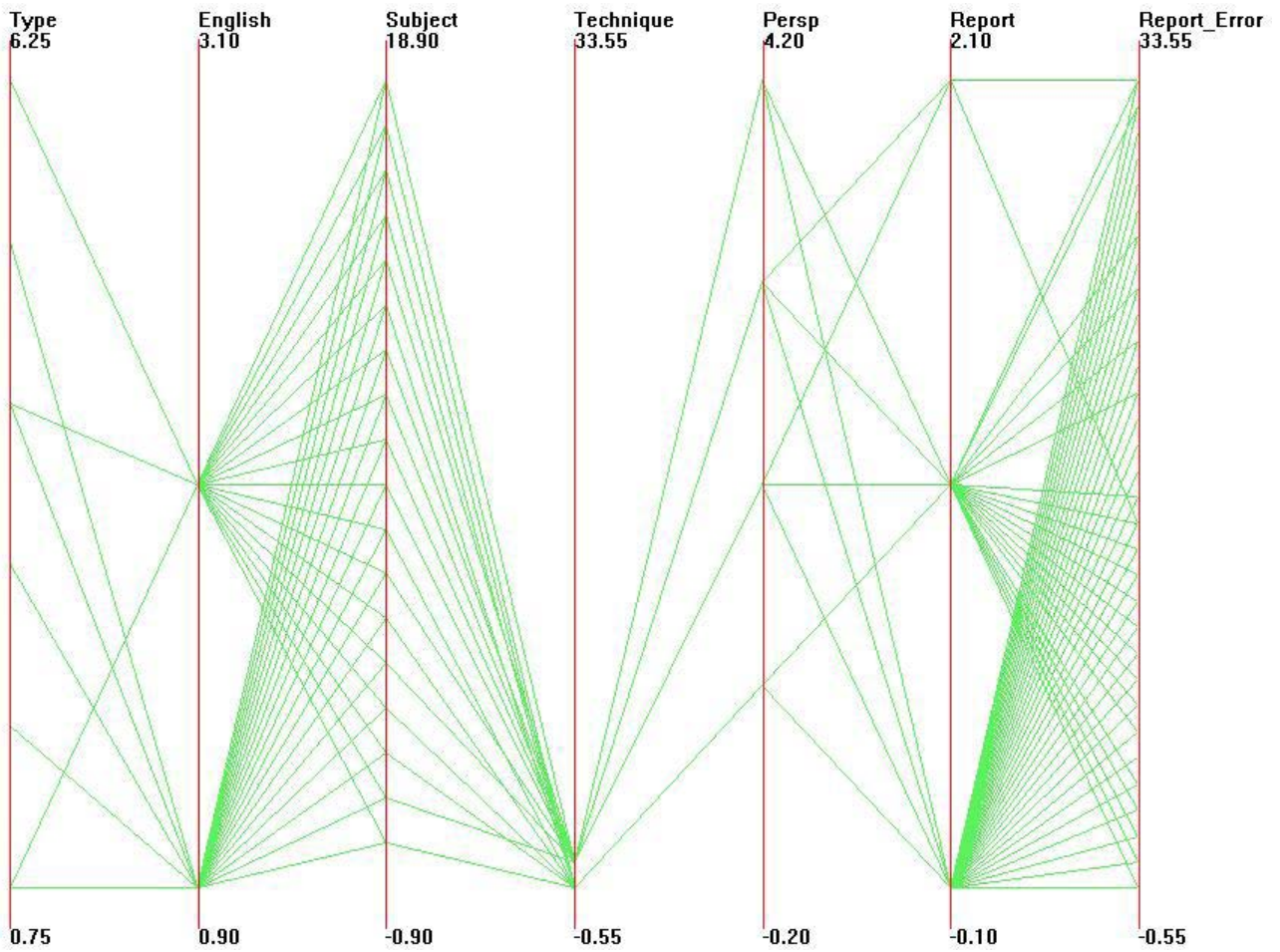


Figura 2.2: Visualização pela técnica de coordenadas paralelas, extraído de [Garcia \(2006\)](#).

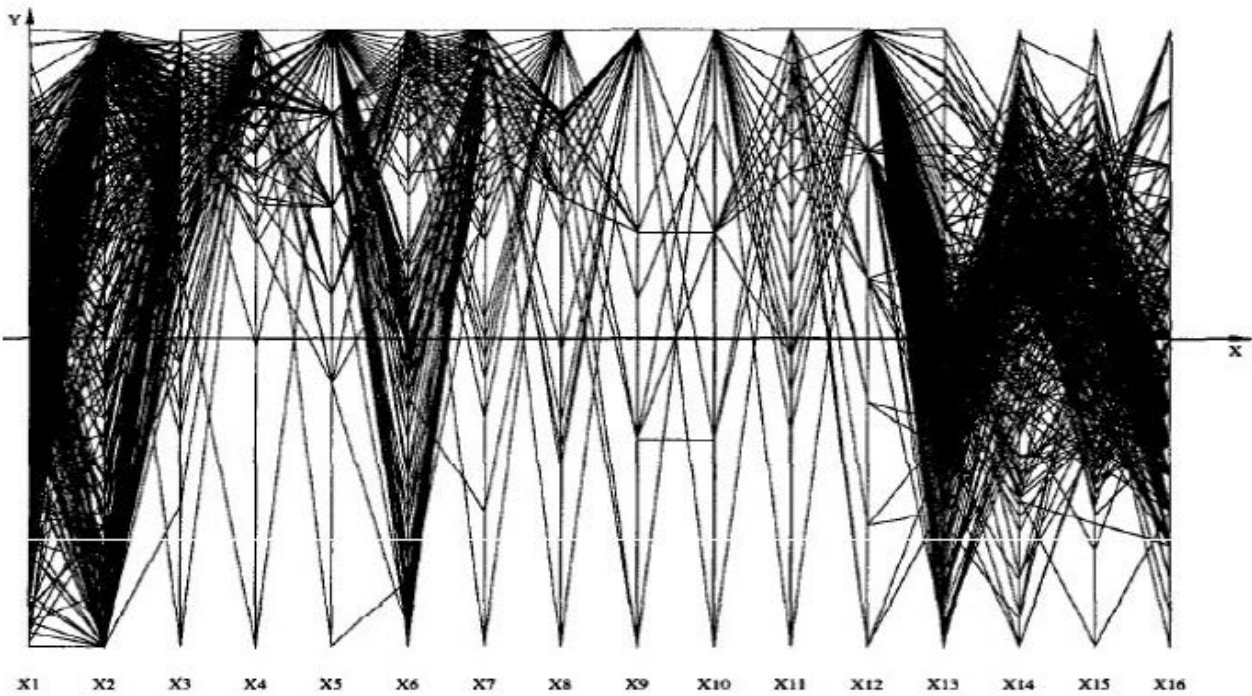


Figura 2.3: Sobreposição de linhas em técnicas de projeções paralelas, extraído de [Shimabukuro \(2004\)](#).

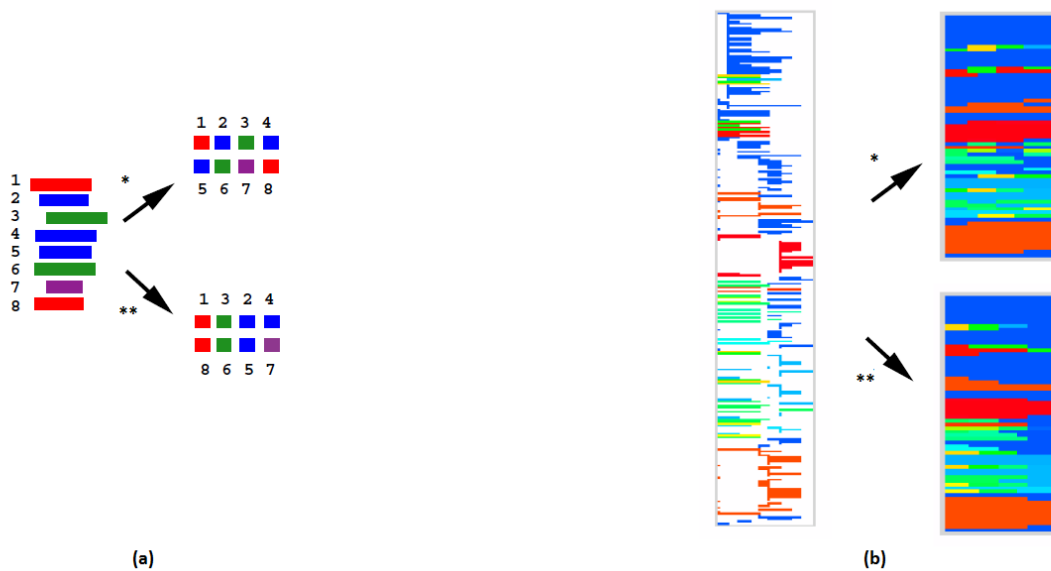


Figura 2.4: Exemplo de duas representações orientadas a *pixel*. Adaptado de Ball e Eick (1996).

pixels devem ser bem organizados no espaço de projeção. Para isso, segundo Keim e Kriege (1996) é necessário considerar os seguintes aspectos:

- Disposição dos *pixels*: ao definir uma organização para os *pixels*, deve-se considerar as dimensões do espaço de projeção, o agrupamento e a percepção de como os dados estão relacionados. Podem ser dispostos em espirais, curvas ou linhas.
- Mapeamento das cores: o uso de cores permite um número maior de JNDs (*Just Noticeable Differences*), ou seja, cores percebidas como diferentes, maximizam o poder de percepção das relações por parte do usuário.
- Formato do espaço de projeção: o formato retangular para as janelas garante o bom uso do espaço de projeção do computador. Porém, quanto maior for o número de atributos, menos explícita é a visualização do relacionamento entre os mesmos.

As principais desvantagens são a dependência em relação à resolução do espaço de projeção e o baixo valor semântico da representação: quanto maior a dimensão dos dados representados, menor o número de elementos exibidos simultaneamente (Ware, 2004).

2.2.3 Técnicas iconográficas

Nas técnicas iconográficas, diferentes ícones são utilizados para mapear os atributos de cada elemento multidimensional. A apresentação dos ícones resulta em texturas que dependem das características dos dados mapeados. Um exemplo dessa técnica são as *Stick Figures*, apresentadas por Pickett e Grinstein (1988), que utilizam dois eixos (x/y) para mapear atributos e as demais dimensões são mapeadas em ângulos e comprimentos de segmentos de reta, que formam uma *stick figure*.

Na Figura 2.5 (a) é apresentado um ícone contendo cinco variáveis, na qual uma dimensão é dada pela inclinação do corpo do ícone, e as outras quatro dimensões pelas orientações dos segmentos

de reta. Quando projetados, os ícones formam texturas que dependem das características dos dados. Uma textura é observada na Figura 2.5 (b), formada por um conjunto de ícones que representam cinco variáveis.

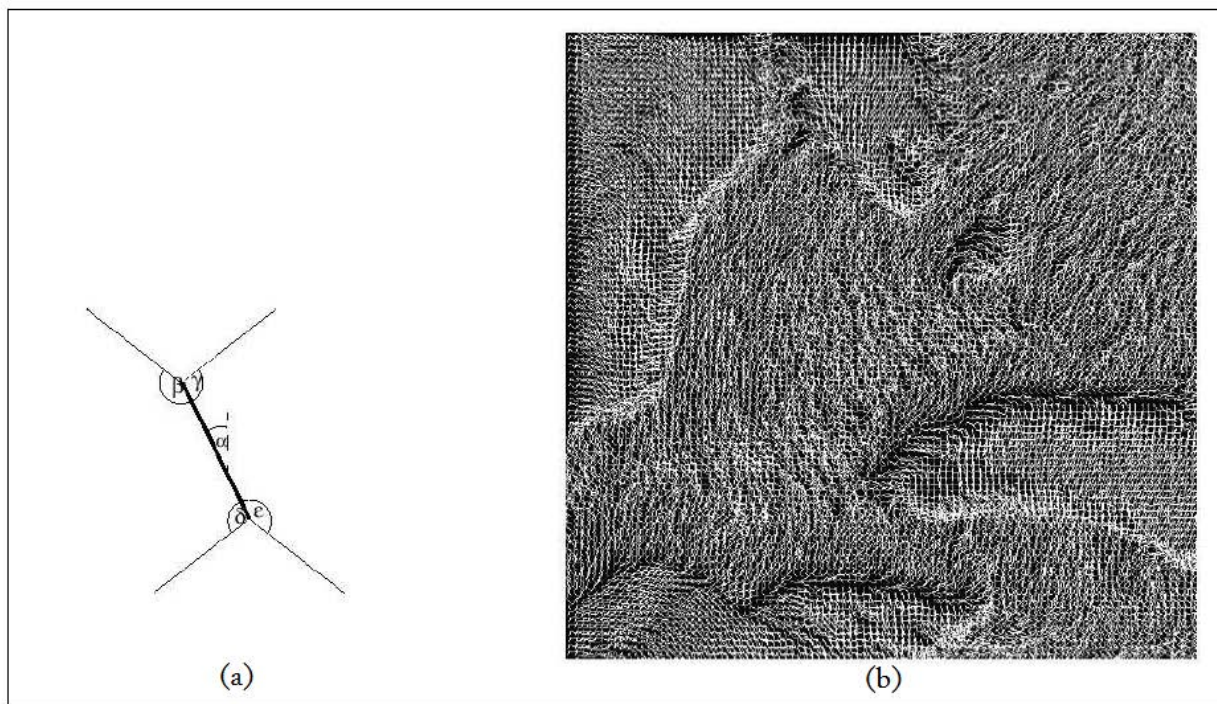


Figura 2.5: Exemplo de representações utilizando técnicas iconográficas; (a) ícone que representa cinco variáveis; (b) imagem com ícones de cinco dimensões. Adaptado de [Pickett e Grinstein \(1988\)](#)

A vantagem dessa técnica é a capacidade de destacar atributos qualitativos em um volume de dados. Porém a quantidade de atributos que pode ser apresentada é limitada pelo ícone escolhido, ou seja, o ícone pode não possuir características perceptíveis ([Pickett e Grinstein, 1988](#)).

2.2.4 Técnicas hierárquicas

Nas técnicas hierárquicas, o espaço n -dimensional dos dados é subdividido em subespaços organizados e exibidos hierarquicamente. Um exemplo dessa técnica é o *Treemap*, ilustrado na Figura 2.6. Nesse exemplo é apresentado um portfólio de uma empresa organizado pelos países e clientes. Desenvolvida por [Shneiderman \(1992\)](#), o *Treemap* mapeia dados estruturados e os apresenta utilizando uma abordagem retangular e de preenchimento de espaço. Sendo assim, é adequada para a representação de estruturas de árvores, mapeando-as diretamente em uma região planar ([North, 2000](#)).

A vantagem dessa técnica é a representação direta das hierarquias contidas em conjunto de dados. Porém, a técnica não é vantajosa quando mapeia dados não organizados adequadamente em hierarquias, sendo necessária a definição de uma organização hierárquica para a exibição.

2.2.5 Técnicas baseadas em grafos

Grafos são representados como um conjunto de pontos (vértices) ligados por segmentos de retas (as arestas) ou, dependendo da aplicação, por arestas direcionadas (arcos). Em técnicas baseadas

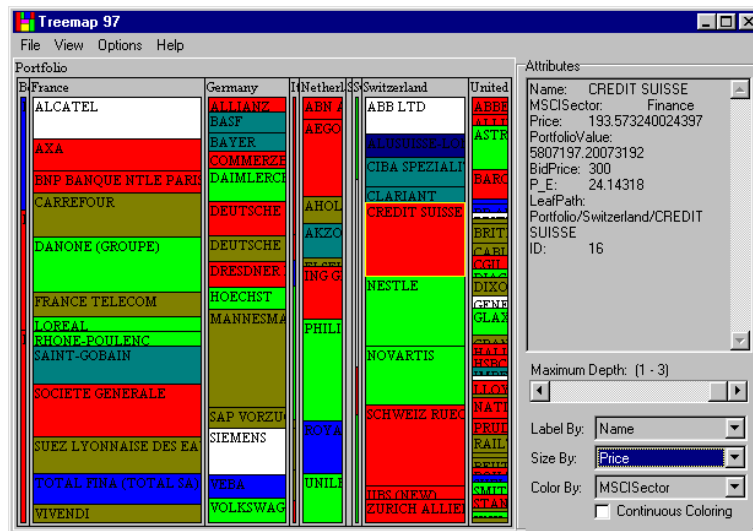


Figura 2.6: Exemplo de *Treemap*, extraído de North (2000)

em grafos, características como direção e orientação são mapeadas e representadas nas ligações entre os vértices. As representações em grafos são importantes na representação gráfica das relações existentes entre os elementos de dados, pois representam conexões ou ligações e apresentam enorme variabilidade na organização das arestas e dos vértices. Um exemplo dessa técnica é apresentado na Fig. 2.7, em que a similaridade entre um conjunto de *malware* é representada.

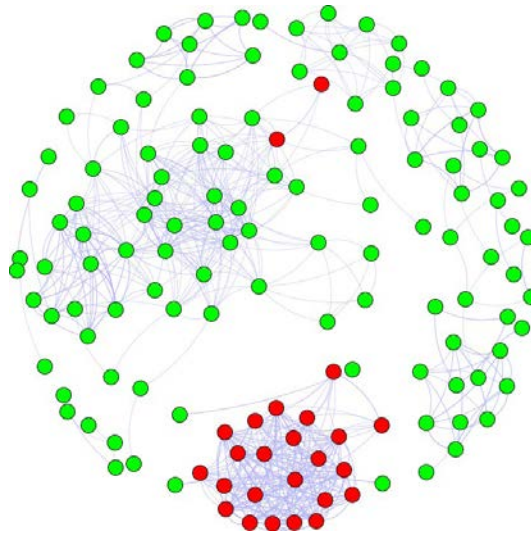


Figura 2.7: Exemplo de técnicas baseadas em grafos: os agrupamentos dos vértices representam similaridade entre *malware*, extraído de Santos (2014)

2.2.6 Técnicas híbridas

Nas técnicas híbridas, diferentes técnicas de visualização são integradas e o resultado é exibido em uma ou mais janelas. Todas as técnicas se utilizam de um conjunto extenso de estratégias de interação para apoiar processos dinâmicos de exploração. Um exemplo dessa técnica é mostrado na Figura 2.8, em que são apresentados resultados de projeções de nuvens de documentos (“*word clouds*”) da ferramenta *ProjCloud*. Nessa ferramenta o usuário pode desenhar interativamente uma região (polígono) contendo um subconjunto de documentos de seu interesse (Figura 2.8(a)). Então,

palavras-chave são extraídas e a partir dessa seleção, as mesmas são projetadas, com variação de cores e tamanhos, dentro da região definida pelo usuário (Figura 2.8(b)) (Paulovich et al., 2012).

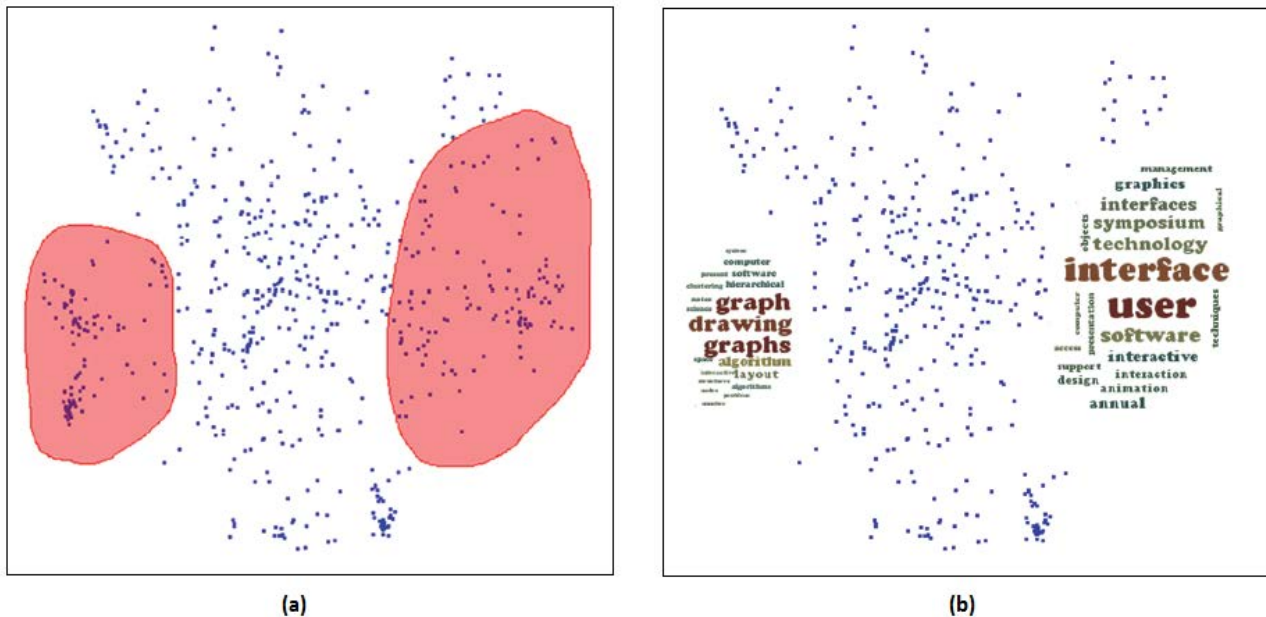


Figura 2.8: Exemplo de técnicas híbridas: (a) seleção dos documentos; (b) projeção das palavras-chave. Adaptado de Paulovich et al. (2012)

2.3 Modelos de coordenação

Questões como sincronização, visualização de relações entre dados e descoberta de informações ocultas, são abordadas por meio de modelos de coordenação (North e Shneiderman, 2000). Modelos propostos por North (2000), Pattison e Phillips (2001), Boukhelifa e Rodgers (2003) e Weaver (2004) são utilizados nos sistemas *Snap-Together Visualization*, *InVision*, *Cviews* e *Improvise*, respectivamente. Em comum, esses modelos procuram abordar os principais desafios da coordenação (Boukhelifa e Rodgers, 2003).

O primeiro aborda a natureza multiplataforma das múltiplas visões, pois as ações realizadas em uma visão nem sempre podem ser aplicadas diretamente a outras visões. O segundo desafio é sobre o espaço de projeção, em que questões como a disposição das janelas das visões ao usuário e quais dessas visões devem ser coordenadas devem ser tratadas. O terceiro aspecto trata da notificação de eventos durante a coordenação, isto é, como as visões destino são notificadas de eventos provocados pela visão origem (Boukhelifa e Rodgers, 2003). Os modelos de coordenação propostos a seguir tratam desses desafios.

2.3.1 *Snap Together Visualization*

O *Snap Together Visualization*, ou simplesmente *Snap* (North, 2000), é baseado no modelo relacional de dados, ou seja, a coordenação é feita com base no esquema relacional da base de dados. É composto de uma interface, de uma arquitetura e um modelo de coordenação que viabiliza a coordenação entre múltiplas visões. Utiliza as tabelas de dados ou resultados de consultas a um banco de dados relacional como entrada do seu processo de coordenação. As saídas geradas são visualizações

das tabelas ou dos relacionamento entre os dados, em que cada item visualizado corresponde a uma tupla.

O propósito do *Snap* é interligar representações visuais e construir interfaces de visualizações coordenadas para exploração de dados e relacionamentos. A visualização é construída utilizando os seguintes conceitos, (ilustrados na Figura 2.9): relação (visualização), tupla (item visualizado), chave primária (identificação do item), junção relacional (coordenação).

