

Fernando Benedet Ghisi

**UM MÉTODO PARA GERAÇÃO SEMIAUTOMÁTICA DE
SUMÁRIOS TEXTUAIS PARA APOIO À DISSEMINAÇÃO DE
CONHECIMENTO E AO PROCESSO DECISÓRIO EM
PROJETOS DE BUSINESS INTELLIGENCE**

Dissertação submetida ao Programa de Pós Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia e Gestão do Conhecimento.

Orientador: Denilson Sell, Dr.

Coorientador: Aran Bey Tcholakian, Dr.

Florianópolis
2013

Ficha de identificação da obra elaborada pelo autor por meio do Programa de Geração Automática da
Biblioteca Universitária da UFSC

Ghisi, Fernando Benedet

UM MÉTODO PARA GERAÇÃO SEMIAUTOMÁTICA DE SUMÁRIOS
TEXTUAIS PARA APOIO À DISSEMINAÇÃO DE CONHECIMENTO E AO
PROCESSO DECISÓRIO EM PROJETOS DE BUSINESS INTELLIGENCE /
Fernando Benedet Ghisi; orientador, Denilson Sell; co-
orientador, Aran Bey Tcholakian. - Florianópolis, SC, 2013.
133 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico. Programa de Pós-Graduação em
Engenharia e Gestão do Conhecimento.

Inclui referências

1. Engenharia e Gestão do Conhecimento. 2. Geração de
linguagem natural. 3. Business Intelligence. 4.
Ontologias. 5. Disseminação de conhecimento. I. Sell,
Denilson. II. Tcholakian, Aran Bey. III. Universidade
Federal de Santa Catarina. Programa de Pós-Graduação em
Engenharia e Gestão do Conhecimento. IV. Título.

Fernando Benedet Ghisi

**UM MÉTODO PARA GERAÇÃO SEMIAUTOMÁTICA DE
SUMÁRIOS TEXTUAIS PARA APOIO À DISSEMINAÇÃO DE
CONHECIMENTO E AO PROCESSO DECISÓRIO EM
PROJETOS DE BUSINESS INTELLIGENCE**

Esta dissertação foi julgada adequada para a obtenção do Título de Mestre em Engenharia e Gestão do Conhecimento, especialidade em Engenharia do Conhecimento, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina.

Florianópolis, 28 de fevereiro de 2013.

Prof. Gregório Jean Varvakis Rados, Dr.
Coordenador do Programa – UFSC

Banca Examinadora:

Prof. Denilson Sell, Dr.
Universidade Federal de Santa
Catarina
Orientador

Prof. José Leomar Todesco, Dr.
Universidade Federal de Santa
Catarina

Prof. Ronaldo dos Santos Mello,
Dr.
Universidade Federal de Santa
Catarina

Prof. Roberto Carlos dos Santos
Pacheco, Dr.
Universidade Federal de Santa
Catarina

Aos meus pais, Adilson e Maria Estela, aos meus avós, Mário (*in memoriam*) e Maria (*in memoriam*), e à minha esposa, Ana.

AGRADECIMENTOS

Para o desenvolvimento desta dissertação, além do meu próprio empenho e esforço, considero que algumas pessoas tiveram papel fundamental, para as quais listo a seguir os meus agradecimentos. Antes de tudo, gostaria de agradecer à minha família, sobretudo à minha esposa, Ana, e aos meus pais, Adilson e Maria Estela, por todo apoio e incentivo dado durante essa caminhada.

Também agradeço ao Professor Dr. Denilson Sell pelo excelente trabalho de orientação que foi realizado, pelos *insights* gerados em nossas conversas e pelos direcionamentos – sempre objetivos e eficientes – que me foram dados.

Agradeço à banca examinadora composta pelos professores Dr. Ronaldo dos Santos Mello, Dr. José Leomar Todesco e Dr. Roberto Carlos dos Santos Pacheco pelas críticas sempre construtivas feitas durante o desenvolvimento e a defesa deste trabalho.

Ao Instituto Stela, agradeço pela confiança, pelo incentivo e pela flexibilidade de horários, fatores que permitiram a minha participação nas atividades deste Programa de Pós-Graduação.

Agradeço também aos meus colegas de trabalho, que sempre me apoiaram, inclusive nos momentos mais difíceis dessa caminhada. Em especial, gostaria de agradecer aos amigos Flávio Ceci, Júlio Gonçalves Reinaldo, Marcio Napoli e Luyane Cardoso (que fez a revisão profissional deste documento).

Ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, deixo o meu agradecimento pela oportunidade de participar de tão prestigiado curso.

Por fim, agradeço às demais pessoas que participaram direta ou indiretamente do desenvolvimento deste trabalho.

Palavra puxa palavra, uma ideia traz outra, e assim se faz um livro, um governo, ou uma revolução, alguns dizem que assim é que a natureza compôs as suas espécies.

(Machado de Assis, 1883)

RESUMO

Nos sistemas de BI, é comum a apresentação de informações em gráficos, em tabelas e em indicadores de desempenho. Contudo, essas formas de visualização nem sempre são facilmente compreensíveis pelo tomador de decisão que precisa extrair o conhecimento relevante a partir das informações apresentadas. Muitas vezes, a apresentação de informações em um simples texto pode se mostrar mais efetiva como apoio ao processo de tomada de decisão do que através de formas gráficas. Nesse contexto, o presente trabalho teve como desafio estabelecer um método que possibilitasse a geração de sumários textuais visando apoiar a disseminação de conhecimento e os processos decisórios no âmbito de soluções de *Business Intelligence* (BI). Para demonstração da viabilidade do método proposto, fez-se a codificação do protótipo de um serviço de geração de sumários textuais analíticos, utilizando os recursos da Plataforma SBI – uma plataforma que faz uso de tecnologias semânticas. Nessa codificação, duas versões do protótipo foram implementadas – uma baseada em *templates* textuais e outra em *templates* estruturais. Por fim, foram realizadas simulações utilizando as implementações dos protótipos em cenários hipotéticos de BI, demonstrando o processo dinâmico de geração textual em diferentes contextos, e utilizando regras criadas por especialistas de domínio.

Palavras-chave: Geração de linguagem natural. Business Intelligence. Ontologias. Disseminação de conhecimento.

ABSTRACT

In BI systems, it is common presenting information through charts, tables or performance indicators. However, these forms of display are not always easily understood by the decision maker, who needs to extract the relevant knowledge from the information presented. Often, the presentation of information through a simple text may prove to be more effective to support the process of decision making than through graphical forms. In this context, the present work had as a challenge to establish an approach that enables the generation of textual summaries to support the dissemination of knowledge and decision-making processes within Business Intelligence solutions. To demonstrate the feasibility of the proposed approach, a prototype of service that generates analytical textual summaries was constructed, using resources of the SBI Platform - a platform that makes use of semantic technologies. In fact, two versions of this prototype have been implemented - one based on textual templates and another based on structural templates. Finally, simulations were performed using the prototype implementations in hypothetical BI scenarios, demonstrating the dynamic process of generating text in different contexts and using rules created by domain experts.

Palavras-chave: Natural Language Generation. Business Intelligence. Ontologies. Knowledge Dissemination.

LISTA DE FIGURAS

Figura 1 – Quadrantes de Morgan.....	35
Figura 3 – Exemplo de um modelo dimensional relacionado ao processo de pedidos de uma organização.....	45
Figura 4 – Ligação entre uma tabela de fato e uma tabela dimensional, ambas relacionadas a um processo de vendas.....	45
Figura 5 – Uma arquitetura para SGLN.....	55
Figura 6 – Tipos de ontologia (as setas representam relações de especialização).....	64
Figura 7 – Arquitetura SBI.....	68
Figura 8 – Os principais elementos da ontologia de BI.....	69
Figura 9 – Abordagem para geração textual utilizando recursos da Plataforma SBI.....	73
Figura 10 – Exemplo da utilização de uma propriedade booleana (egc:aumentoEhBom) para qualificar instâncias de medidas (sbi:measure) em uma base de conhecimento.....	76
Figura 11 – Exemplos de elementos de contexto envolvidos em um cenário de utilização de uma ferramenta de BI.....	77
Figura 12 – Exemplo da utilização de propriedades e conceitos linguísticos em anotações semânticas sobre instâncias de medidas (sbi:measure).....	80
Figura 13 – Exemplo de ambiente para configuração de regras de produção e de encadeamento de sentenças textuais.....	99
Figura 15 – Representação esquemática da codificação da arquitetura básica do componente.....	104
Figura 16 – Exemplo de um possível tipo de saída do componente de geração textual: um documento HTML.....	105
Figura 18 – Organização da codificação dos elementos básicos da arquitetura proposta.....	108
Figura 19 – Classe que representa o elemento orquestrador dos módulos da arquitetura.....	109
Figura 20 – Estruturação e fluxo de informações no protótipo desenvolvido.....	112

LISTA DE QUADROS

Quadro 1 – Exemplo de regras com alto grau de especificidade.....	84
Quadro 2 – Exemplo de regras com alto grau de generalidade.....	85
Quadro 3 – Exemplo de regras encadeadas.....	86
Quadro 4 – Exemplo de funcionamento de uma especificação de documento.....	89
Quadro 5 – Exemplo de funcionamento do módulo de microplanejamento do documento.....	89
Quadro 6 – Exemplo de das atividades de escolha léxica e geração de expressões para referência.....	90
Quadro 7 – Exemplo de funções representando pontos de flexibilidade em <i>templates</i> textuais.....	92
Quadro 8 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em <i>templates</i> textuais – exemplo de uma regra codificada e de um <i>template</i> estrutural.....	94
Quadro 9 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em <i>templates</i> textuais – exemplo de um contexto hipotético.....	94
Quadro 10 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em <i>templates</i> textuais – exemplo de um plano de documento.....	95
Quadro 11 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em <i>templates</i> textuais – exemplo de uma especificação de documento.....	95
Quadro 12 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em <i>templates</i> textuais – exemplo de um documento gerado.....	96
Quadro 13 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em <i>templates</i> textuais – exemplo de um <i>template</i> textual que geraria a mesma sentença.....	96
Quadro 14 – Propriedades indicando os nomes das classes com a implementação concreta dos módulos a serem injetados no serviço...	110
Quadro 15 – Regras codificadas no módulo de planejamento de documento dos protótipos.....	111
Quadro 16 – Parte do <i>lexicon</i> utilizado na abordagem não baseada em <i>templates</i> textuais.....	115

LISTA DE TABELAS

Tabela 1 – Aspectos relacionados à aquisição de conhecimento a partir de informações numéricas representadas por sumários textuais e por gráficos.....	28
Tabela 2 – Comparação dos ambientes OLTP e OLAP	47
Tabela 3 – Tarefas executadas por cada módulo.....	57
Tabela 5 – Tipos de OWL.....	65
Tabela 6 – Descrição dos conceitos da ontologia de BI.....	70
Tabela 7 – Elementos de contexto típicos em uma aplicação de BI e algumas informações que eles podem agregar como entrada às regras para geração textual.....	78
Tabela 8 – Representação dos templates textuais criados	114
Tabela 9 – Massa de dados do contexto 1	117
Tabela 10 – Massa de dados do contexto 2	118
Tabela 11 – Massa de dados do contexto 3	119
Tabela 12 – Massa de dados do contexto 4	120
Tabela 13 – Frases geradas pelo protótipo com abordagem baseada em <i>templates</i> textuais	120
Tabela 14 – Frases geradas pelo protótipo com abordagem não baseada em <i>templates</i> textuais	121

LISTA DE ABREVIATURAS E SIGLAS

BI – Business Intelligence
BPM – Business Performance Management
BSC – Balanced Score Card
CRM – Customer Relationship Management
DBA – Database Administrator
DSS – Decision Support System
DM – Data Mining
DW – Data Warehouse
EGC – Engenharia e Gestão do Conhecimento
EIS – Sistema de Informações Executivas
ERP – Enterprise Resource Planning
ETL – Extraction, Transformation and Loading
GUI – Graphical user interface
HTML – HyperText Markup Language
MER – Modelo Entidade-Relacionamento
NLG – Natural Language Generation
OLAP – Online Analytical Processing
OLTP – Online Transactional Processing
OWL – Ontology Web Language
SAD – Sistema de Apoio à Decisão
SBC – Sistema Baseado em Conhecimento
SBI – Semantic Business Intelligence
SCM – Supply Chain Management
SGBD – Sistema de Gerenciamento de Banco de Dados
SGI – Sistemas de Informações Gerenciais
SGLN – Sistema Gerador de Linguagem Natural
SQL – Structured Query Language
TI – Tecnologia da Informação
URI – Uniform Resource Identifier
XML – eXtensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	25
1.1	PERGUNTA DE PESQUISA	27
1.2	MOTIVAÇÃO.....	27
1.3	OBJETIVOS.....	32
1.3.1	Objetivo geral	32
1.3.2	Objetivos específicos	32
1.4	ADERÊNCIA AO OBJETO DE PESQUISA DO PROGRAMA	33
1.5	ESCOPO.....	34
1.6	METODOLOGIA.....	35
1.7	ESTRUTURA DO TRABALHO	36
2	FUNDAMENTAÇÃO TEÓRICA	39
2.1	BUSINESS INTELLIGENCE (BI).....	39
2.1.1	Histórico do BI	40
2.1.2	Arquitetura e componentes de BI	42
2.1.3	Data Warehouse	43
2.1.4	Ambiente de análises	46
2.1.5	Business Performance Management	50
2.1.6	Interface de usuário: dashboards e outras ferramentas de visualização	51
2.1.7	Algumas considerações sobre BI	52
2.2	GERAÇÃO AUTOMÁTICA DE TEXTOS	53
2.2.1	A Linguística e o estudo das linguagens	53
2.2.2	Geração automática de linguagem natural	54
2.2.3	Uma macroarquitetura para geração automática de textos	55
2.2.4	Considerações sobre a etapa de determinação do conteúdo	60
2.2.5	Algumas considerações sobre a geração de linguagem natural	61
2.3	REPRESENTAÇÃO DE CONHECIMENTO E TECNOLOGIAS SEMÂNTICAS.....	62
2.3.1	Ontologias	63
2.3.2	Business Intelligence semântico	67
3	MÉTODO PROPOSTO	73
3.1	ONTOLOGIAS NA REPRESENTAÇÃO DE CONHECIMENTO PARA APOIAR A GERAÇÃO TEXTUAL	75
3.1.1	Representação de conhecimento relacionado ao domínio da aplicação e aos elementos de BI	76

3.1.2	Representação de conhecimento relacionado a questões linguísticas	80
3.1.3	Utilização de conhecimento inferido.....	81
3.1.4	Considerações sobre a utilização de ontologias no método para geração textual	82
3.2	MÓDULO DE PLANEJAMENTO DO DOCUMENTO	82
3.2.1	A criação de regras para geração textual no contexto de BI.....	84
3.2.2	Considerações sobre o módulo de planejamento do documento.....	87
3.3	MÓDULO DE MICROPLANEJAMENTO.....	88
3.4	MÓDULO DE REALIZAÇÃO TEXTUAL	91
3.4.1	Implementações baseadas em templates textuais....	92
3.4.2	Implementações não baseadas em templates textuais.....	93
3.4.3	Considerações gerais sobre a implementação do módulo de realização textual.....	97
3.5	ESTRUTURAÇÃO DE AMBIENTES PARA CONFIGURAÇÃO DE REGRAS E DE TEMPLATES	98
3.6	VISÃO GERAL DO MÉTODO PROPOSTO	100
3.7	OUTRAS ABORDAGENS SEMELHANTES	102
4	VERIFICAÇÃO DO MÉTODO PROPOSTO	103
4.1	ESTRUTURAÇÃO BÁSICA DE UM SERVIÇO PARA GERAÇÃO DE TEXTOS	103
4.2	VISÃO GERAL DA IMPLEMENTAÇÃO DO PROTÓTIPO DO MÉTODO PROPOSTO.....	108
4.3	SIMULAÇÃO DE FUNCIONAMENTO DO MÉTODO PROPOSTO	116
4.3.1	Implementação baseada em templates.....	120
4.3.2	Implementação não baseada em templates.....	121
4.4	CONSIDERAÇÕES SOBRE A AVALIAÇÃO DA PROPOSTA.....	122
5	CONSIDERAÇÕES FINAIS.....	123
5.1	CONTRIBUIÇÕES.....	123
5.2	LIMITAÇÕES E TRABALHOS FUTUROS	123
	REFERÊNCIAS BIBLIOGRÁFICAS.....	127

1 INTRODUÇÃO

Na fase atual da economia mundial, o conhecimento assumiu o papel do fator de produção mais importante, ultrapassando o capital e a mão de obra (EVERS, 2001). Pode-se dizer que o ambiente de negócios sobre o qual as empresas operam atualmente está se tornando cada vez mais complexo e mutante. As empresas, privadas ou públicas, sentem as crescentes pressões e são forçadas a responder rapidamente a condições que estão sempre mudando, além de precisarem ser inovadoras na maneira que operam.

Diante dessa realidade, as empresas necessitam de agilidade nas frequentes tomadas de decisão (sejam elas estratégicas, táticas ou operacionais). Algumas dessas tomadas de decisão são muito complexas e podem exigir uma quantidade considerável de informações, bem como conhecimentos oportunos e relevantes (TURBAN et al., 2009). Assim, uma adequada gestão do conhecimento nas organizações é um importante meio para atingir efetividade e competitividade empresarial, pois a correta utilização de seus ativos de conhecimento é um ponto crítico para o sucesso da organização (STUDER; BENJAMINS; FENSEL, 1998; WIIG, 1997).

Nesse contexto, a literatura demonstra o valor que as soluções de *Business Intelligence* (BI) podem agregar às organizações por meio da explicitação do conhecimento potencialmente útil ao processo decisório (INMON, 2005 e KIMBALL et al., 2008). Contudo, Sell (2006) relata que a implantação e a utilização desse tipo de recurso nas organizações é, muitas vezes, malsucedida em virtude das limitações das soluções de BI tradicionais. Entre essas limitações, estão: (1) a não utilização da perspectiva semântica do negócio no apoio ao processamento analítico (sua terminologia e suas regras), e (2) a falta de flexibilidade para a extensão das funcionalidades exploratórias de acordo com as especificidades de cada organização. Segundo Kimball et al. (2008), os sistemas de BI precisam evoluir tão rapidamente quanto as organizações que o utilizam, o que torna necessário técnicas mais flexíveis e adaptáveis para o desenvolvimento e a evolução desse tipo de sistema.

Como evolução para as soluções de *Business Intelligence*, Sell (2006) propõe a utilização de tecnologias relacionadas à Web Semântica (BERNERS-LEE; HANDLER; LASSILA, 2001). Segundo o autor, essas tecnologias são utilizadas de diferentes maneiras para lidar com questões relacionadas a sistemas de informação, mas não têm uso específico no contexto de ferramentas analíticas. Dessa forma, Sell (2006) apresenta a Arquitetura SBI, uma abordagem para o

desenvolvimento de aplicações de BI baseadas na semântica do negócio. Essa arquitetura serviu, posteriormente, como base para o desenvolvimento da Plataforma SBI (SELL et al., 2008).

A Plataforma SBI apresenta as tecnologias semânticas – em especial, as ontologias – ao universo de BI e, com isso, abre novas possibilidades de desenvolvimento de estratégias para utilização do conhecimento da organização nos processos decisórios de forma potencialmente mais eficaz. Um dos trabalhos derivados dessa proposta foi apresentado por Silva (2011) e tem como foco a interpretação de perguntas feitas em linguagem natural dentro de sistemas de apoio à decisão, reconhecendo estruturas semânticas e transformando-as em consultas que retornam resultados relativos à questão elaborada pelo analista. Contudo, uma das limitações desse trabalho é que ele não chegou a ser uma abordagem *question-answering* (HIRSCHMAN; GAIZAUSKAS, 2001) completa, pois não apresentou respostas na forma de informações em linguagem natural, e sim na forma de resultados tabulados (semelhante a um relatório).

Segundo Fenn (2011), sistemas baseados em perguntas e respostas em linguagem natural (*natural-language question-answering*) suportarão, a longo prazo, a tomada de decisão de forma radicalmente mais rápida em serviços médicos, legais, financeiros e de relacionamento com o cliente. A expectativa é de que serviços enriquecidos com contexto (semântica) e sistemas baseados em perguntas e respostas em linguagem natural estejam entre as principais forças tecnológicas em um horizonte de cinco anos (FENN; LEHONG, 2011).

Parte da importância desse tipo de tecnologia – que modifica a forma de interação dos usuários com os sistemas de apoio à decisão – está no fato de que, nos sistemas de BI, é comum a apresentação de informações em gráficos, em tabelas e em indicadores de desempenho. Contudo, essas formas de visualização da informação nem sempre são facilmente compreensíveis pelo tomador de decisão, que precisa extrair conhecimento relevante das informações apresentadas. Estudos, como o de Law et al. (2005), demonstram que muitas vezes a apresentação de informações através de um simples texto pode mostrar-se mais efetiva como apoio ao processo de tomada de decisão do que a mesma informação apresentada de forma gráfica. Além disso, de acordo com Kacprzyc e Zadrozny (2009), os sumários textuais podem ser uma ferramenta poderosa para propiciar *insights* sobre relações existentes em bases de dados, fator que pode ser relevante em algumas atividades de negócio envolvendo processos decisórios.

Nesse cenário, um dos desafios que se apresenta na área de *Business Intelligence* é o desenvolvimento de uma abordagem sistematizada para a geração de sumários textuais que complementem a estratégia de difusão de informação e de disseminação de conhecimento¹ para apoio ao processo decisório. A complexidade dessa abordagem – além da dificuldade inerente à manipulação da linguagem natural na geração de textos – está relacionada às características dos projetos de BI, como o fato de utilizarem dados dinâmicos e heterogêneos provenientes dos mais diferentes domínios e de lidarem com regras de negócio que podem ser alteradas constantemente.

1.1 PERGUNTA DE PESQUISA

Como sistematizar a produção de sumários textuais para apoiar o processo de disseminação de conhecimento e os processos decisórios em projetos de *Business Intelligence*?

1.2 MOTIVAÇÃO

A produção de sumários textuais pelas ferramentas de apoio à decisão se apresenta como uma alternativa aos recursos gráficos para explicitação e disseminação de conhecimento nas organizações. Para verificar o potencial valor dos sumários textuais, Ghisi, Ceci e Sell (2011) compilaram algumas das principais questões encontradas na literatura relacionadas ao processo de visualização de informação e de disseminação de conhecimento a partir de representações textuais e gráficas de informações numéricas. Nesse trabalho, os autores consideraram tanto os aspectos cognitivos do ser humano, como as características gerais inerentes às duas formas de apresentação da informação.

Na Tabela 1, apresenta-se um resumo dos aspectos avaliados sob a ótica de utilização dos sumários textuais e dos gráficos, sendo eles: (i) naturalidade, (ii) escalabilidade, (iii) necessidade de conhecimento prévio, (iv) relevância das informações, (v) restrição de informações e (vi) acessibilidade.

¹ Distribuição de conhecimento (ou disseminação de conhecimento), “diz respeito ao processo de manter disponível o conhecimento para acesso fácil e rápido por aqueles que deles necessitam na organização ou em sua cadeia produtiva” (STEIL, 2007, informação oral).

Tabela 1 – Aspectos relacionados à aquisição de conhecimento a partir de informações numéricas representadas por sumários textuais e por gráficos

Aspecto	Sumários textuais	Representações gráficas
Naturalidade	Uso de linguagem natural é o único meio completamente natural de articulação e de comunicação.	Relacionada à prática de processamento perceptivo de estímulos visuais e ao aprendizado de convenções gráficas.
Escalabilidade	Escalabilidade máxima através do uso de sumarizadores linguísticos (ex.: todos, a maioria, grande parte etc.).	Limitações relacionadas à sobrecarga de informações e às características da mídia utilizada.
Necessidade de conhecimento prévio	Pode ser reduzida com a explicitação de informações complementares eventualmente necessárias.	O conhecimento prévio é crítico, afetando a seleção de informações e os processos de interpretação e inferência do indivíduo.
Relevância das informações	Pode sumarizar aspectos mais relevantes do conjunto de dados, embora possa ser difícil prever o que de fato é relevante para um indivíduo.	Dados irrelevantes podem mascarar dados importantes.
Restrição de informações	Quando sumarizado, o texto pode omitir informações importantes ou induzir a interpretações errôneas.	Elementos gráficos utilizados podem restringir ou forçar interpretações (restrição gráfica).
Acessibilidade	Possibilidade de utilização de mídias mais simples e de leitores de tela.	Necessidade de utilização de mídias com mais requisitos e pouca acessibilidade para pessoas com limitações visuais.

Fonte: Ghisi, Ceci e Sell (2011, p. 3)

No que diz respeito à **naturalidade**, o estudo aponta que uma questão importante a ser considerada quando se aborda a comunicação, é a naturalidade com que esse processo é estabelecido. Dessa forma, a utilização de um meio pouco natural pode prejudicar a sua eficácia. Nesse sentido, a linguagem natural – diferentemente das representações gráficas – é o único meio completamente natural de articulação e de comunicação entre seres humanos (KACPRZYK; ZADROZNY, 2009). Com relação aos gráficos, muitos dos benefícios presumidos das representações gráficas à moda antiga (por exemplo, diagramas estatísticos) foram considerados devido aos anos de prática de processamento perceptivo de estímulos visuais e ao aprendizado de convenções gráficas (SCAIFE, 1996). Dessa forma, a utilização de representações gráficas não convencionais poderia impactar negativamente na eficácia dos processos de comunicação e de aquisição de conhecimento, enquanto que a utilização de representações textuais (isto é, verbalizar) para apresentar a essência de um conjunto de dados poderia favorecer esses processos.

Em se tratando da **escalabilidade**, o estudo demonstra que, pela utilização da linguagem natural, pode-se atingir um nível de escalabilidade máximo, expressando informações de uma forma humanamente compreensível, sem se importar com o tamanho do conjunto de dados. Assim, termos linguísticos simples (como ‘muitos’, ‘poucos’ e ‘a maioria’) podem ser utilizados semanticamente como representações de conjuntos de dados de qualquer tamanho, promovendo um meio uniforme e facilmente compreensível de sumarizar os dados e de transmitir todas as intenções e necessidades de informação do indivíduo (KACPRZYK; ZADROZNY, 2009). Já em relação à escalabilidade das representações gráficas, ela pode ser reduzida com o excesso de informações visuais (poluição visual) e com as limitações cognitivas do ser humano, bem como com as limitações da própria mídia de apresentação das informações (características de tamanho e de resolução de tela, por exemplo).