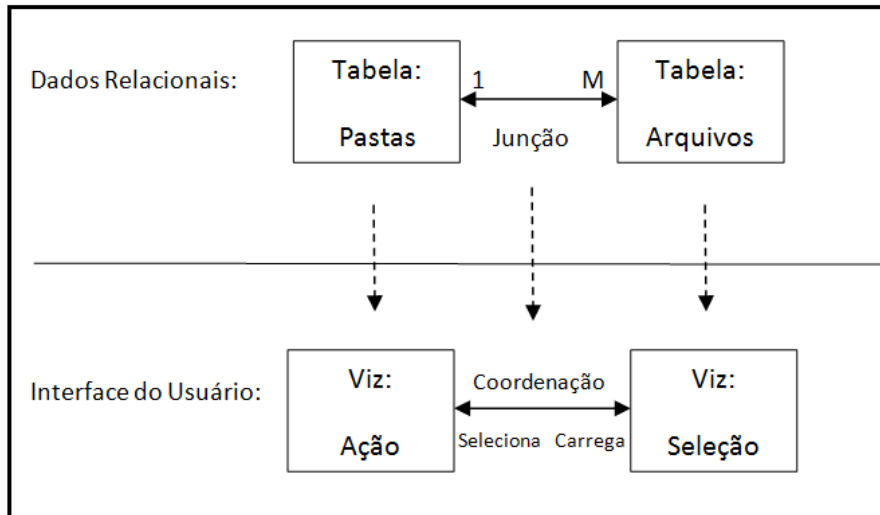


Figura 2.9: Modelo conceitual do *Snap*. Adaptado de North (2000).

A coordenação de visões é feita utilizando junções relacionais das tuplas da base de dados. Após a realização de uma ação sobre uma visualização, são disparadas outras ações nas tuplas associadas pela junção relacional, gerando outras visualizações. Durante esse processo é construído o grafo de coordenação, responsável pela gerência e coordenação entre as visualizações. A sintaxe para a definição da coordenação é:

$$((vis1,ação1,idItem) (vis2,ação2,idItem))$$

cuja semântica é: *ação1* em uma visualização *vis1* envolvendo os itens identificados por *idItem*, provoca uma ação correspondente *ação2* em uma visualização *vis2* envolvendo os itens identificados pelas mesmas chaves. O caminho inverso também é válido.

A relação entre os elementos presentes em um par de visões é estabelecida pela realização de ações acopladas. A relação de junção determina qual das três combinações possíveis entre chave primária e chave estrangeira podem ser utilizadas. As possibilidades são ilustradas na Figura 2.10.

- Chave primária para chave primária (um-para-um): essa relação é frequentemente o resultado da exibição de diferentes projeções de uma mesma tabela em múltiplas visualizações. A chave da ação em questão é acoplada a uma chave de uma outra ação, ligando os seus valores. Assim, quando uma das ações é invocada, a outra (com o mesmo valor da chave) também é invocada;
- Chave primária para chave estrangeira (um-para-muitos): essa combinação liga o valor de chave primária de uma ação ao valor de uma chave estrangeira de uma outra ação. Quando a ação

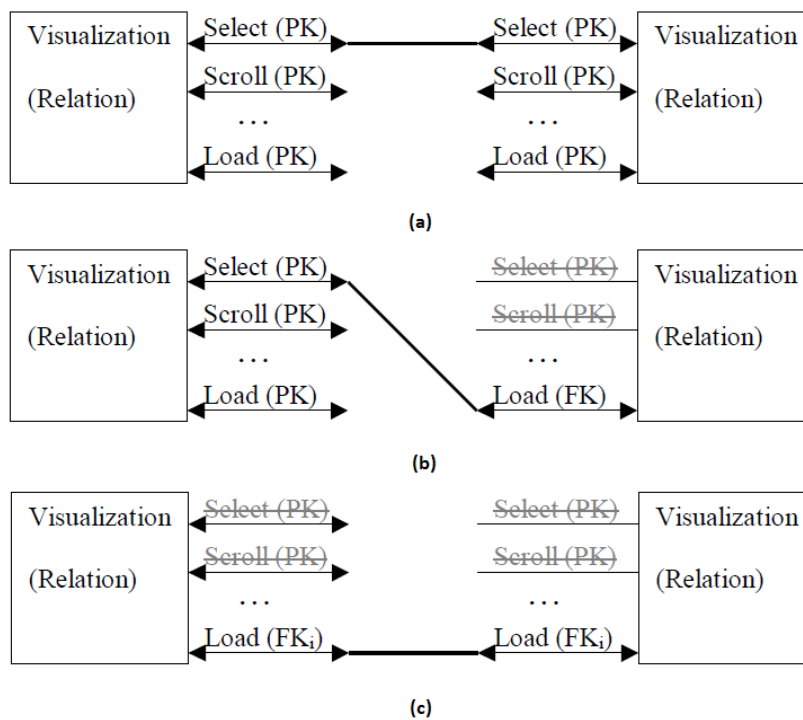


Figura 2.10: Coordenação entre chaves. Adaptado de North (2000).

identificada pela chave primária é invocada, outra ação identificada pela chave estrangeira também é invocada, usando o valor da chave primária como a chave estrangeira;

- Chave estrangeira para chave estrangeira (muitos-para-um-para-muitos): é uma relação que ocorre quando duas relações compartilham uma chave estrangeira em comum como atributo.

2.3.1.1 Arquitetura do *Snap*

A arquitetura do *Snap*, ilustrada na Figura 2.11, é baseada em eventos. Também, atua como um intermediário entre as visualizações e como mediação entre as mesmas e a base de dados. Além disso, a arquitetura mantém as visualizações isoladas dos outros componentes do sistema. Isso faz com que o processo de visualização seja independente do processo de coordenação, e obedecendo ao seguinte fluxo de execução:

1. A visualização notifica o *Snap* da existência de uma ação e do valor da chave primária da tupla, por meio do evento **ação chave primária**;
2. Inicia-se o percurso no grafo de coordenação, a partir de uma ação originária na *Visualização 1*;
3. Para cada visualização encontrada no percurso, inicia-se uma ação fortemente acoplada: se a ação não é uma ação do tipo *load*, então o valor de chave primária é passado como parâmetro, por meio da **ação chave primária** (indicada na Figura 2.11). Por outro lado, se a ação for do tipo *load*, o **Gerenciador de Visualização** executa uma consulta usando o valor da chave primária como parâmetro e carrega os resultados para a *Visualização 2*.

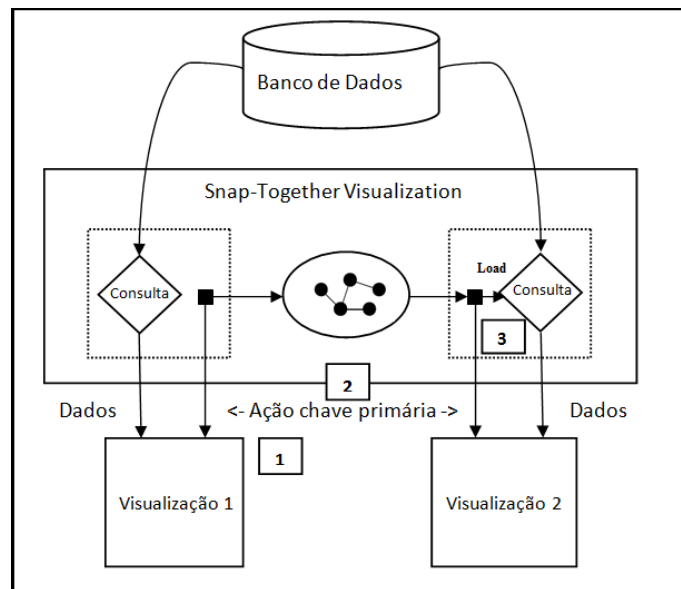


Figura 2.11: Visão da arquitetura da coordenação. Adaptado de North (2000).

2.3.2 Modelo proposto por Pattison e Phillips

A arquitetura para a coordenação de visões apresentada por Pattison e Phillips (2001) propõe a separação das responsabilidades de apresentação e renderização de uma visão, da sua representação. Essa separação é feita por meio da utilização dos componentes de **Apresentação** e de **Modelo**, exibidos na Figura 2.12.

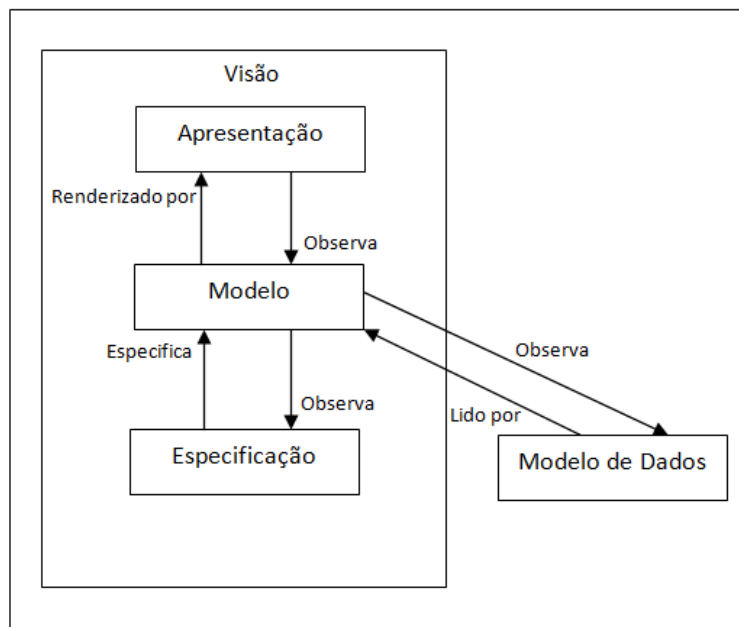


Figura 2.12: Arquitetura de uma visão proposta por Pattison e Phillips (2001)

O componente da camada **Modelo** é responsável por mapear os dados em um modelo de exibição. Então, cada componente da camada de apresentação observa o componente correspondente da camada **Modelo**, esperando por suas mudanças e atualizações. Adicionalmente, monitora os componentes das camadas **Especificação** e de **Modelo de Dados** em busca de mudanças. Isso garante que as mudanças

na especificação da visão sejam propagadas para o componente da camada de apresentação.

A coordenação entre as visões é mediada por componentes de coordenação. Esses componentes procuram mudanças de estado em uma visão e disparam as mudanças resultantes na visão dependente. Além disso, encapsulam funcionalidades que se referem especificamente às ligações entre as visões. Por fim, a especificação da coordenação exige no mínimo a nomeação da:

- A visão a ser observada pelo componente da coordenação;
- O tipo de mudança na visão observada responsável por desencadear a coordenação;
- A exibição destino;
- A reação exibida na visão destino após o disparo da coordenação.

Com o propósito de facilitar essa exploração, [Pattison e Phillips \(2001\)](#) apresentam uma arquitetura que fornece suporte genérico para arquiteturas de coordenações de visões, ilustrada na Figura 2.13. O diferencial desse modelo é não se restringir a coordenar elementos correspondentes, propondo uma coordenação aos pares de elementos.

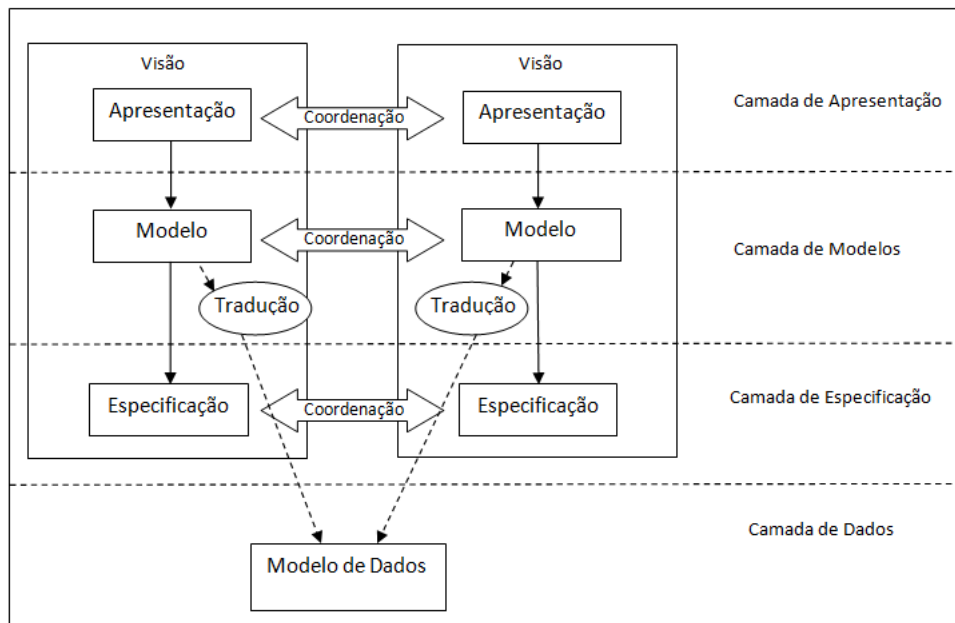


Figura 2.13: Arquitetura geral proposta por [Pattison e Phillips \(2001\)](#)

2.3.3 Modelo proposto por Boukhelifa e Rodgers

O modelo proposto por [Boukhelifa e Rodgers \(2003\)](#) inclui a definição de **objetos de coordenação**, contidos em um espaço de coordenação. Esses objetos são responsáveis por gerenciar combinações das entidades que controlam os aspectos relacionados às visões, que são coordenadas compartilhando um objeto de coordenação.

Na Figura 2.14 é ilustrado o processo de coordenação. O primeiro passo é a definição das **funções de tradução** das visões a serem coordenadas, representadas por $f_{1,1}$ e $f_{2,1}$ na Figura 2.14. Ao

ocorrer um evento, como uma ação iniciada por um usuário, o objeto da coordenação é alterado e as visões relacionadas recebem uma notificação. Essas visões são notificadas da mudança e usam as informações fornecidas pelo objeto de coordenação para a criação ou atualização de uma nova visão.

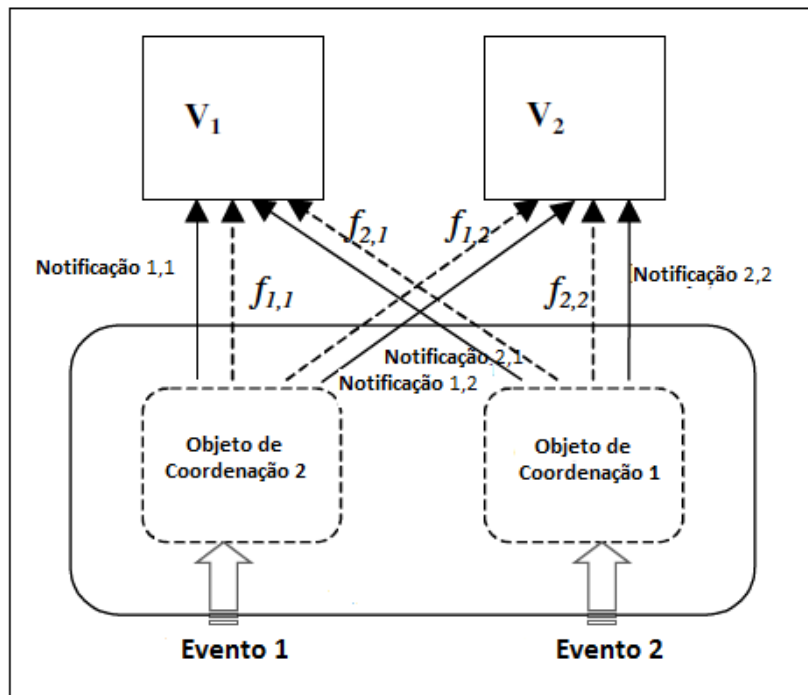


Figura 2.14: Coordenação no sistema *Cviews*, adaptado de Boukhelifa e Rodgers (2003).

Nesse modelo, as visões não são dependentes umas das outras no processo de coordenação. A técnica de coordenação empregada é a baseada em identificadores, pois novas visões compartilham o mesmo parâmetro selecionado da visão origem. As novas visões são resultados das mudanças de parâmetros para o processo de visualização.

2.3.4 Modelo Proposto por Weaver

O modelo proposto por Weaver (2004) também é baseado no modelo relacional de dados. É um sistema que coordena múltiplas visões usando um mecanismo simples de coordenação, junto de uma linguagem de abstração visual. Permite aos usuários carregar dados, criar exibições, especificar abstrações visuais, e estabelecer coordenações interativamente (Weaver, 2004).

A sua arquitetura e a implementação da coordenação são divididas em duas partes principais. A primeira é um mecanismo de coordenação simples e direto chamado *live properties* – sua função é criar dependências interativas entre as visões. Além disso, *live properties* coordenam múltiplas visões utilizando mecanismos de atualização simétrica e de notificação, que são responsáveis pela ligação das visões e de outros controles por meio de objetos compartilhados. Os controles são responsáveis por formatar informações como cores, formatos, fontes, além de realizar operações de consulta de dados com a utilização de projeções e filtros. A arquitetura desse modelo é capaz de coordenar diretamente visões, por meio de objetos compartilhados chamados *variáveis*. Alterações nessas variáveis são propagadas por intermédio de suas *live properties*.

A execução ocorre na sequência ilustrada na Figura 2.15. Após uma interação, um controle altera

o valor de uma de suas *live properties* (1). Um novo valor é atribuído à variável da *live property* alterada (2). A variável envia uma notificação de alteração para todas as *live properties* ligadas a ela (3), que por sua vez, notificam seus respectivos controles da mudança (4). Assim os controles são atualizados adequadamente.

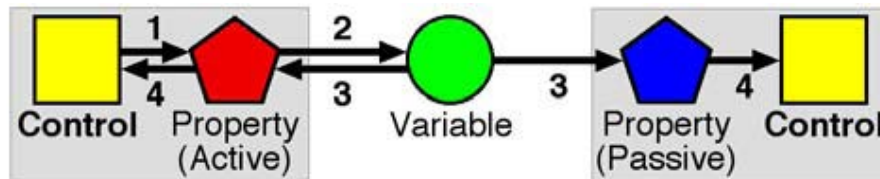


Figura 2.15: Coordenação direta no sistema *Improvise*. Extraído de Weaver (2004).

A segunda parte é um mecanismo de coordenação indireta, chamado *coordinated queries*. O diferencial desse mecanismo de coordenação é tornar viável aos usuários a definição de dependências complexas entre as visões. Isso é realizado por meio das expressões definidas pelo usuário, que por sua vez, podem depender da interação entre as visões. Essas expressões podem ser dos tipos: **filtros** ou **projeções**.

Filtros são usados para calcular um valor para cada registro de entrada e usam uma única expressão booleana. Com o uso de filtros, apenas os registros com valor verdadeiro são processados pelas visões. Por outro lado, as projeções são usadas para calcular atributos de uma tupla de saída para cada uma das tuplas de entrada, usando uma ou mais expressões. Com isso, as visões mapeiam as tuplas usando informações gráficas contidas nos atributos da tupla de saída.

Coordenações indiretas são criadas usando variáveis de operadores. Sempre que uma expressão é avaliada, esses operadores assumem o valor atual da sua variável correspondente. Quando um controle possui uma dependência direta sobre uma variável, que contém uma projeção ou filtro, também depende indiretamente de quaisquer variáveis referenciadas pelas expressões da projeção ou do filtro.

Por meio dos operadores das variáveis, as expressões não dependem somente dos parâmetros de navegação e seleção de uma visão, mas também de suas projeções, filtros e do conjunto de dados. Essa dependência é utilizada para a agregação, agrupamento e indexação das visões aninhadas e para outros tipos de consultas. O funcionamento da propagação da interação em uma coordenação indireta é mostrado na Figura 2.16. Um objeto propaga a mudança de um valor para uma variável (1). A variável notifica todos valores léxicos que contêm expressões que a referenciam (2). Cada expressão notifica as variáveis que tenham valores atribuídos (3). Por fim, a variável envia uma notificação de alteração a todos os outros objetos (4).

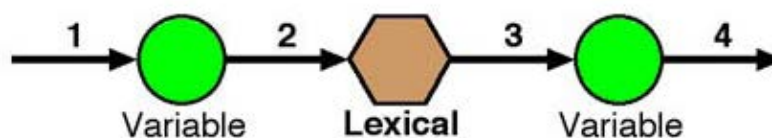


Figura 2.16: Coordenação indireta no sistema *Improvise*. Extraído de Weaver (2004).

2.4 Técnicas de coordenação entre mapas de similaridade

Uma técnica de coordenação faz um mapeamento entre múltiplas visões, relacionando elementos visuais presentes em cada visão, que por sua vez representam elementos dos dados projetados (North, 2000). Eler (2011) propôs novas técnicas de coordenação para permitir a exploração de conjuntos de dados a partir de mapas de similaridade. As técnicas de coordenação propostas foram implementadas na *Projection Explorer* (PEX), uma ferramenta desenvolvida em Java usada para criar e explorar representações visuais de, por exemplo, coleções de documentos. Na PEX cada documento é representado por um círculo projetado em um plano. Se dois círculos são próximos, o conteúdo dos documentos por eles representados é semelhante. Se os círculos estão distantes os documentos apresentam baixa correlação. Outro recurso disponibilizado pela PEX é a projeção simultânea de duas ou mais visões. Essas projeções podem ser de uma mesma base de dados ou de bases distintas (Paulovich et al., 2007).

Para exemplificar, na Figura 2.17 são apresentadas duas projeções de uma coleção de documentos. Ao selecionar um conjunto de documentos projetados na visão da esquerda, são identificados na visão da direita os documentos relacionados por meio de uma técnica de coordenação. Usando essa representação gráfica é possível identificar grupos de documentos similares e verificar a correlação entre diferentes áreas do conhecimento. Além disso, é possível selecionar grupos de documentos e criar rótulos para tais grupos, identificando os tópicos a serem discutidos (Paulovich et al., 2007).

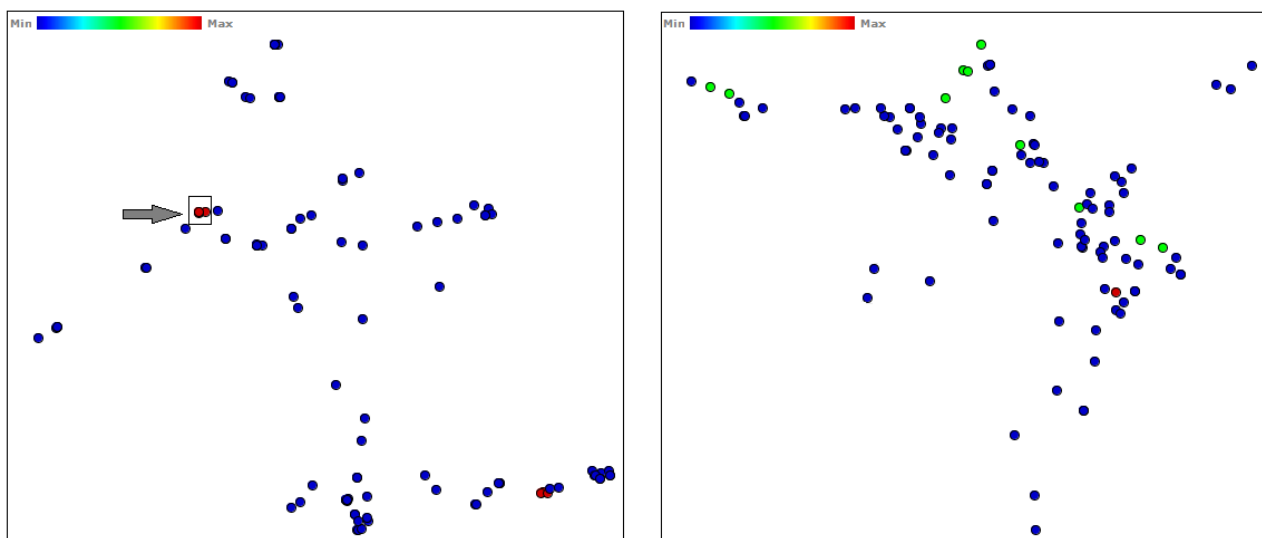


Figura 2.17: Projeção de um conjunto de documentos pela PEX

O modelo de coordenação responsável por definir e gerenciar as técnicas de coordenação presentes na PEX é adaptado do modelo proposto por Boukhelifa e Rodgers (2003). O diferencial está na inclusão de novas técnicas de coordenação que possibilitam a criação de diferentes tipos de mapeamentos (relações criadas entre as instâncias).