Com relação à **necessidade de conhecimento prévio**, indica-se que ela poderia ser reduzida com a utilização de sumários textuais, pois com eles é possível explicitar informações complementares eventualmente necessárias. Por outro lado, em se tratando de representações gráficas, as diferenças individuais – especialmente o conhecimento prévio – são críticas no impacto que uma representação visual tem nas estruturas e nos processos cognitivos dos indivíduos, determinando a facilidade com que essa representação será percebida e

interpretada (COOK, 2006). Em gráficos complexos, o conhecimento do domínio pode afetar a seleção de informações, e os processos de interpretação e de inferência pelos indivíduos (CANHAM; HEGARTY, 2010).

Quanto à **relevância das informações**, as considerações do estudo vão no sentido de que a variedade e a complexidade com as quais as informações são apresentadas em alguns ambientes podem distrair as pessoas que não estão acostumadas com certo tipo de visualização. Isto é, pode deixá-las perdidas com detalhes irrelevantes, e gerar dificuldades para extrair as informações cruciais (STERN; APREA; EBNER, 2003). Por outro lado, os sumários textuais poderiam apresentar os aspectos mais relevantes do conjunto de dados. Contudo, é difícil prever o que é de fato relevante para os indivíduos, pois cada um de nós tem diferentes necessidades, está em momentos e contextos de utilização distintos, e possui razões particulares ao buscar informações. Os sumários textuais poderiam, por exemplo, apresentar o resultado de deduções, generalizações, extrapolações ou predições (TRAFTON et al., 2000).

Já com relação ao aspecto de **restrição das informações**, um problema que pode ocorrer com o uso de gráficos é a chamada restrição gráfica. Essa característica refere-se à forma como elementos gráficos podem restringir os tipos de inferências que podem ser feitas sobre os cenários representados. Isso ocorre quando os elementos gráficos não são capazes de mapear todas as relações entre as questões envolvidas em um espaço de problema, de forma que eles restringem (ou forçam) os tipos de interpretação que podem ser feitas (SCAIFE, 1996). Nesse mesmo sentido, argumenta-se sobre a utilização das representações textuais que, quando sumarizadas, podem também omitir informações importantes ou induzir a interpretações errôneas.

Por fim, em se tratando do nível de **acessibilidade** provido, o estudo evidencia duas questões para observação: (1) as limitações das mídias utilizadas para apresentação de informações e (2) as limitações dos consumidores dessas informações. Para o primeiro caso, ressaltam-se os requisitos de mídia impostos para visualização de informações de forma gráfica – geralmente ambientes digitais coloridos, com tamanho e resolução adequados são requeridos – e a possibilidade de utilização de mídias mais simples para apresentação de informações em forma textual. Já em relação ao segundo caso, pode-se dizer que, apesar do grande uso de gráficos para visualização de informações numéricas, esse é um meio problemático para pessoas com limitações visuais (FERRES et al., 2006). A comunidade cega, principalmente, é efetivamente

excluída de um vasto repositório de informações disponível, além de possuir oportunidades seriamente restritas na educação e em empregos que lidem com dados quantitativos (FERRES; LINDGAARD; SUMEGI, 2010). Com a utilização de sumários textuais, ampliam-se as possibilidades de acesso à informação, principalmente com a utilização de leitores de tela em ambientes digitais.

De forma geral, o estudo de Ghisi, Ceci e Sell (2011) evidencia que, em algumas circunstâncias e sob a ótica de aspectos como naturalidade, escalabilidade, necessidade de conhecimento prévio, relevância das informações, restrição de informações e acessibilidade, a simplicidade de uma descrição textual pode ser mais efetiva que a representação gráfica nos processos de comunicação e de aquisição de conhecimento. Dessa forma, pode-se dizer que os sumários textuais podem ser utilizados como um insumo potencialmente relevante para apoiar a disseminação de conhecimento e os processos decisórios nas organizações.

Pode-se dizer que, por possuírem características diferentes, a união das duas abordagens em ferramentas de *Business Intelligence* – sumários textuais complementando e suportando a visualização de informações gráficas – poderia proporcionar um sinergismo no apoio ao processo decisório. Essa terceira abordagem parece relevante e algumas vezes até indispensável quando está em pauta a visualização de informações complexas ou quando o consumidor potencial das informações não tem suficiente conhecimento do domínio em questão.

O aumento no tamanho e na complexidade dos dados torna a utilização de informações complementares para apoiar um processo de visualização da informação uma necessidade, em vez de uma opção (CHEN et al., 2009). Estratégias de visualização complexas costumam requerer que os indivíduos tenham não somente um grande conhecimento do domínio, mas também que decidam como e quais informações visualizar, que informações extrair, e o que fazer com esse conhecimento (TRAFTON et al., 2000).

Além disso, a falta de certos conhecimentos das pessoas é com frequência um grande obstáculo na implantação de técnicas de visualização gráfica. O indivíduo pode não ter recebido treinamento adequado para especificar configurações de visualização, ou pode não ter tempo suficiente ou habilidades de navegação para explorar todas as possibilidades de visualização possíveis. Ambos os cenários sugerem a necessidade de visualização assistida por conhecimento (CHEN et al., 2009). Nesse ponto, no contexto de *Business Intelligence*, a utilização de

sumários textuais pelas ferramentas de apoio à decisão, explicitando e disseminando conhecimento (presente nas bases de dados e de conhecimento da organização), e assistindo o processo de visualização de informações e de tomada de decisão parece se mostrar importante.

Ainda, em convergência com essa necessidade, a evolução científica e tecnológica parece tornar mais factível a apresentação de informações e a explicitação de conhecimento possivelmente relevante para o processo decisório por meio de textos gerados pelos sistemas de informação. Nessa linha, mostram-se promissoras as evoluções na área de Processamento de Linguagem Natural (uma subárea da Inteligência Artificial), mais especificamente na Geração de Linguagem Natural (REITER; DALE, 2000). Também se mostra importante o advento e o amadurecimento das tecnologias semânticas com todo o seu potencial de representação e armazenamento de conhecimento – principalmente com o desenvolvimento de arquiteturas para sistemas de apoio à decisão que se sustentam nessas tecnologias (SELL, 2006; SELL et al., 2008).

Dessa forma, esta pesquisa procura sistematizar uma abordagem que pode ser utilizada para disseminar conhecimento na forma de linguagem natural, buscando apoiar a disseminação de conhecimento e os processos decisórios no contexto de projetos de *Business Intelligence* que utilizam tecnologias semânticas.

1.3 OBJETIVOS

1.3.1 Objetivo geral

Estabelecer um método que possibilite a geração de sumários textuais visando apoiar a disseminação de conhecimento e os processos decisórios no âmbito de soluções de *Business Intelligence* que utilizam tecnologias semânticas.

1.3.2 Objetivos específicos

- Identificar o estágio atual de evolução dos sistemas de apoio à decisão no que diz respeito à apresentação de informações e à disseminação de conhecimento para o tomador de decisão, estabelecendo uma plataforma-base de *Business Intelligence* para o desenvolvimento de um método de geração de sumários textuais.
- Identificar, a partir das estratégias existentes para geração de linguagem natural, as abordagens que têm maior possibilidade de se adequar aos propósitos deste trabalho,

levando em consideração questões como: flexibilidade, extensibilidade e facilidade de uso, e integração.

- Desenvolver um método que possibilite a geração de sumários textuais dinâmicos baseados em informações disponibilizadas por sistemas de apoio à decisão e em modelos criados por especialistas de domínio.
- Desenvolver um protótipo funcional do método proposto, no qual – a partir de um contexto informacional dinâmico e de regras pré-definidas por especialistas – sejam gerados sumários textuais realçando algumas informações possivelmente relevantes no cenário apresentado.

1.4 ADERÊNCIA AO OBJETO DE PESQUISA DO PROGRAMA

Dentro das áreas de concentração do Programa de Pós-graduação em Engenharia e Gestão do Conhecimento (EGC) da Universidade Federal de Santa Catarina, este trabalho se insere na área de Engenharia do Conhecimento. Essa área tem como parte de seus objetivos o desenvolvimento de técnicas e ferramentas para a formalização, a codificação e gestão do conhecimento, e a aplicação de sistemas de conhecimento à gestão do conhecimento organizacional (EGC, 2012). Em relação à linha de pesquisa, este trabalho está inserido na categoria Teoria e Prática em Engenharia do Conhecimento, na qual se estuda metodologias e técnicas de Engenharia do Conhecimento e de Inteligência Computacional e suas relações com as áreas de Gestão do Conhecimento e de Mídia e Conhecimento (EGC, 2012).

De acordo com Studer, Benjamins e Fensel (1998), a nova Engenharia do Conhecimento pressupõe a existência de processos de modelagem de conhecimento em sistemas, os quais devem ser aplicados em atendimento a demandas da área de Gestão do Conhecimento. Além disso, de forma geral, em um sistema ou em uma solução de Engenharia do Conhecimento, assume-se que há alguma representação explícita de conhecimento inclusa no sistema (SCHREIBER, et. al., 2000). Finalmente, como finalidade, os sistemas de conhecimento podem ser destinados a apoiar as decisões de modo mais rápido e com maior qualidade, podendo, dessa forma, aumentar a produtividade das organizações.

Assim, no método proposto neste trabalho, serão utilizadas técnicas para modelagem de conhecimento e de sistemas. Também será concebida uma solução de Engenharia de Conhecimento para apoiar a

disseminação de conhecimento e os processos decisórios no contexto de *Business Intelligence*, visando atender, de forma direta, a demandas da área de Gestão do Conhecimento.

Por fim, é importante ressaltar que o presente trabalho apresenta uma vertente interdisciplinar, pois é fundamentado em áreas como *Business Intelligence*, Engenharia de Ontologias e Visualização da Informação, abordando ainda questões tratadas no âmbito da Linguística. Dessa forma, pode-se dizer que ele também está alinhado com o caráter interdisciplinar do Programa de Pós-graduação em Engenharia e Gestão do Conhecimento.

1.5 ESCOPO

Os objetivos deste trabalho não estão focados em um processo que busque perfeição linguística na produção de textos em linguagem natural, mas sim na possibilidade de apoiar a análise de dados e a tomada de decisão, disseminando conhecimento por sumários textuais gerados automaticamente por ferramentas analíticas no contexto de *Business Intelligence*.

Dessa forma, questões linguísticas supostamente envolvidas na produção de textos em linguagem natural não serão tratadas, em princípio, de forma aprofundada. De outra sorte, serão buscadas técnicas que simplifiquem o processo de geração textual. A manipulação direta de especificidades de uma linguagem sensível ao contexto – como as línguas portuguesa e inglesa – traria um grau de complexidade muito mais elevado ao trabalho e, por consequência, seus objetivos principais poderiam ser comprometidos. Além disso, técnicas para manipular as linguagens naturais de forma mais detalhada já são estudadas em profundidade por disciplinas específicas.

Já com relação ao conteúdo dos sumários textuais que serão gerados, pode-se dizer que, por não existir uma fórmula ideal – única e geral – para determinação de conteúdo relevante para ser apresentado a qualquer pessoa e em qualquer contexto, serão feitas apenas algumas propostas nesse sentido, cabendo às implementações específicas de cada projeto tratar dessas questões.

De forma geral, procura-se elaborar a proposta de um método que, embora inicialmente simplificada para efeitos didáticos e de demonstração de viabilidade, seja modularizada e extensível. Dessa forma, todos os seus componentes poderão ser definidos de forma mais adequada em termos de funcionalidade e de versatilidade, conforme as necessidades, recursos e limitações de cada projeto de *Business Intelligence*.

Além disso, é importante ressaltar que a intenção com o método proposto não é de gerar sumários textuais para substituir o raciocínio exploratório e as análises realizadas pelos tomadores de decisão. Segundo Kacprzyk e Zadrozny (2009), para resolver problemas complexos do mundo real efetivamente e eficientemente, os tomadores de decisão deveriam ser sustentados por sistemas de apoio à decisão (SADs), mas nunca substituídos por eles. Assim, os sumários textuais gerados neste trabalho estão estruturados como um meio para apoiar a análise de dados e a disseminação de conhecimento, visando suportar os processos decisórios nas organizações mediante a apresentação de informações complementares aos recursos visuais tradicionalmente utilizados pelas ferramentas analíticas.

Fora o já exposto, a qualidade e a relevância dos textos gerados dependem de uma série de fatores envolvendo, principalmente, métodos e técnicas específicos de geração textual, algoritmos para determinação do conteúdo utilizados pelo sistema, bem como a qualidade e a abrangência do conhecimento linguístico e de domínio codificados no sistema.

1.6 METODOLOGIA

Segundo a classificação apresentada por Silva e Menezes (2001), quanto à natureza, este trabalho é considerado uma pesquisa aplicada, pois tem como objetivo criar uma aplicação prática dirigida à solução de problemas específicos. Já em relação à abordagem do problema, é considerado qualitativo, pois está baseado em um modelo de desenvolvimento para o qual não é necessária análise estatística para qualquer comprovação. Do ponto de vista de seus objetivos, a pesquisa é exploratória, já que visa proporcionar uma maior relação com o problema por meio do levantamento bibliográfico e de exemplos já existentes.

Em relação à visão de mundo, utilizando como referência os Quadrantes de Morgan, visualizados na Figura 1, o presente trabalho se enquadra no paradigma funcionalista, pois procura examinar a regularidade e as relações que levam a generalizações e a princípios universais (BURREL; MORGAN, 1979).

Figura 1 – Quadrantes de Morgan



Fonte: adaptado de Burrell e Morgan (1979, p.22)

Por fim, como modalidade de pesquisa, este trabalho se enquadra na pesquisa tecnológica, na qual não há teorização, pois se tem como objetivo a criação de um instrumento tecnológico visando uma solução para uma demanda ou para uma necessidade pré-estabelecida.

1.7 ESTRUTURA DO TRABALHO

Esta dissertação está estruturada em cinco capítulos. Após este primeiro capítulo, organizou-se os restantes da seguinte forma:

- Capítulo 2 – Fundamentação teórica
- Capítulo 3 – Método proposto
- Capítulo 4 – Demonstração da viabilidade do método proposto
- Capítulo 5 – Considerações finais

Dessa forma, no Capítulo 2, são apresentadas as definições que orientaram o desenvolvimento do trabalho, por meio do resgate dos fundamentos sobre sistemas de apoio à decisão e *Business Intelligence*. A fundamentação teórica envolveu ainda os principais aspectos relacionados à área de geração automática de linguagem natural, questões sobre a representação de conhecimento, e tecnologias semânticas – com enfoque na plataforma SBI (*Business Intelligence Semântico*). No Capítulo 3, é apresentado o método proposto com vistas a atender os objetivos do presente trabalho e, no Capítulo 4, é feita uma verificação da viabilidade desse método através de mecanismos de

prototipação. Por fim, no Capítulo 5, são apresentadas algumas considerações finais e as possibilidades de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Tendo em vista os objetivos deste trabalho, faz-se necessário um estudo da literatura dirigido para o desenvolvimento de uma abordagem que possibilite a geração de sumários textuais relacionados às informações disponibilizadas por sistemas de apoio à decisão. Dessa forma, neste capítulo, é feito um estudo do estágio atual de evolução dos sistemas de apoio à decisão, envolvendo as questões de apresentação de informações e de disseminação de conhecimento para o tomador de decisão.

Também foi realizada uma investigação sobre as estratégias existentes para geração de textos em linguagem natural, observando e selecionando as que têm maior possibilidade de se adequar aos propósitos deste trabalho. Além disso, são estudadas formas para representação e para armazenamento de conhecimento de especialistas de domínio, visando subsidiar a geração textual no contexto de *Business Intelligence*.

Por fim, espera-se estudar e estabelecer uma plataforma-base de *Business Intelligence* para o desenvolvimento do método de geração de sumários textuais.

2.1 BUSINESS INTELLIGENCE (BI)

Para Turban et al. (2009), *Business Intelligence* (BI) é um termo guarda-chuva que inclui arquiteturas, ferramentas, bancos de dados, aplicações e metodologias. É uma expressão livre de conteúdo, portanto, significa coisas diferentes para pessoas diferentes. Segundo os autores, parte da confusão relacionada ao BI é causada pela enxurrada de acrônimos e de palavras da moda associadas a ele e à suas ferramentas, como, por exemplo, o *Business Performance Management* – BPM.

De acordo com Khan (2012), o BI pode ser apresentado como uma arquitetura, uma ferramenta, uma tecnologia ou um sistema que coleta e armazena dados, analisa-os utilizando ferramentas analíticas, propicia a criação de relatórios e consultas, e entrega informação ou o conhecimento com a finalidade de melhorar a tomada de decisão das organizações. Já para Golfarelli et al. (2004), o BI é o processo que transforma dados em informações, e então, em conhecimento.

De forma geral, os principais objetivos do BI são: (1) permitir o acesso interativo aos dados (por vezes, em tempo real), (2) proporcionar a manipulação desses dados e (3) fornecer aos gestores e analistas de negócios a capacidade de realizar análises adequadas ao se depararem

com dados, situações e desempenhos históricos e atuais, pois, ao analisarem esse vasto material, os tomadores de decisão conseguem valiosos *insights*, que podem servir como base para decisões melhores e mais informadas (TURBAN et al., 2009). Khan (2012) ressalta que o objetivo dos sistemas de BI é melhorar o tempo de resposta e a qualidade da informação, e assim possibilitar aos gestores entender melhor a posição das suas empresas em relação aos competidores. Aplicações e tecnologias de BI podem ajudar organizações a analisar mudanças de tendências no mercado, mudanças no comportamento dos clientes e dos padrões de gastos, preferências dos clientes, capacidades da organização e condições de mercado.

2.1.1 Histórico do BI

Para falar sobre BI, é importante apresentar um pouco do seu histórico. Segundo Turban et al. (2009), o termo BI foi cunhado pelo *Gartner Group* em meados da década de 1990. Contudo, o conceito nasceu muito antes, suas raízes estão nos sistemas de geração de relatórios SIG (Sistemas de Informações Gerenciais), datado de 1970. Nesse período, os sistemas de geração de relatórios eram estáticos, bidimensionais e não possuíam recursos de análise.

Ainda segundo o autor, no início dos anos 80, surgiu o conceito de sistemas de informações executivas (EIS). Esse conceito expandiu o suporte computadorizado aos gerentes e executivos de nível superior. Alguns recursos introduzidos foram sistemas de geração de relatórios dinâmicos multidimensionais (*ad hoc* ou sob demanda), prognósticos e previsões, análise de tendências, detalhamentos, acesso a *status*, e fatores críticos de sucesso. Esses recursos apareceram em dezenas de produtos comerciais até meados da década de 90. Mais tarde, os mesmos recursos e alguns recursos novos apareceram sob o nome de BI. Hoje se reconhece que todas as informações de que os executivos necessitam podem estar em um bom sistema de informações empresariais baseado em BI. Assim, o conceito original de EIS foi transformado em BI.

De acordo com Inmon (2005), a ideia por trás dos EIS era a apresentação de informações sem o real entendimento da infraestrutura necessária para criar essas informações. Pode-se dizer que, de certa forma, quando o conceito de *Data Warehouse* (DW)² apareceu, os EIS

² O conceito e os componentes de um *Data Warehouse* (DW) serão apresentados na seção 2.1.3.

foram transformados em BI. Já em 2005, os sistemas de BI começaram a incluir o recurso de inteligência artificial bem como recursos poderosos de análise (TURBAN et al., 2009). Somando esses recursos a outros conceitos e paradigmas, inicia-se o surgimento da segunda geração do BI, o BI 2.0.

Segundo logiXML (2007), o conceito de BI 2.0 segue a ideia de Web 2.0, que se refere à segunda geração da Web. Seu foco, assim como o da Web 2.0, é nas pessoas – possibilitando aos usuários expressar a sua criatividade e permitindo que eles acessem a informação livremente, produzindo algo significativa a partir dela. Ao mesmo tempo, o BI 2.0 está focado no compartilhamento de informações, na comunicação e na colaboração. Essa segunda geração representa uma coleção de abordagens tecnológicas, de funcionalidades de BI e de modelos de licenciamento inovadores, dinâmicos, adaptativos e colaborativos, com o intuito de levar o BI para as massas, tornando a informação disponível a qualquer hora, em qualquer lugar, para qualquer pessoa.

O objetivo do BI 2.0 é auxiliar ainda mais no processo de tomada de decisão, dando à pessoa certa exatamente a informação de que ela precisa, com a preocupação de que essa informação seja completa. Além disso, um foco muito forte do BI 2.0 é a colaboração. Muitas decisões, especialmente as mais importantes, envolvendo toda a empresa e a sua estratégia geral, não são tomadas em uma fração de tempo por uma única pessoa. Essas decisões dependem de muitos fatores e pessoas diferentes da empresa precisam participar delas. Assim, a ideia é melhorar a comunicação e o compartilhamento de informações através das ferramentas de BI – dessa forma, todos os envolvidos no processo decisório poderão incluir os seus conhecimentos e os seus pareceres em cada situação. Mas para que essa proposta seja fértil, tem que existir um ambiente que propicie a colaboração ativa e o compartilhamento imediato de informações (HAJNYSZ, 2007).

Para Hajnysz (2007), outra ideia por trás do BI 2.0, é utilizar dados de múltiplas fontes, não importando se elas estão localizadas em um servidor interno ou disponibilizadas na Web. Isso implica em uma mudança fundamental em relação às abordagens tradicionais: o suporte à utilização de dados não estruturados. Além disso, um conceito um pouco mais abstrato está sendo introduzido no BI 2.0: a semântica. Para Raden (2007), localizar a informação adequada para a resolução de um determinado problema tem que ser um processo semântico, sem que se tenha a necessidade de conhecer estruturas de dados ou formas

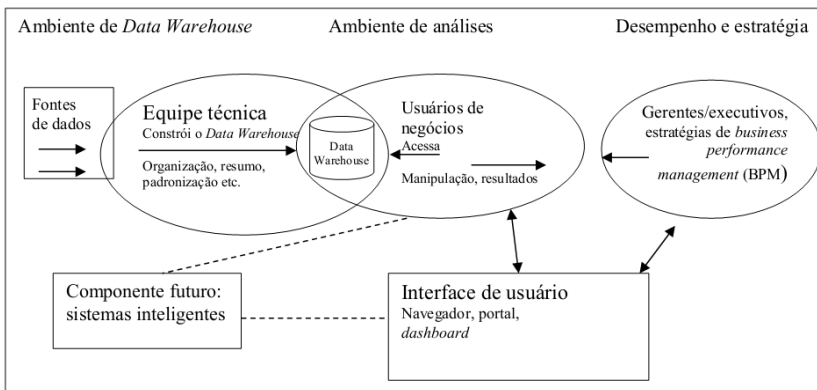
canônicas. A existência de uma camada semântica possibilita também a divulgação das informações constantes nos ambientes analíticos, com uniformização dos conceitos e com uma linguagem muito mais próxima dos usuários, além de um conteúdo muito mais completo e preciso. É aí que entram as tecnologias semânticas dando suporte às ferramentas de BI.

2.1.2 Arquitetura e componentes de BI

Segundo Turban et al. (2009), o BI pode ser apresentado por meio de quatro grandes componentes, conforme está ilustrado na Figura 2.

- Um **Data Warehouse (DW)** com seus dados-fonte.
- Um **ambiente de análises** – coleção de ferramentas para manipular e para analisar os dados no *Data Warehouse* (incluindo ferramentas OLAP e *Data Mining*).
- **Business Performance Management (BPM)** para monitoria e análise do desempenho.
- Uma **interface de usuário**, incluindo *dashboards* e outras ferramentas de visualização de informações.

Figura 2 – Uma arquitetura de alto nível do BI



Fonte: adaptado de Turban et al. (2009)

Dentro desse contexto, o ambiente de *Data Warehousing* é, sobretudo, de responsabilidade de uma equipe técnica, e o ambiente de análises está no âmbito dos usuários de negócios. Qualquer usuário pode se conectar ao sistema por meio de uma interface, por meio de um navegador. Além disso, e a alta administração pode usar o componente

de BPM e também os *dashboards*. A seguir, serão detalhados cada um dos quatro componentes de BI apresentados.

2.1.3 Data Warehouse

Partindo do lado esquerdo da Figura 2, vê-se o fluxo de dados dos sistemas operacionais (por exemplo, CRM, ERP etc.) até o *Data Warehouse* (DW), que é um banco ou um repositório de dados especial, preparado para dar suporte a aplicações de tomada de decisão (TURBAN et al., 2009). Segundo Inmon (2005), o (DW) é uma coleção de dados orientada por assunto, integrada, variável no tempo e não volátil, que proporciona suporte ao processo de tomada de decisões dos gestores. Para Kimball e Ross (2002), as principais características que um DW deve buscar são:

- Tornar as informações da organização acessíveis, de modo intuitivo e facilmente compreensível.
- Tornar as informações da organização consistentes, com alta qualidade e credibilidade.
- Ser adaptável a mudanças, sem comprometer dados históricos ou aplicações existentes.
- Controlar efetivamente o acesso às informações confidenciais da organização.
- Servir como base para as tomadas de decisão.
- Ter aceitação dos gestores, pois, diferentemente dos sistemas operacionais que costumam ser de uso obrigatório para o fluxo de trabalho, o DW normalmente é uma solução opcional a ser utilizada.

Uma das grandes diferenças entre os sistemas de bancos de dados operacionais e os *Data Warehouses* é a forma como os dados são armazenados, o que reflete diretamente no processo de modelagem. Segundo Kimball et al. (2008), no tradicional modelo entidade-relacionamento (MER), a estrutura é otimizada para recuperar, criar e atualizar registros individuais em tempo real, e para preservar a integridade dos dados. No entendimento dos autores, esse modelo não é adequado para a construção de um *Data Warehouse*. O baixo desempenho em consultas e a falta de uma navegabilidade adequada entre as tabelas para a apresentação das informações são alguns dos motivos citados. Para eles, a técnica mais viável (e amplamente aceita) para disponibilização de dados em soluções de *Business Intelligence* é a modelagem dimensional.

Nos ensinamentos de Kimball et al (2008), um modelo dimensional é um modelo de dados estruturado para atingir a máxima *performance* em consultas e com facilidade de uso, já que em um DW essas consultas são realizadas recuperando normalmente um grande número de registros e sumarizando seus dados segundo diferentes perspectivas. Assim, é utilizada uma estrutura que normalmente não é normalizada para evitar junções de muitas tabelas, obtendo-se assim uma *performance* superior. O modelo dimensional divide o mundo em duas partes: medidas e contexto. As medidas são capturadas pelos processos de organização e pelos sistemas operacionais que dão suporte a eles – elas geralmente são valores numéricos e costumam ser chamadas de fatos. Os fatos são envolvidos por um amplo contexto textual no momento em eles são armazenados. Esse contexto é intuitivamente dividido em grupos lógicos chamados de dimensões. As dimensões, por sua vez, descrevem contextos como: quem, o que, quando, onde, por que e como.