As principais classes desse modelo de coordenação são ilustradas na Figura 2.18. A classe **Viewer** representa uma visão do sistema, enquanto a classe **Mapping** representa uma técnica de coordenação responsável por criar e aplicar um mapeamento entre instâncias de duas visões (**outer** e **local**).

A responsabilidade por ecoar os eventos de coordenação ativados pelas interações do usuário é

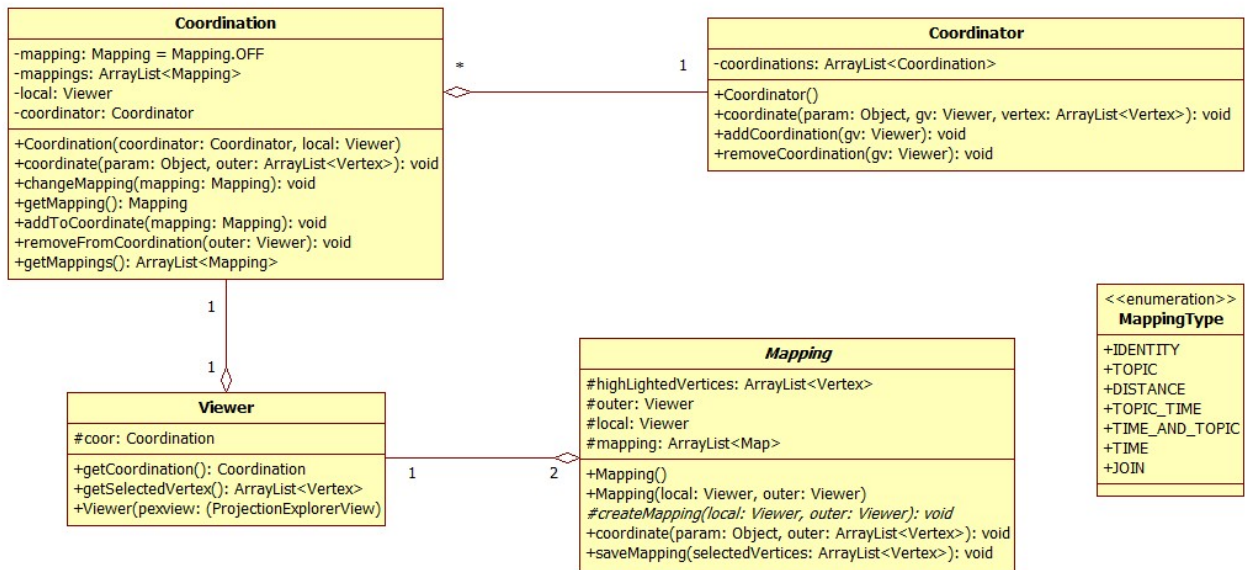


Figura 2.18: Classes do modelo de coordenação

da classe **Coordinator**, que possui uma lista das técnicas de coordenação definidas, apresentadas a seguir:

- *Identity Coordination* (Coordenação baseada em identificadores): é a técnica de coordenação mais simples. Utiliza identificadores das instâncias da visão origem para destacar itens que possuem o mesmo identificador na visão destino;
- *Distance Coordination* (Coordenação baseada em distâncias): a partir de uma seleção de um grupo de instâncias na visão origem, são destacadas na visão destino as k instâncias mais próximas das que tiverem sido selecionadas inicialmente;
- *Topic Coordination* (Coordenação baseada em tópicos): são selecionados documentos na visão origem, então, a partir dessa seleção são gerados tópicos. Durante o processo de extração dos tópicos é criada uma matriz de documentos e termos, em que cada linha representa um documento e cada coluna um termo ocorrente nesse documento. O valor dos elementos da matriz indica a frequência de ocorrência do termo. Assim, se os tópicos são encontrados em instâncias na visão destino, elas são destacadas;
- *Topic Time Coordination* (Coordenação baseada em coordenação de tópicos no tempo): utilizam-se do tempo para avaliar a evolução das instâncias de um conjunto de dados, ou seja, identificam tópicos que venham a ter relações durante a adição de novos documentos à coleção;
- *Time Coordination* (Coordenação baseada em atributos temporais): esta técnica explora o fato de que muitos dados possuem informações temporais. Realiza comparações (igual, menor, maior, menor ou igual, maior ou igual) entre os atributos temporais na visão origem, com os atributos das visões destino. Um exemplo é a seleção de documentos em que o atributo ‘ano’ seja maior que o da instância selecionada;

- *Join Coordination* (Combinação de coordenações): combina diversas técnicas de mapeamento em uma única coordenação. O usuário seleciona as técnicas que deseja combinar e, se necessário, fornece parâmetros.

Um exemplo de exploração de coleções de documentos utilizando a técnica *Topic Coordination* é apresentado na Figura 2.19. Primeiro o usuário seleciona um grupo de documentos na visão origem (Figura 2.19 (a)), a partir do qual é obtido um tópico a partir do conteúdo dos documentos. Em seguida, uma técnica de coordenação relaciona na visão destino (Figura 2.19 (b)) documentos que contém as palavras (termos) do tópico gerado.

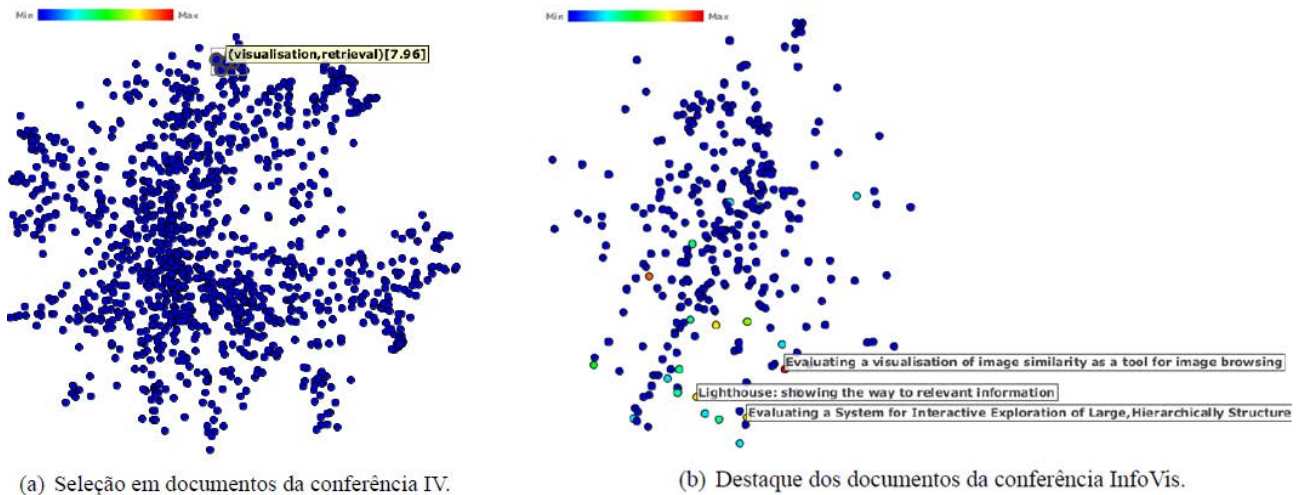


Figura 2.19: Coordenação entre dois mapas de similaridade. Seleção de documentos em (a) e destaque de documentos em (b), extraído de [Eler \(2011\)](#)

2.5 Aplicações de múltiplas visões coordenadas

Na literatura são encontradas diferentes aplicações de CMV em diversas áreas de pesquisa. São apresentadas nas próximas subseções algumas aplicações na exploração de dados textuais, com e sem o uso de ontologias no processo exploratório.

2.5.1 Aplicações em dados textuais

Dentre as aplicações de CMV em exploração de textos, a proposta de [Chang et al. \(2007\)](#) utiliza técnicas de coordenação para analisar dados, estruturados ou não, de transações financeiras na ferramenta chamada *Wire Vis*. Nessa abordagem, é usada uma lista de palavras-chave desenvolvida por pesquisadores que indicam risco de fraude. Adicionalmente são coordenadas diferentes visões contendo informações financeiras. São elas: *Keyword Network View* e *Strings and Beads*.

- *Keyword Network View*: usada para representar as relações entre as palavras-chave, que são conectadas por um grafo. Quando uma palavra-chave é selecionada, são destacadas as palavras relacionadas, e então traçadas linhas entre a seleção e os destaques, como ilustrado na Figura 2.20;

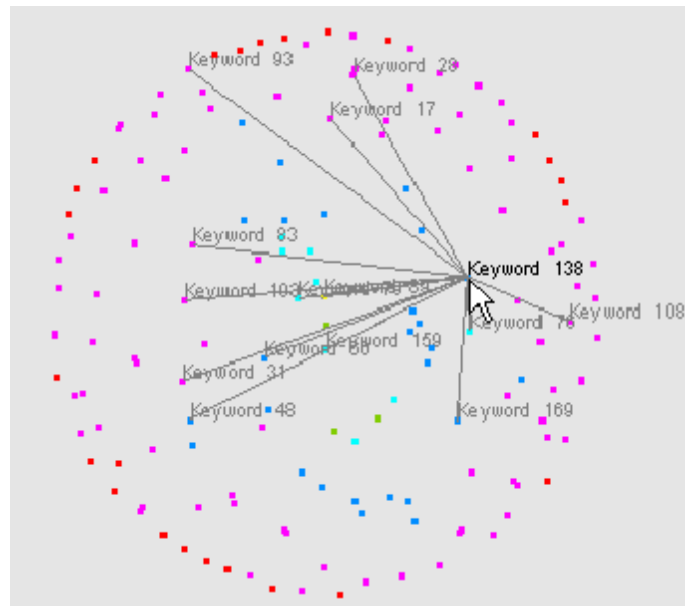


Figura 2.20: *Keyword Network View*: mostra relacionamentos entre palavras-chave. Extraído de Chang et al. (2007)

- *Strings and Beads*: são descritas as operações financeiras ao longo do tempo. As linhas do gráfico (*strings*) representam transações sobre o tempo – a menor porção de linha (*bead*) representa as transações realizadas em um dia determinado. Quando um *bead* é selecionado, as transações agrupadas por ele são selecionadas em uma visão separada, identificando quais palavras-chave estão relacionadas, como ilustrado na Figura 2.21.

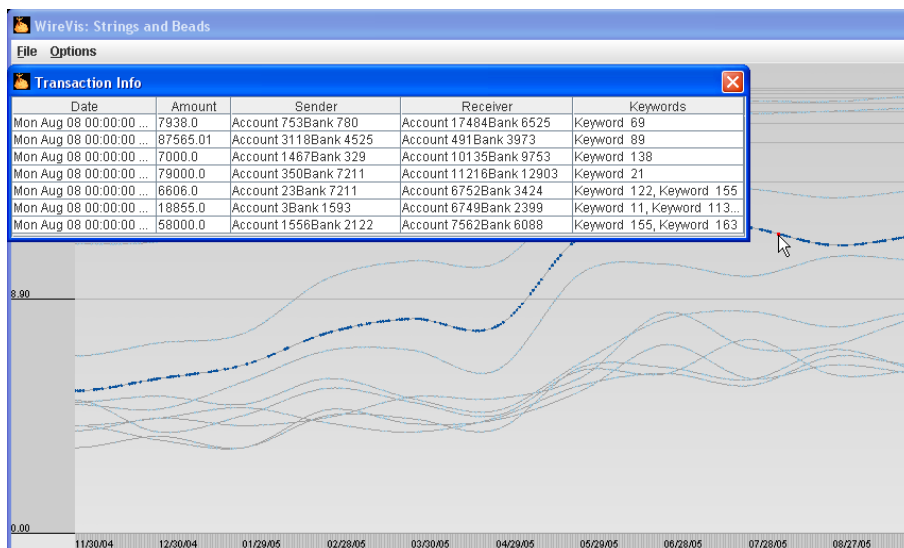


Figura 2.21: Visão *String and Bead*: clicando duas vezes sobre um *bead* específico mostra as transações relacionadas em uma visão separada. Extraído de Chang et al. (2007)

Outra ferramenta baseada em CMV é a *VisTools*, usada para a análise visual de coleções de documentos (Sabot et al., 2009). Para explorar dados, a *VisTools* fornece componentes visuais e também proporciona meios para combinar e integrar dados à interfaces do usuário. Um exemplo de exploração de documentos após a escolha de termos de interesse do usuário é ilustrado na Figura 2.22. Nesse exemplo, quatro visões são integradas a uma única interface visual coordenada.

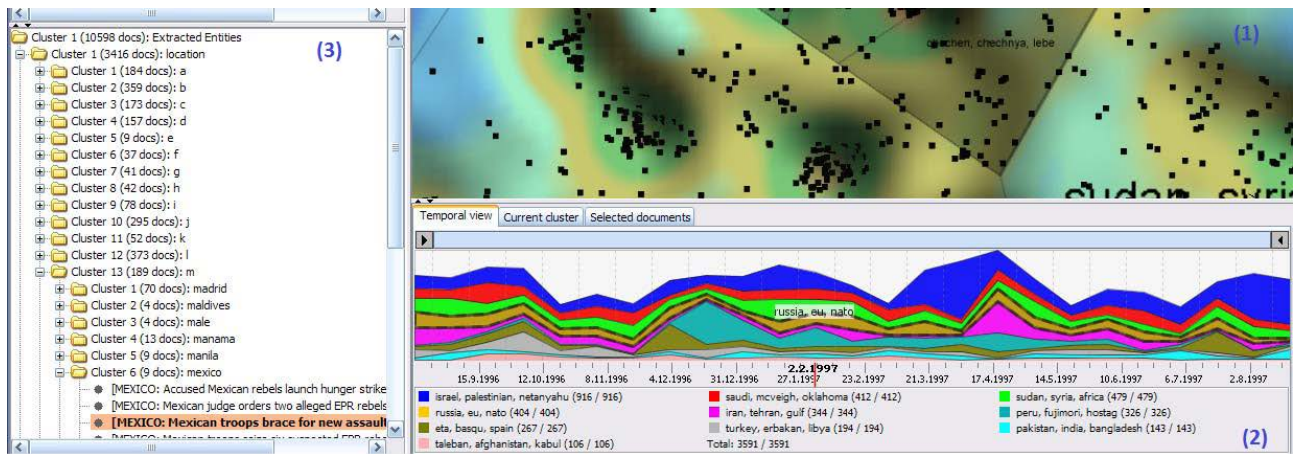


Figura 2.22: Exploração de uma coleção de documentos por meio do *framework VisTools*, extraído de Sabol et al. (2009)

Na visão indicada por (1) são mostrados tópicos relacionados em formato de paisagem. A visão indicada por (2), baseada na técnica *Theme River* (Havre et al., 2002), consiste em uma visão do tipo *StreamView*, que mostra a evolução temporal dos tópicos. A visão indicada por (3) é constituída de duas árvores (*TreeView*), a primeira mostra um aglomerado dos tópicos, e a segunda exhibe as entidades nomeadas agrupadas por sua classe (como pessoas ou localizações). Além dessas visões, o *framework* disponibiliza duas visões de tabelas (escondidas pelo painel de separadores na Figura 2.22), sendo que uma mostra sub-grupos dos tópicos selecionados e a outra os documentos selecionados.

Essas visões são integradas por uma coordenação que promove uma análise simultânea e temporal dos conjunto de dados. O usuário pode interagir modificando as propriedades visuais em qualquer visão. Enquanto o usuário interage, os itens visualizados (cor, seleção, ícone e tamanho) permanecem consistentes em todas as visões. No exemplo ilustrado na Figura 2.22, os tópicos “russia”, “ue”, “nato” foram escolhidos pelo usuário como termos de seu interesse. Como consequência, esses termos são destacados na *TreeView* e na *StreamView*. A atualização de características visuais, como cor e seleção de documentos, também pode ser vista na *TreeView*.

2.5.2 Aplicações com o uso de ontologias

A seguir são apresentados trabalhos relacionados que aplicam ontologias em CMV, embora não explorem a estrutura de relações que uma ontologia fornece. Seeling e Becks (2004) propõem um sistema chamado *SWAPit*, no qual ontologias são utilizadas para categorizar os temas que definem cada documento, baseados em similaridade, categorias e atributos relacionados. O sistema é projetado para auxiliar os usuários a explorar as associações entre os dados, incluindo informações textuais, ontologias, metadados do documento, e conjuntos de dados relacionais.

Conceitualmente, as visões são acopladas pelas associações entre os diferentes itens de dados. Além disso, o sistema proporciona a exploração simultânea de um repositório de textos categorizados na similaridade, nas descrições de metadados de documentos (compreendendo pares atributo–valor) e em dados relacionais associados aos documentos. De acordo com os diferentes tipos de informação que precisam ser integrados no ambiente multivisão, são utilizados os seguintes componentes:

- *Similarity View*: indicada na Figura 2.23 por *Mapa de Documentos*, auxilia o usuário a explorar

um conjunto de documentos textuais não estruturados com base no conteúdo do documento. Exibe associações entre esses documentos com base em uma medida de similaridade entre cada par de documentos. Visualiza a estrutura do espaço dos agrupamentos dos documentos.

- *Tree View*: indicada na Figura 2.23 por *Árvore de Ontologias*, proporciona ao usuário a navegação por coleções de documentos de um domínio específico, ou pelos catálogos dos tópicos. Podem ser atribuídos a cada documento da coleção, um ou mais tópicos desses catálogos.
- *Relational View*: indicada na Figura 2.23 por *Lista de Documentos*, apresenta dados relacionais estruturados associados aos documentos, como tabelas de dados relacionais. Existem dois tipos de dados relacionais: os metadados de documentos e os dados que estão relacionados com os documentos (por exemplo, as vendas ou registros de produtos). Qualquer conjunto de tuplas de dados relacionais, e que são especificadas por uma consulta, podem ser incluídos nesta visão.
- *Tool Area*: indicada na Figura 2.23 por *Ferramentas*, oferece funções para a categorização e navegação dos documentos.

A *Tree View* fornece esquemas para a classificação hierárquica dos documentos, assim cada documento pode ser associado a diferentes classes em um esquema. Pode ser usada para auxiliar a navegação em conjuntos de documentos, permitindo que o usuário identifique esses documentos por meio da sua classificação.

Na Figura 2.23 é apresentada uma interface com quatro painéis visuais, que são codificados por cores para sinalizar a origem de uma seleção. A *Tool Area* é marcada pela cor rosa. O painel do mapa de documentos (*Similarity View*) é cercado por um retângulo amarelo, enquanto que a área de tabelas (*Relational View*) é marcada em vermelho e a área do domínio de ontologias (*Tree View*) é marcada por azul. Sempre que o usuário seleciona um determinado conjunto de documentos em um painel, a seleção é simultaneamente espelhada nas outras visões coordenadas.

Quando o usuário faz alguma seleção na *Tree View*, todos os documentos associados aos nomes selecionados são destacados na *Similarity View* por quadrados azuis, que cercam os respectivos ícones dos documentos. Na *Relational View*, as tuplas associadas aos documentos correspondentes são destacadas em vermelho.

Um segundo sistema, chamado IVEA, é proposto por Thai et al. (2008). Nesse sistema coleções de textos são exploradas com o uso de ontologias e os componentes de visualização representam escopos de interesse do usuário. Assim, os usuários são envolvidos no estágio inicial do processo de visualização, definindo seus escopos de interesse, as entidades importantes e suas relações. A estrutura hierárquica dos escopos definidos na ontologia permite que os usuários interpretem vários aspectos de uma coleção de textos em diferentes níveis de detalhamento.

Cada documento da coleção pode ser representado por uma tupla constituída pelos números das variáveis (atributos), cujos valores correspondem aos valores de relevância do documento, relacionados às instâncias da ontologia. Esse valor de relevância é o valor agregado de um documento em relação a uma classe, no que diz respeito a todas as suas instâncias diretas e de maneira recursiva, de todas as suas subclasses. Os pesos dos termos nos documento de uma instância são utilizados como o valor de relevância do referido documento em relação a essa instância.



Figura 2.23: Interface do sistema SWAPit, adaptado de Seeling e Becks (2004)

A interface do IVEA é apresentada na Figura 2.24. A *collection overview*, localizada no canto superior direito, mostra uma matriz baseada em componentes multi-dimensionais. Também apresenta um painel que contém um visualizador de documentos, permitindo que os usuários leiam o conteúdo dos documentos no IVEA.

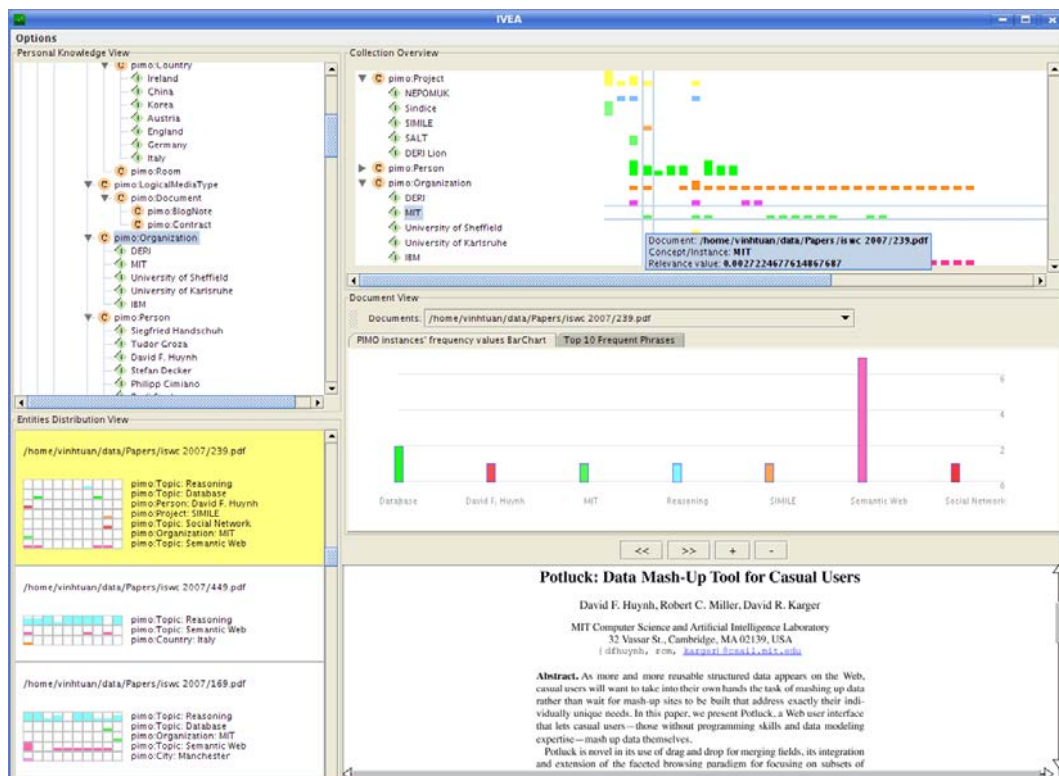


Figura 2.24: Interface do sistema IVEA, extraído de Thai et al. (2008)

Adicionalmente, ao passar o *mouse* sobre uma célula da matriz são exibidas *tooltips* que indicam a associação do nome do documento com base em sua coluna, além da classe ou instância com base na sua linha. Também é exibido o valor de relevância do referido documento no que diz respeito ao seu escopo. Com isso, os usuários podem focar em determinados escopos e, em seguida, visualizar os dados relevantes. Isso pode ser feito arrastando uma classe ou uma instância da *personal knowledge view* e soltando-a para a *collection overview*. Uma vez que as relações hierárquicas são levadas em consideração, há a inclusão de todas as suas ocorrências diretas e, recursivamente, a todas as suas subclasses.

Por exemplo, na Figura 2.24 os usuários podem arrastar a classe “*pimo:Organization*” da *personal knowledge view* e soltando-a na *collection overview*. Nesse caso, todas as organizações definidas como instâncias de “*pimo:Organization*” são incluídas automaticamente na hierarquia da *personal knowledge view*. Analisando as colunas, os usuários podem descobrir quais organizações de seu interesse cada documento menciona, bem como a distribuição das entidades da organização nos documentos, analisando as linhas.

Os usuários podem usar valores agregados e relevantes para a classe “*pimo:Organization*” para ver qual documento é o mais relevante para todas as organizações em questão. Com isso, a relação hierárquica entre os elementos da ontologia permite inspecionar uma coleção de textos sob diferentes níveis de granularidade, expandindo ou minimizando os nós que representam classes na estrutura hierárquica.

2.6 Considerações Finais

Neste capítulo são apresentados os conceitos relativos a técnicas de visualização exploratória, assim como conceitos de múltiplas visões coordenadas (Coordinated and Multiples Views - CMV), com foco no processo de coordenação dessas visões. Ainda, são exibidos modelos de coordenação que implementam técnicas responsáveis pelo processo de coordenação entre as múltiplas visões.

Os modelos apresentados possuem diferenças principalmente quanto à arquitetura. Em comum, apenas a existência de uma técnica de coordenação baseada em identificadores que mapeia elementos em visões destino que apresentem um mesmo identificador dos elementos selecionados da visão origem. Uma aplicação de CMV encontrada na literatura é a exploração de coleções de documentos. Nesse contexto, Eler (2011) desenvolveu técnicas de coordenação implementadas na ferramenta *Projection Explorer (PEX)*. Em comum, essas técnicas também utilizam identificadores das instâncias para realizar o mapeamento – durante o processo de coordenação é realizado um mapeamento léxico entre os elementos.

A realização de um mapeamento semântico no processo de coordenação entre as visões pode associar itens não relacionados sintaticamente. Para isso, **ontologias** apresentam um conjunto de características que podem ser utilizadas na execução do mapeamento semântico. Contudo, considerando o uso de ontologias nas abordagens descritas neste capítulo, apenas alguns dos seus recursos são explorados: principalmente a hierarquia, utilizada para contextualizar os documentos em seus respectivos tópicos. No próximo capítulo são apresentados conceitos de ontologias, em especial o processo de *matching*, utilizado para relacionar elementos externos com elementos presentes em ontologias.

Ontologia

3.1 Considerações iniciais

O termo *ontologia*, introduzido por Aristóteles no início século XVII, era originalmente relacionado a questões filosóficas, como a existência dos entes (Campos et al., 2011). Na área da Ciência da Computação, e especificamente da Inteligência Artificial, está relacionado com a organização de dados e com a representação formal do modelo de uma porção da realidade (Noy e McGuinness, 2001).

Ontologias modelam e especificam um domínio de conhecimento por meio de conceitos e axiomas lógicos, descrevendo o significado de um vocabulário formal (Garcia et al., 2008, Guarino et al., 2009). Os axiomas lógicos representam hierarquias, conceitos e seus relacionamentos, podendo ser usados para restringir a informação, verificar sua correção, ou deduzir novas informações (Gruber, 1995, Guarino, 1998, Guarino et al., 2009). Ainda segundo esses autores, os benefícios de se usar ontologias são:

- Possibilitar o compartilhamento de conhecimento, ou seja, proporcionar o entendimento comum de um conceito. Para isso, pode-se usar ontologias em aplicações de mesmo domínio de conhecimento;
- Fornecer um vocabulário para a representação de conhecimento, evitando interpretações ambíguas;
- Organizar dados para que a informação seja recuperada de maneira hábil. O uso de ontologias permite a utilização de anotações semânticas, facilitando a busca por informação quando os dados se encontram de maneira não estruturada;
- Relacionar informações semelhantes provenientes de fontes diversas, por meio da definição de conceitos e suas relações;

- Obter novas informações a partir de axiomas.

Neste capítulo é dada uma visão geral em ontologias, fundamentais para a técnica de coordenação apresentada no capítulo a seguir. Na próxima seção são apresentadas linguagens para definir ontologias. Em seguida, na Seção 3.3, é exibida a estrutura de uma ontologia. Na Seção 3.4 são apresentados conceitos do processo de *matching*. Na Seção 3.5, são mostradas técnicas de *matching* entre elementos externos e ontologias. Por fim, na Seção 3.6 são expostas as considerações finais.

3.2 Linguagens para Definir Ontologias

Uma ontologia deve ser passível de ser processada por uma máquina, por isso há a necessidade de uma linguagem para representá-la. O início dessas linguagens se deu no começo dos anos 90 e foi uma evolução das linguagens de representação do conhecimento existentes. Com isso, linguagens como **RDF**, **OIL** e **OWL** passaram a ter a sintaxe baseada em linguagens de marcação existentes, como HTML e XML (Guarino et al., 2009).

Na Linguagem **RDF** (*Resource Description Framework*), desenvolvida pelo W3C (*World Wide Web Consortium*), o conhecimento é representado utilizando redes semânticas. Objetos e suas relações são representados em sintaxe XML, permitindo a especificação da semântica dos dados de uma maneira padronizada. O modelo de dados é definido por meio de três componentes: recursos, propriedades e declarações. *Recursos* são descritos por expressões RDF e são sempre nomeados por URI's ¹. As *propriedades* definem aspectos específicos, características, atributos, ou relações usadas para descrever um recurso. As *declarações* atribuem a um recurso específico (*sujeito*), um valor para uma propriedade (*predicado*). Os valores das propriedades são denominados *objetos*, que podem ser *strings* simples ou outro tipo de dado primitivo, definido por XML ².

A linguagem RDF é um modelo usado para representar propriedades nomeadas e seus valores. Um dos seus objetivos é tornar possível a especificação da semântica dos dados com base no XML, definindo o significado da estrutura por meio das *tags* especificadas em seu modelo. *Tags* identificam classes, subclasses e propriedades, além de estabelecer restrições do domínio do conhecimento descrito. Adicionalmente, possui um sistema de classes similar à programação orientada a objetos e a sistemas de modelagem, em que uma coleção de classes é chamada de esquema. Essas classes são organizadas em uma hierarquia, e proporcionam extensibilidade por meio do refinamento das subclasses.

O modelo de dados RDF não fornece mecanismos para definir as relações entre as propriedades (atributos) e recursos, por outro lado a linguagem **RDFS** (*RDF Schema*) define os conjuntos de recursos descritos, não apenas as propriedades dos recursos, mas também seus objetos e suas classes.

A linguagem **OIL** (*Ontology Interchange Language*), desenvolvida pelo projeto *OntoKnowledge*, permite a interoperabilidade semântica entre recursos Web. Estende o modelo RDF, combinando primitivas de modelagem das linguagens baseadas em *frames* com lógica descritiva (Fensel, 2001). Também representa a semântica da informação para o uso na Web, não apenas a sintaxe. Assim, é possível realizar pesquisas em torno do conceito semântico, contido nas páginas Web, e não apenas

¹Uniform Resource Identifier - string de caracteres usada para identificar um recurso

²<http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

por palavras-chave.

3.2.1 OWL (*Web Ontology Language*)

A recomendação atual W3C para linguagens para ontologias é a **OWL** (*Web Ontology Language*), uma linguagem baseada nas linguagens DAML e OIL e apoiada pelas arquiteturas XML e RDF. Apresenta novas estruturas específicas para ontologias e adiciona propriedades como cardinalidade, equivalência e restrições. A partir das descrições é possível descrever classes, propriedades (funcional, transitiva, simétrica ou inversa) e suas instâncias. Além disso, OWL é baseada na lógica de descrições, em que um modelo lógico define conceitos, podendo descrever relações por meio de conjuntos de operadores lógicos. Também possibilita o uso formal de semântica, permitindo usar um sistema que computa inferências com base no modelo lógico (*reasoner*), por meio das propriedades das classes. Alguns exemplos de inferências que podem ser computadas são³:

- Associação de classes: Se x é uma instância de uma classe C , e C é uma subclasse de outra classe D , pode-se inferir que x é uma instância de D ;
- Equivalência de classes: Se a classe A é equivalente à classe B e a classe B é equivalente à classe C ; então A é equivalente a C ;
- Consistência: se x é declarada uma instância da classe A , que por sua vez é uma subclasse de B , e se A e B são declaradas disjuntas, tem-se uma inconsistência que aponta para um provável erro na ontologia;
- Classificação: se é declarado que certas propriedades são condições suficientes para que um indivíduo x seja membro da classe A , e se esse indivíduo x satisfaz estas condições, então o mesmo deve ser uma instância de A .

OWL fornece três sublinguagens diferentes, em que cada uma provê um conjunto de características com o intuito de cumprir diferentes aspectos para o seu uso. Dentre seus usuários, podem-se destacar as comunidades específicas de implementadores, que se utilizam das seguintes linguagens:

- OWL Full: é como a linguagem completa é chamada. Sua vantagem é ser inteiramente compatível com a linguagem RDF, tanto sintática como semanticamente, ou seja, qualquer documento RDF também é um documento OWL Full;
- OWL DL: é assim chamada por sua correspondência com a lógica descritiva (*Description Logic*). Também inclui todas as construções da linguagem OWL, mas que apenas podem ser usadas em certas restrições (por exemplo: enquanto uma classe pode ser uma subclasse de muitas classes, uma classe não pode ser uma instância de outra classe). A vantagem deste procedimento é que ele permite um apoio eficiente ao *reasoning*. A desvantagem é a falta de compatibilidade com RDF;

³<http://www.w3.org/TR/owl-features/>

- **OWL Lite:** oferece suporte aos usuários que necessitam de uma classificação por meio de hierarquias e de restrições simples. Exclui classes enumeradas, afirmações disjuntas e cardinalidade arbitrária. Apresenta como vantagem um poder maior de compreensão para os usuários e maior facilidade de implementação para os desenvolvedores. Por outro lado, sua expressividade é restrita.

3.2.1.1 Estrutura de um documento OWL

A estrutura de um documento em OWL se baseia na sintaxe da linguagem RDF (também baseado no XML) e as instâncias são definidas usando descrições RDF. A semântica formal e o suporte ao *reasoning* são fornecidos por meio do mapeamento OWL em lógicas de predicado e descritiva. A seguir são apresentados os principais elementos da estrutura OWL.

O *header* contém elementos como a raiz e asserções. Documentos OWL são geralmente chamados de ontologias OWL, e também são documentos RDF. Assim, o elemento raiz de uma ontologia OWL é identificado por um elemento *rdf:RDF*, que também especifica os *namespaces*, como exemplifica o trecho de Código 3.1.

Código 3.1: Exemplo de uma raiz de uma ontologia OWL

```
<rdf:RDF
  xmlns:owl = " http: //www. w3. org /2002/07/ owl#"
  xmlns:rdf = " http: //www. w3. org /1999/02/22 - rdf -syntax -ns#"
  xmlns:rdfs = " http: //www. w3. org /2000/01/ rdf -schema#"
  xmlns:xsd = " http: //www. w3. org /2001/ XLMSchema#">
```

Uma ontologia OWL pode ter seu início constituído por uma coleção de afirmações agrupadas sob um elemento *owl:Ontology*, composto de comentários, controle de versão e inclusão de outras ontologias – cujos conteúdos passam a ser considerados parte da ontologia atual (por meio da asserção *owl:imports*). Um exemplo desse conjunto de afirmações é mostrado no trecho de código 3.2

Código 3.2: Exemplo de um conjunto de afirmações em uma ontologia OWL

```
<owl:Ontology rdf:about="">
  <rdfs:comment>An example OWL ontology</rdfs:comment>
  <owl:priorVersion
    rdf:resource=" http: //www. mydomain . org / uni -ns -old " />
  <owl:imports rdf:resource=
    " http: //www. mydomain . org / persons " />
  <rdfs:label>University Ontology</rdfs:label>
</owl:Ontology>
```

Classes são definidas por um elemento *owl:Class* e durante a sua declaração podem ser incluídas definições – por exemplo, declarar que duas classes são equivalentes (pelo uso do elemento *owl:equivalentClass*) ou disjuntas (pelo uso do elemento *owl:disjointWith*). Uma nova classe também pode ser obtida por meio de combinações de outras classes. No Código 3.3 as classes *professor* e *professorAssistente* são declaradas disjuntas. Além disso, existem duas classes predefinidas: *owl:Thing*, a classe

mais geral que contém tudo; e *owl:Nothing*, uma classe vazia. Assim, cada classe é uma subclasse de *owl:Thing* e uma superclasse de *owl:Nothing*.

Também são explorados os conceitos de *subclasses*, especificadas por meio do elemento *rdfs:subClassOf*, em que é especificado que uma classe *C* é uma subclasse de outra classe *C'*. Isso implica em declarar que cada instância de *C* também é uma instância de *C'*.

Código 3.3: Exemplo de disjunção de classes

```
<owl:Class rdf:about="#professorAdjunto">
  <owl:disjointWith rdf:resource="#professor"/>
  <owl:disjointWith rdf:resource="#professorAssistente"/>
</owl:Class>
```

Os elementos representados em OWL possuem propriedades que podem ser classificadas em dois tipos: propriedades do objeto (representado por *owl:ObjectProperty*), que relacionam objetos (instâncias de classes) com outros objetos; e propriedades dos tipos de dados (representado por *owl:DatatypeProperty*), que relacionam objetos com valores para os tipos de dados. É possível relacionar “propriedades inversas”, por meio do elemento *owl:inverseOf*. Um exemplo desse tipo de propriedade inversa é apresentado no trecho de Código 3.4, em que as propriedades “leciona” e “ensinadoPor” são declaradas inversas.

Código 3.4: Exemplo de propriedades de projeto

```
<owl:ObjectProperty rdf:ID="leciona">
  <rdfs:range rdf:resource="#curso"/>
  <rdfs:domain rdf:resource="#membroCorpoDocente"/>
  <owl:inverseOf rdf:resource="#ensinadoPor"/>
</owl:ObjectProperty>
```

As propriedades dos elementos também podem possuir restrições, representadas pelos elementos *owl:Restriction* e *owl:onProperty*, somadas a uma ou mais declarações de restrição – com isso, pode-se criar conceitos ou classes. Declaração de restrição é aquela que define restrições sobre os tipos dos valores das propriedades:

- *owl:allValuesFrom*: usada para especificar quais são os valores possíveis das propriedades de uma classe. São referidas por *owl:onProperty*;
- *owl:hasValue*: declara um valor específico que a propriedade especificada por *owl:onProperty* deve ter;
- *owl:someValuesFrom*: a diferença entre esta e a restrição anterior é a não obrigatoriedade de todos os valores virem da classe em questão, ou seja, apenas alguns valores podem ser importados. No exemplo ilustrado no Código 3.5, a classe *membroEquipeAcademica* requer a existência de um curso de graduação, lecionado por uma instância da classe que representa um membro da equipe acadêmica.