No ambiente dos SGBD relacionais, uma tabela-fato é baseada em um evento de medição, a qual geralmente possui um registro para cada medição distinta. Essa tabela-fato possui uma chave composta que faz a ligação com tabelas dimensionais, cada uma com uma única chave primária, a qual descreve precisamente o que é conhecido dentro do contexto dos registros de medição.

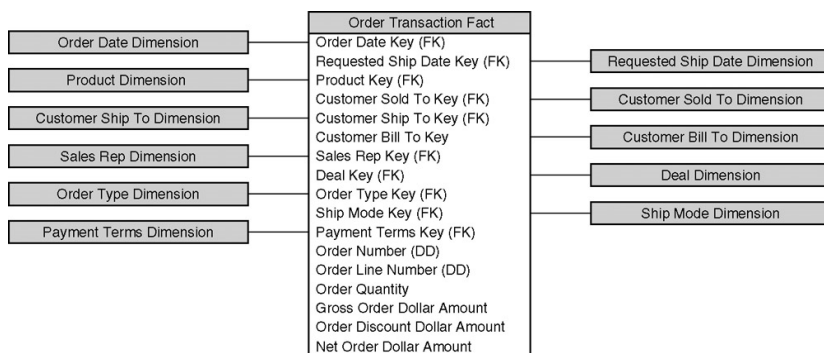
De forma geral, cada processo de negócio de uma organização pode ser representado por meio de um modelo dimensional, que consiste em uma tabela-fato contendo as medidas numéricas envolvida por um conjunto de tabelas dimensionais. Devido a essa estrutura característica, um modelo dimensional geralmente é chamado de esquema-estrela (KIMBALL et al, 2008).

Para Inmom (2005), um esquema-estrela é a base para um projeto multidimensional de um *Data Mart* – que ele define como sendo uma estrutura de dados departamentizada que compõe o *Data Warehouse*.

Dessa forma, no *Data Mart*, os dados são denormalizados³ com base nas necessidades de informação do departamento. Kimball et al. (2008) relatam que, embora também utilizassem frequentemente o termo *Data Mart*, não o fazem mais, já que o termo está caindo em desuso por representar conjuntos de dados sumarizados por departamento, independentes e não integrados com o restante da arquitetura. Por isso, os autores preferem adotar o termo *business process dimensional models* (ou modelos dimensionais dos processos de negócio).

Nas Figuras Figura 3 e Figura 4, podemos observar, respectivamente, um modelo dimensional de um processo de negócios relacionado aos pedidos da organização, e parte de outro modelo dimensional relacionado ao processo de vendas. Neste último, é apresentada a ligação entre uma tabela de fatos e uma dimensão relativa aos produtos associados às vendas.

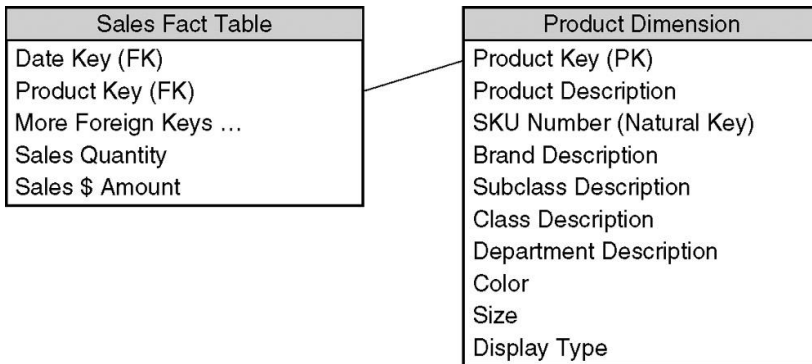
Figura 3 – Exemplo de um modelo dimensional relacionado ao processo de pedidos de uma organização



Fonte: Kimball et al. (2008, p. 235)

³ Os sistemas de banco de dados tradicionais utilizam como boa prática a normalização de suas tabelas, visando garantir a consistência dos dados, a diminuição de redundâncias e a minimização do espaço de armazenamento necessário. Contudo, algumas transações e consultas em bases de dados normalizadas podem se tornar lentas devido às operações de junção entre tabelas. O modelo dimensional, utilizado em DW, utiliza a denormalização de suas tabelas, visando o aumento do desempenho das consultas e a facilidade de utilização pelos usuários finais (KIMBALL; ROSS, 2002).

Figura 4 – Ligação entre uma tabela de fato e uma tabela dimensional, ambas relacionadas a um processo de vendas



Fonte: Kimball et al. (2008, p. 245)

De forma geral, o DW é construído seguindo algumas metodologias. Entre elas, cita-se o ETL (*Extract, Transformation and Load*), processo de extração, transformação e carga dos dados das diversas fontes de dados da organização para as tabelas do modelo dimensional (KIMBALL; ROSS, 2002; INMON, 2005). No processo de ETL, ocorre a consolidação e a integração dos dados organizacionais (limpeza, eliminação de redundâncias, validação etc.).

Para Kimball et al (2008), modelos dimensionais se demonstraram fáceis de entender, previsíveis, estendíveis e altamente responsivos a demandas *ad hoc* devido à sua natureza simétrica previsível. Além disso, eles são a base de muitas melhorias de *performance* dos SGBDs, incluindo potentes abordagens de indexação e agregação, e de desenvolvimento incremental distribuído de *Data Warehouses* mediante o uso de dimensões e de fatos adequados. Os modelos dimensionais são também a fundamentação lógica de todos os sistemas OLAP.

2.1.4 Ambiente de análises

Há muitas ferramentas de *software* que permitem aos usuários criar relatórios e consultas sob demanda, e realizar análise de dados. Elas surgiram originalmente com o nome de processamento analítico *online* (OLAP). Com essas ferramentas, os usuários podem, por exemplo, analisar diferentes dimensões de dados, como uma série temporal de vendas em cada região, e fazer análise de tendências.

Segundo Turban et al. (2009), durante muitos anos a TI se concentrou em construir sistemas essenciais que permitissem, principalmente, o processamento de transações corporativas. Tais sistemas deveriam ser praticamente tolerantes a falhas e oferecer execução eficiente e resposta rápida. O processamento de transações *online* (OLTP) ofereceu uma solução eficaz, que girou em torno de atividades repetitivas e de rotina, usando um ambiente de banco de dados relacional distribuído. Os últimos desdobramentos nessa área apontam para o uso de *softwares* de ERP e de gestão da cadeia de fornecimento (SCM) para tarefas de processamento de transações, aplicações de gerenciamento de relacionamento com o cliente (CRM) e integração com tecnologias baseadas na Web e em intranets.

Segundo o mesmo autor, o termo processamento analítico *online* (OLAP) se refere a uma variedade de atividades normalmente executadas por usuários finais em sistemas *online*. Não há consenso sobre quais atividades são consideradas OLAP. Mas, normalmente, OLAP inclui atividades como geração e resposta de consultas, solicitação e execução de relatórios e gráficos *ad hoc*, realização de análises estatísticas, e construção de apresentações visuais. Muitas pessoas também pensam em análise e apresentações multidimensionais, EIS e *Data Mining*, como atividades de OLAP. Basicamente, os produtos de OLAP oferecem recursos de modelagem, análise e visualização de grandes conjuntos de dados para sistemas de gerenciamento de banco de dados (SGBD) ou, mais frequentemente, para sistemas de *Data Warehouse*. Os produtos de OLAP oferecem também uma visão conceitual e multidimensional dos dados.

Segundo Pinheiro (2008), a melhor forma de entender um ambiente OLAP proposto através da implementação de um *Data Warehouse* é fazendo uma comparação direta com seu antecessor, o ambiente OLTP. Desse modo, segue uma tabela comparativa entre as duas abordagens.

Tabela 2 – Comparação dos ambientes OLTP e OLAP

OLTP	OLAP
<p>Controle do processo Em um ambiente operacional, as aplicações de banco de dados devem controlar o processo como um todo, envolvendo as diversas etapas de uma entrada ou a</p>	<p>Não controla o processo Em um ambiente de <i>Data Warehouse</i> não existe controle do processo, pois eles são todos controlados no ambiente operacional, onde já deveriam ter</p>

<p>atualização de dados.</p> <p>Grande volume de pequenas transações realizadas diariamente: o foco é na transação.</p> <p>Como essas aplicações devem controlar os processos, as transações pertinentes a esses processos devem ser atômicas (todas as operações de banco de dados são associadas a uma transação e devem ser efetuadas como um todo).</p> <p>Um registro por transação.</p> <p>Cada transação insere, atualiza ou remove apenas um registro da base de dados por vez. Mesmo existindo diversas operações em uma mesma transação, os registros das diversas tabelas da base de dados serão manipulados isoladamente, um de cada vez.</p> <p>Consistência microscópica dos dados.</p> <p>Devido ao conceito de atomicidade da transação, uma operação de banco de dados deve ser verificada completamente. Isso implica que todos os campos do registro que está sendo inserido, atualizado ou removido podem ser consistidos de forma isolada e, em caso de falha de apenas um deles, toda a transação pode falhar.</p> <p>Escopo temporal é momentâneo.</p> <p>Todas as informações que estão armazenadas no banco de dados</p>	<p>sido validados e consistidos.</p> <p>Pequeno volume de grandes transações realizadas diariamente: o foco é no conjunto de dados.</p> <p>Os dados oriundos do ambiente operacional podem ser sumarizados e agregados. Sendo assim, o que interessa não é o dado isolado, mas a sumarização deles.</p> <p>Milhares de registros por transação.</p> <p>A carga do ambiente operacional para um DW é caracterizada por pequenas quantidades de transações com grandes volumes de dados. Assim, dentro de uma transação no ambiente de DW, que é a carga do operacional, um grande número de registros é passado de um ambiente para o outro.</p> <p>Não há consistência, apenas consultas.</p> <p>A grande finalidade de um DW é proporcionar uma maneira eficaz de realizar consultas estratégicas para o processo de tomada de decisões. Como o ambiente é apenas para consulta, as informações nele contidas não precisam ser validadas, elas já deveriam ter sido, quando estavam no ambiente operacional.</p> <p>Escopo temporal é histórico.</p> <p>Conforme já mencionado, a finalidade do ambiente de DW é</p>
---	---

possuem um escopo temporal limitado, ou seja, em algum momento elas não terão mais importância ou validade e, conseqüentemente, serão deslocadas para uma área de armazenamento secundária (como um arquivo morto, p. ex.). Um sistema de banco de dados não poderia realmente armazenar indefinidamente todas as informações que são inseridas, pois isso poderia implicar em uma queda de *performance* considerável do sistema com o passar do tempo, podendo comprometer as novas operações.

Consultas pré-concebidas.

Em um ambiente operacional, as consultas à base de dados são pré-concebidas e implementadas. Essa implementação pode demandar um tempo considerável, já que as consultas devem ser especificadas pelo usuário e, posteriormente, codificadas, implementadas e testadas por um analista, programador ou DBA. O tempo de implementação de novas consultas pode ser proibitivo com relação à expectativa de resposta do usuário.

Ambiente estático.

Pelo fato das consultas serem pré-concebidas ou demandarem um tempo considerável para sua implementação, o ambiente operacional torna-se um ambiente estático, ou seja, ele é uma fotografia do estado corrente da aplicação de banco de dados

proporcionar um meio eficiente de consultar os dados de interesse para auxílio no processo de tomada de decisões. Essas consultas normalmente são complexas e envolvem comparações históricas. Essas comparações, no decorrer do tempo, determinam as estratégias a serem adotadas em determinado segmento dos negócios. Portanto, o tempo é uma dimensão fundamental em um ambiente de DW e, por isso, as informações nele contidas são armazenadas por um longo período de tempo.

Consultas *ad hoc*.

As consultas em um ambiente de DW podem ser realizadas de forma instantânea, ou seja, na medida em que elas vão sendo necessárias, o usuário compõe as diversas dimensões do DW para efetuar as consultas desejadas. Isso é possível graças ao modelo multidimensional fornecido pelo ambiente OLAP.

Ambiente dinâmico.

Como as consultas podem ser montadas na hora em que se fazem necessárias, o ambiente de DW torna-se um ambiente dinâmico. Isto implica num ganho substancial de tempo no processo de tomada de decisão. Algumas análises podem até continuar

utilizada. Qualquer alteração nessa aplicação ou em relatórios e consultas nela inseridos necessita ser implementada como um novo processo sistêmico.	sendo pré-concebidas (as mais solicitadas, por exemplo), mas sempre existirá a opção das consultas <i>ad hoc</i> .
---	--

Fonte: compilado de Pinheiro (2008)

Segundo Turban et al. (2009), o OLTP é voltado para o processamento de transações repetitivas em grandes quantidades e manipulações simples. O OLAP envolve o exame de muitos itens de dados (milhares ou até milhões) em relacionamentos complexos. Além de responder às consultas dos usuários, o OLAP consegue analisar esses relacionamentos e buscar padrões, tendências e exceções. Em outras palavras, OLAP é um método direto de suporte à decisão.

Outra classe de ferramentas presentes nos ambientes de análises das soluções de *Business Intelligence* são as soluções de *Data Mining* (DM). Elas são utilizadas para análise de informações, são baseadas em banco de dados, e buscam padrões ocultos em uma coleção de dados, os quais poderão ser utilizados para prever comportamentos futuros. O DM pode, por exemplo, ajudar empresas de varejo a encontrar clientes com interesses comuns. As ferramentas de *Data Mining* são utilizadas para substituir ou para aprimorar a inteligência humana devido à sua capacidade de verificar um volume enorme de dados. Dessa forma, elas descobrem novas e significativas correlações, padrões e tendências através de tecnologias de reconhecimento de padrões e de métodos estatísticos avançados (TURBAN et al., 2009).

2.1.5 Business Performance Management

O componente final do processo de BI é o *business performance management* (BPM), também conhecido por *corporate performance management*. Esse componente se baseia na metodologia *balanced scorecard*, a qual se trata de uma estrutura para definir, implementar e gerenciar a estratégia de negócios de uma empresa, conectando objetivos a medidas factuais. Em outras palavras, é uma estratégia para conectar métricas de nível superior, como as informações financeiras criadas por um diretor financeiro, a desempenhos reais de todos os níveis hierárquicos da corporação. O BPM usa a análise, a geração de relatórios e as consultas de BI. Seu objetivo é otimizar o desempenho geral de uma organização (TURBAN et al., 2009).

Para Neubauer Filho e Assad (2007), o BPM é, na sua essência, um conjunto de tecnologias que visa permitir um controle maior sobre regras de negócio, separando-as do fluxo de processo e controlando as atividades, sejam elas compostas por tarefas automatizadas ou executadas por elementos humanos. O BPM integra as atividades em redes de processos que, por vezes, têm partes executadas por fornecedores ou clientes, extrapolando a fronteira organizacional através da Web.

Segundo Turban et al. (2009), atualmente, a maioria das suítes de BI permite a aplicação de *balanced scorecards* pela capacidade de comparar imediatamente o desempenho dos negócios às metas estabelecidas. Além disso, segundo os autores, elas também oferecem uma plataforma para compartilhamento de metas de desempenho e de resultados em toda a empresa, permitindo que os gestores rapidamente compreendam como vão os negócios.

Esse controle sobre a operação é primordial para o alinhamento entre o mundo gerencial e o mundo operacional das organizações, pois esses dois mundos, não raramente, se encontram distanciados e até divergem em objetivos e direcionamentos (NEUBAUER FILHO; ASSAD, 2007).

2.1.6 Interface de usuário: dashboards e outras ferramentas de visualização

Os sistemas de BI normalmente incluem, em suas interfaces com o usuário, painéis analíticos (ou *dashboards*), que proporcionam uma visão rápida e abrangente do desempenho corporativo por meio de recursos gráficos. Esses painéis podem conter diferentes recursos e ferramentas de visualização – desde apresentações dos dados em cubos multidimensionais, passando pelos tradicionais gráficos, e incluindo, até mesmo, recursos avançados como realidade aumentada ou realidade virtual.

De qualquer forma, conforme já foi comentado, o estudo de Ghisi, Ceci e Sell (2011) evidencia que, em circunstâncias específicas e sob a ótica de alguns aspectos, a simplicidade de uma descrição textual pode ser mais efetiva do que a representação gráfica nos processos de comunicação e de aquisição de conhecimento. Dessa forma, os sumários textuais podem ser utilizados como um insumo potencialmente relevante para disseminação de conhecimento para o apoio a processos decisórios nas organizações.

De forma geral, o termo ‘visualização de dados’ refere-se às tecnologias que dão suporte à visualização e, algumas vezes, à interpretação de dados e de informações em vários pontos ao longo da cadeia de processamento de dados (FAYYAD et al., 2002 *apud* TURBAN et al., 2009). Essas ferramentas podem ajudar a identificar relações, como, por exemplo, tendências – sendo que a capacidade de identificar de maneira rápida tendências importantes em dados corporativos e de mercado pode fornecer vantagem competitiva (TURBAN et al., 2009). Segundo Turban et al. (2009), desde o fim dos anos 90, a visualização de dados evoluiu tanto na computação convencional, em que é integrada às ferramentas e às aplicações de suporte à decisão, como na visualização inteligente, que inclui a interpretação de dados (informação).

2.1.7 Algumas considerações sobre BI

No ambiente altamente competitivo de hoje, a qualidade e a pontualidade da informação de negócios para uma organização não é a escolha entre o lucro ou a perda, mas pode ser uma questão de sobrevivência ou de falência (TURBAN et al., 2009). Nesse contexto, a tecnologia de BI objetiva ajudar as pessoas a tomar melhores decisões de negócios por meio da disponibilização de informações relevantes, atualizadas e precisas, sempre que necessário (KHAN, 2012).

As organizações atuais estão obtendo mais valor com BI por estenderem as informações a muitos tipos de funcionários e, assim, maximizarem o uso dos ativos de dados existentes. Ferramentas de visualização que incluem *dashboards* são utilizadas por produtores, varejistas, governos e órgãos especiais. Mais e mais ferramentas de análise específicas do setor irão invadir o mercado para executar praticamente qualquer tipo de análise e para facilitar a tomada de decisões, desde o nível mais estratégico até o nível dos usuários (TURBAN et al., 2009).

Por outro lado, as soluções tradicionais de *Business Intelligence* – fortemente estruturadas e orientadas à rápida recuperação de informações pré-estabelecidas – ainda carecem de melhorias em alguns aspectos. Segundo Kimball et al. (2008), os sistemas BI precisam evoluir tão rapidamente quanto as organizações que os utilizam, o que torna necessário o desenvolvimento de novas técnicas, mais flexíveis e adaptáveis, para o desenvolvimento e para a evolução desse tipo de sistema.

Nesse sentido, a utilização de sumários textuais como meio de apresentação de informações complementares aos recursos

tradicionalmente utilizados pelas ferramentas analíticas (como gráficos, tabelas e indicadores de desempenho) pode se mostrar importante. Segundo Kacprzyk e Zadrozny (2009), a geração de sumários textuais a partir dos dados disponíveis em um sistema de apoio à decisão é uma abordagem interessante e promissora.

2.2 GERAÇÃO AUTOMÁTICA DE TEXTOS

Conforme já foi mencionado, a apresentação de informações por um simples texto em linguagem natural pode se mostrar mais efetiva como apoio ao processo decisório do que a mesma informação apresentada de forma gráfica. No contexto de sistemas para *Business Intelligence*, a dinamicidade e a variedade das fontes de informação, bem como os diferentes contextos e possibilidades de utilização trazem a necessidade de que os textos utilizados como mecanismos de visualização de informações sejam gerados dinamicamente, de forma automática, pelo sistema.

Antes de ser abordada efetivamente a geração automática de textos, na seção a seguir, é feita uma introdução sobre a Linguística e aspectos da linguagem natural.

2.2.1 A Linguística e o estudo das linguagens

A área de conhecimento que estuda as linguagens é chamada de Linguística – é uma área complexa, que possui diferentes enfoques, como a linguística propriamente dita, a psicolinguística, a neurolinguística e a linguística social (RADFORD et al., 2009). Segundo Chomsky (1978), um notório pesquisador da área de Linguística, conhecer uma linguagem significa ter uma representação mental de sua gramática.

De forma geral, uma gramática define como as palavras devem ser combinadas para formar expressões e sentenças – assim, um falante nativo de um idioma teria acesso a um sistema cognitivo que, de alguma forma, especifica essas possibilidades de combinação. Um aspecto muito importante dessa visão é que ela mostra como o sistema cognitivo (necessariamente finito, já que é representado no cérebro) pode, de alguma forma, caracterizar uma lista infinita de objetos: as expressões e sentenças em linguagem natural (RADFORD et al., 2009).

Ainda segundo Radford et al. (2009), a gramática de uma linguagem é formada por pelo menos quatro componentes: um dicionário (léxico), um componente sintático, um componente fonético e um componente lógico.

- O dicionário (léxicon) lista todas as palavras encontradas na linguagem. Cada entrada léxica (cada item listado no dicionário) informa sobre as propriedades linguísticas da palavra – por exemplo, suas propriedades fonológicas (como é a sua pronúncia), gramaticais (substantivo, verbo etc.), e semânticas (significado).
- O componente sintático especifica como combinar as palavras para formar expressões e sentenças.
- O componente fonético⁴ especifica como as palavras são pronunciadas. Muitas palavras mudam sua forma fonética na fala conectada, pois os fonemas sofrem determinação dos sons vizinhos dentro da própria palavra, expressão ou sentença.
- O componente lógico determina a forma lógica das sentenças (qual o seu significado ou como ela deve ser interpretada).

Alguns desses componentes são amplamente utilizados por sistemas geradores de linguagem natural, os quais serão abordados a seguir.

2.2.2 Geração automática de linguagem natural

A geração automática de linguagem natural, ou *natural language generation* (NLG), é um subcampo da Inteligência Artificial e da Linguística Computacional que se preocupa com a construção de sistemas computacionais que possam produzir textos compreensíveis em linguagem humana a partir de representações não linguísticas da informação. Sistemas geradores de linguagem natural combinam o conhecimento sobre a linguagem e o domínio de aplicação para, automaticamente, produzir documentos, relatórios, explicações, mensagens de ajuda, entre outros tipos de textos (REITER; DALE, 1997, 2000).

Com relação às abordagens existentes para geração automática de linguagem natural, pode-se dizer que não existe uma solução única e totalmente genérica para a geração de textos. Existem diferentes abordagens que vão desde as mais simples – utilizar sentenças predefinidas (*canned texts*) ou modelos (*templates*) que mesclam textos

⁴ O componente fonético não faz parte do escopo desse trabalho, já que este visa, inicialmente, a produção de textos escritos e não falados.

fixos e variáveis – até as mais complexas – gerar documentos analíticos completos e totalmente dinâmicos (específicos para cada cenário informacional apresentado) utilizando técnicas de Inteligência Artificial e de análise de dados, além de conhecimentos linguísticos e de domínio.

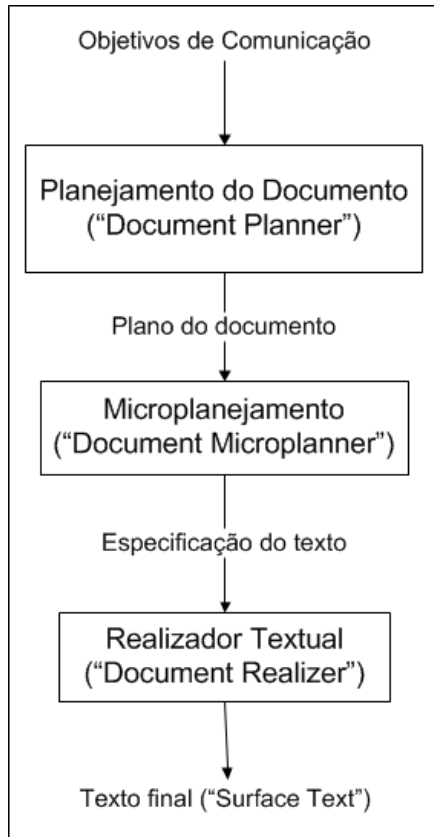
Uma das vantagens de trabalhar com abordagens que utilizam textos predefinidos é a menor necessidade de lidar com as complexidades envolvendo a gramática do idioma e outras questões inerentes às linguagens naturais, como as ambiguidades. Além disso, na maioria das vezes, a utilização de textos pré-definidos faz com que sejam apresentadas sentenças com formas de escrita mais naturais; contudo, por causa dessa simplicidade no processo de geração do texto, as sentenças acabam sendo mais inflexíveis (CULLEN et al., 2009). É nas abordagens mais complexas que ocorrerem os processos de geração de linguagem natural de fato (de forma mais automática e com menor dependência humana), utilizando informações não linguísticas como parâmetros de entrada e informações linguísticas (como gramáticas) para estruturação das sentenças.

Diante das abordagens expostas, pode-se dizer que, dependendo das necessidades, das limitações existentes e, inclusive, da arquitetura de *software* e da arquitetura de informações disponíveis, os sistemas geradores de linguagem natural podem utilizar as abordagens mais simples ou as mais complexas nos diferentes projetos.

2.2.3 Uma macroarquitetura para geração automática de textos

Conforme já foi mencionado, não existe uma abordagem única e totalmente genérica para geração de textos. Dessa forma, Reiter e Dale (2000) apresentam a proposta de uma arquitetura abstrata para sistemas geradores de linguagem natural, conforme pode ser observado na Figura 5. Para os autores, muitos dos sistemas geradores de linguagem natural (SGLN) seriam formados basicamente por três módulos conectados sequencialmente, no estilo *pipeline* (no qual a saída de um módulo serve como entrada do módulo subsequente).

Figura 5 – Uma arquitetura para SGLN



Fonte: adaptado de Reiter e Dale (2000)

Dessa forma, a saída do módulo **Planejamento do documento** serve de entrada para o módulo **Microplanejamento**, e a saída desse módulo, por sua vez, serve como entrada para o **Realizador de texto**. Além disso, cada um desses três grandes módulos executam tarefas específicas, relacionadas ao tratamento do conteúdo ou da estrutura dos documentos a serem gerados, conforme pode ser observado na Tabela 3⁵.

⁵ Nem todas as tarefas elencadas são sempre realizadas pelos módulos abstratos correspondentes; e, quando são, nem sempre são realizadas de forma isolada e com uma ordem fixa de execução (algumas vezes são, inclusive, realizadas em conjunto durante um mesmo processo).

Tabela 3 – Tarefas executadas por cada módulo

Módulo	Tarefas (conteúdo)	Tarefas (estrutura)
Planejamento do documento	Determinação de conteúdo	Estruturação do documento
Microplanejamento	Lexicalização Geração de expressões de referência	Agregação
Realização textual	Realização linguística	Realização estrutural

Fonte: adaptado de Reiter e Dale (2000)

Assim, o módulo de planejamento do documento (*Document Planner*) engloba as tarefas de **determinação de conteúdo** e de **estruturação do documento**. A determinação de conteúdo é a tarefa de decidir quais informações deveriam ser comunicadas no documento a ser gerado – o que irá depender dos objetivos de comunicação traçados, do público-alvo que irá utilizar o documento, do contexto informacional apresentado e de restrições impostas, como o tamanho máximo de texto permitido. A estruturação, por sua vez, é a tarefa de decidir como o conteúdo será apresentado dentro do documento – em que ordem ele será apresentado, como ele será agrupado e como serão feitos os relacionamentos desses grupos.

Já o módulo de microplanejamento (*Microplanner*) pode conter tarefas como a de **lexicalização**, **geração de expressões de referência** e **agregação**. A lexicalização é a tarefa de decidir quais palavras específicas (como substantivos, verbos, adjetivos e advérbios) ou estruturas sintáticas serão utilizadas para expressar o conteúdo selecionado pelo módulo de planejamento. A geração de expressões de referência é a tarefa que decide que expressões serão utilizadas para referenciar as entidades (concretas e abstratas) dentro do contexto apresentado. Já a agregação é a tarefa de estruturação do módulo de microplanejamento, decidindo como a estruturação criada pelo módulo de planejamento será mapeada em estruturas linguísticas, como sentenças e parágrafos.

Por fim, o módulo de realização (*Surface Realizer*) engloba as tarefas de **realização linguística** e **estrutural**. Na primeira, as

representações abstratas das sentenças são convertidas em texto real, utilizando as regras gramaticais da linguagem. Já na segunda, as representações estruturais abstratas, como de sentenças e parágrafos, são convertidas em marcações (metadados) que serão utilizadas futuramente na formatação dos documentos pelo componente que fará a apresentação.

A Tabela 4, apresentada a seguir, mostra um resumo dos módulos apresentados.

Tabela 4 – Resumo dos módulos da arquitetura

Planejamento do documento	Determinação de conteúdo	Que informações irão aparecer no texto de saída?
	Estruturação do documento	Em que ordem e de que forma as diferentes informações serão apresentadas nesse documento? Por exemplo, ao apresentar informações meteorológicas, pode-se primeiramente definir que seja apresentada a temperatura, para depois apresentar o volume de precipitação da chuva.
Microplanejamento	Lexicalização	Que termos específicos (substantivos, verbos, adjetivos, advérbios) serão utilizados para expressar o conteúdo definido anteriormente? Obs.: esses termos podem ser tirados de um <i>Lexicon</i> . Em algumas situações, algumas estruturas sintáticas também podem ser definidas neste módulo. Por exemplo, pode-se dizer “o carro que é de propriedade da Maria” ou “o carro da Maria”.

	<p>Geração de expressões de referência</p>	<p>Como referenciar as entidades (concretas e abstratas) presentes no documento?</p> <p>Obs.: a mesma entidade pode ser representada de várias formas. Por exemplo, o mês de março do ano passado pode ser referenciado como: “Março de 2012”, “Mês de Março”, “Março do ano passado”, “Nesta ocasião”, entre outros (dependendo do contexto e das outras sentenças presentes no documento).</p>
	<p>Agregação</p>	<p>Como apresentar de forma fluída e natural as estruturas estabelecidas no plano do documento na forma de estruturas linguísticas como sentenças e parágrafos?</p> <p>Por exemplo, duas informações podem ser expressas em duas sentenças ou em apenas uma: “O mês foi mais frio do que na média. O mês foi mais úmido do que na média.” ou, simplesmente, “O mês foi mais frio e mais úmido do que na média.”.</p>
<p>Realização textual</p>	<p>Realização linguística</p>	<p>Geração do texto propriamente dito (às vezes utilizando gramáticas) a partir de representações abstratas</p>

		definidas na especificação do documento.
	Realização estrutural	Conversão de estruturas abstratas, como parágrafos, em símbolos de marcação que serão utilizados para guiar a renderização e a forma visual de apresentação do texto.

Fonte: adaptado de Reiter e Dale (2000)

É importante salientar que a arquitetura apresentada é do tipo abstrata, prevendo uma série de tarefas que poderiam ser implementadas de várias formas pelos diferentes sistemas. Além disso, ela é apenas uma proposta e, conforme foi constatado por Mellish et al. (2006), não é adotada em sua totalidade por muitos dos sistemas que fazem geração automática de linguagem natural. De qualquer forma, os módulos e as tarefas propostas por Reiter e Dale (2000) são bem definidos e foram validados em diferentes ambientes, podendo – com as devidas adaptações requeridas por cada contexto de implementação – ser utilizados para embasar a construção de serviços de geração automática de textos para sistemas de apoio à decisão.

2.2.4 Considerações sobre a etapa de determinação do conteúdo

Uma etapa muito importante de um processo de geração textual, em que possivelmente estariam algumas tarefas intensivas em conhecimento (SCHREIBER et al., 2000), é a fase de determinação do conteúdo a ser produzido.

Primeiramente, uma questão importante a se abordar na etapa de determinação de conteúdo é o objetivo e a comunicação para a qual um texto em linguagem natural será escrito – o que pode ser, em parte, explicitado pela Teoria dos Atos de Fala (AUSTIN, 1962). Segundo Freitas (2003), a comunicação humana tem sido modelada por essa teoria, que considera que a linguagem falada tem por objetivo engendrar ações e provocar mudanças no ambiente. Segundo o autor, os Atos de Fala, anteriormente estudados em Processamento de Linguagem Natural, uma área de Inteligência Artificial, são classificados como assertivos (informar), diretivos (pedir ou consultar), comissivos (prometer ou comprometer-se), proibitivos, declarativos (causar eventos

para o próprio comunicador) e expressivos (emoções). Dessa forma, para os propósitos de geração de texto escrito para apoio a processos decisórios, a categoria da teoria que se enquadra melhor, pelo menos em um primeiro momento, é o ato assertivo (informar).

Outra questão relevante a ser tratada nesta etapa é a dinamicidade (adaptabilidade) de um sistema gerador de textos em relação ao contexto informacional apresentado (parâmetros de entrada). Segundo Bontcheva (2005), as técnicas de geração de linguagem natural utilizam modelos que incluem informações do contexto e do usuário para a seleção das estratégias mais adequadas de apresentação das informações textuais. Assim, pode-se dizer que, quanto mais dinâmico e adaptável for o sistema de geração de linguagem natural em relação ao contexto apresentado, mais surpreendente ele pode ser, e mais valor ele pode agregar para um processo decisório.

Nessa linha, no que diz respeito à adaptabilidade dos textos gerados em função do público-alvo (ou do perfil do usuário envolvido em um processo decisório), pode-se utilizar estudos da área da Linguística como embasamento da forma como utilizar a linguagem em diferentes situações. Mais especificamente, existe um ramo da linguística especializado em estudar a forma como a linguagem é utilizada em diferentes contextos. Segundo Radford (2009), a Sociolinguística é o estudo da relação entre o uso da linguagem e a estrutura da sociedade. Ela leva em consideração fatores como os *backgrounds* sociais do interlocutor e do público-alvo (por exemplo, sua idade, sexo, classe social ou etnia), a relação entre o interlocutor e o público-alvo (como amigos, empregador-empregado ou professor-aluno) e o contexto e forma de interação (por exemplo, em casa, na empresa, no supermercado, por telefone ou por e-mail), argumentando que esses fatores são cruciais para entender a estrutura e a função da linguagem utilizada em cada situação.

2.2.5 Algumas considerações sobre a geração de linguagem natural

É importante ressaltar que, atualmente, a tecnologia de geração de linguagem natural se encontra em um estado paradoxal: por um lado, pesquisas já elucidaram muitas questões fundamentais e conceberam soluções que são suficientemente robustas e escaláveis para uso prático. Mas, por outro lado, muito poucos sistemas de geração de linguagem natural estão implantados de forma produtiva nas organizações. A tecnologia de geração de linguagem natural tem permanecido

virtualmente desconhecida e inutilizada no meio profissional (MACEDO, 2010).

Reiter e Sripada (2003) argumentam que os sistemas geradores de linguagem natural, assim como muitos sistemas de Inteligência Artificial, precisam de uma quantidade substancial de conhecimento. Segundo os autores, as experiências que eles tiveram em dois projetos sugeriram que é difícil adquirir o conhecimento correto para esses sistemas – de fato, toda técnica de aquisição de conhecimento que foi testada teve problemas significativos. Em termos gerais, esses problemas ocorreram devido à complexidade, ineditismo, e pouca compreensão da natureza das tarefas que o sistema utilizava, e foram piorados pelo fato de as pessoas escreverem de maneiras muito diferentes. Isso significou, para eles, que as abordagens de aquisição de conhecimento baseadas em *corpus* de documentos tiveram problemas, porque era impossível montar um *corpus* considerável de textos de alta qualidade, escritos manualmente, dentro de seus domínios. Além disso, técnicas estruturadas de aquisição de conhecimento orientada a especialistas tiveram problemas, porque os especialistas discordaram entre si e porque os pesquisadores não conseguiram informações suficientes sobre os casos especiais e não usuais para a construção de um sistema robusto.

2.3 REPRESENTAÇÃO DE CONHECIMENTO E TECNOLOGIAS SEMÂNTICAS

Conforme já mencionado, os sistemas geradores de linguagem natural precisam de uma quantidade substancial de conhecimento. Dessa forma, são necessários mecanismos para a formalização e para o armazenamento desse conhecimento, seja o conhecimento relacionado ao domínio das organizações (conceitos e regras de negócio) ou relacionado às próprias questões linguísticas.

De acordo com Takeuchi e Nonaka (2008), o conhecimento em si é formado por dois componentes dicotômicos e aparentemente opostos: o conhecimento explícito e o conhecimento tácito. O conhecimento explícito pode ser expresso em palavras, números ou sons e compartilhado na forma de dados, fórmulas científicas, recursos visuais, fitas de áudio, especificações de produtos ou manuais. O conhecimento tácito, por outro lado, não é facilmente visível e explicável, pelo contrário, é altamente pessoal e difícil de formalizar, tornando-se de comunicação e compartilhamento difícil.

Uma das formas amplamente difundidas para explicitar e formalizar conhecimentos, principalmente após o advento da Web

Semântica (BERNERS-LEE, HANDLER e LASSILA, 2001) são as ontologias, que serão apresentadas a seguir.

2.3.1 Ontologias

Nos dias de hoje, a noção de ontologia é vista sob duas perspectivas: uma mais tradicional, da filosofia, e uma mais recente, das ciências da computação. A ontologia filosófica surgiu, provavelmente, com os filósofos gregos, há mais de 2400 anos; a ontologia computacional (alguma vezes chamada de engenharia ontológica ou de ontologias) surgiu por volta de 15 anos atrás. Segundo Poli, Healy e Kameas (2010), a perspectiva filosófica das ontologias tem como foco a análise categórica – isto é, pensar em quais são as entidades do mundo e quais são as categorias dessas entidades. Nesse sentido, em um primeiro momento, a intenção da análise categórica é fazer um ‘inventário’ da realidade. Já a perspectiva das ciências da computação sobre ontologias (isto é, ontologia como tecnologia), ainda segundo os autores, tem as mesmas questões como foco, mas a intenção é distinta: criar modelos de engenharia sobre a realidade, gerando artefatos que podem ser manipulados por *software* e, possivelmente, diretamente interpretados por raciocínios realizados via *softwares* especiais chamados motores de inferência (agregando ao *software*, dessa forma, a semântica humana).

Uma das definições clássicas de ontologias foi dada por Gruber (1993), e diz que uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada. Ele ainda define a conceitualização como uma visão abstrata e simplificada do mundo que se quer representar por algum propósito. Segundo Guarino (1998), uma ontologia se refere a um artefato de engenharia constituído por um vocabulário específico e usado para descrever certa realidade, mais um conjunto de assertivas explícitas relacionadas ao significado pretendido das palavras no vocabulário. Segundo o autor, esse grupo de assertivas tem, usualmente, a forma de uma teoria lógica de primeira-ordem, em que palavras do vocabulário aparecem como nomes de predicados unários ou binários chamados, respectivamente, de conceitos e relações. No caso mais simples, uma ontologia descreve uma hierarquia de conceitos relacionados por relações de classificação. Em casos mais sofisticados, axiomas apropriados são adicionados com o intuito de expressar outras relações entre conceitos e para restringir a sua interpretação conforme o planejado.

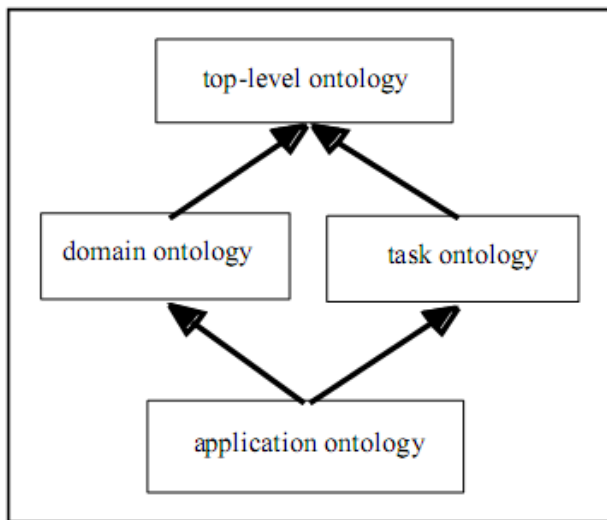
No que diz respeito à Engenharia de Ontologias, a concepção de ontologias deve ser conduzida como qualquer outro projeto de *software*,

no sentido de serem tomadas decisões de projeto que determinam sua qualidade, a qual se baseia em critérios como eficiência, legibilidade, portabilidade, extensibilidade, interoperabilidade e reuso. Por isso, tal concepção deve basear-se em seu futuro emprego, e não somente em aspectos filosóficos do conhecimento acerca do domínio representado (FREITAS, 2003).

2.3.1.1 Classificação das ontologias

Segundo Guarino (1998), as ontologias podem ser classificadas em diferentes tipos de acordo com o seu nível de generalização – ou nível de dependência em relação a uma tarefa ou ponto de vista.

Figura 6 – Tipos de ontologia (as setas representam relações de especialização)



Fonte: Guarino (2008, p.9)

Segundo o autor, as ontologias *top-level*, de alto nível, descrevem conceitos gerais – como espaço, tempo, matéria, objeto, evento e ação –, que são independentes de um problema ou domínio particular. Sendo assim, parece razoável, pelo menos na teoria, que se tivessem ontologias de alto nível unificadas, disponíveis para toda a comunidade de usuários.

Já as ontologias de domínio e de tarefa descrevem, respectivamente, o vocabulário relacionado a um domínio genérico (como medicina ou automóveis) e uma tarefa genérica ou atividade

(como diagnosticar ou vender), especializando os termos introduzidos na ontologia de alto nível.

Por fim, as ontologias de aplicação descrevem conceitos dependentes de domínios e tarefas específicos, sendo geralmente uma especialização das duas ontologias combinadas. Esses conceitos correspondem com frequência a papéis desempenhados por entidades de domínio na execução de certas atividades.

2.3.1.2 OWL

Segundo Smith, Welty e McGuinness (2004), a OWL (*Web Ontology Language* ou, em português, Linguagem de Ontologia da Web) foi concebida com intuito de prover uma linguagem que pode ser utilizada para descrever as classes e as suas relações, que estão inerentes a documentos Web e a aplicações.

Uma ontologia OWL pode incluir descrições de classes, de propriedades e das suas instâncias. Dada uma ontologia desse tipo, a semântica formal da OWL especifica como derivar suas consequências lógicas, ou seja, fatos não presentes literalmente dentro da ontologia, mas escondidos na semântica. Esses fatos poderiam estar baseados em um único documento ou em múltiplos documentos distribuídos, que foram combinados utilizando mecanismos da OWL.

A linguagem OWL fornece três sublinguagens com expressividade incremental, projetadas para a utilização de comunidades específicas de usuários e de desenvolvedores, conforme pode ser observado na tabela a seguir.

Tabela 5 – Tipos de OWL

Tipo de sublinguagem	Descrição
OWL Lite	Suporta usuários que precisam primariamente de uma hierarquia de classificação e de alguns mecanismos de restrições. Por exemplo, mesmo suportando restrições de cardinalidade, ela só permite valores 0 ou 1. Dessa forma, ficaria mais fácil desenvolver ferramentas para suportar a OWL Lite, em relação aos tipos mais expressivos de OWL, e promover um caminho de migração rápida para tesouros e outras taxonomias.
	Suporta usuários que querem expressividade

OWL DL	<p>máxima sem perder as características de computabilidade dos sistemas raciocinadores, como a completude (garantia de que todas as conclusões serão tomadas) e a decibilidade (todas as conclusões serão efetuadas em tempo finito).</p> <p>A OWL DL inclui todas as propriedades da OWL com algumas restrições, como de tipo (uma classe não pode ser também um indivíduo ou uma propriedade, e uma propriedade não pode ser também um indivíduo ou uma classe).</p> <p>O OWL DL recebeu esse nome – acrônimo DL – devido a sua correspondência com a lógica descritiva (campo de pesquisa que tem estudado um fragmento da lógica de primeira ordem e que tem a característica de poder ser sempre decidido). Por essa correspondência, a OWL DL tem as características computacionais desejáveis para sistemas raciocinadores.</p>
OWL Full	<p>Suporta usuários que querem a máxima expressividade e liberdade sintática do RDF, sem garantias computacionais. Por exemplo, uma classe pode ser tratada ao mesmo tempo como uma coleção de indivíduos e como um único indivíduo.</p> <p>A OWL Full permite ainda que uma ontologia sobrescreva o significado dos vocabulários pré-definidos (por RDF ou OWL), dentre outras questões. Assim, é improvável que os sistemas raciocinadores sejam capazes de suportar todas as características da OWL Full, tornando-se imprevisíveis.</p>

Fonte: compilado de Smith, Welty e McGuinness (2004)

Além do que já foi apresentado, cada sublinguagem é uma extensão da sua predecessora, de modo que uma ontologia OWL Lite é uma ontologia OWL DL, e uma ontologia OWL DL é uma ontologia OWL Full. Pelas características de expressividade e computabilidade, a OWL DL acaba sendo bastante interessante para uso em sistemas de conhecimento que se utilizam de ontologias.

Recentemente foi publicada a especificação da evolução da OWL, a OWL 2, que adiciona novas funcionalidades à linguagem, mas mantém a compatibilidade com a versão predecessora (W3C OWL WORKING GROUP, 2012).

2.3.1.3 Considerações sobre o uso das ontologias

Nos anos recentes, as ontologias têm sido adotadas em muitas organizações e em comunidades científicas como um meio de compartilhar, reutilizar e processar o conhecimento de domínio. As ontologias são agora a base de muitas aplicações, como portais de conhecimento científico, sistemas de gestão da informação e de integração, comércio eletrônico e *webservices* semânticos (STANFORD CENTER FOR BIOMEDICAL INFORMATICS RESEARCH, 2012). Dentro do contexto de sistemas de apoio à decisão, seguindo essa mesma tendência, surgiu recentemente o conceito de *Business Intelligence* semântico.

2.3.2 Business Intelligence semântico

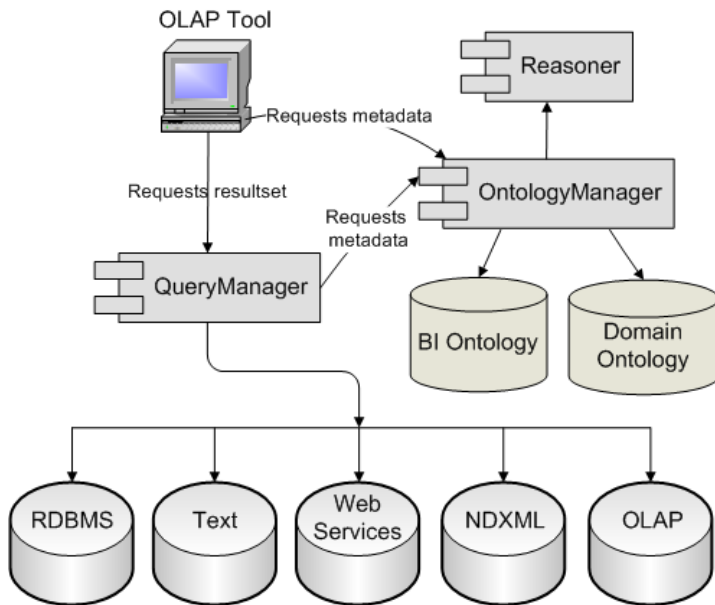
Já foi comentado, na seção 2.1, sobre a importância das ferramentas de BI para as organizações. Apesar disso, ainda existem desafios que têm que ser trabalhados para aumentar o impacto do BI nos processos decisórios. Esses desafios incluem as dificuldades para estender as ferramentas de acordo com os requisitos do negócio, a falta de suporte para analisar e interpretar os dados, e a falta de flexibilidade para customizar a apresentação das informações de acordo com o perfil dos usuários (SELL et al., 2012). Segundo os autores, essas questões ocorrem pela falta de integração da semântica do negócio à infraestrutura das ferramentas analíticas.

Apesar disso, conforme já mencionado, a utilização da semântica do negócio já faz parte da infraestrutura de algumas aplicações. As tecnologias relacionadas à Web Semântica (BERNERS-LEE; HANDLER; LASSILA, 2001) – também chamadas de tecnologias semânticas – têm sido aplicadas de diferentes maneiras no apoio aos sistemas de informação. Guarino (1998) enumera algumas referências na literatura sobre a utilização de ontologias em áreas como Engenharia do Conhecimento, modelagem de banco de dados, modelagem e integração de informações, análise orientada a objetos, extração e recuperação de informações, e modelagem de sistemas baseados em agentes.

Segundo Sell (2006), as tecnologias da Web Semântica eram utilizadas de diferentes maneiras para lidar com questões relacionadas a sistemas de informação, mas não especificamente no contexto de ferramentas analíticas. O autor propôs então, uma arquitetura para aplicações de *Business Intelligence* baseada na semântica do negócio, a Arquitetura SBI. Essa arquitetura serviu posteriormente como base para o desenvolvimento da Plataforma SBI (SELL et al., 2008).

Ainda de acordo com Sell et al. (2008), a Plataforma SBI é composta por um conjunto de módulos fracamente acoplados. Parte da arquitetura dessa plataforma pode ser observada na Figura 7.

Figura 7 – Arquitetura SBI



Fonte: Sell et al. (2008, p.3)

As ontologias utilizadas pela Plataforma SBI incluem a semântica do negócio (*Domain Ontology*) e descrevem as relações entre essa semântica, a terminologia de BI, e as fontes de dados da organização (*BI Ontology*). As ontologias são utilizadas por um módulo chamado *QueryManager* para interpretar as requisições de informações realizadas pelas ferramentas analíticas e executá-las em fontes de dados heterogêneas (ilustradas na parte inferior da figura), possibilitando assim

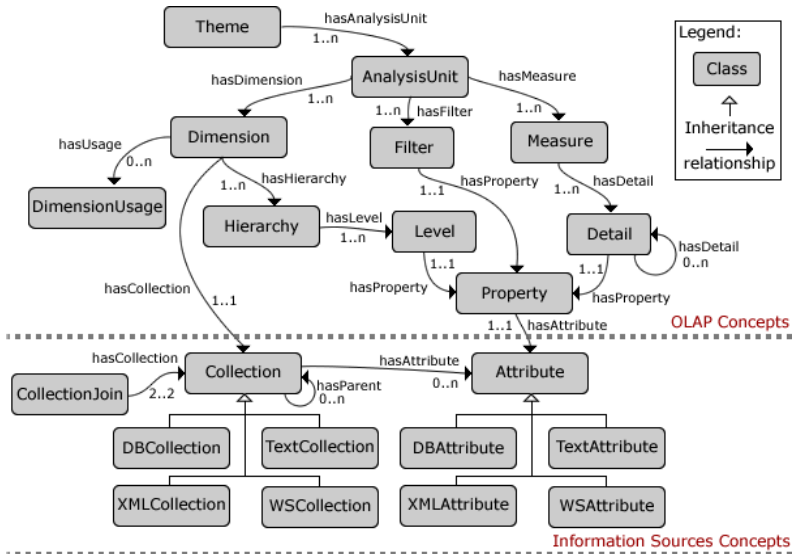
a combinação de dados estruturados e não estruturados nas mesmas análises.

O *OntologyManager* é o módulo que possibilita o acesso às ontologias utilizadas pela plataforma. Esse módulo conta com um *reasoner* (interpretador), que suporta inferências *on-the-fly* (em tempo de execução) ou em *batch* (pré-processadas) sobre a semântica do negócio. Essas inferências ampliam os resultados das consultas e tornam possíveis operações de *slice* e de *drill*⁶ guiadas pela semântica e baseadas nas regras do negócio.

Pode-se dizer que a ontologia de BI é o elemento central dessa plataforma, servindo de elo entre os diferentes componentes da aplicação. Além de propiciar a realização das inferências semânticas, a ontologia BI propicia a apresentação dos resultados das consultas utilizando a terminologia do negócio e provê – por conceitos de *Business Intelligence* que estão modelados nela – uma abstração sobre as fontes de dados da organização. Esse comportamento pode guiar a interação do tomador de decisão na exploração analítica dos dados. Os principais elementos da ontologia de BI estão apresentados na Figura 8.

Figura 8 – Os principais elementos da ontologia de BI

⁶ Através do processo chamado *drill*, o usuário pode aumentar (*drill down*) ou diminuir (*drill up*) o nível de detalhamento (granularidade) dos dados. Já por meio dos processos de *slice and disse* é possível criar visões dos dados por meio de sua reorganização, de forma que eles possam ser examinados sob diferentes perspectivas (KIMBALL; ROSS, 2002).



Fonte: Sell et al. (2012, p. 36)

Em resumo, a ontologia de BI apresenta conceitos OLAP – utilizados pelas ferramentas analíticas – e conceitos relacionados às fontes de informação – representando estruturas de fontes de dados e mapeando essas estruturas com os conceitos de domínio da organização (representados na ontologia de domínio da plataforma). A Tabela 6 apresenta maiores detalhes sobre cada um dos elementos ilustrados na Figura 8.

Tabela 6 – Descrição dos conceitos da ontologia de BI

Conceito	Descrição
Tema (<i>Theme</i>)	Um tema representa um conjunto de documentos e de tabelas de fato e de dimensões associadas a um processo de negócio, como, por exemplo, Pesquisa e Desenvolvimento (P&D).
Unidade de análise (<i>AnalysisUnit</i>)	Define coleções de documentos e tabelas de fato relacionados a assuntos específicos de um tema. Em um tema de P&D, por exemplo, poderia existir uma unidade de análise relacionada à evasão escolar. Uma unidade de análise pode conter diferentes medidas e dimensões associadas.