Código 3.5: Exemplo de restrição de propriedades: todos os membros da equipe acadêmica deve lecionar pelo menos um curso de graduação.

```
<owl:Class rdf:about="#membroEquipeAcademica">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#leciona"/>
      <owl:someValuesFrom rdf:resource
        ="#cursoGraduacao"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Outros tipos de restrições são de cardinalidade, expressas por: *owl:minCardinality* (usada para limitar inferiormente), *owl:maxCardinality* (usada para limitar superiormente), ou ainda *owl:cardinality* para expressar um número exato de cardinalidade. Existem propriedades especiais que podem ser definidas diretamente:

- *owl:TransitiveProperty*: define uma propriedade transitiva, como “maisAltoQue”, “ancestralDe”, etc;
- *owl:SymmetricProperty*: define uma propriedade simétrica, como “temMesmoGrauQue”, “irmãoDe”, etc;
- *owl:FunctionalProperty*: define uma propriedade que tem um único valor para cada objeto, como “idade”, “altura”, etc;
- *owl:InverseFunctionalProperty*: define uma propriedade para a qual dois objetos diferentes não podem ter o mesmo valor, por exemplo, a propriedade “numeroCPF” (o CPF é um número único para cada pessoa).

3.3 Estrutura de uma ontologia

O conhecimento em uma ontologia pode ser especificado por meio de entidades como classes, conceitos, taxonomias, relações, funções e instâncias (Calero et al., 2006). Além disso, axiomas também podem ser utilizados para restringir a semântica das ontologias. Por exemplo, o seguinte axioma descreve que uma autobiografia é uma biografia cujo tema é o autor:

$$\forall x, \text{Autobiografia}(x) \Rightarrow \exists y; \text{Pessoa}(y) \wedge \text{autor}(x, y) \wedge \text{tópico}(x, y)$$

Uma ontologia pode ser expressa por uma tupla: $o = (C, I, R, T, V, \leq, \perp, \in, =)$ (Euzenat e Shvaiko, 2007), em que:

C : conjunto de classes;

I : conjunto de instâncias;

R : conjunto de relações;

T : conjunto de tipos de dados;

V : conjunto de valores (C, I, R, T, V sendo disjuntos aos pares);

\leq : relação em $(C \times C) \cup (R \times R) \cup (T \times T)$ chamada especialização;

\perp : relação em $(C \times C) \cup (R \times R) \cup (T \times T)$ chamada exclusão;

\in : relação sobre $(I \times C) \cup (V \times T)$ chamada instanciação;

$=$: relação sobre $I \times R \times (I \cup V)$ chamada atribuição.

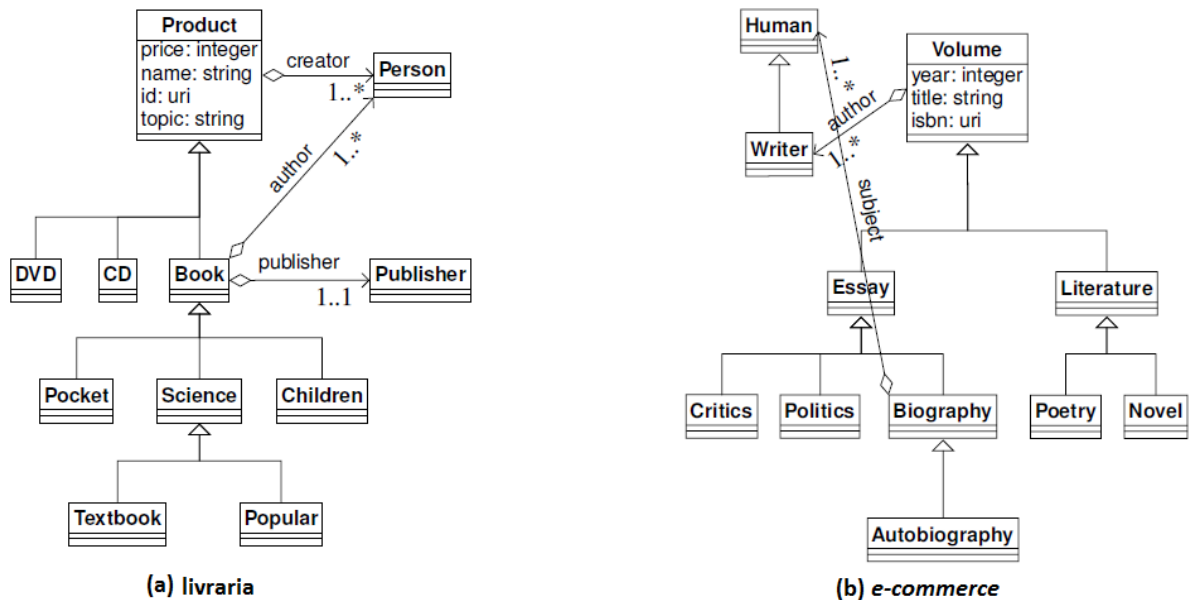


Figura 3.1: Dois modelos conceituais UML, extraído de *Euzenat e Shvaiko (2007)*

Na Figura 3.1 são apresentados dois diagramas de classes UML para ilustrar a estrutura de uma ontologia. À esquerda é mostrada a representação de uma livraria (indicada por **(a)**) e à direita um site *e-commerce* de livros (indicada por **(b)**). Em ambas as representações são descritos os itens por meio da especificação de suas propriedades e de uma taxonomia de classes, além de expressar as relações entre as classes. Nota-se que as classes *Writer* (Autor) e *Person* (Pessoa) são correspondentes, assim como as classes *Volume* (Volume) e *Book* (Livro) (*Euzenat e Shvaiko, 2007*).

Os fragmentos de ontologias da Figura 3.2 correspondem aos modelos UML apresentados na Figura 3.1. A diferença entre esses fragmentos é a existência de semântica, que fornece as regras para a interpretação de sua sintaxe. Essas regras não fornecem o significado dos conceitos diretamente, mas sim por meio de restrições.

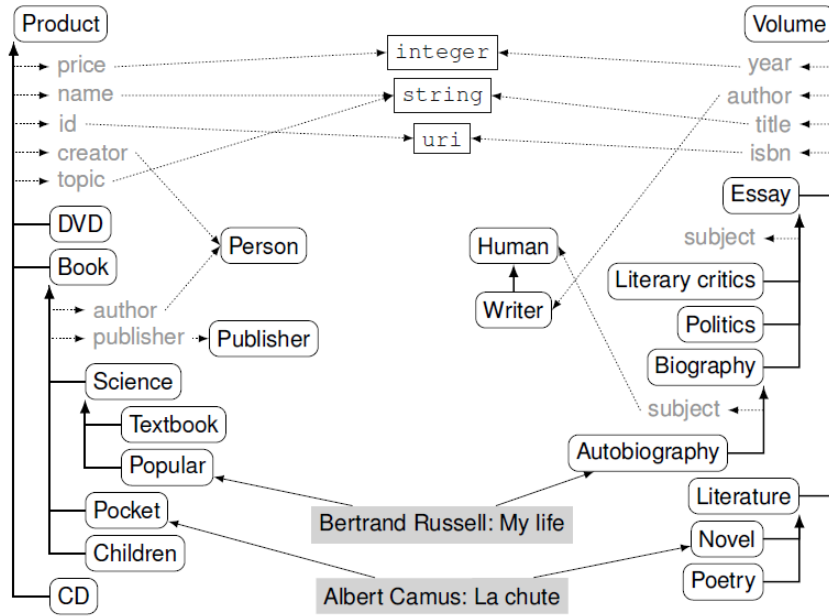


Figura 3.2: Fragmentos de ontologias, extraído de Euzenat e Shvaiko (2007)

3.4 Processo de *matching* de ontologias

No processo de *matching* um conjunto de correspondências deve ser encontrado, determinando um alinhamento A (conjunto de correspondências entre duas ou mais ontologias) para um par de ontologias o e o' . Alinhamentos expressam as correspondências entre entidades de ontologias diferentes. Além dos alinhamentos, há outros parâmetros envolvidos: pesos, *thresholds* e outros recursos externos usados pelo processo, como um tesouro de um domínio específico (Euzenat e Shvaiko, 2007).

O processo pode ser definido por uma função f , na qual a partir de um par de ontologias o e o' , um alinhamento de entrada A , um conjunto de parâmetros p e um conjunto de recursos de r , retorna um alinhamento A' entre essas ontologias, conforme ilustrado na Figura 3.3 (Euzenat e Shvaiko, 2007):

$$A' = f(o, o', A, p, r)$$

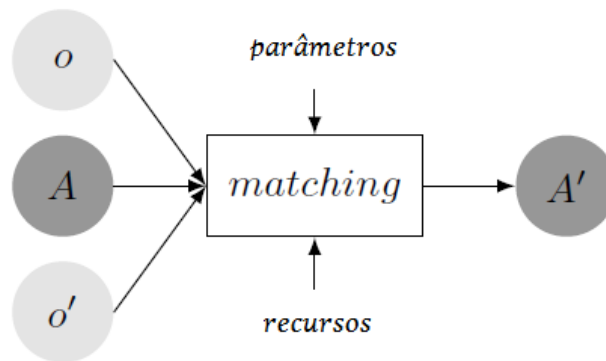


Figura 3.3: Processo de *matching*, adaptado de Euzenat e Shvaiko (2007)

Para a combinação de duas ou mais ontologias existe o processo de *matching* múltiplo. Esse processo pode ser visto como uma função f , na qual a partir de um conjunto de ontologias o_1, o_2, \dots, o_n , um alinhamento A , um conjunto de parâmetros p e um conjunto de recursos de r , retorna um alinhamento A' entre essas ontologias (Euzenat e Shvaiko, 2007):

$$A' = f(o_1, \dots, o_n, A, p, r)$$

Para realizar a correspondência entre duas ontologias se deve considerar as duas entidades correspondentes e a relação que é supostamente mantida entre elas. As correspondências, encontradas pelos algoritmos de *matching*, podem ser definidas formalmente. Dadas duas ontologias o e o' , a descrição da correspondência entre elas pode ser representada por uma 5-tupla (Euzenat e Shvaiko, 2007):

$$\langle id, e, e', r, n \rangle$$

em que:

id : identificador único da correspondência;

e / e' : entidades relacionadas à o e o' , respectivamente;

r : relação que pode ser de equivalência (\equiv), subsunção (\sqsubseteq) ou disjunção (\perp);

n : nível de confiança (contido no intervalo $[0, 1]$) para a correspondência entre e e e' .

A correspondência entre a 5-tupla afirma que existe uma relação r entre as entidades de ontologia e e e' com nível de confiança n . Para ilustrar esse conceito, é mostrado no exemplo a seguir as correspondências entre as entidades **Book** (e) e **Volume** (e'), com um nível de confiança máximo (1.0).

<http://book.ontologymatching.org/example/culture-shop.owl#Book> =_{1.0}
<http://book.ontologymatching.org/example/library.owl#Volume>

Um exemplo, com maior complexidade, expressa o caso de uma entidade *Livro* x escrito por um *Escritor* w . O autor de x na primeira ontologia é identificado pela concatenação do primeiro e do último nome de w . Então, o nível de confiança, nesse caso, é de 0.85 (esse cálculo depende das técnicas apresentadas na próxima seção). A partir de um conjunto de correspondências entre ontologias, pode-se definir um alinhamento. Dadas duas ontologias o e o' , Um alinhamento é realizado de um conjunto das correspondências entre pares de entidades pertencentes à o e o' .

Livro = _{1.0} Volume id ≥ _{0.9} isbn Pessoa = _{0.9} Humano	name ≥ _{1.0} title Autor = _{1.0} Autor Ciências ≤ _{0.8} Ensaio
---	---

O alinhamento entre as ontologias descrito acima está ilustrado na Figura 3.4, em que pares de entidades com nível de confiança alto são considerados correspondentes.

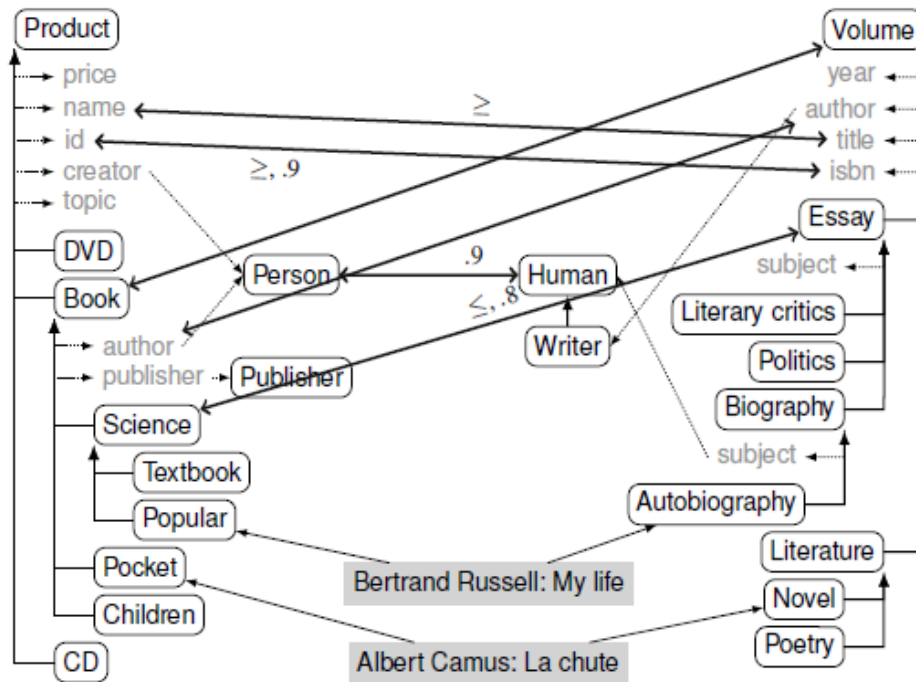


Figura 3.4: Exemplo de alinhamento entre ontologias, extraído de Euzenat e Shvaiko (2007)

3.5 Técnicas de *matching* de ontologias

Matching de esquemas/ontologias é um processo em que são encontradas relações ou correspondências entre fragmentos de informação que podem ser modelados usando diferentes estruturas e formatos, como esquemas de banco de dados, documentos XML ou ontologias (Euzenat e Shvaiko, 2007, Bellahsene et al., 2011). Segundo Euzenat e Shvaiko (2007) as técnicas de *matching* podem ser classificadas de acordo com as seguintes propriedades:

- Dimensões de entrada: relacionadas ao tipo de entrada sobre a qual os algoritmos de *matching* operam. Na primeira dimensão, os algoritmos podem ser classificados de acordo com os dados ou modelos conceituais em que as ontologias são expressas. A segunda dimensão depende dos tipos de dados que os algoritmos exploram;
- Dimensões do processo: baseiam-se nas propriedades gerais do processo e dependem da aproximação ou da natureza exata de seu cálculo. Outra perspectiva para a análise dos algoritmos de *matching* é baseada na maneira de interpretar os dados de entrada. Existem três grandes classes: *sintática* (com base na entrada intrínseca), *externa* (com base nos recursos externos), ou *semântica* (com base em alguma teoria semântica das entidades em questão);
- Dimensões de saída: relacionadas às maneiras em que resultados são produzidos. Por exemplo, a correspondência resultante pode ter como resultado um nível de confiança de 78%, cujo cálculo envolve fatores como a entrada e a técnica de *matching* utilizadas.

A classificação das técnicas de *matching* é ilustrada na Figura 3.5. A camada superior se baseia na granularidade e na interpretação da entrada, a inferior se baseia no tipo de entrada, enquanto que a

camada do meio apresenta as classes das técnicas básicas.

No primeiro nível, a camada é dividida em **níveis de elemento** (*element-level*) e **níveis de estrutura** (*structure-level*). No nível de elemento o cálculo das correspondências é feito analisando as entidades ou instâncias dessas entidades, ignorando suas relações com outras entidades ou suas instâncias. No nível de estrutura as correspondências são calculadas analisando-as como entidades.

No segundo nível, a classificação é dividida em: **sintático** (*syntactic*), **externo** (*external*) ou **semântico** (*semantic*). O nível sintático interpreta a entrada analisando apenas sua estrutura e seguindo algum algoritmo. O nível externo explora recursos auxiliares de um domínio e do conhecimento comum, a fim de interpretar a entrada. Esses recursos podem ser entradas do usuário ou algum dicionário, que expressem as relações entre os termos. Já o nível semântico usa semânticas formais, como o modelo teórico-semântico para interpretar a entrada e justificar os seus resultados. No caso de um sistema baseado no *matching* semântico os algoritmos são completos, pois garantem uma descoberta de todos os alinhamentos possíveis.

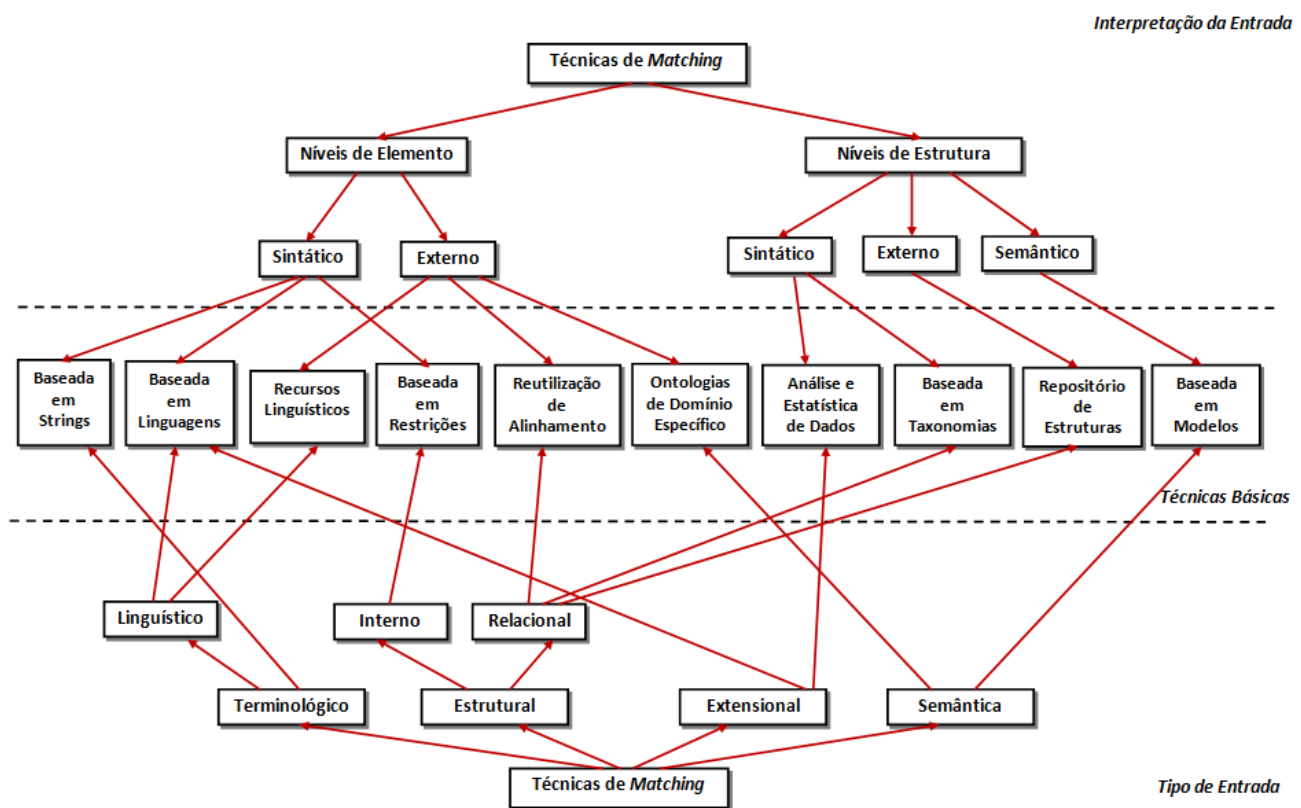


Figura 3.5: Técnicas de *Matching*, adaptado de [Euzenat e Shvaiko \(2007\)](#)

A camada inferior, que contém o **tipo de entrada** (*kind of input*), está relacionada com o tipo de entrada considerada por uma técnica particular. O primeiro nível é classificado de acordo com o tipo de dado em que os algoritmos trabalham, podendo ser terminológico (*strings*), estrutural (estruturas), semântico (modelos) ou extensional (instâncias). O segundo nível dessa classificação decompõe, se necessário, as categorias descritas acima em métodos terminológicos ou estruturais. Os métodos terminológicos podem ser baseados em *strings* ou na interpretação desses termos como objetos linguísticos. Os métodos estruturais são divididos em dois tipos de métodos: os que consideram a estrutura interna das entidades, como atributos e seus tipos (interno), e os que consideram a relação

entre as entidades (relacional).

A camada intermediária é constituída das técnicas básicas, que diferem umas das outras na maneira em que interpretam a informação de entrada, podendo ser interpretadas como *strings* (sequência de caracteres), frases em alguma linguagem natural, hierarquias (grafos) ou taxonomias. São denominadas **técnicas básicas**. A seguir são apresentadas as técnicas de *matching*.

3.5.1 Técnicas baseadas em *strings*

As técnicas baseadas em *strings* realizam o *matching* entre os nomes e as descrições dos nomes das entidades de ontologias. Quanto maior for a similaridade entre as *strings*, maior é a probabilidade das mesmas compartilharem os mesmos conceitos. Normalmente o cálculo é feito por funções que mapeiam um par de *strings* e retornam um número real. Quanto maior for o valor desse número, maior a similaridade entre as *strings*. Existem várias técnicas para se comparar *strings* (Euzenat e Shvaiko, 2007):

- Normalização: normaliza as *strings* antes da comparação. Todos os caracteres podem ser convertidos para caracteres minúsculos (por exemplo, DVD se torna dvd), a acentuação e hifenização retirada (anti-inflamatório se torna anti inflamatório), dígitos suprimidos (livro467 se torna livro) ou a pontuação é eliminada (RG. se torna RG). Após a normalização, *strings* são comparadas e se forem idênticas, a função de comparação retorna 1, caso contrário, retorna 0;
- Teste de *substrings*: duas *strings* são similares quando uma é uma *substring* da outra. A comparação de strings pode ser realizado pela similaridade *n-gram*, que calcula o número de sequências de *n* caracteres que são comuns às duas *strings*. Por exemplo, os *trigrams* ($n = 3$) para a *string* *paper* são: *pap*, *ape* e *per*. O cálculo da similaridade *n-gram* para duas *strings* *s* e *t* é definida pela equação:

$$\bar{\sigma}(s, t) = \frac{|ngram(s, n) \cap ngram(t, n)|}{\min(|s|, |t|) - n + 1}$$

- Distância de edição: calcula o custo mínimo de operações aplicadas em um objeto a fim de se obter o outro. Essas operações podem ser de inserção, deleção ou substituição de um caractere. A seguinte equação calcula o menor custo (por meio de um conjunto *Op* de operações) para transformar uma string *s* em uma string *t*, em que *w* é uma função para atribuir o custo da transformação.

$$\delta(s, t) = \min_{(op_i)_1:op_n(\dots op_1(s))=t} \left(\sum_{i \in I} w_{op_i} \right)$$

- Distância baseada em *token*: consideram uma *string* como um conjunto de palavras (*bag of words*), isto é, um conjunto em que um determinado item pode aparecer diversas vezes. Essa abordagem funciona melhor em textos com muitas palavras. Em geral, o conjunto de palavras *s* é considerado como um vetor \vec{s} , pertencente a um espaço vetorial *V* de *n* dimensões, em que cada dimensão corresponde a um *token* e cada posição do vetor corresponde a um número de ocorrências do *token* no conjunto de palavras;

- Comparação de caminhos: comparam não apenas os rótulos dos objetos, mas também a sequência dos rótulos das entidades. Por exemplo, na ontologia ilustrada na Figura 3.2 a classe *Science*, é identificada pelo caminho *Product:Book:Science*. Outro exemplo é a concatenação de nomes de classes e superclasses, em que o resultado é dependente da comparação de cada cadeia de *strings*.

3.5.2 Técnicas baseadas em linguagens