Medida (<i>Measure</i>)	Esse conceito é utilizado para representar valores quantitativos, agregações ou sumarizações relacionadas ao conteúdo de uma unidade de análise (por exemplo, número de estudantes).
Filtro (<i>Filter</i>)	Os filtros são atributos de dimensões que podem ser aplicados em operações de <i>slice</i> ou <i>dice</i> sobre os dados relacionados a uma unidade de análise (por exemplo, idade dos estudantes, gênero, entre outros).
Dimensão (<i>Dimension</i>)	Descreve as dimensões de uma unidade de análise. Pode envolver diferentes hierarquias e propriedades (por exemplo, Unidade federativa).
Hierarquia (<i>Hierarchy</i>)	Esse conceito descreve hierarquias de dimensões. Cada hierarquia é composta por um ou mais níveis (por exemplo, Cidade, Unidade federativa e País).
Nível (<i>Level</i>)	Representa um nível hierárquico que pode ser utilizado em operações de <i>drill-up</i> e de <i>drill-down</i> .
Detalhamento (<i>Detail</i>)	Descreve como unidades de análise podem ser detalhadas ou apresentadas em seu nível atômico (por exemplo, nome e matrícula).
Propriedade (<i>Property</i>)	Identifica a terminologia utilizada para identificar uma unidade de informação. Também mapeia instâncias do conceito de atributo e instâncias dos conceitos de detalhamento, filtro, nível e medidas.
Uso da dimensão (<i>DimensionUsage</i>)	Descreve como coleções de dados são relacionadas a unidades de análise.
Coleção (<i>Collection</i>)	Esse conceito representa uma coleção de dados ou um provedor de dados, e descreve como as fontes de dados se relacionam com os conceitos representados na ontologia de domínio.
Junção da coleção (<i>CollectionJoin</i>)	Descreve como uma coleção pode ser relacionada com outra coleção. Também identifica quais propriedades e quais operações são utilizadas para relacionar essas duas coleções.
Atributo (<i>Attribute</i>)	Corresponde aos itens contidos nas coleções, como os campos de uma tabela, os elementos XML, as entidades extraídas de documentos ou as colunas de planilhas eletrônicas. Esse conceito também associa os itens citados com conceitos representados na ontologia de domínio.

Fonte: adaptado de Sell et al. (2012)

O arcabouço tecnológico apresentado pela Plataforma SBI (o qual é baseado em tecnologias semânticas) possibilita que os desenvolvedores customizem as soluções de BI de acordo com os requisitos analíticos específicos de cada organização, o que pode tornar possível a oferta de serviços inovadores para guiar os tomadores de decisão durante a análise de seus negócios – as soluções poderiam incluir a geração de sumários textuais analíticos (possivelmente customizados de acordo com o perfil do analista) ou, até mesmo, uma abordagem para acessar as informações do negócio através de perguntas e respostas (*question-answering*). Em suma, a Plataforma SBI pode suportar uma nova geração de ferramentas analíticas que poderá guiar os tomadores de decisão desde a investigação de seus negócios até a implementação de ações de acordo com os *insights* obtidos nessas investigações.

Diante do exposto, por possuir um arcabouço tecnológico que se vislumbra necessário e adequado, a Plataforma SBI e suas ontologias são utilizadas como base para o desenvolvimento e para os testes do método para geração de sumários textuais para apoio à decisão que é proposto neste trabalho.

3 MÉTODO PROPOSTO

A arquitetura abstrata para sistemas geradores de linguagem natural proposta por Reiter e Dale (2000) – seção 2.2.3 –, somada aos recursos oferecidos pela Plataforma SBI – seção 2.3.2 – representa o arcabouço do método para operacionalização de sistemas geradores de sumários textuais para apoiar a disseminação de conhecimento e os processos decisórios no contexto de *Business Intelligence*.

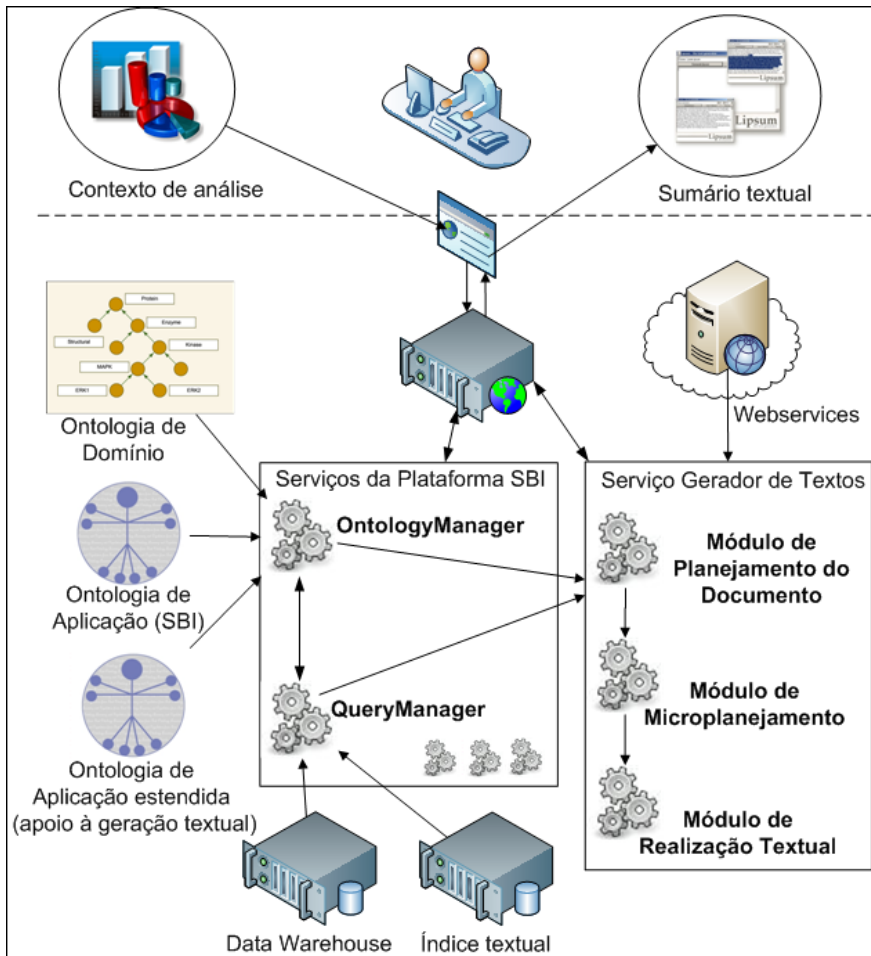
Dessa forma, a abordagem proposta no presente trabalho é constituída por 3 módulos conectados sequencialmente, na qual a saída de um módulo serve como entrada do módulo subsequente – no estilo *pipeline*. Esses módulos são: (1) Planejamento do documento, (2) Microplanejamento e (3) Realização do texto.

Na arquitetura proposta por Reiter e Dale (2000), as ferramentas e métodos utilizados por cada módulo e as estruturas de dados utilizadas na comunicação entre eles (entradas e saídas dos módulos) não estão definidas e podem variar em cada abordagem de implementação desenvolvida. Assim, neste capítulo, será proposto um método que considere a divisão das responsabilidades descritas para os três módulos e o cenário de geração de sumários textuais para apoiar a disseminação do conhecimento e os processos decisórios, no contexto de uma plataforma de *Business Intelligence Semântica*.

Tem-se também como objetivo deste trabalho o desenvolvimento um método flexível e extensível (adaptável às necessidades e limitações de cada projeto). Dentro da estrutura apresentada, serão expostas diferentes propostas e alternativas para a operacionalização de cada módulo, utilizando como insumo os recursos oferecidos pela Plataforma SBI.

De forma geral, o método proposto integrado com alguns recursos da Plataforma SBI, pode ser visualizado na Figura 9.

Figura 9 – Abordagem para geração textual utilizando recursos da Plataforma SBI



Fonte: elaborado pelo autor

A seguir, é apresentado um detalhamento dos módulos e recursos envolvidos no serviço gerador de textos do método proposto, bem como as perspectivas de interação com os serviços e recursos disponibilizados pela Plataforma SBI. Primeiramente, são apresentadas as ontologias como recurso para representação do conhecimento que é utilizado no processo de geração textual. Em seguida, são então apresentados os módulos (1) Planejamento do documento, (2) Microplanejamento e (3) Realização textual, os quais utilizam as ontologias e os serviços da Plataforma SBI, além de outros recursos externos, como *webservices*.

3.1 ONTOLOGIAS NA REPRESENTAÇÃO DE CONHECIMENTO PARA APOIAR A GERAÇÃO TEXTUAL

Conforme apresentado na seção 2.3.2, na Plataforma SBI, as ontologias utilizadas incluem: (1) a semântica do negócio (Ontologia de Domínio) e a descrição das relações entre essa semântica, (2) a terminologia de BI, (3) e as fontes de dados da organização (Ontologia de Aplicação (SBI)) – estejam elas em repositórios estruturados (como em um *Data Warehouse*) ou em repositórios não estruturados (como em índices textuais).

Partindo-se do pressuposto que as soluções de *Business Intelligence* têm algumas características comuns e independentes do domínio de aplicação (incluindo, por exemplo, alguns recursos, nomenclaturas e práticas), é possível sugerir possibilidades genéricas envolvendo a representação de conhecimento com vistas à geração de sumários textuais nesse contexto.

Assumindo essa função, as ontologias da Plataforma SBI, complementadas – quando necessário – por outra ontologia para apoio direto aos módulos de geração textual (Ontologia de Aplicação Estendida), formam a base de conhecimento que será utilizada para suportar um serviço de geração de sumários textuais analíticos. Nessa linha, quando uma Ontologia de Aplicação Estendida é utilizada⁷, os elementos presentes nela fazem referência direta aos elementos das ontologias da Plataforma SBI, estejam eles na Ontologia de Domínio ou na Ontologia de Aplicação (como os temas de análise, as medidas, as dimensões e os filtros).

Por outro lado, além das ontologias da Plataforma SBI formarem a base de conhecimento que irá apoiar diretamente o processo de geração textual, propõe-se que elementos nelas mapeados componham o contexto de análise que servirá de entrada para o serviço de geração textual (por exemplo, o perfil do usuário, as características dos recursos visualizados nos painéis analíticos e os filtros selecionados nesses painéis, entre outros).

De forma geral, dependendo das características de cada projeto (como seus requisitos, recursos e limitações), o conhecimento

⁷ Essa ontologia pode ou não existir e pode ter o mais variado nível de complexidade, dependendo das necessidades de cada implementação e de cada projeto.

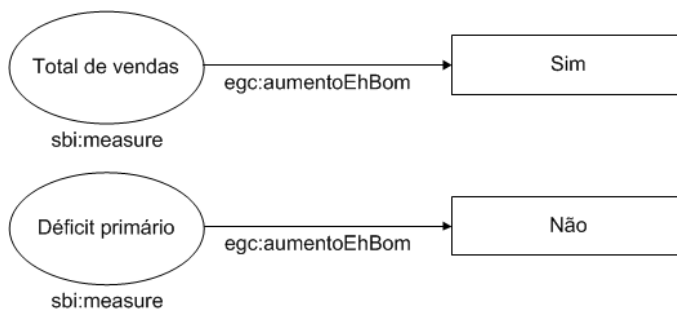
necessário para geração textual pode envolver tanto questões relacionadas ao domínio da aplicação e aos elementos de BI, quanto às próprias questões linguísticas. A seguir, serão apresentados alguns exemplos e considerações envolvendo a representação de conhecimento envolvendo esses dois tipos de questões.

3.1.1 Representação de conhecimento relacionado ao domínio da aplicação e aos elementos de BI

Em relação à representação de conhecimento do domínio da aplicação, um exemplo de informação potencialmente importante envolvendo as unidades de informação manipuladas por um sistema de BI, que pode ser utilizado futuramente em regras criadas para a geração textual, é o fato de uma determinada medida estar maior. Essa condição indica um aspecto positivo ou negativo do domínio. Assim, para uma medida como ‘total de vendas’, um valor mais alto representa um aspecto positivo, já para uma medida como ‘déficit primário’, o fato um valor mais alto aponta para um aspecto negativo.

Na ontologia, o conhecimento citado no exemplo pode ser representado através de uma propriedade booleana na classe que representa uma medida (elemento *Measure* na Ontologia de Aplicação da plataforma SBI), conforme é ilustrado na Figura 10.

Figura 10 – Exemplo da utilização de uma propriedade booleana (egc:aumentoEhBom) para qualificar instâncias de medidas (sbi:measure) em uma base de conhecimento



Fonte: elaborado pelo autor

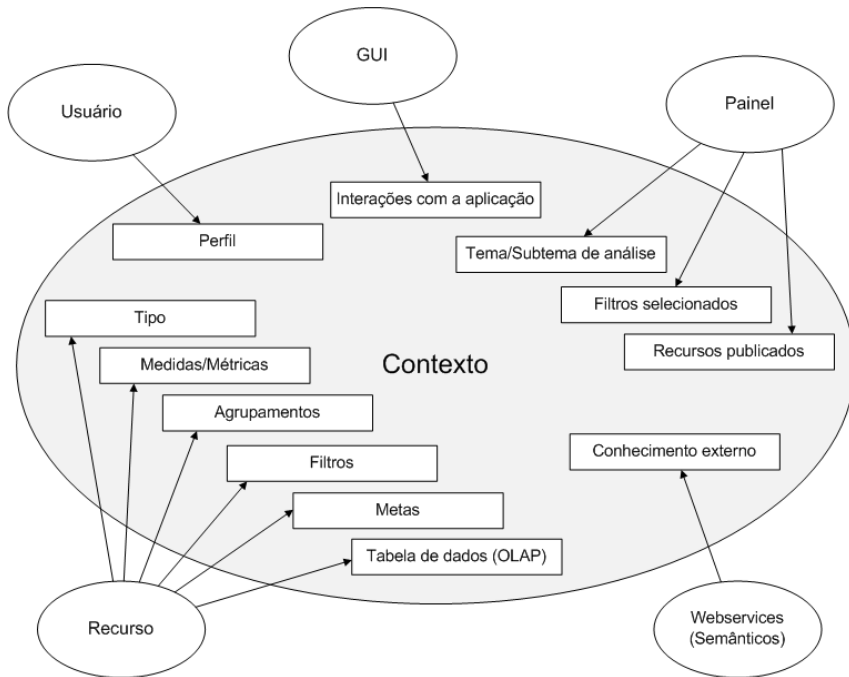
Dessa forma, no serviço de geração textual do módulo de planejamento do documento, em um cenário em que é verificado que a medida ‘total de vendas’ teve um aumento (observando dados de uma série temporal), poderia existir uma regra que, por exemplo, estrutura

uma frase para dizer que houve um aumento nas vendas e marca essa frase com uma anotação (metainformação) representando seu caráter positivo (baseando-se no valor do atributo ‘egc:aumentoEhBom’ mapeado na ontologia). Assim, essa marcação pode ser interpretada futuramente pelo módulo de realização textual, que utilizará a metainformação para incluir outros tipos de marcações que afetarão a forma de visualização do texto final na interface do usuário, como, por exemplo, *tags* HTML alterando a cor do texto: o verde para uma frase positiva e o vermelho para uma frase negativa. Outra implementação de regra que poderia ser aplicada no módulo de planejamento do documento, seria desprezar todas as sentenças positivas e separar apenas as negativas para serem apresentadas e realçadas. Tudo depende dos objetivos de comunicação estabelecidos em um determinado cenário de geração textual.

Esses foram apenas exemplos do que poderia ser explorado utilizando o conhecimento de domínio presente na ontologia, mais especificamente, o conhecimento relacionado com as unidades de informação manipuladas pelo sistema de *Business Intelligence* (mapeados na Ontologia de Aplicação da Plataforma SBI).

Em relação à representação do conhecimento relativo ao domínio da aplicação, aos elementos de BI, e às unidades de informação utilizadas pelo sistema, pode-se afirmar que o conhecimento do contexto sobre o qual o texto será gerado é potencialmente importante. Dessa forma, pode ser explorado no processo de geração textual, servindo de entrada para o módulo de planejamento do documento. Em uma aplicação típica de BI, pode-se utilizar informação contextual em diferentes frentes, conforme pode ser visualizado na Figura 11.

Figura 11 – Exemplos de elementos de contexto envolvidos em um cenário de utilização de uma ferramenta de BI



Fonte: elaborado pelo autor

Assim, no módulo de planejamento do documento do serviço de geração textual, propõe-se que, com vistas à personalização e à maior especificidade das sentenças geradas, possa ser feito uso das informações desses diferentes elementos, conforme listado na Tabela 7.

Tabela 7 – Elementos de contexto típicos em uma aplicação de BI e algumas informações que eles podem agregar como entrada às regras para geração textual

Elemento de contexto	Informação envolvida	Observações
Usuário do sistema (tomador de decisão)	Perfil	Por exemplo: sexo, idade, localização, cargo etc.
Interface de aplicação (GUI)	Interações com a aplicação	Demonstra indicativos de interesse do usuário.
Painel	Tema/Subtema de	O painel está vinculado a um

<i>(Dashboard)</i>	análise	tema ou a um subtema de análise (como, por exemplo, “Financeiro”, “Marketing”, “Perfil de compradores da loja” etc.).
	Filtros selecionados	Valores selecionados para filtros gerais que se aplicam a todos os elementos de um painel.
	Recursos publicados	Quais recursos estão sendo visualizados no momento?
Recurso	Tipo	Tipo do recurso (como, por exemplo, gráfico, indicador, relatório etc.).
	Medidas/Métricas	Medidas e métricas presentes na consulta associada ao recurso.
	Agrupamentos	Agrupamentos presentes na consulta associada ao recurso.
	Filtros	Filtros presentes na consulta associada ao recurso. Os valores desses filtros podem ser predefinidos pelo criador do recurso ou podem ser dinâmicos, tendo seus valores definidos em tempo de execução (seja pela aplicação ou manualmente pelo próprio usuário).
	Metas	Metas associadas aos dados presentes no recurso.
	Tabela de dados (OLAP)	Tabela (OLAP) com os dados, resultantes da consulta associada ao recurso, atualizados.
Ambiente externo <i>(Webservices semânticos)</i>	Conhecimento externo	Informações advindas de fora da aplicação, possivelmente por <i>Webservices</i> semânticos (como, por exemplo, a taxa

		do dólar, a previsão do tempo, os produtos mais procurados pelos consumidores em determinados sites etc.).
--	--	--

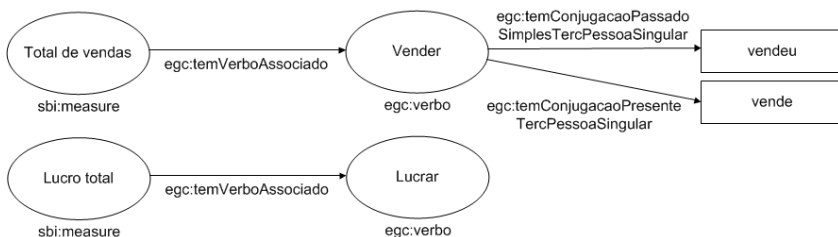
Fonte: elaborado pelo autor

Conforme já foi citado, dependendo das características de cada projeto, o conhecimento necessário para geração textual pode envolver, além das questões relacionadas ao domínio de aplicação e aos elementos de BI, as próprias questões linguísticas.

3.1.2 Representação de conhecimento relacionado a questões linguísticas

No que diz respeito à representação de conhecimento relativo a questões linguísticas, uma possibilidade, na geração textual não baseada em *templates* textuais, é definir os verbos que estão mais associados a cada medida do *Data Warehouse*. Dessa forma, para uma medida como ‘total de vendas’, o verbo que poderia ser utilizado em uma sentença é vender. Além disso, as diferentes possibilidades de conjugação desse verbo podem estar na própria ontologia (Ontologia de Aplicação Estendida) ou – externamente a ela – em um *lexicon* (ver seção 2.2.1). A Figura 12 ilustra o exemplo citado.

Figura 12 – Exemplo da utilização de propriedades e conceitos linguísticos em anotações semânticas sobre instâncias de medidas (sbi:measure)



Fonte: elaborado pelo autor

Diante do que foi exposto, é importante deixar claro que, da mesma forma como foram apresentados aspectos passíveis de serem explorados na representação de conhecimento relacionado ao domínio e aos elementos de BI (na seção 3.1.1), esta seção expõe apenas um exemplo envolvendo a representação de conhecimento relacionado a questões linguísticas. O universo de possibilidades, levando-se em conta

as necessidades, características e limitações de cada projeto, é muito grande e impossível de ser generalizado.

Além desse tipo de representação direta de conhecimento, conforme os exemplos das seções 3.1.1 e 3.1.2, a utilização de ontologias propicia a utilização de outro poderoso recurso, o qual pode ampliar as potencialidades de uma ferramenta de BI e, conseqüentemente, de um componente para geração de textos: as inferências semânticas.

3.1.3 Utilização de conhecimento inferido

A utilização das ontologias para armazenamento de conhecimento de forma direta não é a única aplicação disponível, também é possível fazer uso de outra de suas potencialidades: as inferências semânticas – que são consideradas, muitas vezes, a maior potencialidade das ontologias.

Um exemplo de tipo inferência possível de ser utilizada em um processo de geração textual é a utilização das relações de hierarquia entre as classes da ontologia de domínio. Nesse caso, seria possível, por exemplo, ter uma sentença genérica que apresenta o nome do funcionário que realizou um maior número de horas extras na empresa – levando em consideração que, na ontologia de domínio, os funcionários da empresa são instâncias de subclasses da classe “Funcionário” (como “Recepcionista”, “Administrador”, “Programador” e “Gerente de projeto”). Dessa forma, independentemente do cargo que a pessoa ocupa, ela será reconhecida por mecanismos de inferência aplicados sobre a ontologia, como um funcionário da empresa, tendo suas horas extras verificadas no processamento que irá selecionar o funcionário que mais trabalhou.

Ainda em relação à representação de conhecimento relativo ao domínio da aplicação, tem-se a possibilidade de representar lógicas para definir, por exemplo, conceitos relativos, como ‘o melhor’ e ‘o pior’, e utilizá-los em sentenças textuais. Por exemplo, o conceito “o melhor vendedor” poderia ser definido como relativo à entidade “vendedor”, que possui a medida “total de vendas” com melhor média de valor no ano corrente. Dessa forma, seria possível gerar uma frase informando quem é o melhor vendedor da organização.

3.1.4 Considerações sobre a utilização de ontologias no método para geração textual

Diante dos exemplos expostos na seção 3.2.1, pode-se dizer que o universo de possibilidades disponibilizado pela utilização e pela exploração das potencialidades das ontologias é grande. Dessa forma, quanto melhores forem as bases de conhecimento utilizadas e os serviços que fazem uso dessas bases, mais eficaz pode se tornar o processo de geração de textos relevantes que possam apoiar, efetivamente, a disseminação de conhecimento e os processos decisórios nas organizações.

Dentro do método proposto, já foram apresentados os aspectos relacionados à utilização das ontologias da Plataforma SBI, bem como de uma possível Ontologia de Aplicação Estendida para suportar demandas específicas do processo de geração textual. A seguir, será apresentado um detalhamento dos módulos envolvidos no serviço gerador de textos: (1) planejamento do documento, (2) microplanejamento e (3) realização textual. Além disso, são expostas as possibilidades de integração desses módulos com os serviços e recursos disponibilizados pela Plataforma SBI, além de outros recursos externos, como *webservices*.

3.2 MÓDULO DE PLANEJAMENTO DO DOCUMENTO

No módulo de planejamento do documento, etapa de determinação do conteúdo, é realizada a verificação do conteúdo (ou do conhecimento) que pode ser relevante para o tomador de decisão e que deverá ser apresentado nas ferramentas analíticas. Além disso, na etapa de estruturação do documento, é definida a forma como esse conteúdo será exposto.

Durante o planejamento do documento, questões como determinação de relevância, seleção de conteúdo e, até mesmo, criação de novos conhecimentos⁸ – derivados de informações e de outros conhecimentos já disponíveis na organização – podem estar envolvidas. Dessa forma, é importante salientar que essa etapa é potencialmente complexa de ser desenvolvida, e que alguns dos desafios da área de

⁸ O processo de criação de conhecimento pode ocorrer por meio da organização de conhecimento anterior em novas formas, da combinação de informações relevantes, ou mesmo por meio de *insights* acerca da aplicação de conhecimento existente em novos contextos (CALHOUN; STARBUCK, 2005).

Engenharia e Gestão do Conhecimento (SCHREIBER et al., 2000) podem estar presentes. Particularmente, no módulo de planejamento do documento, por se tratar do módulo em que provavelmente estaria a maior inteligência do sistema, é onde se encontrariam a maior parte das tarefas intensivas de conhecimento (SCHREIBER et al., 2000) de um sistema gerador de textos.

Diante disso, pode-se dizer que não existe uma abordagem ideal e totalmente genérica para implementação do módulo responsável pela etapa de planejamento do documento, mas vislumbra-se que diferentes técnicas possam ser utilizadas, dependendo das necessidades, dos recursos e das limitações de cada projeto. Segue a lista de algumas das limitações supostas.

- Regras específicas criadas previamente por especialistas e armazenadas em bases de conhecimento.
- Heurísticas codificadas para tratar, de forma genérica, casos que envolvam alguns aspectos gerais, independentemente do domínio da aplicação.
- Técnicas de análise de dados e de inteligência artificial para descoberta ou criação de novos conhecimentos.
- Mecanismos de utilização de *feedbacks* de usuários sobre a ‘corretude’ e relevância do conteúdo gerado, retroalimentando e adaptando o sistema.

Independentemente das técnicas utilizadas, é importante ressaltar que sistemas de geração de linguagem natural, assim como a maioria dos sistemas que utilizam inteligência artificial, precisam de uma quantidade substancial de conhecimento (REITER; SRIPADA, 2003). A utilização de uma base de conhecimento adequada⁹ é, portanto, um fator crítico nessa etapa do processo de geração de textos.

Nessa linha, um aspecto técnico central da Gestão do Conhecimento é a utilização de uma memória organizacional (WALSH e UNGSON, 1991) como um meio para conservação, distribuição e reuso de conhecimento (STUDER; BENJAMINS; FENSEL, 1998). Assim, o desenvolvimento de um método para geração de sumários analíticos que possibilite a utilização de um sistema de memória organizacional (OLIVERA, 2000) visando preservar e reutilizar

⁹ Os métodos e as tecnologias utilizadas para a criação e para a manutenção de uma base de conhecimento não são o foco deste trabalho.

conhecimento do negócio poderia ser extremamente estratégico para potencializar a relevância dos textos gerados.

Assim, utilizando uma base de conhecimento que apoie o processo de geração textual, seção 3.1, pela utilização das ontologias, o módulo de planejamento do documento deverá possuir a inteligência de, com base em um dado contexto informacional dinâmico, definir o que será gerado (conteúdo) e de que forma isso será apresentado (estrutura). Propõe-se que essas questões estejam codificadas em regras de produção textual – independentemente da complexidade que possuírem as regras e de como elas serão representadas.

3.2.1 A criação de regras para geração textual no contexto de BI

Conforme já foi mencionado, diferentes técnicas podem ser utilizadas no módulo de planejamento do documento para decidir quais informações devem ser comunicadas no documento em que será gerado, e de que forma elas devem estar apresentadas.

Dessa forma, regras de diferentes complexidades e representadas de diferentes formas podem ser criadas e utilizadas para determinar o conteúdo que será apresentado ao tomador de decisão, e de que forma esse conteúdo será apresentado. Para isso, serão utilizados os conhecimentos representados na base de conhecimento da organização (sejam relacionados a questões linguísticas, aos elementos de BI ou ao domínio da aplicação, conforme apresentado na seção 3.1) e as informações dinâmicas advindas do contexto do processo decisório.

Essas regras, também chamadas de heurísticas, podem ser utilizadas para tratar de forma especializada cenários peculiares, como também tratar cenários envolvendo aspectos gerais ou relações genéricas, independentemente do domínio da aplicação. As heurísticas com alto grau de especificidade provavelmente serão produzidas por especialistas da organização ou de um domínio específico. Como exemplo, têm-se as regras apresentadas no Quadro 1 (explicitadas em linguagem natural, por questões de simplificação).

Quadro 1 – Exemplo de regras com alto grau de especificidade

- | |
|--|
| <ol style="list-style-type: none">a. Quando o usuário que está utilizando o sistema for o presidente da empresa, e se as ações da empresa caíram mais de X% no mês corrente, apresente a sentença Y no painel de situação W.b. No painel Z, mostre sempre a sentença X. Para essa sentença X (que utiliza como provedor de dados uma consulta que utiliza a |
|--|

dimensão “Pessoa”), quando a nacionalidade da instância de pessoa for “Brasileira”, apresentar o CPF dessa pessoa na sua identificação na sentença; caso contrário, apresentar o número do seu passaporte.

Fonte: elaborado pelo autor

Já as regras com um alto grau de generalidade podem ser produzidas independentemente do domínio da aplicação, podendo ser reaproveitadas em diferentes situações de forma mais automatizada e, portanto, com maior reusabilidade. Essas regras podem ser desenvolvidas utilizando aspectos gerais, conforme os exemplos apresentados no Quadro 2.

Quadro 2 – Exemplo de regras com alto grau de generalidade

- Quando uma consulta envolver dados dispostos em séries temporais, deverá ser gerada uma frase fazendo a predição do próximo valor futuro.
- Quando uma consulta envolver elementos que possuam relações hierárquicas (todo-parte) com outros elementos, deverá ser gerada uma frase apresentando o total resultante dos dados agregados no elemento-pai e outra frase com os dados segregados de alguns dos elementos-filho (segundo outro critério, como os elementos de destaque ou o contexto).

Fonte: elaborado pelo autor

Apesar das regras com maior generalidade possuírem um alto grau de reusabilidade em diferentes domínios da aplicação, a utilização de regras específicas, criadas por especialistas do domínio, mostra-se cada vez mais importante. Segundo Davenport, Marchand e Dickson (2004), devido à avalanche de informações a que estão submetidas as pessoas de negócio, alguém dentro da empresa deve determinar quais informações são importantes internamente, pois isso depende do ambiente de negócios e da estratégia. Assim, alguém que esteja familiarizado com os mercados, com as ofertas e com a orientação estratégica da empresa deve ser o responsável pela determinação de qual informação privilegiar.

Outro ponto a se observar com a utilização de regras para geração textual, além das características de generalidade ou especificidade mencionadas anteriormente, é a característica de possibilidade de conjunção entre diferentes regras. Nessa linha, em vez de permitir o encadeamento de diferentes regras, os serviços de geração textual

poderiam utilizar uma abordagem mais simplificada, utilizando as regras de forma atômica. Assim, cada regra seria processada em relação ao contexto apresentado, gerando ou não sentenças associadas como resultado. Dessa forma, as regras atuariam como gatilhos, verificando condições para determinadas sentenças serem apresentadas nos painéis analíticos. As regras apresentadas nos exemplos até o momento seguem essa abordagem. A outra abordagem, um pouco mais complexa, possibilitaria uma conjunção de diferentes regras. Assim, seria possível fazer o encadeamento delas por meio de estruturas semelhantes a uma ‘árvore de decisão’. No Quadro 3, é apresentado um exemplo ilustrativo dessa abordagem.

Quadro 3 – Exemplo de regras encadeadas

- | |
|--|
| <ol style="list-style-type: none">a. Quando o usuário que está utilizando o sistema for o presidente da empresa, apresente a sentença “Bom dia, presidente!”, e vá para o item (d); caso contrário, vá para o item (b).b. Apresente a sentença “Bom dia, [fulano]!” e vá para o item (c) – a variável [fulano] representa o nome do usuário que está utilizando o sistema.c. Se as vendas do departamento do usuário que está utilizando o sistema aumentaram, apresente a sentença X; caso contrário, apresente a sentença Y e vá para o item (e).d. Se as ações da empresa caíram mais de $X\%$ no mês corrente, apresente a sentença W; se caíram menos de $X\%$, apresente a Z; e, se subiram, apresente a sentença J.e. Se o usuário foi um dos maiores responsáveis pelo aumento nas vendas, apresente a sentença “Parabéns, continue assim!”. |
|--|

Fonte: elaborado pelo autor

Seguindo essa abordagem, vislumbra-se que é possível que os especialistas do domínio configurem, de forma sistemática, textos analíticos representando sequências lógicas de como eles fariam as análises dos cenários informacionais apresentados. Assim, cada questão levantada em um processo de análise poderia ser mapeada em uma regra, e as possíveis conclusões ou comentários envolvendo essas questões, em sentenças. Essas regras e sentenças podem englobar funções de grande complexidade, fazendo cálculos matemáticos e estatísticos com os dados disponíveis e buscando, até mesmo, novas informações complementares, da mesma forma que um especialista

humano poderia fazer. O documento final gerado seria equivalente a uma verdadeira leitura do cenário informacional apresentado pelas ferramentas analíticas (verbalização da análise).

Vale ressaltar que, no contexto de utilização de regras criadas por especialistas, mais especificamente no caso em que ocorre a utilização de regras conjugadas, é importante disponibilizar um meio para que os especialistas do domínio criem e configurem as regras sem possuir conhecimentos técnicos a respeito de formalismos para representá-las, conforme apresentado na seção 3.5.

Por fim, é importante deixar claro que as propostas expostas nesta seção devem ser implementadas de acordo com o contexto de cada cenário de implantação dos projetos de *Business Intelligence*, conforme apregoado por Sell (2006) na sua proposta de aplicação de ontologias para estender as possibilidades de extração de conhecimento e de apoio ao processo decisório. Assim, em cada implementação baseada no método proposto neste trabalho, devem ser tomadas decisões relacionadas ao formalismo de representação das regras, à sua complexidade, aos recursos que elas poderão manipular (elementos da base de conhecimento, consultas sobre o DW, consultas a informações externas, entre outros) e à interpretação das regras em tempo de execução. Enfim, a instanciação desse módulo no espaço de cada projeto deverá ser orientada pelas suas necessidades, seus recursos e suas limitações.

3.2.2 Considerações sobre o módulo de planejamento do documento

Conforme já apresentado, pode-se dizer que não existe uma abordagem ideal e totalmente genérica para a implementação do módulo responsável pela etapa Planejamento do documento. Basicamente, dependendo dos requisitos, dos recursos disponíveis e das limitações de cada projeto, esse módulo poderá ser muito simples ou muito complexo.

Se aumentarmos um pouco a complexidade, vislumbra-se que esse módulo pode atuar em processos de descoberta de conhecimento, utilizando algoritmos de *Data Mining* para analisar uma massa de dados e buscar padrões possivelmente interessantes para serem apresentados nas ferramentas analíticas. Conforme também já mencionado, segundo Kacprzyk e Zadrozny (2009), sumários linguísticos gerados a partir de dados podem ser uma poderosa ferramenta para obter *insights* sobre que relações existem entre esses dados que podem ser relevantes para uma atividade particular do negócio e para a tomada de decisão envolvida.

De forma geral, a entrada do módulo de planejamento do documento será um objeto que representa – possivelmente por elementos de ontologias da Plataforma SBI – um contexto informacional envolvido em um ambiente de tomada de decisão. A saída, baseada em objetivos de comunicação refletidos nas regras codificadas, será um plano do documento a ser gerado (que também pode ser representado das mais variadas formas, dependendo das características de cada projeto). A saída do módulo de planejamento do documento serve de entrada para o módulo de microplanejamento, que será abordado a seguir.

3.3 MÓDULO DE MICROPLANEJAMENTO

A etapa de microplanejamento envolve, como ora abordado, questões como a agregação de sentenças semelhantes, escolhas léxicas e geração de expressões para referenciar as entidades presentes nas sentenças. Dessa forma, diz-se que essa etapa pode ser mais ou menos complexa, dependendo das estratégias adotadas para a geração textual.

Considerando duas abordagens de implementação principais e divergentes, utilizando e não utilizando *templates* textuais (conforme foi exposto na seção 2.2.2), e levando em consideração a proposta de estruturação de um serviço para geração de sumários textuais dentro do contexto de uma plataforma de BI que utilize tecnologias semânticas, como a Plataforma SBI, faz-se, neste trabalho, algumas proposições sobre práticas que poderiam ser adotadas no desenvolvimento de um módulo de microplanejamento visando agregar melhorias no processo de geração de textos e potencializar a experiência final do usuário envolvido nos processos decisórios.

De forma geral, em um sistema que adota uma abordagem de implementação que não utiliza *templates* textuais, a etapa de microplanejamento pode se tornar essencial, pois nela estariam as atividades que definem – de fato – os termos e as expressões que farão parte das sentenças. Por exemplo, em um contexto de BI, o módulo de planejamento do documento pode definir para o plano do documento – através de algum formalismo estabelecido – que seja apresentado um documento contendo uma sentença que informe o valor de determinada medida (*sbi:measure*) como, por exemplo, “total de vendas”. Essa medida deve ser analisada sob uma dimensão temporal (*sbi:dimension*), como “mês”. Finalmente, é apresentado o valor da medida (valor no mês atual). Dessa forma, o módulo de microplanejamento teria que, baseado nesse plano do documento, definir os termos e expressões que seriam

utilizados na sentença, gerando como saída a especificação do documento, conforme ilustrado de forma simplificada no Quadro 4.

Quadro 4 – Exemplo de funcionamento de uma especificação de documento

- **Plano do documento:** apresentar uma sentença contendo [o valor da medida “total de vendas”] [na dimensão “mês” com o valor corrente (mês = X)].
- **Especificação do documento:** [verbo vender no passado (voz passiva)], [neste mês], [valor da medida “Total de vendas” no mês X].
- **Sentença final (após realização textual):** Foi vendido, neste mês, R\$435.520,00.

Fonte: elaborado pelo autor

Nessa etapa é possível proporcionar uma maior naturalidade e fluidez entre diferentes sentenças de um mesmo documento. Por exemplo, se o módulo de planejamento do documento – de forma isolada – estabeleceu que devem ser apresentadas duas sentenças, o módulo de microplanejamento poderia, por sua vez, alterar a segunda sentença de forma que a leitura das duas sentenças de forma sequencial ficasse mais natural. O exemplo apresentado no Quadro 5 (representado de forma didática e simplificada já em linguagem natural) demonstra essa situação, na qual a sentença 2 é alterada para a 2a via módulo de microplanejamento.

Quadro 5 – Exemplo de funcionamento do módulo de microplanejamento do documento

- **Sentença 1:** A equipe A vendeu, neste mês, 23% dos produtos.
- **Sentença 2:** A equipe B vendeu, neste mês, 11% dos produtos.
- **Sentença 2a:** Já a equipe B vendeu, **no mesmo mês**, 11% dos produtos.

Fonte: elaborado pelo autor

Vale ressaltar que, em uma abordagem baseada em *templates* textuais, um mesmo modelo já poderia prever o encadeamento das duas frases e proporcionar a fluidez entre elas, sem que fosse necessário esse tipo de processamento e essas adequações. No exemplo apresentado, por exemplo, o *template* textual já poderia ter uma estrutura contendo, de forma fixa, os termos “Já” e “no mesmo mês” (esse *template* seria algo

como “A equipe X vendeu, neste mês, x% dos produtos. Já a equipe Y vendeu, no mesmo mês, y% dos produtos.”, sendo que somente as variáveis X, x, Y e y seriam calculadas e substituídas dinamicamente durante o processo de geração textual).

Com base nesse exemplo, pode-se dizer que, em um sistema que adota uma abordagem de implementação baseada em *templates* textuais, a etapa de microplanejamento pode ser praticamente suprimida, visto que a maioria dos aspectos abordados por ela podem ser contemplados e estabelecidos nos próprios *templates* (nos textos fixos). No entanto, mesmo para essa abordagem, dentro de um contexto envolvendo uma plataforma de BI semântico, como a Plataforma SBI, propõe-se que alguns aspectos ainda sejam explorados, visando – principalmente – uma maior adaptabilidade e personalização dos documentos gerados, deixando os textos mais amigáveis para o usuário final.

Assim, principalmente no que diz respeito às atividades de escolha léxica e de geração de expressões para referência, pode-se fazer uso do contexto do processo decisório e da base de conhecimento da organização (representada pelas ontologias) para adaptar dinamicamente os documentos pelo serviço de geração textual. Por exemplo, conforme apresentado no Quadro 6, em uma situação em que será gerada uma sentença apresentando os resultados obtidos pela equipe A (conforme a sentença 1); se for verificado pelo contexto de entrada e pela base de conhecimento que o usuário que está utilizando o sistema participa dessa equipe, a sentença pode referenciá-lo de maneira diferenciada, tornando-se mais pessoal e amigável (conforme a sentença 1a).

Quadro 6 – Exemplo de das atividades de escolha léxica e geração de expressões para referência

- | |
|---|
| <ul style="list-style-type: none"> • Sentença 1: <u>A equipe A</u> vendeu, no mês de dezembro, 23% dos produtos. • Sentença 1a: <u>Sua equipe</u> vendeu, no mês de dezembro, 23% dos produtos. |
|---|

Fonte: elaborado pelo autor

Para as abordagens não baseadas em *templates* textuais, que são bem mais flexíveis, esse aspecto parece mais factível de ser implementado. Mesmo assim, é possível explorar essas questões em abordagens baseadas em *templates* textuais com a utilização de alguns artifícios. O que se propõe para esses casos, é que alguns pontos de flexibilidade sejam incluídos nos modelos para que assim alguns termos ou expressões de referência específicos possam ser injetados

dinamicamente nesses pontos, conforme o contexto da geração textual. Os pontos de flexibilidade poderiam ser representados por funções ou por marcações especiais presentes nos *templates*, as quais seriam interpretadas pelo módulo de microplanejamento para análise e decisões dos termos a serem utilizados, injetando-os no *template* final, que é utilizado pelo módulo de realização textual.

Vale ressaltar que, para esse tipo de codificação, o módulo de microplanejamento também tem que ter acesso às informações do contexto e às ontologias que formam a base de conhecimento da organização, por meio das ontologias na Plataforma SBI pelo serviço OntologyManager. Além disso, também cabe salientar que apesar do conhecimento codificado necessário para o funcionamento do módulo de microplanejamento ser menor do que do módulo de determinação do conteúdo (no qual estaria a maior inteligência do sistema), ele possui certo grau de complexidade e de dependência do domínio da aplicação, principalmente nas abordagens não baseadas em *templates* textuais.

Assim, componentes genéricos desenvolvidos para apoio a processos de geração textual – como é o caso da biblioteca *SimpleNLG* (SIMPLENLG, 2012) – costumam não abranger o esse módulo de microplanejamento, ficando responsáveis apenas pela etapa de realização textual, que é mais automatizável. Dessa forma, os aspectos tratados pelo módulo de microplanejamento acabam tendo que ser desenvolvidos – quando o são – dentro dos próprios componentes ou dos projetos específicos.

Conforme já mencionado, a saída do módulo de microplanejamento – que pode ser chamada de especificação do texto serve como entrada para o realizador textual, que será apresentado a seguir.

3.4 MÓDULO DE REALIZAÇÃO TEXTUAL

Na etapa de realização textual, o texto final que será apresentado ao tomador de decisão é efetivamente gerado. Nesse momento, todas as definições de conteúdo e de estrutura do texto já foram feitas pelo módulo de planejamento do documento, e algumas adaptações na forma como esse conteúdo será apresentado já foram definidas pelo módulo de microplanejamento.

Dessa forma, seguindo as definições feitas pelos outros módulos e formalizadas na especificação do documento, resta fazer a realização ou execução da especificação, gerando o texto propriamente dito em linguagem natural para ser apresentado na interface do usuário. Além

disso, conforme já comentado, existe também a possibilidade de que a saída desse módulo seja um texto contendo marcações (como, por exemplo, *tags* HTML) que podem ser interpretadas e utilizadas na interface com o usuário, alterando, por exemplo, a apresentação visual do texto.

Existem duas abordagens principais e divergentes para geração de linguagem natural (baseada e não baseada em *templates* textuais). Em última instância, o tipo de abordagem utilizado reflete diretamente na implementação do módulo de realização textual. A seguir, são feitas algumas considerações a respeito das implementações envolvendo os dois tipos de abordagens e, na sequência, considerações gerais sobre a implementação desse módulo, independentemente do tipo de abordagem adotado.

3.4.1 Implementações baseadas em templates textuais

Nos sistemas que utilizam a abordagem baseada em *templates* textuais, a tarefa do módulo de realização textual costuma ser relativamente simples: substituir dinamicamente variáveis presentes em um *template* textual por outros segmentos de textos, gerando assim os versão final que será apresentada.

Apesar de uma notável inflexibilidade existente nos textos produzidos por essa abordagem, as variáveis presentes nos *templates* proporcionam pontos de flexibilidade, que podem ser bastante explorados, conforme exemplificado na seção 3.3, para conferir uma maior versatilidade e reusabilidade desses *templates*. Na prática, cada uma dessas variáveis é representada por uma função ou marcação, que é calculada dinamicamente, em tempo de execução da realização de cada frase. O resultado pode ser injetado no local ocupado pela variável no *template*. Dessa forma, seria possível criar *templates* compostos por funções complexas, deixando-os – em última instância – completamente flexíveis. Como exemplo da combinação de um *template* textual que possua um ponto de flexibilidade representado por uma função, pode-se citar um *template* que gera sentenças conforme os modelos apresentados no Quadro 7.

Quadro 7 – Exemplo de funções representando pontos de flexibilidade em *templates* textuais

- **Sentença 1:** De acordo com os números dos últimos 5 anos, a tendência é que as vendas umentem em 16% no próximo ano.
- **Sentença 2:** De acordo com os números dos últimos 5 anos, a

tendência é que as vendas sejam reduzidas em 5% no próximo ano.

- **Sentença 3:** De acordo com os números dos últimos 5 anos, a tendência é que as vendas se mantenham estáveis no próximo ano.

Fonte: elaborado pelo autor

Assim, utilizando um *template* com dois trechos de texto fixo “De acordo com os números dos últimos 5 anos, a tendência é que as vendas” e “no próximo ano”, intercalados com uma função que faz a predição matemática de um valor se baseando em uma série temporal de valores e que calcula o percentual de diferença entre o último valor da série e o valor previsto para o futuro, pode-se gerar os três modelos de frases.

Nesse caso, a função deveria ser configurada pelo especialista responsável por criar o *template* por meio da indicação da consulta que gera a massa de dados utilizada no cálculo e dos elementos de BI presentes na consulta, os quais serão envolvidos no cálculo (medidas e dimensões mapeadas na ontologia da Plataforma SBI). Além disso, os termos a serem utilizados em caso de aumento, diminuição ou estabilidade do percentual calculado (“aumentem em”, “sejam reduzidas em” e “se mantenham estáveis”, no exemplo citado) podem ser parametrizáveis. Em outros casos, algumas informações podem vir das próprias ontologias.

Esse foi apenas um exemplo apresentando a possibilidade de construção de frases adaptáveis baseadas em *templates* textuais que utilizam funções pré-definidas e configurações feitas pelo especialista do domínio, referenciando – quando necessário – elementos das ontologias da organização. A seguir, serão feitas considerações relacionadas à utilização da abordagem não baseada em *templates* textuais.

3.4.2 Implementações não baseadas em templates textuais

Com relação aos sistemas que utilizam uma abordagem não baseada em *templates* textuais, a tarefa do módulo de realização textual costuma ser um pouco mais complexa, pois a maioria das definições linguísticas, envolvendo diferentes elementos léxicos e sintáticos, tem que ser estabelecida *a priori*. Nessa linha, o módulo de microplanejamento, apresentado na seção 3.3, poderia fazer todo esse papel, mas a complexidade envolvida no processo seria grande (principalmente na geração de frases naturais, ou seja, que pareçam ser criadas diretamente por seres humanos).

Uma proposta alternativa para atender a esse tipo de questão é que, para implementações não baseadas em *templates* textuais, seja feita uma codificação que propicie a criação de *templates* estruturais pelos especialistas da organização. Esses *templates* estruturais conteriam um esqueleto com algumas definições léxicas e sintáticas necessárias para a estruturação das sentenças. Além disso, eles também poderiam receber elementos representando funções para injetar informações calculadas dinamicamente, relacionadas ao contexto do processo decisório, conforme exemplificado na abordagem baseada em *templates* textuais, na seção 3.4.1.

A seguir, é apresentada (de forma didática, sem formalismos) uma simulação ilustrativa do funcionamento de um cenário completo envolvendo um serviço de geração textual que utiliza a abordagem não baseada em *templates* textuais. Dessa forma, o **Quadro 8** apresenta um exemplo de regra codificada e de um *template* estrutural utilizado por ela.

Quadro 8 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em *templates* textuais – exemplo de uma regra codificada e de um *template* estrutural

- **Regra codificada no módulo de planejamento do documento:** quando a consulta que gera os dados associados a um recurso visualizado contiver uma medida e uma dimensão temporal, utilizar o *template* W, passando como parâmetro essa medida e o valor mais atual dessa dimensão.
- **Template W:** [estrutura verbal (definida na ontologia de aplicação estendida) associada a uma medida], [estrutura (definida na ontologia de aplicação estendida) indicativa do valor de uma dimensão], [descritivo (definida na ontologia de aplicação estendida) associado à totalização de uma medida] [valor de uma medida sumarizado pelo valor de uma dimensão].

Fonte: elaborado pelo autor

O **Quadro 9**, por sua vez, apresenta um contexto hipotético da simulação ilustrativa.

Quadro 9 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em *templates* textuais – exemplo de um contexto hipotético

- **Contexto:** no painel analítico que está sendo visualizado pelo usuário, existe um gráfico mostrando o total de vendas realizado

por mês.

Fonte: elaborado pelo autor

O módulo de planejamento do documento identifica pela ontologia que “mês” é uma dimensão temporal, e que o contexto se enquadra em uma das regras codificadas (apresentadas no Quadro 8). Dessa forma, ele gera o plano do documento conforme especificado pela regra e parametriza o *template* estrutural referenciado por ela, conforme apresentado no Quadro 10, a seguir.

Quadro 10 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em *templates* textuais – exemplo de um plano de documento

- **Plano do documento:** apresentar uma sentença baseada no *template* W, utilizando como parâmetros a medida “total de vendas” e o valor mais atual da dimensão “mês”.
- **Template W parametrizado:** [estrutura verbal associada à medida “total de vendas”], [estrutura indicativa da dimensão “mês” com valor igual a X], [descritivo associado à totalização da medida “total de vendas”] [valor da medida “total de vendas” sumarizada pela dimensão “mês” com valor igual a X].

Fonte: elaborado pelo autor

Na sequência, o módulo de microplanejamento, utilizando o plano do documento e o *template* associado, faz alguns ajustes no modelo e gera a especificação do documento, conforme apresentado no Quadro 11.

Quadro 11 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em *templates* textuais – exemplo de uma especificação de documento

- **Especificação do documento:** o próprio *template* W com alguns ajustes feitos pelo módulo de microplanejamento.
- **Template W especificado:** [verbo vender no passado (voz passiva)], [neste mês], [descritivo associado à totalização da medida “total de vendas”] [valor da medida “total de vendas” sumarizada pela dimensão “mês” com valor igual a X].

Fonte: elaborado pelo autor

Por fim, o módulo de realização textual, utilizando a especificação do documento gerada, cria o documento final: um texto em linguagem natural, conforme apresentado no Quadro 12.

Quadro 12 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em *templates* textuais – exemplo de um documento gerado

- **Sentença final:** Foi vendido, neste mês, o valor de R\$435.520,00.

Fonte: elaborado pelo autor

É importante ressaltar que o *template* estrutural utilizado nessa simulação ilustrativa poderia ser usado com outras medidas e dimensões, gerando sentenças como: “Foi gerada, neste ano, uma receita de R\$210.000,00.” e “Teve-se como índice, no ano de 2012, o valor de 1,75.”.

Vislumbra-se, portanto, que essa abordagem, no que diz respeito à potencialidade de reaproveitamento de sentenças genéricas em diferentes cenários, mostra-se até mais promissora que uma abordagem baseada unicamente em *templates* textuais (com textos fixos). Pois, no caso de uma abordagem baseada em *templates* textuais, possivelmente o modelo do exemplo já conteria uma estrutura semelhante à apresentada no Quadro 13 impossibilitando a sua utilização com outras medidas e dimensões.

Quadro 13 – Simulação ilustrativa do funcionamento de uma abordagem não baseada em *templates* textuais – exemplo de um *template* textual que geraria a mesma sentença

- **Template textual:** Foi vendido, neste mês, o valor de [Função X].

Sendo que o trecho sublinhado representa um texto fixo, e a Função X representa, por exemplo, uma função na qual um conjunto de dados formado por uma medida e por uma dimensão retorna o valor da medida sumarizada pelo valor mais atual dessa dimensão.

Fonte: elaborado pelo autor

Por fim, também se vislumbra que uma abordagem híbrida – com textos fixos e estruturas textuais dinâmicas (além das funções relatadas anteriormente) – possa se mostrar promissora. A seguir, serão apresentadas algumas considerações gerais sobre a implementação do módulo de realização textual, independentemente do tipo de abordagem utilizada.