Essas técnicas consideram *strings* como palavras de alguma linguagem natural. São baseadas em técnicas de processamento de linguagem natural, que exploram morfológicamente as propriedades das *strings* de entrada. Diferentes *strings* podem representar os mesmos termos, por exemplo, *theory papers*, *theoretical paper* e *theoretical papers*. Por isso, é necessária a aplicação de métodos para distinção desses termos, por meio da **normalização linguística**, que visa a redução de um termo à alguma forma padronizada para que seja facilmente reconhecido. Em geral, são utilizadas técnicas para se chegar à normalização, apresentadas por [Euzenat e Shvaiko \(2007\)](#):

- Tokenização: separa uma *string* em uma sequência de *tokens*, reconhecendo pontuação, maiúsculas/minúsculas, espaços e dígitos. Por exemplo, *peer-reviewed periodic publication* se torna *<peer;reviewed;periodic;publication>*;
- Lematização: transforma os termos em seus radicais, eliminando tempos verbais, gênero e número. Por exemplo *reviewed* se torna *review*;
- Eliminação de termos: extratores identificam termos provenientes da repetição de frases morfológicamente semelhantes nos textos e a utilização de padrões. Por exemplo, *theory paper* é o mesmo que *paper theory*;
- Eliminação de *stop words*: *stop words* são palavras que aparecem com muita frequência em textos, mas que não apresentam significado adicional, como artigos e preposições. Por exemplo, *collection of article* se torna *collection article*.

3.5.3 Técnicas baseadas na estrutura

As estruturas das entidades encontradas nas ontologias podem ser comparadas. A comparação pode ser feita internamente, comparando atributos ou entidades com as quais são relacionados, ou comparando as estruturas relacionais (o conjunto de relações que uma entidade possui com outras entidades). A seguir são apresentados dois métodos de comparação baseados em estruturas ([Euzenat e Shvaiko, 2007](#)).

3.5.3.1 Métodos baseados em estruturas internas

Métodos baseados em estruturas internas analisam a estrutura interna das entidades e usam critérios como o conjunto e alcance de suas propriedades, sua cardinalidade ou multiplicidade, além dos tipos de relacionamento para calcular a semelhança entre elas. Elementos de duas estruturas são mostrados na Figura 3.6, porém métodos terminológicos de similaridade não classificam essas duas

estruturas como semelhantes. Nesse caso, a aplicação de métodos linguísticos pode ser mais vantajosa na procura de uma relação entre *creator* e *author*, por exemplo. A comparação entre elementos de estruturas internas, usa os seguintes componentes (Euzenat e Shvaiko, 2007):

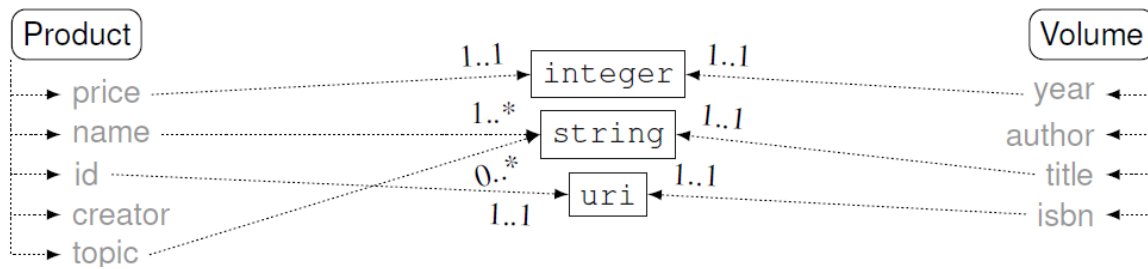


Figura 3.6: Comparação de dois conjuntos de propriedades, extraído de Euzenat e Shvaiko (2007)

- Comparação entre propriedades e chaves: classes são identificadas pelas chaves. Se há similaridades de mesmo tipo, há grande possibilidade das classes serem equivalentes;
- Comparação entre tipos de dados: entidades similares provavelmente apresentam atributos com tipos de dados similares. A proximidade entre os tipos de dados é máxima, quando esses dados são de mesmos tipos. É média quando os tipos são compatíveis, por exemplo, os tipos *integer* e *float*, pois podem ser convertidos um para o outro. E é considerada baixa quando não são compatíveis.

3.5.3.2 Métodos baseados em estruturas relacionais

Ontologias podem ser consideradas grafos, cujas arestas representam as relações entre as entidades (Euzenat e Shvaiko, 2007). A busca por correspondências é realizada por meio de buscas em subgrafos comuns entre as ontologias. Correspondência entre ontologias a partir de suas estruturas relacionais é mais eficiente que as demais técnicas, pois permite que todas as relações entre as entidades tenham sido consideradas. Existem três tipos de relações em técnicas de estruturas relacionais (Euzenat e Shvaiko, 2007):

- Estrutura de taxonomias: é o esqueleto das ontologias, construída por um grafo com relações do tipo *subClassOf*. Existem várias medidas propostas para comparação entre classes com base na estrutura taxonômica. As mais comuns são baseadas na contagem do número de arestas da taxonomia entre duas classes;
- Estruturas mereológicas: o termo mereológico vem da palavra *mereologia*, que significa o estudo lógico-matemático das relações entre as partes e o todo. Essa estrutura corresponde a uma relação do tipo *part-of*. Se é possível detectar as relações que apoiam a essa estrutura, então elas podem ser usadas para a computação da similaridade entre classes – serão mais semelhantes, se compartilharem partes semelhantes. Esses métodos são de grande utilidade quando as extensões das classes são comparadas, pois a partir disso é possível inferir que os objetos que partilham as mesmas partes são os mesmos;

Relações também podem ser extraídas da semelhança entre os nós dos grafos. Por exemplo, considerando as ontologias ilustradas na Figura 3.2. Se em uma ontologia, a classe *Book* está relacionada com a classe *Human* pela relação *author*; e se em outra ontologia, as classes *Volume* e *Writer* estão relacionadas por uma relação similar *author*. Então, sabendo que as classes *Book* e *Volume* são semelhantes, pode-se inferir que as classes *Human* e *Writer* podem ser semelhantes também.

3.5.4 Técnicas extensionais

Comparar instâncias de ontologias é uma boa estratégia para realizar o *matching*. Por exemplo, se duas classes compartilham exatamente o mesmo conjunto de instâncias, então pode-se presumir que os conceitos que essas classes representam são correspondentes. Métodos extensionais são divididos em três categorias: os que se aplicam à ontologias com conjuntos de instâncias comuns, os que propõem técnicas de identificação individual, e os que não necessitam de identificação, ou seja, que trabalham com conjuntos heterogêneos de instâncias (Euzenat e Shvaiko, 2007):

- Comparação comum de extensões: testam a intersecção do conjunto de instâncias de duas classes A e B . São consideradas muito similares quando $A \cap B = A = B$, ou mais genéricas quando $A \cap B = A$ ou $A \cap B = B$. Porém, a técnica apresenta problemas quanto à habilidade de lidar com falhas: pequenas quantidades de dados incorretos podem conduzir o sistema a uma conclusão errada sobre as relações de domínio;
- Técnicas de identificação de instâncias: podem ser utilizadas quando um conjunto comum de instâncias não existe, pois é possível tentar identificar qual instância de um conjunto, corresponde à outra instância de outro conjunto. Por exemplo, pode-se comparar chaves de base dados, ou usar dados da instância para comparar os valores das propriedades;
- Comparação de extensões disjuntas: pode ser utilizada quando não é possível inferir diretamente sobre um conjunto de dados comum para duas ontologias, pois é mais fácil usar técnicas de aproximação para comparar as extensões das classes. Esses métodos podem ser baseados em medidas estatísticas sobre as características dos membros da classe, nas similaridades computadas entre instâncias de classes ou baseados na correspondência entre conjuntos de entidades.

3.5.5 Técnicas baseadas em semântica

Técnicas baseadas em semântica não podem encontrar correspondências por si mesmas. No entanto, elas são úteis quando correspondências são geradas a fim de assegurar a completude, ou seja, encontrar correspondências que levam à inconsistência do alinhamento. Quando é necessário realizar o *matching* entre duas ontologias, muitas vezes não há fundamentos comuns sobre os quais as comparações podem ser baseadas. Esses fundamentos podem ser encontrados relacionando as ontologias a recursos externos, como uma ontologia intermediária, podendo definir um contexto comum para as duas ontologias (Aleksovski et al., 2006, Euzenat e Shvaiko, 2007).

O esquema do *matching* semântico é representado na Figura 3.7. A tarefa é combinar o vocabulário das ontologias 1 e 2 por meio de *âncoras* à uma ontologia que detém um *background* sobre

o domínio. Essas âncoras são responsáveis por ligar os conceitos das ontologias iniciais ao *background*. Quanto maior o número de âncoras, mais complexas se tornam as comparações, por outro lado, reforçam a idéia de que as ontologias estão relacionadas.

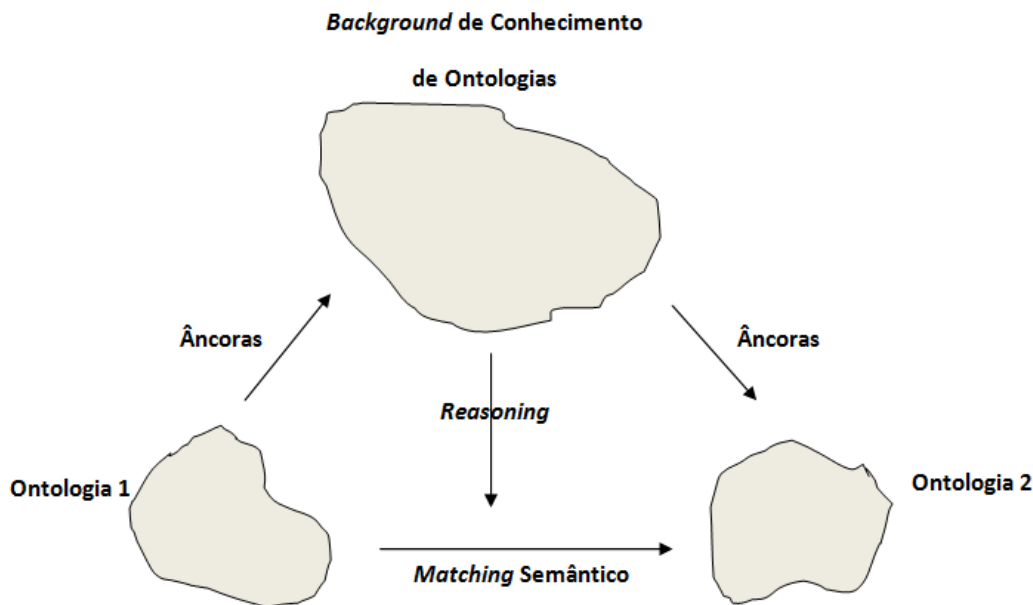


Figura 3.7: *Matching* entre duas ontologias usando um *background* de conhecimento de ontologias. Adaptado de [Aleksovski et al. \(2006\)](#)

No exemplo ilustrado na Figura 3.8 há a tentativa de realizar o *matching* entre os conceitos “Intoxicação por heroína” e “Overdose por drogas”. De acordo com o *background* de conhecimento, “Heroína” é uma espécie de “Drogas”, enquanto “Overdose” é uma espécie de “Intoxicação”, ou seja, os dois aspectos possuem uma relação. Assim, apesar de os conceitos não serem equivalentes e nem possuírem qualquer semelhança léxica, possuem uma sobreposição semântica. Seus rótulos descritivos consistem em conjuntos de palavras disjuntos, sendo apenas possível realizar o *matching* utilizando o *background* de conhecimento.

3.6 Considerações Finais

Neste capítulo são apresentados conceitos de ontologias, com foco nas linguagens para representá-las, em especial a linguagem OWL. Também são expostas as técnicas para o cálculo da similaridade entre elementos externos ou presentes na ontologia.

Algumas linguagens de ontologias apresentam limitações quanto à expressividade. Por exemplo, a RDF é limitada a predicados binários, enquanto que a RDFS é limitada a uma hierarquia de subclasses e propriedades, com definições do domínio dessas propriedades. A semântica é um pré-requisito para o apoio ao *reasoning*, com isso derivações podem ser feitas mecanicamente, ao invés de serem feitas manualmente. Além disso, o *reasoning* é importante pois permite verificar a consistência da ontologia, procurar as relações inesperadas entre as classes, e classificar automaticamente as instâncias em suas respectivas classes. A linguagem OWL supre essas limitações, ao usar semântica e proporcionar realização do (*reasoning*).

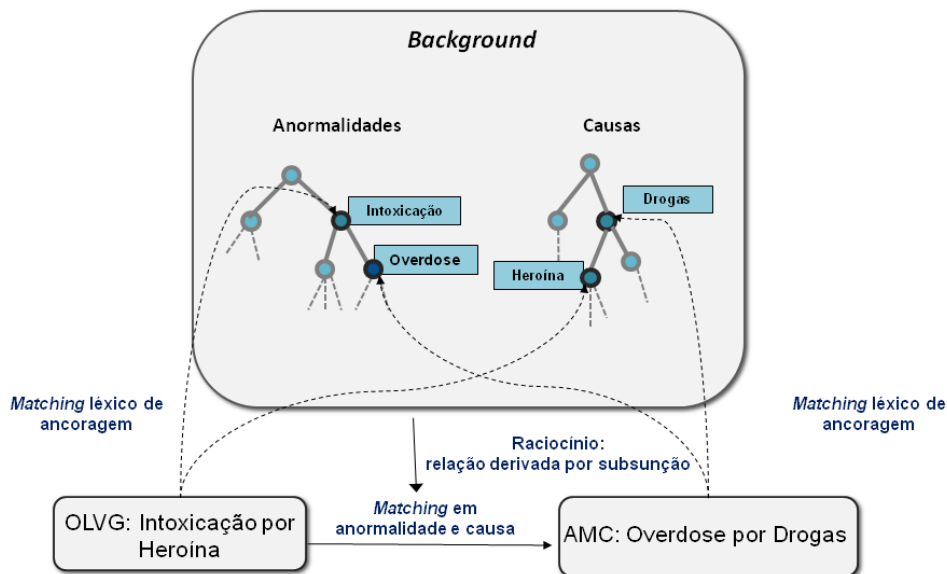


Figura 3.8: *Matching* entre duas ontologias usando um *background* de conhecimento. Adaptado de Aleksovski et al. (2006)

As técnicas utilizadas para analisar se dois esquemas são similares se baseiam em suas características específicas, avaliando a similaridade entre pares de elementos presentes nesses esquemas. Por isso, as técnicas de *matching* devem ser escolhidas tendo em consideração os elementos contidos nos esquemas. Por exemplo, técnicas baseadas em comparações de strings são úteis no caso de strings semelhantes que denotam os mesmos conceitos. Porém, essas técnicas podem causar falsos positivos, pois duas *strings* podem ser muito similares, mas por outro lado, representarem conceitos distintos.

Por outro lado, recursos linguísticos como léxicos e tesouros permitem a interpretação dos termos utilizados nas expressões contidas nas ontologias. Eles fornecem uma compreensão mais precisa dos seus rótulos descritivos. Técnicas baseadas na estrutura são de fácil implementação, porém a estrutura interna não fornece muita informação para comparar as entidades. Além disso, é possível que diferentes modelos de um conceito, utilizem tipos de dados diferentes e incompatíveis. Técnicas extensionais fornecem uma maneira fácil de comparar a sobreposição entre duas classes. Porém, quando os dados das instâncias não estão disponíveis não é possível aplicar esta técnica. Por último, foram apresentadas técnicas semânticas, que por meio de um *background* de conhecimento é capaz de combinar duas ontologias.

Uma vez que essas técnicas tenham sido aplicadas, as entidades das ontologia são representadas como conjuntos de termos, que podem ser comparados com as mesmas técnicas previamente apresentadas. Com os termos normalizados, é possível realizar consultas a recursos linguísticos externos para encontrar similaridades entre os termos. Esses recursos podem ser dicionários, terminologias ou tesouros e podem ser definidos para um idioma ou domínio específicos.

Os conteúdo exposto neste capítulo é fundamental na proposta apresentada no próximo capítulo. É por meio do relacionamento dos conceitos contidos nas ontologias e das técnicas de *matching* que o mapeamento semântico entre múltiplas visões coordenadas é estabelecido.

Técnica de Coordenação com o Apoio de Ontologias

4.1 Considerações iniciais

Uma das limitações encontradas em técnicas de coordenação utilizadas na exploração de múltiplas visões é ocasionada pela utilização de mapeamentos léxicos, pois apenas são relacionados elementos de mesma sintaxe. Assim, o processo de exploração fica limitado, pois não são reconhecidos elementos de mesmo significado e sintaxes distintas. A proposta deste trabalho é a concepção de uma técnica de coordenação com o apoio de ontologias, denominada *Semantic Coordination*, para realizar um mapeamento semântico entre visões, aumentando a capacidade de exploração do usuário.

A técnica de coordenação *Semantic Coordination*, foi incorporada às técnicas de coordenação da *Projection Explorer* (PEX), apresentada na Seção 2.4. Por meio da PEX são criadas as projeções de mapas de similaridade usando coleções de documentos. Então, o usuário pode definir quais as técnicas de coordenação para apoiá-lo no processo de exploração desses mapas (Paulovich et al., 2007).

A *Semantic Coordination* é uma extensão da *Topic Coordination*. A diferença consiste em como os tópicos são gerados. Na *Topic Coordination*, os tópicos de interesse são gerados após a seleção do usuário em documentos de seu interesse. Por meio da coordenação os tópicos são propagados às demais visões, destacando os documentos cobertos por esses tópicos. O processo de extração dos termos se utiliza da covariância dos termos, em que é gerada uma matriz de *termos x documentos*. Cada linha da matriz representa um documento selecionado e as colunas correspondem aos termos ocorrentes nos documentos. Os valores presentes nos elementos da matriz são normalizados entre 0.0 e 1.0, e indicam a frequência de ocorrência de um determinado termo no seu respectivo documento, como ilustrado na Figura 4.1.

Para detalhar a *Semantic Coordination*, o restante deste capítulo está organizado da seguinte maneira. Na próxima seção é descrito o processo de coordenação proposto: desde a extração dos termos, até o processo de *matching* com o apoio de ontologias. Na Seção 4.3 é apresentada a avaliação da téc-

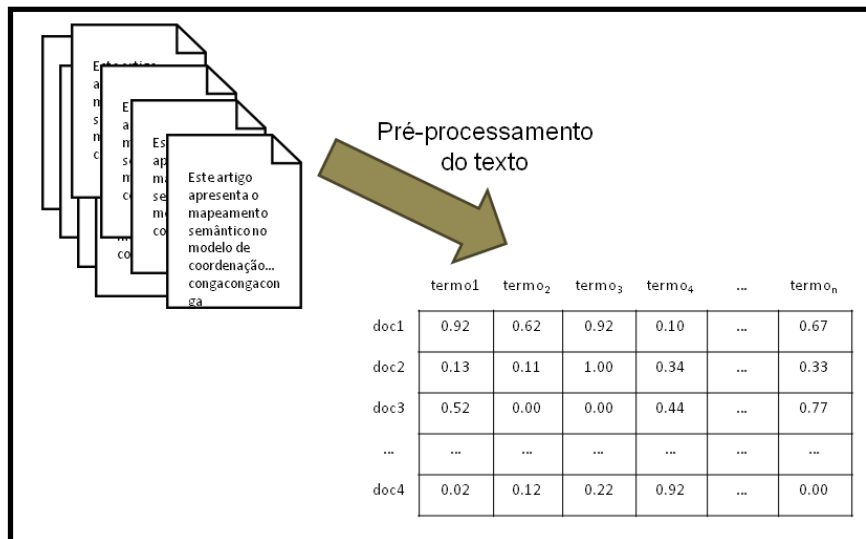


Figura 4.1: Processo de geração de tópicos

nica proposta por meio de dois estudos de caso, contidos em dois domínios de conhecimento distintos. Por fim, são expostas as considerações finais.

4.2 Esquema de coordenação com o apoio de ontologias