3.4.3 Considerações gerais sobre a implementação do módulo de realização textual

Diante de tudo o que foi exposto na seção 3.4, pode-se dizer que o universo de possibilidades na implementação do módulo de realização textual é grande. Assim, independentemente da abordagem e das técnicas escolhidas, é importante para esse módulo que, em cada implementação, seja estabelecido um formalismo por meio do qual os módulos anteriores possam comunicar ao módulo de realização de texto, de forma padronizada, o que será apresentado, e de que forma isso será feito. Vale ressaltar que o formalismo representado na especificação do texto deve ser estabelecido em cada implementação, porque ele pode variar muito – dependendo das características de cada projeto, da abordagem utilizada (baseada ou não baseada em *templates* textuais) e, possivelmente, das especificidades de componentes de *software* que venham a ser utilizados.

Nesse contexto, existem algumas bibliotecas de *software* disponíveis no mercado que podem auxiliar na etapa da realização textual propriamente dita. Essas bibliotecas trabalham de diferentes formas e possuem diferentes formatos de entrada, sendo que a saída é sempre um texto em linguagem natural. Algumas dessas bibliotecas utilizam abordagens que são mais simples, trabalhando com *templates* textuais e injetando informações de forma dinâmica em variáveis presentes nesses *templates*, como o *Apache Velocity Engine* (THE APACHE VELOCITY PROJECT, 2012)¹⁰. Outras atuam gerando textos de fato – algumas dessas manipulando gramáticas complexas e outras procurando simplificar um pouco esse processo, como é o caso da biblioteca *SimpleNLG* (SIMPLENLG, 2012).

Assim, dependendo do tipo de abordagem e da biblioteca de *software* utilizada, a especificação do documento – gerada pelo módulo de microplanejamento, que será a entrada do módulo de realização textual – pode ser completamente diferente. Dessa forma, sugere-se que, em cada implementação da estrutura do serviço de geração textual

¹⁰ O *Apache Velocity Engine* não é uma biblioteca que tem como finalidade a realização de textos, mas suas potencialidades permitem que ela também seja utilizada para essa função.

proposto no presente trabalho, já seja definido, *a priori*, o tipo de abordagem e até mesmo a biblioteca de apoio que será utilizada¹¹.

Além disso, para facilitar a criação e a configuração dos *templates* pelos especialistas de domínio, sem que eles tenham que conhecer questões técnicas de forma aprofundada e para que eles possam estabelecer o vínculo desses *templates* com as regras criadas no contexto dos módulos anteriores (de acordo com o que é sugerido na seção 3.2.1), propõe-se que seja estruturado um ambiente para apoiar a realização do processo, conforme é apresentado a seguir.

3.5 ESTRUTURAÇÃO DE AMBIENTES PARA CONFIGURAÇÃO DE REGRAS E DE TEMPLATES

Na abordagem proposta no presente trabalho para o módulo de planejamento de documento, na seção 3.2.1, sugeriu-se que regras de produção textual fossem criadas pelos especialistas de domínio. Com essas regras, no contexto do processo decisório (mapeado pela ferramenta analítica), e com as bases de conhecimento da organização como insumo, os sumários textuais seriam gerados dinamicamente, visando o apoio ao processo decisório. Além disso, foi sugerida a possibilidade de criação de árvores de decisão envolvendo regras conjugadas que, dependendo do resultado de análises e de testes lógicos, apresentam ou não determinadas sentenças.

Nesse contexto, embora não tenha sido proposto nenhum formalismo específico para a representação e para o armazenamento das regras criadas (deixando essa questão a critério de cada abordagem de implementação desenvolvida), é importante que seja estruturado e disponibilizado um meio para que os especialistas da organização criem e configurem as regras que serão utilizadas, sem que tenham que possuir conhecimentos técnicos a respeito dos formalismos utilizados.

A Figura 13 apresenta um exemplo de ambiente (*wireframe*) para configuração de regras de produção e para o encadeamento de sentenças textuais por meio de árvores de decisão criadas pelos especialistas de

¹¹ Outra possibilidade seria deixar a especificação do documento estruturada da forma mais genérica possível e, para cada biblioteca de *software* utilizada, criar também uma lógica para adequar esse formalismo ao formato de entrada esperado pela biblioteca (codificação de *drivers* ou de classes *wrapper*). De qualquer forma, ainda assim, seria complexo ter uma especificação muito genérica, pois os componentes trabalham de forma muito diferente.

domínio utilizando estruturas lógicas condicionais e *templates* de sentenças previamente criados e categorizados por uma equipe técnica.

Figura 13 – Exemplo de ambiente para configuração de regras de produção e de encadeamento de sentenças textuais

Analizys Configuration

Categoria de análise:

Tipo de análise:

Templates disponíveis:

- Maior número com menor número
- Menor número com maior número
- Específico com maior número
- Específico com menor número
- Específico com pai (somente se relação "todo-parto")
- Específico com indivíduo específico
- Específico com lista de indivíduos específicos

Brache node (create rule)
Merge nodes (qdo 2 selec)
Create new node
Delete node

Regra: R1

Condição E OU () Caso contrário Vá para
> => <=< = função matemática função estatística

'Se' (Vendas/Santa Catarina) > (Vendas/Paraná) 'Vá para' (2a)
'Caso contrário' 'Vá para' (2b)

Consulta: Vendas e Unidade por Estado

Estado	Vendas	Unidades
Santa Catarina	R\$1500,00	73
Paraná	R\$2300,00	102

Nodo: 2a

[Texto livre]
(Santa Catarina) superou o (Paraná) nas vendas.

Fonte: elaborado pelo autor

Além desse ambiente para configuração das regras, sugere-se também a estruturação e a disponibilização de outro ambiente: o ambiente para criação e configuração dos *templates* que serão utilizados pelas regras criadas – sejam esses *templates* textuais ou estruturais (baseados na abordagem de geração de textos não baseada em *templates* textuais). E para facilitar a utilização desses *templates*, sugere-se que, ao serem criados pelos especialistas, eles sejam organizados em diferentes categorias qualitativas, conforme o tipo de análise realizada por eles ou o tipo de conteúdo que apresentam. A Figura 14 ilustra um ambiente para configuração de *templates* estruturais de sentenças analíticas.

Figura 14 – Exemplo de ambiente para configuração de templates estruturais para abordagens de codificação não baseadas em templates textuais

Templates Configuration

Categoria de análise: **Análise Comparativa**

Tipo de análise: **Comparativa com outro indivíduo**

Templates disponíveis:

- Maior número com menor número
- Menor número com maior número
- Específico com maior número
- Específico com menor número
- Específico com pai (somente se relação "todo-parte")
- Específico com indivíduo específico
- Específico com lista de indivíduos específicos

Macons disponíveis: **Localização de indivíduos**

Localizar valor máximo

Localizar valor mínimo

Localizar elemento específico

Funções disponíveis: **Tendência central**

Média

Moda

Mediana

Template selecionado [Específico com maior número]

[Localizar elemento específico: ("Santa Catarina")]

[Imprimir elemento com descritor: ("O estado de")]

[Verbo: ("vendeu")]

[Imprimir valor formatado]

[Texto livre: (" enquanto que")]

[Localizar valor máximo]

[Imprimir elemento com descritor: ("Artigo")]

[Verbo: ("obteve um total de vendas de")]

[Imprimir valor formatado]

*[Optional (If "Año");("no ano de " + XXXX)]

Medidas e métricas do tema: **Financeiro**

Total de vendas

Total de compras

Verbos associados

vendeu

obteve um total de vendas de

Agrupamentos: **Localização geográfica**

País

Estado

Município

Descritores

Artigo

"O estado de"

Consultar teste: Vendas e Unidade por Estado

Estado	Vendas	Unidades
Santa Catarina	R\$1500,00	73
Paraná	R\$2300,00	102

Preview

O estado de Santa Catarina vendeu R\$1500,00, enquanto que o Paraná obteve um total de R\$2300,00.

Medidas e métricas do tema: **Financeiro**

Total de vendas

Total de compras

Verbos associados

vendeu

obteve um total de vendas de

Agrupamentos: **Localização geográfica**

País

Estado

Município

Descritores

Artigo

"O estado de"

Listar valores

Paraná

Rio Grande do Sul

Santa Catarina

São Paulo

Fonte: elaborado pelo autor

É importante observar que foram apresentados apenas exemplos ilustrativos de ambientes para configuração de regras e de *templates* pelos especialistas de domínio, mas, em implementações reais, esses ambientes podem ser muito mais simples ou até mais complexos, dependendo das características da implementação de cada abordagem proposta no presente trabalho. Além disso, esses ambientes poderão ser continuamente aprimorados pela equipe técnica que desenvolve a implementação à medida que os módulos de geração textual e as bases de conhecimento da organização também são aprimorados – fornecendo cada vez mais poder para que os especialistas de domínio mapeiem cenários de análise mais complexos através da produção dos sumários textuais.

Expostas as ideias gerais do método deste trabalho, as questões específicas relacionadas a cada módulo, e a importância da estruturação de ambientes para apoiar os especialistas de domínio na criação de regras e de *templates* para geração textual, será apresentada uma compilação dos principais pontos que foram propostos até agora.

3.6 VISÃO GERAL DO MÉTODO PROPOSTO

Para o desenvolvimento de um serviço de geração textual, propõe-se uma abordagem que – fazendo uso dos recursos de uma plataforma de *Business Intelligence* que utiliza tecnologias semânticas, a Plataforma SBI – possa apoiar a disseminação de conhecimento e os

processos decisórios de maneira flexível e extensível dependendo das necessidades, recursos e limitações de diferentes projetos.

De forma geral, propõe-se que esse serviço seja estruturado inicialmente por módulos, separando responsabilidades, conforme sugerido por Reiter e Dale (2000). Além disso, as ontologias da Plataforma SBI podem ser utilizadas como forma de armazenamento de conhecimento relacionado a questões de domínio da aplicação e de BI. Quando necessário, a utilização de uma Ontologia de Aplicação Estendida representando conhecimento linguístico e visando o apoio direto ao processo de geração textual, também é pertinente.

De forma mais específica, considerando os três principais módulos abordados, faz-se ainda algumas proposições:

(1) Em relação à entrada do módulo de planejamento do documento, sugere-se uma estrutura que represente o contexto do processo decisório, por exemplo, o perfil do usuário, os elementos de BI (como os filtros selecionados em um painel analítico), e os recursos visuais envolvidos). Dessa forma, partindo dos objetivos de comunicação traçados e utilizando como parâmetro de entrada o contexto dinâmico mapeado pela ferramenta analítica, o módulo pode determinar o plano do documento (contemplando o conteúdo a ser gerado e a sua estrutura de apresentação). O plano do documento, por sua vez, baseia-se em heurísticas genéricas ou em regras especializadas pré-determinadas pelos especialistas do domínio.

(2) Já em relação ao módulo de microplanejamento, dentro do contexto da Plataforma SBI, sugere-se que – além das funcionalidades básicas que são responsabilidade desse módulo – sejam exploradas, ainda mais, as atividades de escolha léxica e de geração de expressões para referência. Assim, fazendo uso do contexto do processo decisório e da base de conhecimento da organização, pode-se buscar uma maior adaptabilidade e personalização dos documentos gerados, deixando os textos mais naturais e amigáveis para o usuário final. Nesse sentido, ressalta-se ainda que, embora o módulo de microplanejamento dê, *a priori*, uma maior importância para as abordagens de implementações não baseada em *templates* textuais, pode-se também aplicar essa ideia a abordagens baseadas em *templates* textuais, mediante a inserção dos chamados pontos de flexibilidade. Esses pontos de flexibilidade seriam representados por marcações ou funções especiais que poderiam ser interpretadas pelo módulo de microplanejamento, o qual faria, em tempo de execução da geração textual, a substituição dessas marcações pelos termos ou expressões mais adequados.

(3) Por fim, em relação ao módulo de realização textual, deve ser feita uma codificação que, a partir da especificação do documento (gerada pelo módulo de microplanejamento) gere o texto propriamente dito em linguagem natural ou um texto contendo marcações, como, por exemplo, *tags* HTML. Essas marcações podem ser interpretadas e utilizadas na interface do usuário, alterando a apresentação visual do texto. Além disso, como a especificação que serve de entrada para esse módulo deve conter certo nível de detalhamento léxico e sintático, propõe-se que a definição do tipo de implementação (baseada ou não baseada em *templates* textuais) e, até mesmo, a definição da biblioteca que será utilizada sejam feitas antes mesmo da implementação dos módulos anteriores.

3.7 OUTRAS ABORDAGENS SEMELHANTES

Dalos et al. (2011) apresentam o sistema *EasyText*, que também utiliza como base a arquitetura proposta por Reiter e Dale (2000). Nesse sistema, no módulo de planejamento do documento, a etapa de determinação de conteúdo é feita a partir das tabelas de dados, sendo feita a detecção de células com dados relevantes através de regras de negócio estabelecidas por especialistas e codificadas de forma fixa (sem nenhum módulo de inferência). Contudo, o método proposto no presente trabalho se mostra mais flexível e adaptável, por utilizar também o potencial da Plataforma SBI e das ontologias.

4 VERIFICAÇÃO DO MÉTODO PROPOSTO

Para a demonstração da viabilidade do método proposto no presente trabalho, fez-se a codificação do protótipo de um serviço de geração de sumários textuais que utiliza recursos da Plataforma SBI. Além disso, como esse tipo de plataforma pode ser utilizado em projetos de *Business Intelligence* das mais diversas complexidades e em diferentes domínios de aplicação, na estruturação do serviço gerador de textos, utilizou-se algumas técnicas de codificação visando uma implementação extensível e adaptável às necessidades, recursos e limitações de diferentes projetos. Dessa forma, na seção 4.1, são apresentadas algumas das técnicas que foram utilizadas na codificação do protótipo. Propõe-se que essas técnicas sejam utilizadas na estruturação básica de outras codificações envolvendo projetos com características semelhantes.

Na sequência, seção 4.2, com base nas considerações feitas sobre as abordagens de implementação baseadas e não baseadas em *templates* textuais, é apresentada uma visão geral da codificação realizada para cada um dos módulos do protótipo (planejamento do documento, microplanejamento e realização textual), os quais foram codificados de duas formas diferentes, conforme cada abordagem de implementação.

Por fim, para demonstração das duas versões do protótipo, foram feitas simulações de geração de frases em contextos de *Business Intelligence* predefinidos (ilustrativos), conforme apresentado na seção 4.3.

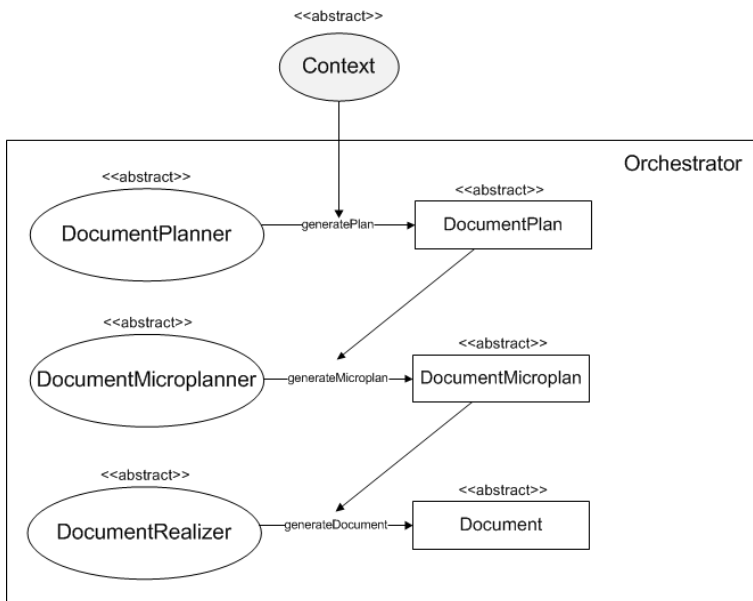
4.1 ESTRUTURAÇÃO BÁSICA DE UM SERVIÇO PARA GERAÇÃO DE TEXTOS

Uma plataforma de BI semântica, como a Plataforma SBI, normalmente é concebida para ser utilizada em projetos diferentes e nos mais variados domínios de aplicação. Dessa forma, ao projetar um serviço para geração de textos que trabalhará de forma integrada com esse tipo de plataforma, faz-se necessária a utilização de um método igualmente flexível e extensível. Além disso, conforme proposto no capítulo 3, é interessante que um serviço de geração textual seja estruturado utilizando a modularização e a divisão de responsabilidades apresentadas na arquitetura abstrata proposta por Reiter e Dale (2000). Assim, será mais fácil aprimorar cada um dos módulos de forma isolada, entre outros benefícios trazidos pelo encapsulamento e baixo acoplamento de módulos (como a possibilidade de eles serem executados, até mesmo, por serviços rodando em servidores diferentes).

Tendo essas questões em mente, propõe-se, antes de mais nada, que seja feita uma implementação-base da estrutura do serviço, separando as responsabilidades de cada módulo e definindo o fluxo de informação entre eles. Nessa linha, ressalta-se que é importante deixar as classes que compõem a codificação-base o mais genéricas possível, permitindo que classes mais específicas possam ser incorporadas na estrutura. Dessa forma, será também possível utilizar as diferentes implementações de cada módulo em outros projetos, dependendo dos seus requisitos e limitações.

A Figura 15 mostra uma possibilidade¹² de implementação da estrutura básica, utilizando sete classes abstratas e uma classe que irá orquestrar o fluxo de informação entre as demais classes (classe *Orchestrator*).

Figura 15 – Representação esquemática da codificação da arquitetura básica do componente



Fonte: elaborado pelo autor

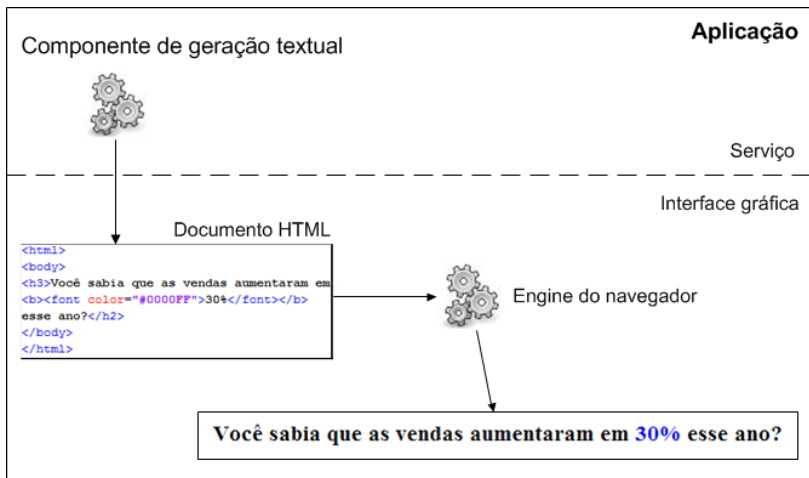
¹² Existem outras possibilidades para projetar e codificar esses elementos, como, por exemplo, interfaces.

Nessa proposta, as classes *DocumentPlanner*, *DocumentMicroplanner* e *DocumentRealizer* representam os três módulos principais da arquitetura (planejamento do documento, microplanejamento e realização textual, respectivamente). Já as classes *DocumentPlan*, *DocumentMicroplan* e *Document* representam as saídas de cada um desses módulos (plano do documento, especificação do texto e o próprio documento).

As saídas dos módulos representam também a entrada dos módulos subsequentes, que devem saber como interpretar essas informações – com exceção do documento que deverá ser interpretado pela própria interface visual do sistema. Assim, o documento poderia ser um texto simples, um código HTML ou qualquer outra representação de texto com metainformações para guiar a renderização na tela.

A Figura 16 representa uma aplicação que possui um componente para geração de textos que tem como saída para a interface gráfica do sistema um documento em formato HTML, o qual será interpretado pelo navegador (*browser*) da máquina do usuário.

Figura 16 – Exemplo de um possível tipo de saída do componente de geração textual: um documento HTML



Fonte: elaborado pelo autor

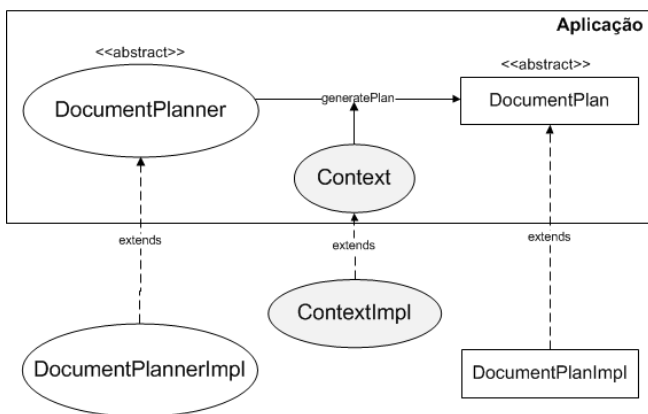
Já a classe *Context*, apresentada na Figura 15, representa o contexto do processo decisório no momento da apresentação das informações na ferramenta de BI, conforme abordado de forma mais

detalhada na seção 3.1.1, e será a entrada para o módulo de planejamento do documento.

Como o nível de detalhamento e a complexidade na representação do contexto de um processo decisório podem variar muito, dependendo das necessidades e das limitações de cada projeto, em termos arquiteturais e técnicos, propõe-se que a codificação-base dessa classe também seja a mais genérica possível. Na implementação apresentada, optou-se pela utilização de uma classe abstrata – a implementação concreta dessa classe abstrata e, conseqüentemente, as informações de contexto que serão utilizadas, dependerão das necessidades de cada projeto e deverão ser feitas no respectivo escopo.

Além disso, como as informações disponíveis no contexto variam nos diferentes projetos, o tratamento dessas informações e o modo como será feito o planejamento do texto a ser gerado (conteúdo e estrutura) também ficam vinculados a cada projeto. Assim, salienta-se a importância de que, na codificação da estrutura-base do serviço de geração textual, seja propiciado um mecanismo para injeção da classe que possuirá a implementação do módulo de planejamento do documento. Essa classe, que estenderia a classe abstrata *DocumentPlanner*, teria como entrada um objeto da classe de contexto concreta definida anteriormente para gerar o plano do documento. A Figura 17 ilustra três classes concretas que estendem as três classes abstratas envolvidas com o módulo de planejamento do documento.

Figura 17 – Classes concretas seguindo a estrutura de codificação-base do módulo de planejamento de documento da aplicação



Fonte: elaborado pelo autor

Seguindo essa abordagem, o plano do documento – que até esse momento também não tem uma estrutura rígida pré-definida – é a entrada para a classe concreta que representa o módulo de microplanejamento (que estende a classe abstrata *DocumentMicroplanner*), a qual, por sua vez, gera um objeto específico: a entrada para a classe que representa o módulo de realização textual (que estende a classe abstrata *DocumentRealizer*).

Assim, seguindo inicialmente a codificação-base do método, as classes concretas que representam os três principais módulos do sistema devem ser injetadas para que manipulem os objetos das classes concretas que representam as unidades de informação que trafegam entre diferentes módulos (implementações específicas de *Context*, *DocumentPlan*, *DocumentMicroplan* e *Document*).

Além dessa codificação básica apresentada, por decisão do projetista do serviço de geração textual, cada um desses módulos já poderia agregar outros submódulos internos e controlar o fluxo de informação entre eles, seguindo as subatividades sugeridas por Reiter e Dale (2000). Por exemplo, a classe abstrata *DocumentPlanner* poderia ter dois atributos de classe representando o módulo de determinação de conteúdo (classe *ContentDeterminator*) e o módulo de estruturação do documento (classe *DocumentStructurer*) – embora as atividades realizadas por esses dois módulos possam ser feitas em um único processo da classe concreta que representará a *DocumentPlanner*¹³.

Apresentadas algumas questões técnicas e a proposta de estruturação do código-base, será exposta a seguir uma visão geral da codificação feita para cada um dos módulos do protótipo – na qual o módulo de planejamento do documento foi implementado de forma genérica e os módulos de microplanejamento e de realização textual foram implementados de duas formas – utilizando e não utilizando *templates* textuais.

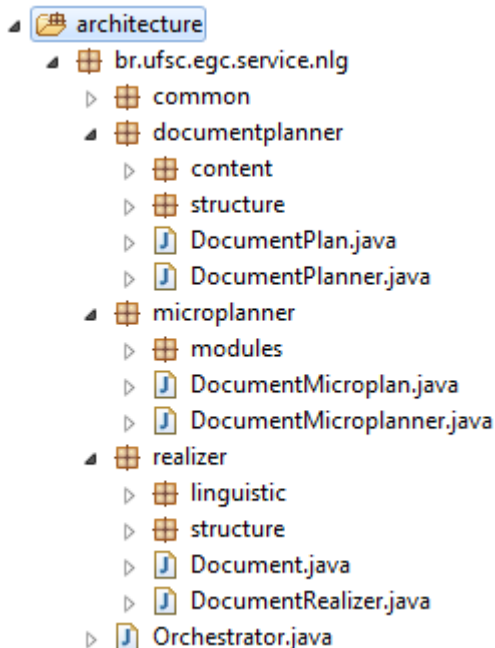
¹³ Essa segunda forma de implementação sugere que *DocumentPlanner* seja uma interface e não uma classe abstrata, já que bastará para esta classe ter um método *generatePlan*, que espera como argumento um objeto do tipo *Context* e que retorna um objeto do tipo *DocumentPlan*.

4.2 VISÃO GERAL DA IMPLEMENTAÇÃO DO PROTÓTIPO DO MÉTODO PROPOSTO

Conforme a estrutura-base definida na seção 4.1, procurou-se fazer a implementação do protótipo de forma que os três módulos principais do serviço de geração textual não tivessem dependência entre si e pudessem trabalhar de forma isolada – podendo, até mesmo, ser substituídos por outras implementações quando necessário.

Em um primeiro momento, codificou-se em Java um conjunto inicial de classes (presentes no *source folder* “*architecture*” do projeto Java), representando os elementos básicos da estrutura, e estabelecendo a interação e o fluxo de informações entre eles (através da classe *Orchestrator*). Além disso, por questões de organização, separou-se o código em três pacotes, os quais representam os três módulos – *DocumentPlanner*, *MicroPlanner* e *Realizer* –, e criou-se o pacote *Common*, onde ficariam as classes utilizadas por mais de um módulo, conforme pode ser observado na Figura 18.