O processo da *Semantic Coordination* se divide basicamente em três etapas, ilustradas na Figura 4.2. Na primeira etapa, após a seleção do usuário, são extraídas da visão explorada os termos frequentes. Na segunda etapa ocorre o processo de *matching* entre os termos extraídos e os termos da ontologia. Por fim, na terceira etapa são mapeados e destacados os itens nas visões destino, que apresentam a ocorrência dos termos oriundos da ontologia.

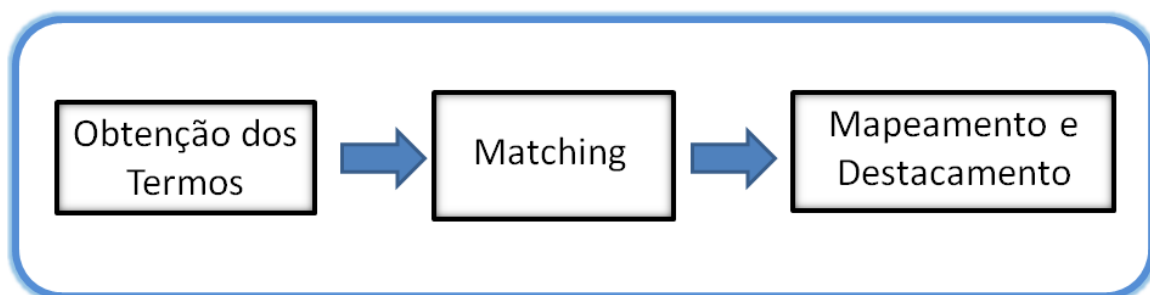


Figura 4.2: Visão geral do esquema de coordenação

4.2.1 Obtenção dos termos frequentes

Na *Semantic Coordination* os tópicos de interesse do usuário não são gerados automaticamente. A primeira etapa do processo de coordenação consiste na extração dos termos frequentes dos documentos selecionados, exemplificada na Figura 4.3. O primeiro passo da extração ocorre após a seleção por parte do usuário de um conjunto de documentos de seu interesse, representados pelos círculos da Parte 1 da Figura 4.3.

Após o pré-processamento, em que a partir da seleção do usuário são extraídos os termos de maior

frequência, é apresentada uma lista de termos em ordem de frequência nos documentos selecionados. Em seguida, os termos extraídos podem ser filtrados pelo usuário conforme o foco da exploração do conjunto de documentos.

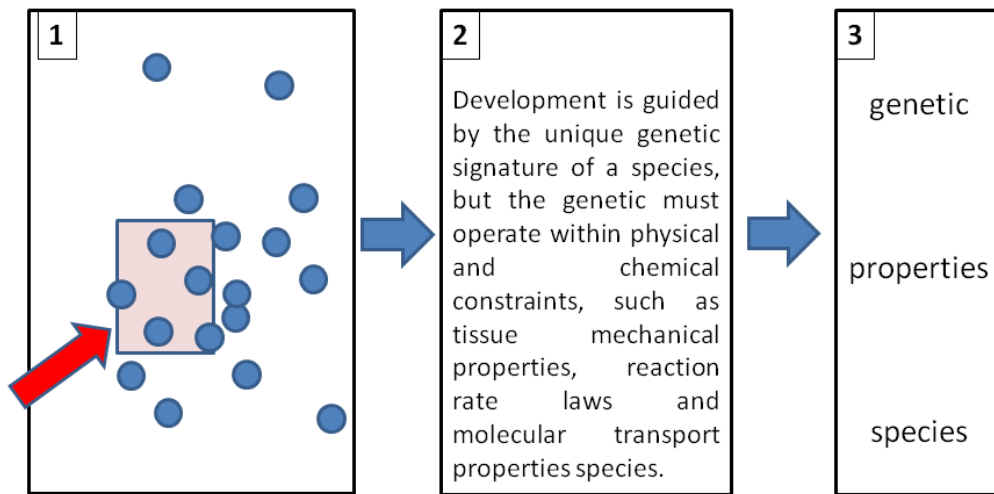


Figura 4.3: Visão geral da obtenção dos termos frequentes

Com a definição do conjunto de termos de interesse do usuário, inicia-se a etapa de *matching*, em que cada um dos termos desse conjunto é utilizado como parâmetro de entrada, como apresentado a seguir.

4.2.2 Etapa de *matching*

A etapa de *matching* é realizada com o suporte de ontologias. O propósito é a formação de um novo conjunto de termos a partir dos termos obtidos da etapa de extração, como ilustrado na Figura 4.4. A técnica utilizada para realizar o *matching* é a técnica baseada em strings, pois é feita a comparação entre elementos externos e elementos da ontologia. Ressalta-se que as ontologias devem ser de mesmo domínio de conhecimento que o do conjunto dos documentos explorados.

A primeira etapa do processo é a busca por ocorrências de *matching* entre cada um dos termos do conjunto de entrada, e os termos da ontologia. Uma vez que ocorra o *matching* entre esses termos, são recuperados da ontologia todos os termos relacionados ao termo de entrada, formando um novo conjunto de termos. O processo obedece aos seguintes passos:

1. É iniciado o processo de busca pelo termo de entrada percorrendo na ontologia todas as suas classes, assim como suas respectivas subclasses;
2. Caso ocorra o *matching* entre o termo de entrada e um termo que represente uma classe ou subclasse, são recuperadas (caso existam) sua superclasse, suas subclasses e as classes relacionadas por meio das propriedades;
3. Caso não ocorra nenhuma ocorrência de *matching*, o usuário é alertado que não há ocorrência do termo de entrada na ontologia.

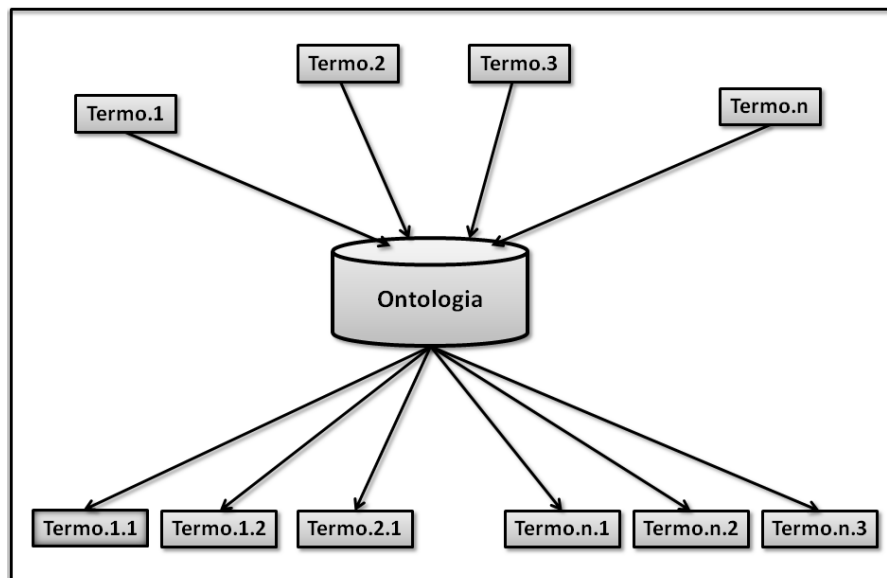


Figura 4.4: Visão geral do processo de *matching*

4.2.3 Mapeamento na visão destino

Após a composição do conjunto de termos relacionados, ocorre o mapeamento dos itens relacionados nas visões destino similar à técnica de coordenação *Topic Coordination*, que utiliza os tópicos gerados para o realizar o mapeamento. A *Semantic Coordination* utiliza os termos oriundos da ontologia para a realização do mapeamento.

Para que isso ocorra, é iniciada uma busca nas visões destino pelos termos gerados após a etapa de *matching*. Então, são procuradas ocorrências desses termos nos documentos em cada uma das visões destino. Caso haja a ocorrência, o item correspondente ao documento na visão destino é destacado, conforme exemplificado na Figura 4.5.

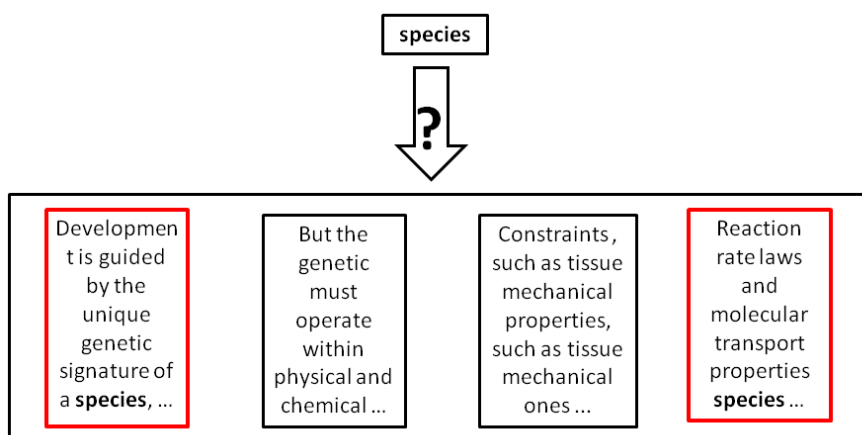


Figura 4.5: Mapeamento e destacamento dos itens na visão destino

A seguir é apresentado um exemplo da aplicação da *Semantic Coordination* na exploração de dados textuais. Após a seleção do usuário, são extraídos os termos com maior frequência dos documentos contidos na seleção. Então o usuário pode escolher quais os termos de seu interesse, ou ainda

inserir termos que não constam nos termos extraídos. Na Figura 4.6 é apresentado um exemplo onde foi extraída e apresentada uma lista de termos após a interação do usuário.

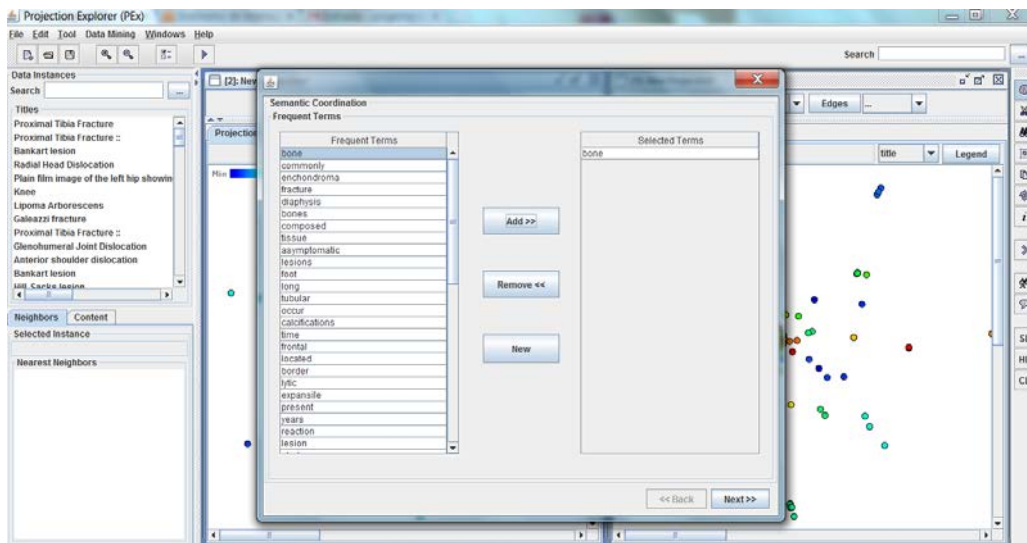


Figura 4.6: Apresentação dos termos frequentes a partir de uma seleção

Após a escolha pelo termo “bone” e da ontologia de interesse do usuário para apoiar a coordenação, é exibida uma lista de termos relacionados aos termos extraídos no passo anterior, como ilustrado na Figura 4.7. Neste exemplo, ao selecionar os termos “vertebra” e “scapula”, o usuário inicia a exploração por documentos cobertos por esses termos.

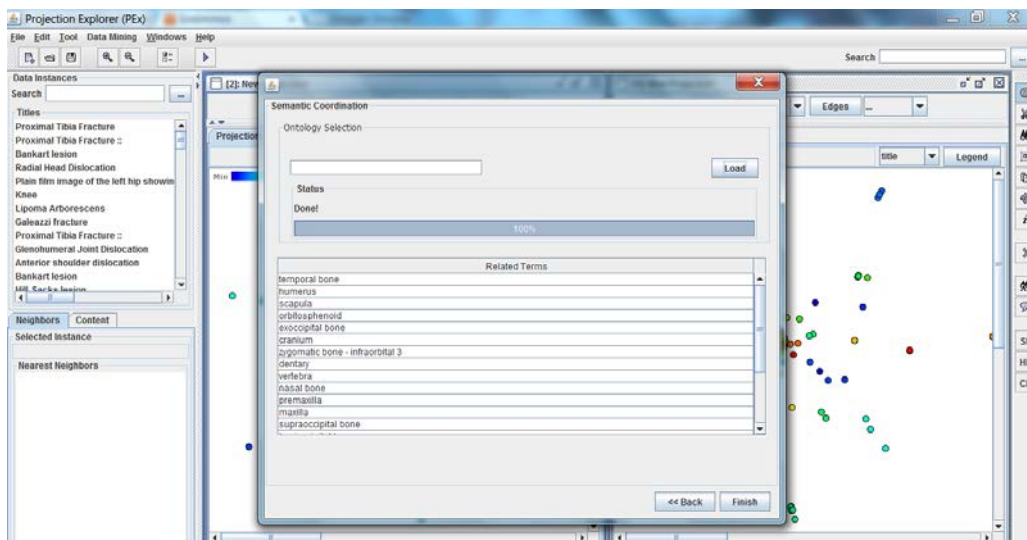


Figura 4.7: Apresentação dos termos relacionados por meio de uma ontologia

Na Figura 4.8 são apresentadas as visões origem (*projection 1*) e destino (*projection 2*), utilizadas no processo de exploração. Na visão destino foram destacados alguns documentos com a ocorrência dos termos, sendo que nesta escala de cores pode ser alterada na interface da PEx. Nesta escala de cor adotada (conhecida como escala *rainbow*) azul indica frequência de ocorrência nula e o vermelho frequência de ocorrência máxima.

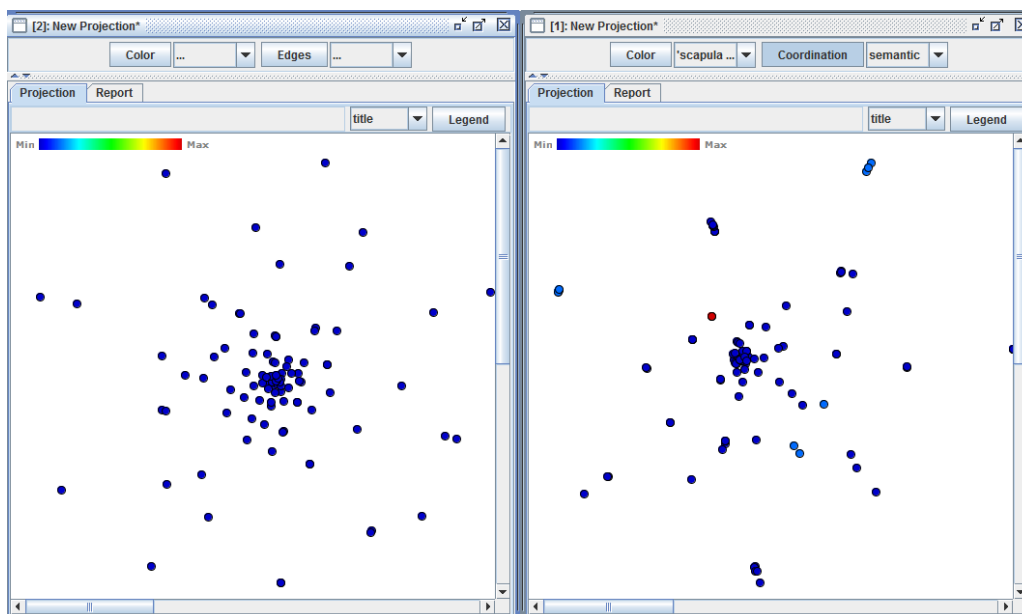


Figura 4.8: Exemplo da aplicação da *Semantic Coordination*

4.3 Avaliação da técnica de coordenação proposta

Para a avaliação da técnica de coordenação proposta são necessários uma ontologia e uma coleção de documentos de mesmo domínio de conhecimento da ontologia selecionada. São apresentados para avaliar a *Semantic Coordination* dois estudos de caso. Para a realização desses estudos de caso, inicialmente foi necessário a definição do domínio de conhecimento abrangido. Após a escolha do domínio, foi iniciada a procura por ontologias contidas no domínio escolhido em repositórios abertos.

4.3.1 Estudo de caso: descrições de imagens médicas

No primeiro estudo de caso é utilizada uma base de dados e uma ontologia da área médica. A ontologia utilizada nesse estudo de caso é uma ontologia de órgãos de grupos de vertebrados, denominada **vHOG** – uma ontologia multiespécie anatômica para a linhagem dos vertebrados, baseada em uma ontologia chamada *Common Anatomy Reference Ontology* (CARO). CARO é uma ontologia desenvolvida para facilitar a interoperabilidade entre ontologias para anatomias existentes para as diferentes espécies, além de fornecer um modelo para a construção de uma nova ontologia para anatomia. A ontologia vHOG inclui 1184 conceitos, usados para comparar padrões de expressão entre as espécies. Também é composta por estruturas que contêm as relações homólogas entre modelos das espécies de vertebrados. Parte de sua estrutura é ilustrada na Figura 4.9.

A base textual deste primeiro estudo de caso é um conjunto de descrições de imagens médicas, extraídas de uma base de dados multi-plataforma chamada **MedPix**¹, composta de imagens e informações textuais.

MedPix é uma base de dados online gratuita de imagens médicas, fornecida pelos Departamentos de Radiologia e Informática Biomédica da *Uniformed Services University (USU)*. A maioria das imagens contem a descrição dos casos, seus diagnósticos e suas patologias. O material é composto

¹<http://rad.usuhs.edu/medpix>

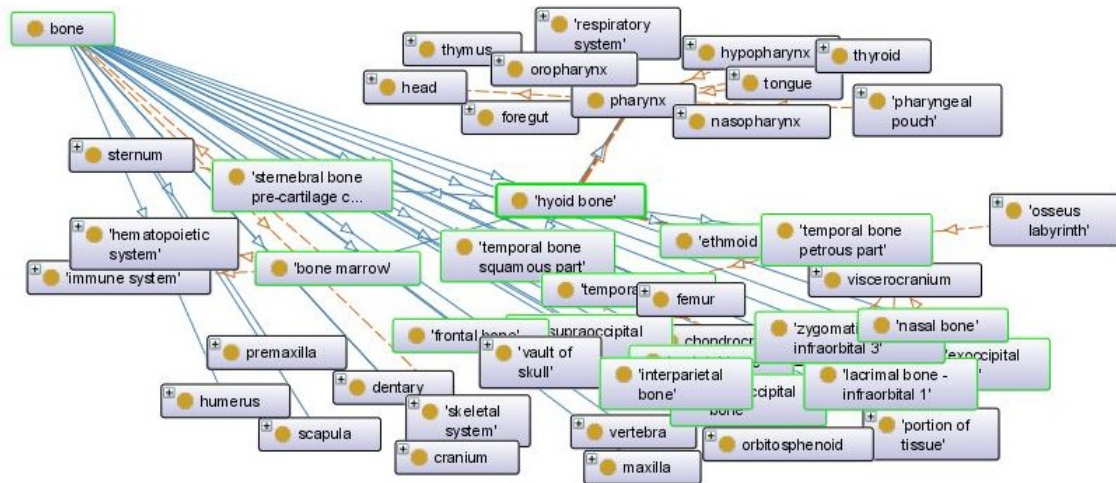


Figura 4.9: Visão parcial da vHOG

por 12.136 casos e 57.271 imagens, e é organizado pela localidade da doença (no sistema de órgãos), categoria da patologia, o perfil dos pacientes e pela classificação de imagem e da sua legenda.

A base de dados é composta por imagens de raio-x e ressonância e arquivos textos. Existe um arquivo texto que contém informações de cada imagem, como diagnóstico e patologia. Na Figura 4.10 é dada uma visão de uma dessas bases, em que arquivos textos e imagens de mesmo número estão relacionadas. As imagens são identificadas por um número, e para cada uma dessas imagens existe um arquivo texto (identificado pelo mesmo número da imagem a que corresponde) contendo informações sobre a patologia do paciente.

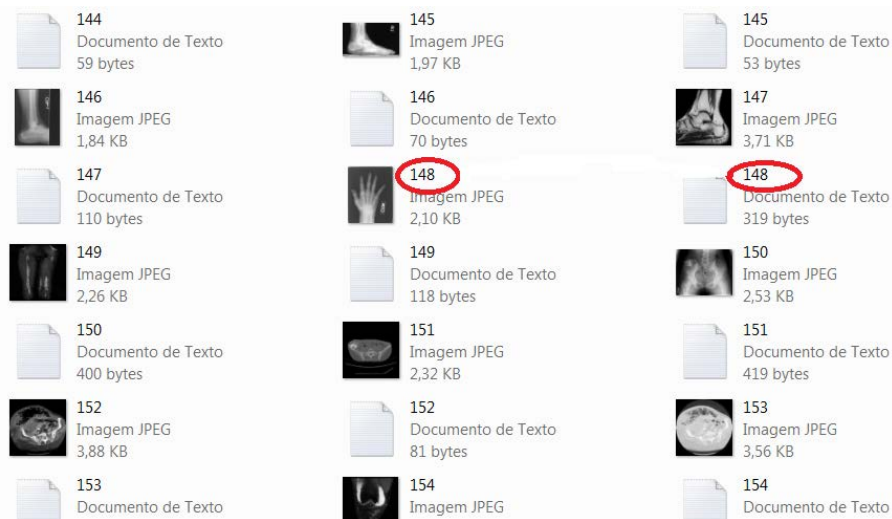


Figura 4.10: Visão geral da base de dados explorada

O primeiro exemplo deste estudo de caso mostra a exploração pelo termo “bone” (osso). Na Figura 4.11 é ilustrada uma exploração do usuário usando a coordenação léxica – após uma seleção na visão origem (indicado por New Projection 2) são destacados os documentos abrangidos pelo termo “bone”. No entanto, há um grande número de documentos destacados, devido à alta ocorrência deste termo na coleção de textos projetada.

Aplicando a *Semantic Coordination*, os usuários são capazes de explorar documentos abrangidos

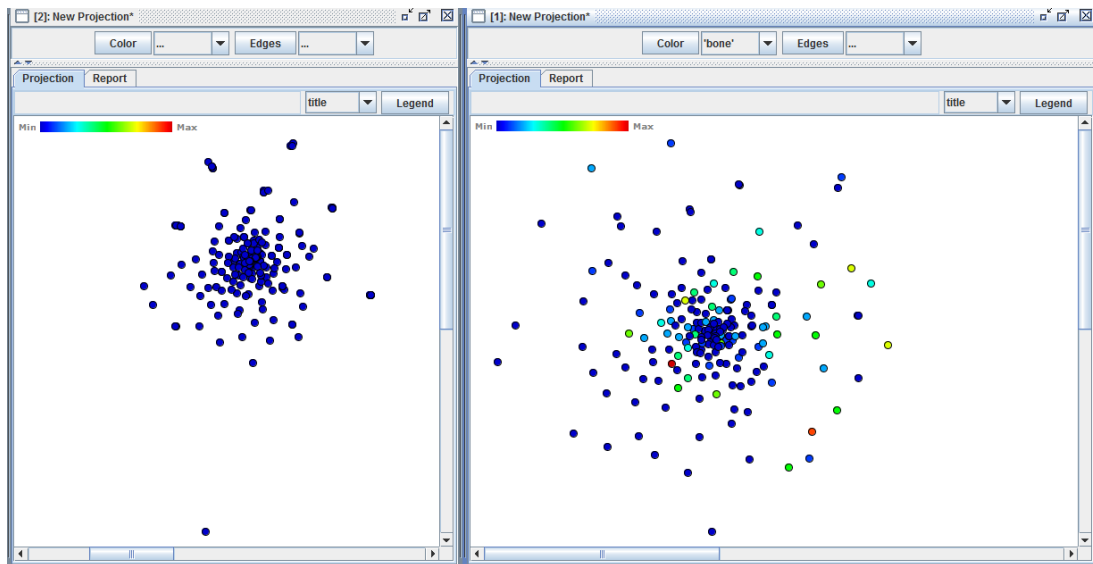


Figura 4.11: Exploração de uma coleção de documentos cobertos pelo termo “bone”

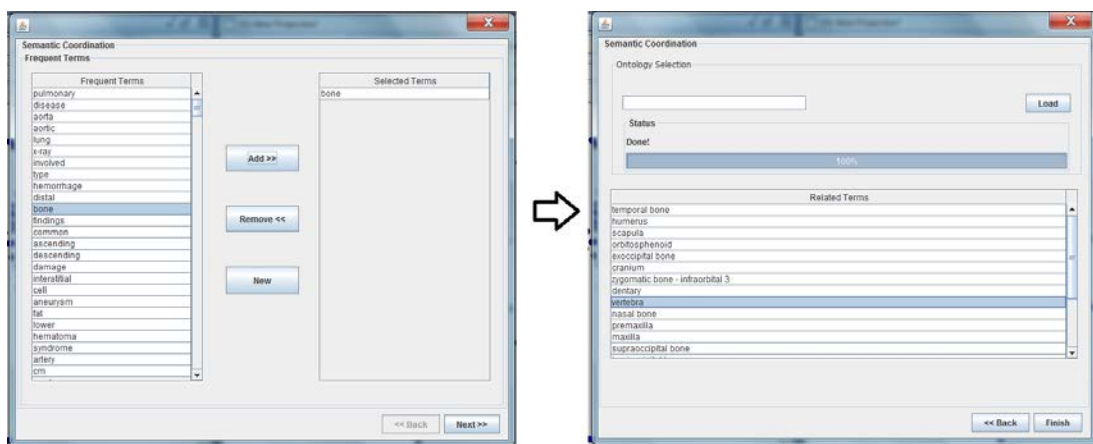


Figura 4.12: Termos relacionados ao termo “bone”

por termos relacionados ao termo “bone”. Em primeiro lugar, é possível filtrar termos do conjunto de termos extraídos após a seleção dos documentos. Após o carregamento da ontologia, os termos relacionados são apresentados em uma interface para que os usuários possam selecionar quais termos são exploradas. Neste exemplo, o termo “vertebra” (qualquer um dos ossos ou segmentos cartilagosos que formam a coluna vertebral), foi selecionado, tal como ilustrado na Figura 4.12.

Por fim, os documentos abrangidos pelo termo “vértebra” estão em destaque na visão destino (indicado por New Projection 1). Assim, o usuário pode efetuar uma filtragem, quando comparado com a projeção na Figura 4.13, assim é possível explorar um número menor de documentos de seu interesse.

No segundo exemplo é demonstrado a exploração pelo termo “aorta” - a principal artéria do corpo humano, que transporta sangue do lado esquerdo do coração para as artérias de todos os membros e órgãos, exceto os pulmões. Na Figura 4.14 é apresentada uma coordenação entre ambas as visões usando um mapeamento léxico, em que são destacados os círculos que representam documentos abrangidos pelo termo “aorta”.

Com o emprego da *Semantic Coordination* são destacados documentos distintos em comparação

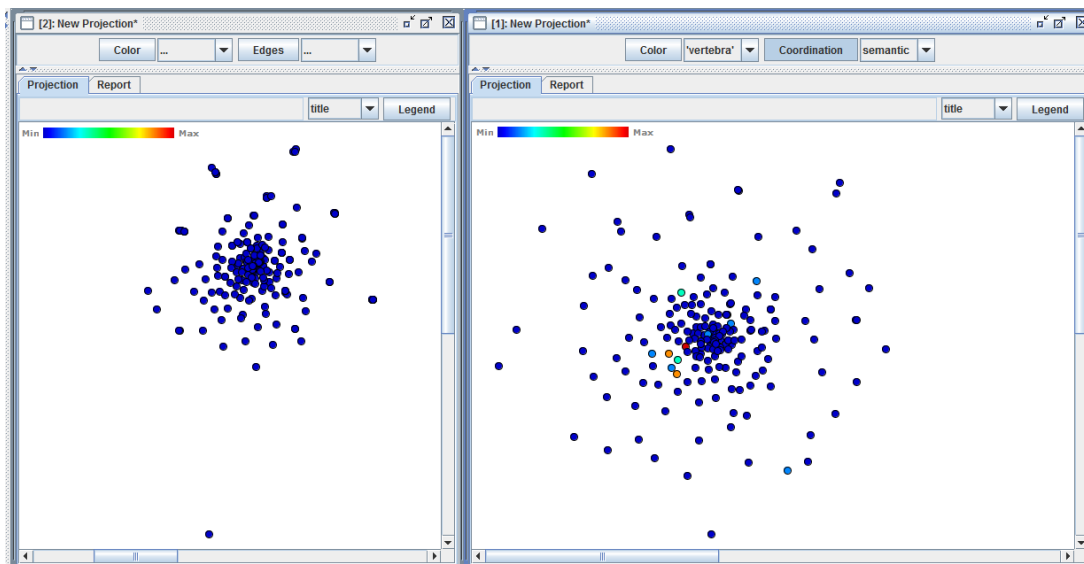


Figura 4.13: Destacamento dos documentos abrangidos pelo termo “vertebra”

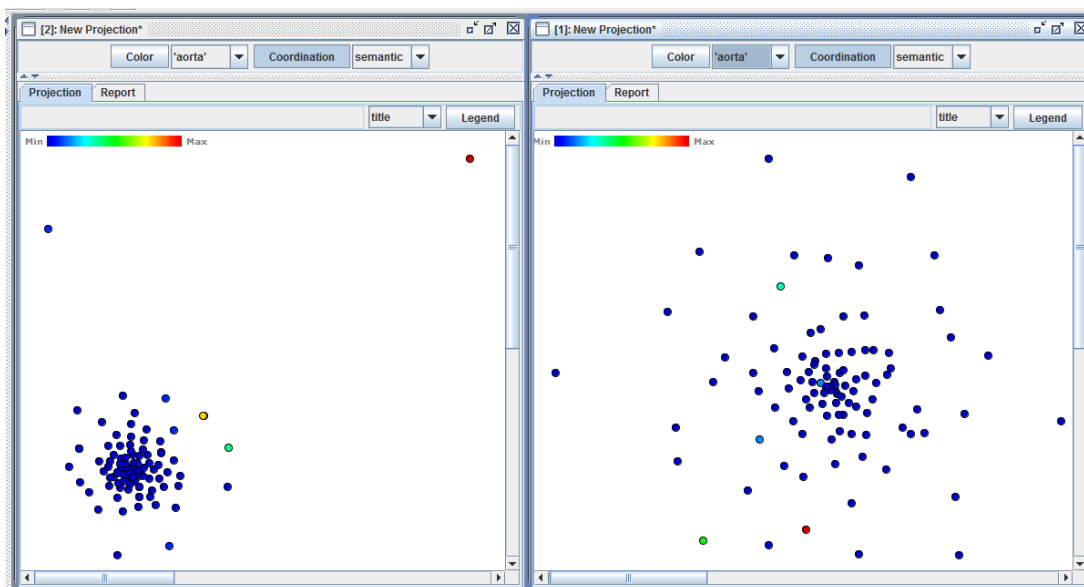


Figura 4.14: Exploração de uma coleção de documentos cobertos pelo termo “aorta”

com o uso da abordagem léxica. Por exemplo, na Figura 4.15 é destacado um círculo que representa um documento que trata sobre edemas pulmonares. Este documento também contém o termo relacionado “artéria”, que por definição é um vaso sanguíneo que carrega sangue a partir dos ventrículos do coração para todas as partes corpo.

No entanto, este documento não foi identificado pelo mapeamento léxico e, conseqüentemente, não seria explorado. Portanto, novas relações entre os documentos são estabelecidas por meio do mapeamento semântico.

Outra vantagem em utilizar o mapeamento semântico entre os elementos é a detecção de um número maior de relações entre eles. O próximo exemplo mostra uma comparação entre o número de documentos destacados utilizando os dois tipos de coordenação. Na Figura 4.16 é mostrada a exploração dos documentos cobertos pelo termo “ligament” (ligamento – um feixe de tecido fibroso, formado por tecido conjuntivo denso modelado, que une entre si duas cabeças ósseas de uma articulação).

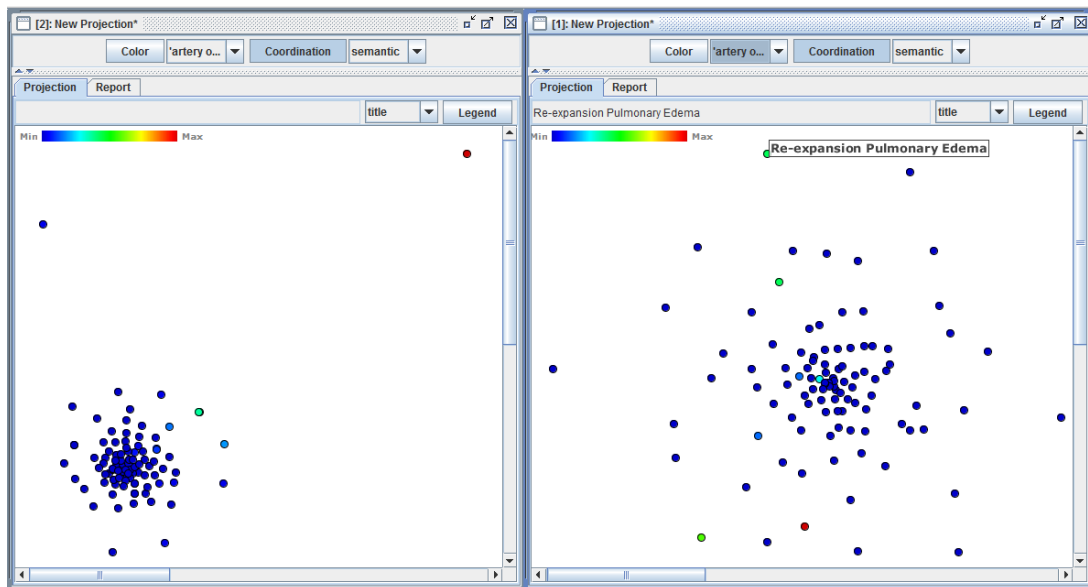


Figura 4.15: Destacamento dos documentos cobertos pelo termo “artery”

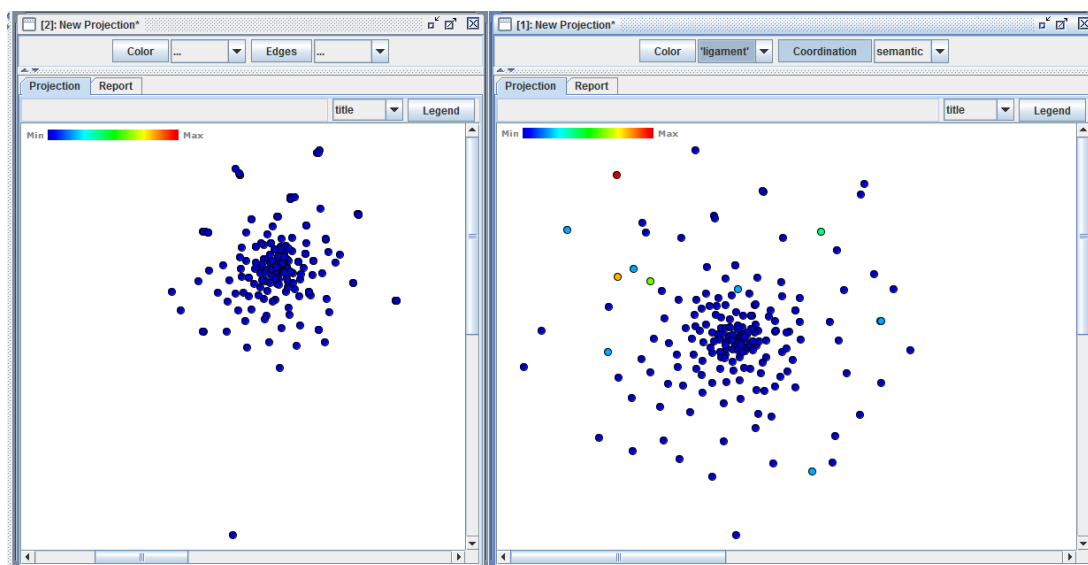


Figura 4.16: Exploração de uma coleção de documentos cobertos pelo termo “ligament”

Com o apoio da ontologia é descoberto que os termos “joint” (articulação ou junta – é a conexão entre duas ou mais peças esqueléticas, ossos ou cartilagens, que permitem o movimento) e “ligament” são relacionados. Assim, o usuário pode explorar um número maior de documentos, como ilustrado na Figura 4.17.

4.3.2 Estudo de caso: artigos de teste de software

Para o segundo estudo de caso é utilizada uma ontologia sobre testes de software chamada *Onto-test*. Essa ontologia foi construída com base na norma ISO/IEC 12207 com o propósito de explorar os diferentes aspectos envolvidos nas atividades de teste. O intuito é definir um vocabulário comum para atividades de teste de software, que pode ser utilizado no desenvolvimento de ferramentas de apoio, bem como aumentar a interoperabilidade entre elas (Barbosa et al., 2006). Do padrão ISO/IEC 12207 são explorados na ontologia aspectos como a definição e a avaliação de critérios de teste, estudos

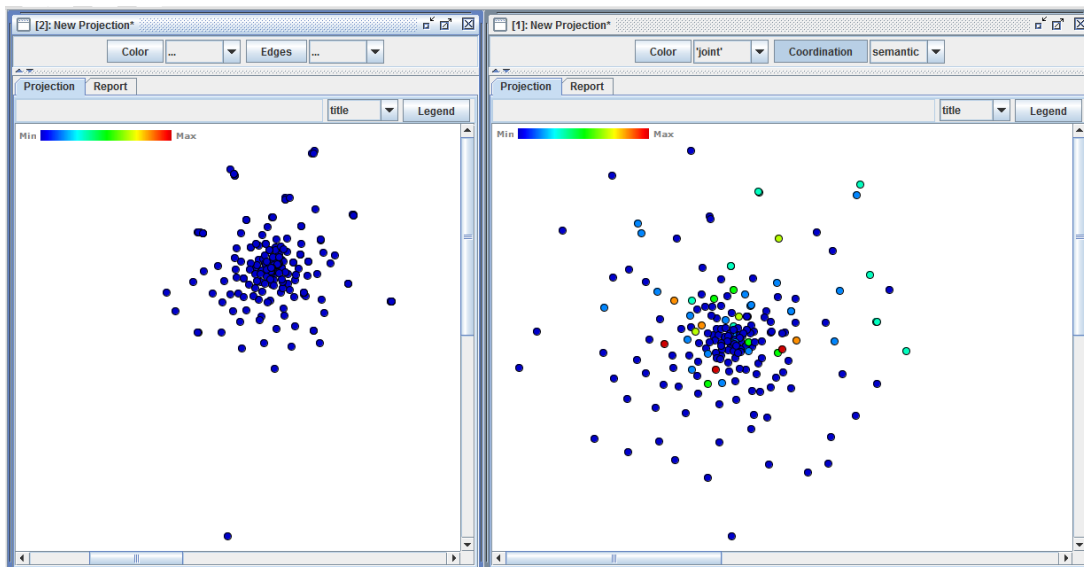


Figura 4.17: Destacamento dos documentos cobertos pelo termo “joint”

teóricos e empíricos que envolvem os testes e o conhecimento e experiência em desenvolvimento de ferramentas (Barbosa et al., 2006).

O desenvolvimento da *OntoTest* é dividido em duas camadas, como ilustrado na Figura 4.18. No nível de ontologia, a ontologia principal de teste de software lida com os principais conceitos e relações associadas ao teste. No nível de sub-ontologia, são tratados os conceitos específicos da ontologia principal, como o processo, as fases, os artefatos, os passos, procedimentos e os recursos para testes.

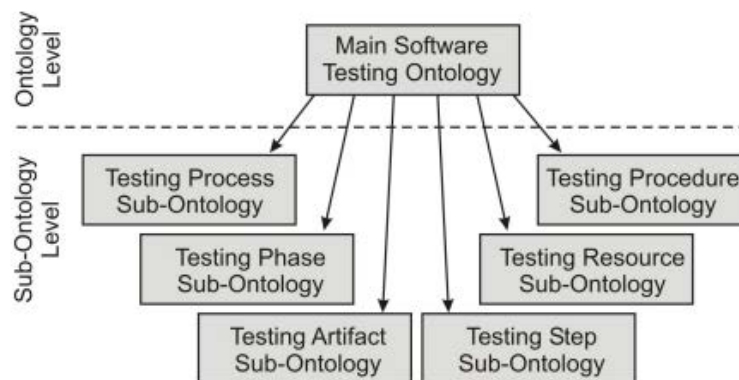


Figura 4.18: Estrutura da *OntoTest*, extraído de (Barbosa et al., 2006)

A coleção de documentos desse estudo de caso é composta por 232 artigos de documentos provenientes de cinco revisões sistemáticas: testes de integração de programas orientados a aspectos; verificação, validação e testes de *web-services*; teste de mutação de programas orientados a aspectos; teste de software em métodos ágeis; e qualidade baseada no desenvolvimento de processos de software (Felizardo et al., 2009).

No primeiro exemplo é mostrada a exploração pelo termo “training” (treinamento), usando um mapeamento léxico. Na Figura 4.19, depois de uma seleção na visão da esquerda, são destacados em ambas as visões os documentos abrangidos por este termo. Mais uma vez, apenas são mapeados

documentos que têm a ocorrência exata do termo “training”. No entanto, os documentos cobertos por termos relacionados não são mapeados e destacados.

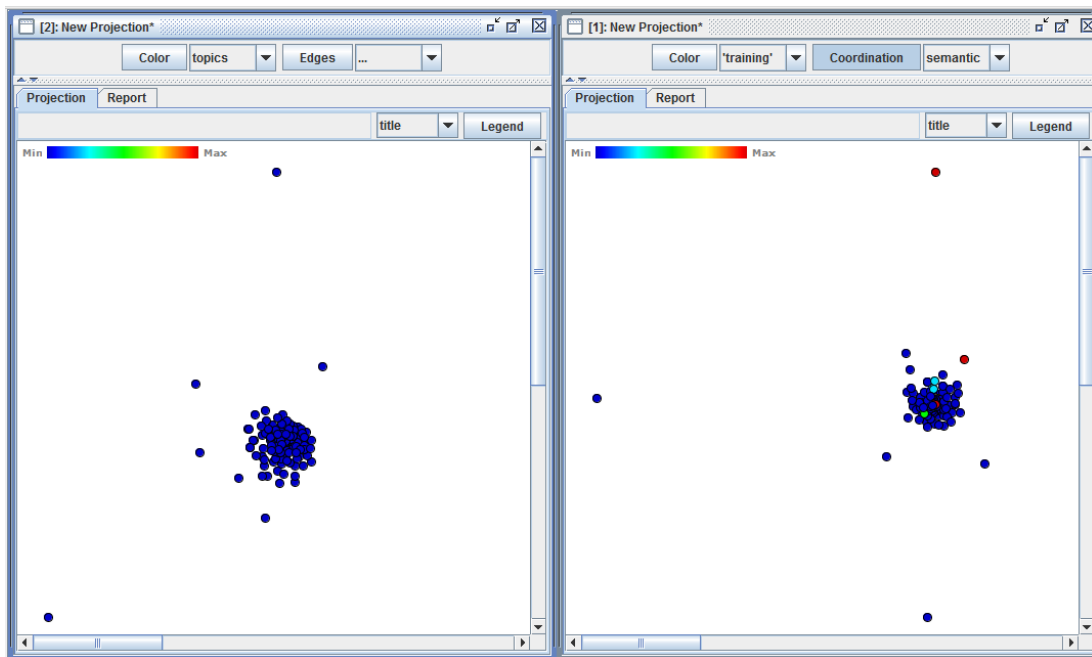


Figura 4.19: Exploração de um conjunto de documentos cobertos pelo termo “training”

Por meio dos relacionamentos da ontologia, verifica-se que o termo “training” está relacionado ao termo “organizational activity”, ou seja, o treinamento é parte de uma atividade organizacional. Então, são mapeados na Figura 4.20 os documentos que apresentam a frequência do termo “organizational activity”. Dessa maneira, são destacados os documentos cobertos por este termo. Assim como no primeiro estudo de caso, com o uso da técnica de coordenação semântica, são detectados elementos distintos do que com o uso da técnica tradicional.

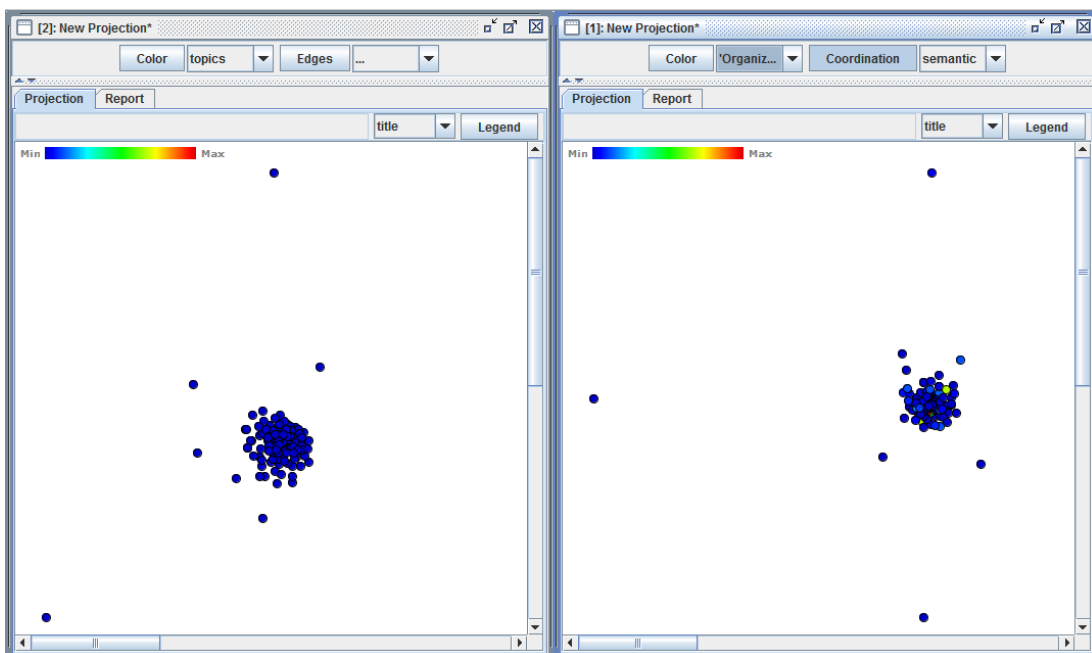


Figura 4.20: Exploração de um conjunto de documentos cobertos pelo termo “organizational activity”

4.4 Considerações finais

A *Semantic Coordination* realiza um mapeamento semântico entre visões com o apoio de ontologias. A técnica é constituída de três etapas: a obtenção dos termos frequentes dos documentos selecionados, a filtragem dos termos obtidos e a realização do *matching* com os termos da ontologia, e a realização do mapeamento e destacamento dos documentos relacionados.

Por meio dos estudos de caso apresentados nesta seção é possível verificar os benefícios de usar a *Semantic Coordination* na exploração de coleções de documentos em CMV. A avaliação apresentada traz indícios que a *Semantic Coordination* pode ser usada como um recurso para a filtragem dos documentos mapeados. Além disso, novas associações entre os elementos podem ser descobertas, como exposto nos exemplos exibidos, aumentando as possibilidades de exploração.

Conclusões

Aplicar mapeamentos léxicos na coordenação de múltiplas visões não revela associações entre elementos de sintaxe distintas. A proposta deste trabalho é aplicar um mapeamento semântico ao coordenar múltiplas visões. Isso é feito com a utilização de ontologias no processo de coordenação, que estabelecem novas associações entre os elementos presentes nas visões coordenadas. Nesta abordagem, a técnica coordenação proposta, denominada *Semantic Coordination*, é implementada como uma extensão das técnicas de coordenação existentes na *Projection Explorer*. Essa técnica é responsável por aplicar um mapeamento que estabelece uma relação específica entre as instâncias de visões distintas. Além disso, fornece um novo mecanismo para realizar a coordenação entre representações de coleções de documentos.

O processo de coordenação da *Semantic Coordination* se inicia após a seleção do usuário, quando são extraídas da visão explorada os termos frequentes. Depois, é iniciado o processo de *matching* entre esses termos extraídos e os termos da ontologia. Por fim, são mapeados e destacados os itens nas visões destino, que apresentam a ocorrência dos termos relacionados oriundos da ontologia e que foram filtrados. Assim, o usuário visualiza os documentos que apresentam ocorrência desse novo conjunto de termos.

Usando uma ontologia como apoio ao processo de mapeamento, as associações entre as instâncias podem ser definidas pelas relações expressas pela ontologia. Contrapondo-se a trabalhos anteriores apresentados, os termos destacados dependem da interação do usuário, pois os termos extraídos são usados para recuperar os tópicos relacionados. Isso é realizado por meio da ontologia empregada, gerando um novo conjunto de termos, utilizado no mapeamento em visões destino. Além disso, as condições podem variar de acordo com a interação do usuário, ao escolher termos de seu interesse no processo de coordenação.

5.1 Contribuições e limitações

A avaliação apresentada no capítulo anterior mostra que são estabelecidas novas associações entre as instâncias presentes nas visões. Isso em comparação à técnicas de coordenação baseada em

identificadores. Por meio do mapeamento semântico, é possível encontrar diferentes documentos de interesse do usuário em visões distintas. Além disso, os testes realizados mostram que a técnica de coordenação proposta oferece uma abordagem de exploração mais eficaz. Isso acontece pela combinação da capacidade cognitiva fornecida pela visualização, com a expressividade das ontologias, que revela conexões ocultas entre os dados.

Geralmente, as técnicas tradicionais limitam o usuário no processo de exploração por informação. Nessas abordagens, o mapeamento entre os elementos de visões distintas são estabelecidos pelos dados provenientes de cada um desses elementos representados nessas visões. Visando suprir esta limitação, neste trabalho são usadas ontologias, que fornecem mecanismos para a execução de um mapeamento semântico. Esse mapeamento expande o número de associações entre os elementos, aumentando o poder de exploração dos dados.

Adicionalmente, o usuário pode selecionar quais os termos extraídos da seleção dos documentos são utilizados no mapeamento. Portanto, ao encontrar documentos com tópicos gerados semelhantes e não apenas idênticos, um maior número de relações entre esses elementos podem ser descobertas durante o mapeamento. Desta maneira, a coordenação semântica aumenta o poder de exploração de coleções de documentos, ao mapear elementos semelhantes em visões distintas no processo de coordenação.

Assim, o conjunto de relações da ontologia é crucial para o sucesso do processo de mapeamento e coordenação. Pois, a utilização de uma representação que incorpora as relações explicitamente, torna o processo de mapeamento mais eficiente.

Uma outra vantagem, é o fato de que uma mesma ontologia pode ser utilizada na exploração de vários conjuntos de dados, desde que estejam contidos em um mesmo domínio de conhecimento. Por outro lado, essa técnica pode ser empregada utilizando diferentes ontologias e conjuntos de dados, na exploração de domínios de conhecimento distintos.

No entanto, essa técnica é dependente da qualidade da ontologia empregada, pois o processo de mapeamento requer que os termos da ontologia sejam bem correlacionados. Se as relações na ontologia não são bem construídas, o processo de mapeamento é comprometido. Também, a ontologia e a base de dados explorada devem estar contidas em um mesmo domínio.

5.2 Trabalhos futuros

Como trabalhos futuros, destaca-se a realização de outros estudos de caso, ratificando a eficácia da técnica proposta. Isso pode ser feito por meio da exploração de outros conjuntos de dados e novas ontologias de mesmo domínio.

Além disso, a *Semantic Coordination* pode ser acurada. Os documentos podem ser mapeados de acordo com o grau do relacionamento proveniente da ontologia. Dessa maneira, viabiliza ao usuário a verificação de como do conjunto de termos fornecidos pela ontologia são relacionados. Também, a realização da extração de termos compostos a partir da seleção do usuário, consiste em uma possível contribuição para o aperfeiçoamento do mapeamento semântico.

5.3 **Produção bibliográfica**

Ao longo do desenvolvimento deste projeto de mestrado, houve três publicações em eventos:

- PRATES, J. M.; SCATALON, L. P.; GARCIA, R. E.; ELER, D. M. Coordinating multiple views using an ontology-based semantic mapping. In: 17th International Conference Information Visualisation, University of London, London, UK: IEEE Computer Society, 2013, p. 192-197.
- ELER, D. M.; PRATES, J. M.; GARCIA, R. E.; MINGHIM, R. Coordinated Multiple Views to Support Image Retrieval. In: 18th International Conference Information Visualisation, University of Paris Descartes, Paris, France, 2014.
- PRATES, J. M.; SCATALON, L. P.; GARCIA, R. E.; ELER, D. M. Ontologies to coordinate multiple views: exploring document collections. In: 16th International Conference on Artificial Intelligence, Las Vegas, USA, 2014.

Referências

- ALEKSOVSKI, Z.; KLEIN, M.; TEN KATE, W.; VAN HARMELEN, F. Matching unstructured vocabularies using a background ontology. In: *Proceedings of the 15th International Conference on Managing Knowledge in a World of Networks*, Springer-Verlag, 2006, p. 182–197.
- BALL, T.; EICK, S. G. Software visualization in the large. *Computer*, v. 29, n. 4, p. 33–43, 1996.
- BARBOSA, E. F.; NAKAGAWA, E. Y.; MALDONADO, J. C. Towards the establishment of an ontology of software testing. In: *18th International Conference on Software Engineering and Knowledge Engineering (SEKE 2006)*, San Francisco, CA: Knowledge Systems Institute, 2006, p. 522–525.
- BELLAHSENE, Z.; BONIFATI, A.; RAHM, E. *Schema matching and mapping*. Springer, 2011.
- BOUKHELIFA, N.; RODGERS, P. J. A model and software system for coordinated and multiple views in exploratory visualization. *Information Visualization*, v. 2, p. 258–269, 2003.
- CALERO, C.; RUIZ, F.; PIATTINI, M. *Ontologies for software engineering and software technology*. Berlin, Heidelberg: Springer-Verlag, 2006.
- CAMPOS, M.; CAMPOS, L. M.; DA SILVA MEDEIROS, J. A representação de domínios de conhecimento e uma teoria de representação: A ontologia de fundamentação. *Inf. Inf. Londrina*, v. 16, p. 140–164, 2011.
- CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B., eds. *Readings in information visualization: using vision to think*. San Francisco, CA, USA: Morgan Kaufmann Pub. Inc., 1999.
- CHANG, R.; CHARLOTTE, U.; GHONIEM, M.; KOSARA, R.; RIBARSKY, W.; YANG, J.; SUMA, E.; CHARLOTTE, U.; KERN, D.; SUDJIANTO, A. Wirevis: Visualization of categorical, time-varying data from financial transactions. In: *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST 2007)*, 2007, p. 155–162.
- CHEN, H. Towards design patterns for dynamic analytical data visualization. In: *Proceedings Of SPIE Visualization and Data Analysis*, 2004, p. 75–86.

- ELER, D. M. *Múltiplas visões coordenadas para exploração de mapas de similaridade*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação (ICMC-USP), 2011.
- ELER, D. M.; PAULOVICH, F. V.; DE OLIVEIRA, M. C. F.; MINGHIM, R. Topic-based coordination for visual analysis of evolving document collections. In: *Proceedings of the 2009 13th International Conference Information Visualisation*, Washington, DC, USA: IEEE Computer Society, 2009, p. 149–155.
- EUZENAT, J.; SHVAIKO, P. *Ontology matching*. Heidelberg (DE): Springer-Verlag, 2007.
- FELIZARDO, K.; MARTINS, R. M.; MALDONADO, J. C.; LOPES, A. A.; MINGHIM, R. Content based visual mining of document collections using ontologies. In: *II Workshop on Web and Text Intelligence (WTI)*, São Carlos, SP, Brazil, 2009, p. 1–8.
- FENSEL, D. *Ontologies: Dynamic networks of formally represented meaning*. Springer-Verlag Berlin Heidelberg, 2001.
- GARCIA, R. E. *Vidaese: processo de visualização exploratória para apoio a estudos empíricos em verificação, validação e teste de software*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação (ICMC-USP), 2006.
- GARCIA, R. E.; HÖHN, E. N.; BARBOSA, E. F.; MALDONADO, J. C. An ontology for controlled experiments on software engineering. In: *20th International Conference on Software Engineering and Knowledge Engineering (SEKE 2008)*, San Francisco, CA: Knowledge Systems Institute, 2008, p. 685–690.
- GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, v. 43, n. 5-6, p. 907–928, 1995.
- GUARINO, N. Formal ontology in information systems. In: *Proceedings of the 1st International Conference*, Trento, Italy: IOS Press, 1998.
- GUARINO, N.; OBERLE, D.; STAAB, S. *Handbook on ontologies*, cáp. What is an ontology? Springer Verlag, p. 201–221, 2009.
- HAVRE, S.; WHITNEY, P.; NOWELL, L. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. on Visualization and Comp. Graphics*, v. 8, p. 9–20, 2002.
- KEIM, D. A. Information visualization and visual data mining. *IEEE Trans. on Visualization and Comp. Graphics*, v. 8, n. 1, p. 1–8, 2002.
- KEIM, D. A.; KRIEGE, H.-P. Visualization techniques for mining large databases: A comparison. *IEEE Trans. on Knowledge and Data Engineering*, v. 8, p. 923–938, 1996.
- NORTH, C.; SHNEIDERMAN, B. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In: *Proceedings of the Working Conference on Advanced Visual Interfaces*, New York, NY, USA: ACM, 2000, p. 128–135.

- NORTH, C. L. *A user interface for coordinating visualizations via. relational schemata: Snap-together visualization.* Tese de Doutorado, University of Maryland, 2000.
- NOY, N. F.; MCGUINNESS, D. L. *Ontology development 101: A guide to creating your first ontology.* Relatório Técnico, Stanford University, 2001.
- OLIVEIRA, M. C. F.; LEVKOWITZ, H. From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics*, v. 9, p. 378–394, 2003.
- PATTISON, T.; PHILLIPS, M. View coordination architecture for information visualisation. In: *Proceedings of the 2001 Asia-Pacific symposium on Information visualisation*, Darlinghurst, Australia: Australian Comp. Soc., Inc., 2001, p. 165–169.
- PAULOVICH, F. V.; OLIVEIRA, M. C. F.; MINGHIM, R. The projection explorer: A flexible tool for projection-based multidimensional visualization. In: *XX Brazilian Symposium on Computer Graphics and Image Processing*, Washington, DC, USA: IEEE Computer Society, 2007, p. 27–36.
- PAULOVICH, F. V.; TOLEDO, F. M. B.; TELLES, G. P.; MINGHIM, R.; NONATO, L. G. Semantic wordification of document collections. *Comp. Graph. Forum*, v. 31, p. 1145–1153, 2012.
- PICKETT, R. M.; GRINSTEIN, G. G. Iconographic displays for visualizing multidimensional data. In: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 1988, p. 514–519.
- SABOL, V.; KIENREICH, W.; MUHR, M.; KLIEBER, W.; GRANITZER, M. Visual knowledge discovery in dynamic enterprise text repositories. In: *Proceedings of the 2009 13th International Conference Information Visualisation*, Washington, DC, USA: IEEE Computer Society, 2009, p. 361–368.
- SANTOS, R. Introdução à mineração de dados com aplicações em ciências espaciais. Disponível em <http://www.lac.inpe.br/rafael.santos/>. Acesso em: Jul/2014., 2014.
- SEELING, C.; BECKS, A. Analysing associations of textual and relational data with a multiple views system. In: *Proceedings of the 2nd International Conf. on Coordinated & Multiple Views in Exploratory Visualization*, Washington, USA: IEEE Computer Society, 2004, p. 61–70.
- SHIMABUKURO, M. H. *Visualizações temporais em uma plataforma de software extensível e adaptável.* Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação (ICMC-USP), 2004.
- SHNEIDERMAN, B. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, v. 11, n. 1, p. 92–99, 1992.

THAI, V.; HANDSCHUH, S.; DECKER, S. Tight coupling of personal interests with multi-dimensional visualization for exploration and analysis of text collections. In: *12th International Conference Information Visualisation*, Washington, DC, USA: IEEE Computer Society, 2008, p. 221–226.

WARE, C. *Information visualization: Perception for design*. San Francisco, CA, USA: Morgan Kaufmann Pub. Inc., 2004.

WEAVER, C. E. Building highly-coordinated visualizations in improvise. In: *Proceedings of the IEEE Symposium on Information Visualization*, Austin, TX: IEEE, 2004, p. 159–166.