Figura 18 – Organização da codificação dos elementos básicos da arquitetura proposta



Fonte: elaborado pelo autor

Conforme proposto na seção 4.1, pode-se dizer que a função do elemento orquestrador dos módulos (*Orchestrator*) é relativamente simples: ele agrega os três módulos – que são passados no seu construtor e armazenados como atributos de classe – e encadeia as responsabilidades de cada um no estilo *pipeline*, de forma que a saída de um módulo é a entrada para o outro. Dessa forma, para gerar um documento (método *generateDocument*) a partir do contexto apresentado (representado por um objeto da classe *Context*), o orquestrador pede para o módulo de planejamento do documento (*DocumentPlanner*) gerar o plano do documento (*DocumentPlan*). Com esse plano, o orquestrador pede para o módulo de microplanejamento (*DocumentMicroplanner*) gerar o documento microplanejado e, finalmente, com esse documento, ele pede para o módulo de realização de texto (*DocumentRealizer*) gerar o documento final, conforme pode ser visto no código representado na Figura 19.

Figura 19 – Classe que representa o elemento orquestrador dos módulos da arquitetura

```
public abstract class Orchestrator {
    protected DocumentPlanner documentPlanner;
    protected DocumentMicroplanner documentMicroplanner;
    protected DocumentRealizer documentRealizer;

    public Orchestrator(DocumentPlanner documentPlanner, DocumentMicroplanner documentMicroplanner,
        DocumentRealizer documentRealizer){
        this.documentPlanner = documentPlanner;
        this.documentMicroplanner = documentMicroplanner;
        this.documentRealizer = documentRealizer;
    }

    public Document generateDocument(Context context){
        DocumentPlan documentPlan = this.documentPlanner.generatePlan(context);
        DocumentMicroplan documentMicroplan = this.documentMicroplanner.generateMicroPlan(documentPlan);
        Document finalDocument = this.documentRealizer.generateDocument(documentMicroplan);
        return finalDocument;
    }
}
```

Fonte: elaborado pelo autor

Um ponto técnico importante a ser realçado é que a classe *Orchestrator* foi codificada como uma classe abstrata, assim como as classes que representam os módulos (*DocumentPlanner*, *DocumentMicroplanner* e *DocumentRealizer*) e as entradas e saídas dos módulos (*Context*, *DocumentPlan*, *DocumentMicroplan* e *Document*). Isso exige que todas essas classes tenham que ter implementações especializadas, e também garante que possam ser feitas diferentes codificações – das mais simples às mais complexas – de um sistema gerador de textos seguindo essa estrutura-base.

Além disso, no protótipo, foram estruturados mecanismos de injeção das classes que representam as implementações concretas de cada um dos módulos, visando uma maior flexibilidade e possibilidade de troca, a qualquer momento, das implementações desses módulos. Assim, os nomes das classes a serem injetadas no sistema e utilizadas pelo serviço são informadas em propriedades específicas de um arquivo de propriedades, conforme o exemplo apresentado no Quadro 14 (sendo que essas classes devem estar também no *buildpath* do projeto).

Quadro 14 – Propriedades indicando os nomes das classes com a implementação concreta dos módulos a serem injetados no serviço

textgeneration.modules.planner.factory.class	=
br.ufsc.egc.service.nlg.documentplanner.PrototypeDocumentPlannerFactory	
textgeneration.modules.microplanner.factory.class	=
br.ufsc.egc.service.nlg.documentplanner.PrototypeDocumentMicroplannerFactory	
textgeneration.modules.realizer.factory.class	=
br.ufsc.egc.service.nlg.documentplanner.PrototypeDocumentRealizerFactory	

Fonte: elaborado pelo autor

Partindo da implementação da estrutura-base do serviço de geração textual, em um segundo momento, fez-se a codificação das classes concretas, já relacionadas ao contexto de BI e fazendo uso dos recursos da Plataforma SBI. Dessa forma, foram criadas as classes *DocumentPlannerImpl* e *DocumentPlanImpl* – representando, respectivamente, a codificação de um módulo de planejamento de documento e de sua saída, o plano do documento. Também se fez a implementação de uma classe representando o contexto que servirá de entrada para o módulo de planejamento do documento codificado.

De forma mais específica, no protótipo desenvolvido, optou-se por fazer uma codificação simulando a necessidade de geração de documentos com informações relacionadas a recursos visuais presentes em painéis analíticos, como gráficos e relatórios. Para isso, fez-se uma codificação simples para servir de exemplo, sendo o mais genérica possível – com regras independentes de recursos específicos, mas relacionadas com algumas características gerais que poderiam ser encontradas em diferentes recursos.

Dessa forma, dentro do módulo de planejamento do documento, foi criada uma classe para determinação do conteúdo e para estruturação do documento que será apresentado. Nessa classe, foram codificadas algumas regras que utilizam como contexto de entrada as características

de um determinado recurso e o perfil do usuário que está utilizando o sistema – essa classe foi chamada de *ResourceBasedDocumentPlanner*. Pela codificação feita, dependendo dos elementos presentes no contexto de entrada, essa classe pode utilizar outras classes de apoio especializadas no tratamento de alguns aspectos gerais por heurísticas genéricas. Por exemplo, os aspectos de temporalidade são tratados pela classe *TemporalElementsDocumentPlanner* e as relações de todo-parte pela classe *WholePartElementsDocumentPlanner*.

Para exemplificar os aspectos tratados pelas regras presentes no módulo de planejamento do documento do protótipo, as quais estão codificadas em Java, elas estão explicitadas no Quadro 15 – didaticamente – em linguagem natural.

Quadro 15 – Regras codificadas no módulo de planejamento de documento dos protótipos

- **Regra 1:** para os recursos baseados em consultas com uma dimensão e uma medida, apresentar uma sentença que mostre os elementos (valores da dimensão) com o maior valor da medida e o com o menor valor da medida. Na sequência, outra sentença com a média geral da medida considerando todo o conjunto de dados (todos os valores da dimensão) deve ser apresentada.
- **Regra 2:** se a consulta vinculada ao recurso possuir uma medida e uma dimensão temporal, apresentar uma sentença com o valor da medida no período mais recente disponível, e uma sentença que, baseando-se nos dados dessa série temporal, faça uma predição do próximo valor que essa medida terá no futuro.
- **Regra 3:** se a consulta vinculada ao recurso contiver uma medida e duas dimensões correlacionadas hierarquicamente (relação do tipo todo-parte), apresentar uma sentença com o valor da medida agrupado pela dimensão superior na hierarquia (valor do todo). Na sequência, apresentar outra sentença que informe qual elemento (valor de dimensão) da dimensão inferior da hierarquia teve o maior valor da medida (valor da maior parte), e outra sentença que diga qual foi esse valor. Por fim, utilizando o contexto do usuário, apresentar o valor do elemento (valor de uma parte específica) que tem relação com o perfil desse usuário.

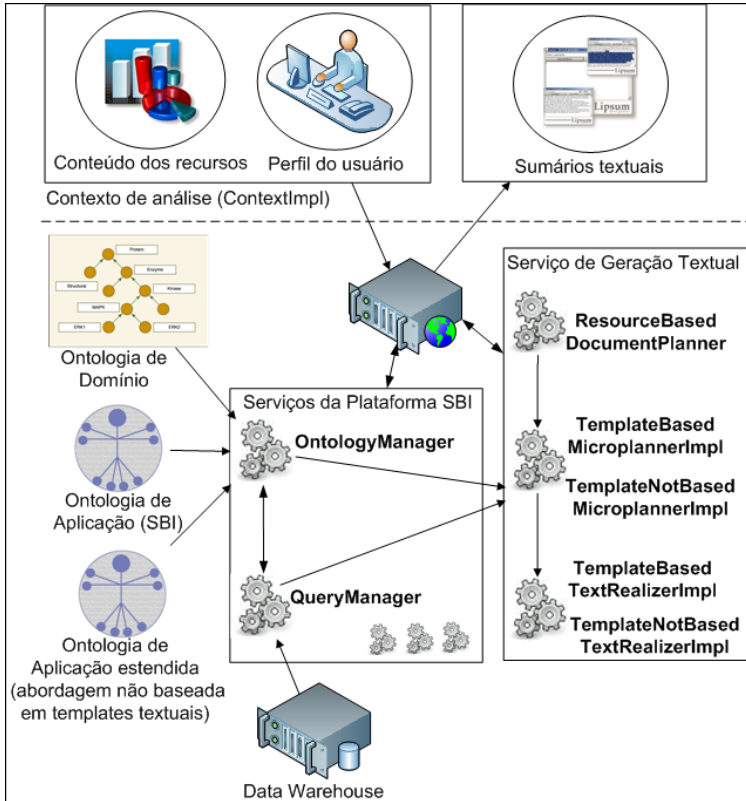
Fonte: elaborado pelo autor

Nesse cenário, em que os recursos visuais estão publicados em um painel analítico visualizado por um usuário (tomador de decisão), se os elementos de BI presentes (associados às consultas que geram o conjunto de dados vinculado a cada recurso) satisfizerem às condições das regras codificadas, sumários textuais pré-determinados serão gerados.

Em relação à codificação das regras, as verificações dos elementos de BI presentes no cenário foram implementadas de forma geral, conforme apresentado anteriormente, pela classe *ResourceBasedDocumentPlanner* e suas classes de apoio. Já os aspectos específicos, considerando uma abordagem não baseada em *templates* textuais e outra baseada neles, foram implementados nas especializações dessas classes. Dessa forma, para cada abordagem, um tipo de plano de documento foi gerado baseado nas necessidades do módulo de microplanejamento específico da abordagem.

A Figura 20 ilustra a estrutura e o funcionamento dos protótipos codificados, bem como a interação do serviço de geração textual com os serviços e as ontologias da Plataforma SBI.

Figura 20 – Estruturação e fluxo de informações no protótipo desenvolvido



Fonte: elaborado pelo autor

Nessa linha, para a abordagem não baseada em *templates* textuais, foram criados e utilizados modelos estruturais (conforme sugerido na seção 3.4.2) e, para a outra abordagem, foram criados e utilizados *templates* textuais. Para cada caso, o respectivo módulo de planejamento do documento tem acesso a esses *templates*, determina quais deles serão utilizados nos diferentes contextos apresentados (dependendo do tipo de conteúdo que cada um provê), e estabelece os parâmetros que serão utilizados.

A Tabela 8 apresenta uma representação dos modelos utilizados pela implementação baseada em *templates* textuais. Nesses modelos, os elementos em **negrito** representam textos fixos e os elementos delimitados entre colchetes representam funções parametrizáveis que retornam valores numéricos ou textuais.

Tabela 8 – Representação dos templates textuais criados

Template 1	➤ [Elemento com maior valor para medida “Total de Vendas”] vendeu R\$ [maior valor da medida “Total de Vendas”], enquanto que [Elemento com menor valor para medida “Total de Vendas”] vendeu R\$ [menor valor da medida “Total de Vendas”].
Template 2	➤ A média de vendas foi de R\$ [média dos valores da medida “Total de Vendas”].
Template 3	➤ R\$ [valor da medida “Total de Vendas” para o ano mais recente] foi vendido em [ano mais recente].
Template 4	➤ É previsto que R\$ [predição do valor da medida “Total de Vendas” para o ano mais recente + 1] seja vendido em [ano mais recente + 1].
Template 5	➤ O índice de qualidade de vida brasileiro foi de [valor da medida “IQV” agrupado pela dimensão “país” (na sumarização é feita a média)].
Template 6	➤ No Brasil , [elemento da dimensão “cidade” com maior valor da medida “IQV”] teve o melhor índice de qualidade de vida.
Template 7	➤ A cidade teve um índice de [valor do elemento da dimensão “cidade” com maior valor da medida “IQV”].
Template 8	➤ [elemento da dimensão “cidade” com valor especificado*] teve um índice de [valor da medida “IQV” para o elemento da dimensão “cidade” com o valor especificado].

Fonte: elaborado pelo autor

Dessa forma, durante a etapa de realização textual executada pela classe *TemplateBasedTextRealizerImpl*, as funções presentes nos *templates* textuais são substituídas dinamicamente por valores presentes no conjunto de dados associados a cada *template* (que representam o resultado de uma consulta OLAP).

Já na implementação não baseada em *templates* textuais, os modelos estruturais foram criados utilizando classes genéricas que equivalem a elementos linguísticos – como, por exemplo, Sentença simples, Sentença composta, Sujeito, Verbo, Objeto e Artigo. Essas

classes possuem atributos para especificar os elementos linguísticos como, o tempo verbal de um objeto Verbo. Assim, após o módulo de planejamento do documento determinar quais *templates* estruturais serão utilizados e quais os elementos de BI envolvidos, o módulo de microplanejamento (representado pela classe *TemplateNotBasedMicroplannerImpl*) determina – através da Ontologia de Aplicação Estendida – quais verbos serão utilizados, de acordo com as medidas envolvidas. Além disso, o módulo de microplanejamento também define como referenciar os demais elementos presentes no *template*.

Posteriormente, já na etapa de realização textual, a classe *TemplateNotBased TextRealizerImpl*, utilizando a classe *SimpleNLGSurfaceRealizator*, faz uma adaptação para a entrada esperada pela biblioteca *SimpleNLG*, convertendo as estruturas linguísticas citadas anteriormente em sentenças formadas por objetos configurados advindos de classes fornecidas pela biblioteca (por exemplo, *SPhraseSpec*, *VPPhraseSpec*, *NPPPhraseSpec*, *CoordinatedPhraseElement* e *PPPPhraseSpec*).

Por fim, as sentenças são transformadas em texto pela classe *Realizer*, que utiliza como apoio um *lexicon* – disponível em formato XML – que contém informações relacionadas a elementos da língua inglesa (por exemplo, conjugação dos verbos e flexão de adjetivos). O Quadro 16 apresenta parte desse *lexicon*, chamado de *default-lexicon.xml*, que é disponibilizado com a biblioteca *SimpleNLG*.

Quadro 16 – Parte do *lexicon* utilizado na abordagem não baseada em *templates* textuais

```

<word>
<base>give</base>
<category>verb</category>
<id>E0029785</id>
<present3s>gives</present3s>
<ditransitive/>
<intransitive/>
<past>gave</past>
<pastParticiple>given</pastParticiple>
<presentParticiple>giving</presentParticiple>
<transitive/>
</word>
<word>
<base>given</base>
<category>adjective</category>
<id>E0029787</id>
<classifying/>
<predicative/>
</word>
<word>
<base>glad</base>
<category>adjective</category>
<id>E0029796</id>
<classifying/>
<comparative>gladder</comparative>
<predicative/>
<qualitative/>
<superlative>gladdest</superlative>
</word>

```

Fonte: elaborado pelo autor

A seguir, será apresentada uma simulação do funcionamento das duas versões do protótipo.

4.3 SIMULAÇÃO DE FUNCIONAMENTO DO MÉTODO PROPOSTO

Para simular o funcionamento do modelo proposto pelas duas versões do protótipo, apresentadas na seção 4.2, criaram-se alguns contextos representando cenários hipotéticos de BI para servirem de entrada para o serviço de geração textual. Dessa forma, dependendo dos elementos presentes nesses contextos, e seguindo as regras pré-estabelecidas no módulo de planejamento do documento, observou-se a geração dos sumários textuais de forma dinâmica.

Foram pensados em quatro cenários para essa simulação, envolvendo recursos criados no contexto da Plataforma SBI e publicados em um painel analítico hipotético. Esses recursos são baseados em consultas OLAP geradas pelo serviço *QueryManager* da Plataforma SBI. Além disso, essas consultas envolvem elementos presentes nas ontologias da plataforma (como medidas, dimensões e filtros). A seguir, são listados os recursos criados para a simulação.

a) Contexto 1

- Gráfico, identificado pelo ID 21, baseado em uma consulta OLAP contendo uma medida e uma dimensão.
 - A medida (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:measure_vendas”) representa um total de vendas (*Sales*).
 - A dimensão (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:header_vendedor”) representa um vendedor (*Seller*).
 - A massa de dados resultante dessa consulta apresentando a medida agregada e sumarizada pela dimensão pode ser visualizada na Tabela 9.

Tabela 9 – Massa de dados do contexto 1

Vendedor	Total de vendas
João da Silva	R\$9.750,00
Fábio Santana	R\$5.600,00
Fábio Júnior	R\$3.400,00
Luis Alberto	R\$1.300,00
Maria Luiza	R\$7.100,00

Fonte: elaborado pelo autor

b) Contexto 2

- Mesmo gráfico do contexto 1, identificado pelo ID 21, baseado em uma consulta OLAP contendo uma medida e uma dimensão.
 - A medida (identificada na ontologia de aplicação da Plataforma SBI com a URI

“ano:measure_vendas”) representa um total de vendas (*Sales*).

- A dimensão (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:header_vendedor”) representa um vendedor (*Seller*).
- A massa de dados, diferente do contexto 1¹⁴, resultante da consulta em outro momento, apresentando a medida agregada e sumarizada pela dimensão pode ser visualizada na Tabela 10.

Tabela 10 – Massa de dados do contexto 2

Vendedor	Total de vendas
João da Silva	R\$5.500,00
Fábio Santana	R\$2.200,00
Fábio Júnior	R\$2.200,00
Luís Alberto	R\$7.300,00
Maria Luiza	R\$8.500,00

Fonte: elaborado pelo autor

c) Contexto 3

- Gráfico, identificado pelo ID 32, baseado em uma consulta OLAP contendo uma medida e uma dimensão.
 - A medida (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:measure_vendas”) representa um total de vendas (*Sales*).
 - A dimensão (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:header_ano”) representa um ano (*Year*).
 - A massa de dados resultante dessa consulta, apresentando a medida agregada e sumarizada pela dimensão pode ser visualizada na Tabela 11.

¹⁴ No contexto 2, propositadamente, ocorre um empate entre os dois vendedores que tiveram um valor menor de vendas.

Tabela 11 – Massa de dados do contexto 3

Ano	Total de vendas
2008	R\$28.000,00
2009	R\$32.000,00
2010	R\$35.500,00
2011	R\$39.600,00

Fonte: elaborado pelo autor

d) Contexto 4

- Usuário utilizando o sistema é vinculado à cidade de Florianópolis.
- Gráfico, identificado pelo ID 46, baseado em uma consulta OLAP contendo uma medida e duas dimensões.
 - A medida (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:life_quality_indicator”) representa um indicador de qualidade de vida (*LifeQualityIndicator*).
 - A primeira dimensão (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:header_pais”) representa um país (*Country*).
 - A segunda dimensão (identificada na ontologia de aplicação da Plataforma SBI com a URI “ano:header_cidade”) representa uma cidade (*City*).
 - A massa de dados resultante dessa consulta apresentando a medida agregada e sumarizada pelas dimensões pode ser visualizada na Tabela 12.

Tabela 12 – Massa de dados do contexto 4

País	Cidade	Índice de Qualidade de vida¹⁵
Brasil	São Paulo	4,8
	Florianópolis	6,5
	Salvador	3,7
	Curitiba	6,7
Argentina	Buenos Aires	6,8
	Córdoba	2,3

Fonte: elaborado pelo autor

Diante desses quatro cenários pré-estabelecidos e utilizando as regras descritas na seção 4.2, são apresentados a seguir os textos gerados pelas duas versões do protótipo desenvolvidas – utilizando e não utilizando *templates* textuais – e algumas considerações sobre o processo de geração textual realizado por cada uma delas.

4.3.1 Implementação baseada em templates

Utilizando a implementação do protótipo que foi baseada em uma abordagem baseada em *templates* textuais, são apresentadas, na Tabela 13, as frases geradas a partir dos contextos apresentados na seção 4.3 e das regras codificadas, expostas no Quadro 15.

Tabela 13 – Frases geradas pelo protótipo com abordagem baseada em *templates* textuais

Contexto	Frases geradas
1	<ul style="list-style-type: none"> ➤ João da Silva vendeu R\$9.720,00, enquanto que Luís Alberto vendeu R\$1.300,00. ➤ A média de vendas foi de R\$5.430,00.
2	<ul style="list-style-type: none"> ➤ João da Silva vendeu R\$9.720,00, enquanto que Luís Alberto vendeu R\$1.300,00. ➤ A média de vendas foi de R\$5.430,00.
3	<ul style="list-style-type: none"> ➤ R\$39.600,00 foi vendido em 2011. ➤ É previsto que R\$43.350,00 seja vendido em 2012.
4	<ul style="list-style-type: none"> ➤ O índice de qualidade de vida brasileiro foi de 5,69.

¹⁵ Os valores são meramente ilustrativos e não fazem referência aos indicadores reais.

	<ul style="list-style-type: none"> ➤ No Brasil, Curitiba teve o melhor índice de qualidade de vida. ➤ A cidade teve um índice de 6,7. ➤ Florianópolis teve um índice de 6,5.
--	---

Fonte: elaborado pelo autor

4.3.2 Implementação não baseada em templates

Utilizando a implementação do protótipo que foi fundamentada em uma abordagem não baseada em *templates* textuais, são apresentadas, na Tabela 14, a seguir, as frases geradas a partir dos contextos apresentados na seção 4.3 e das regras codificadas expostas no Quadro 15.

Tabela 14 – Frases geradas pelo protótipo com abordagem não baseada em *templates* textuais

Contexto	Frases geradas
1	<ul style="list-style-type: none"> ➤ João da Silva sells R\$9.750,00 while Luis Alberto sells R\$1.300,00. ➤ The mean of sales was R\$5.430,00.
2	<ul style="list-style-type: none"> ➤ Maria Luiza sells R\$8.500,00 while Fábio Santana and Fábio Júnior sold R\$2.200,00. ➤ The mean of sales was R\$5.140,00.
3	<ul style="list-style-type: none"> ➤ R\$39.600,00 was sold in 2011. ➤ Is predicted that R\$43.350,00 will be sold in 2012.
4	<ul style="list-style-type: none"> ➤ The Life Quality Indicator for Brazil was 5.6875. ➤ In Brazil, Curitiba had the best Life Quality Indicator. ➤ The city had 6.7. ➤ Florianópolis had a Life Quality Indicator of 6.5

Fonte: elaborado pelo autor

É importante salientar que as frases estão em inglês devido ao fato dessa implementação ter feito uso, no módulo de realização textual, da biblioteca *SimpleNLG* (SIMPLENLG, 2012) – que atualmente não tem suporte para a língua portuguesa.

A seguir, são feitas algumas considerações sobre a avaliação proposta no presente trabalho.

4.4 CONSIDERAÇÕES SOBRE A AVALIAÇÃO DA PROPOSTA

O objetivo deste capítulo foi o desenvolvimento de um protótipo para ilustrar o funcionamento de um serviço codificado seguindo o método proposto neste trabalho: geração de sumários textuais que apresentam informações dinâmicas em linguagem natural relacionados a contextos informacionais ilustrativos, seguindo regras pré-definidas e utilizando uma base de conhecimento representada em ontologias. Esse protótipo foi codificado de duas formas – utilizando e não utilizando *templates* textuais –, mostrando, assim, a independência dos tipos específicos de implementações.

Além disso, procurou-se também estruturar uma codificação-base envolvendo os diferentes módulos e o fluxo de informação entre os protótipos, permitindo uma flexibilidade na evolução e na troca das implementações (através de mecanismos de injeção de classes) – conforme as necessidades, recursos e limitações de cada projeto.

Por fim, utilizando as duas versões codificadas do protótipo, apresentou-se a simulação de um cenário hipotético envolvendo a geração de sumários textuais para apoiar a disseminação de conhecimento e os processos decisórios no contexto de *Business Intelligence*. Vale ressaltar que o objetivo dessa simulação foi demonstrar o funcionamento e a viabilidade da abordagem proposta, não envolvendo uma avaliação de sua efetividade real no apoio a processos decisórios – o que seria complexo de ser realizado. Segundo Turban et al. (2009), o valor para o negócio do próprio BI como um todo é complexo de ser avaliado, devido aos seus muitos benefícios intangíveis.

5 CONSIDERAÇÕES FINAIS

5.1 CONTRIBUIÇÕES

No presente trabalho, foi apresentado um método a ser utilizado na geração de sumários textuais para apoiar a disseminação de conhecimento e os processos decisórios no contexto de utilização de uma plataforma de *Business Intelligence* que utiliza tecnologias semânticas, a Plataforma SBI. Esses sumários textuais podem representar um meio de apresentação de informações e de disseminação de conhecimento complementar aos recursos analíticos disponibilizados pela plataforma.

Segundo o método proposto, durante o processo de geração textual, as sentenças são estruturadas de forma dinâmica, de acordo com as informações disponíveis em cada cenário informacional apresentado (contexto), seguindo regras de produção pré-estabelecidas por especialistas e utilizando o conhecimento disponível nas bases de conhecimento das organizações. Essas bases de conhecimento são representadas pelas ontologias da Plataforma SBI e, se necessário, por uma Ontologia de Aplicação Estendida (específica para apoio ao processo de geração textual).

Além disso, procurou-se apresentar um método flexível e extensível, de forma que diferentes tipos de implementações possam ser utilizadas, dependendo das necessidades, recursos e limitações de cada projeto de *Business Intelligence* que utilize a Plataforma SBI.

Nas simulações realizadas com as duas versões codificadas de protótipos, uma utilizando *templates* textuais e outra *templates* estruturais, ficou demonstrada a viabilidade de implementação do método proposto, sendo gerados sumários textuais de forma dinâmica a partir de contextos informacionais apresentados e de regras pré-estabelecidas.

Vale ressaltar novamente que o universo de possibilidades disponibilizado com a utilização e exploração das potencialidades das ontologias é grande, e que quanto melhores forem as bases de conhecimento utilizadas e os serviços que delas fazem uso, mais eficaz pode se tornar o processo de geração de textos relevantes que possam apoiar, efetivamente, um processo decisório.

5.2 LIMITAÇÕES E TRABALHOS FUTUROS

O método proposto no presente trabalho visa provar, através de sumários textuais, um meio de apresentação de informações e

disseminação de conhecimento complementar aos recursos tradicionalmente apresentados pelas ferramentas analíticas, auxiliando potencialmente nos processos decisórios das organizações dentro do contexto de *Business Intelligence*. Para isso, faz uso das potencialidades de uma plataforma de BI que utiliza tecnologias semânticas, a Plataforma SBI, e de regras criadas por especialista de domínio.

De qualquer forma, mesmo considerando as potencialidades dos sumários textuais nos processos de disseminação e de aquisição de conhecimento (conforme apresentado na seção 1.2), não se pode garantir a sua eficácia real no apoio a processos decisórios. Em função disso, outros trabalhos, fazendo estudos de caso em organizações para a verificação dessa questão, têm que ser realizados.

Em relação ao protótipo apresentado, ele teve como objetivo demonstrar a viabilidade de implementação do método proposto, sendo gerados sumários textuais de forma dinâmica a partir de contextos informacionais apresentados e de regras pré-estabelecidas. Contudo, a implementação dos módulos desse protótipo, bem como os cenários e as regras codificados e as sentenças geradas, são relativamente simples em relação aos requisitos comumente envolvidos em um projeto real, no qual se poderia ter cada uma dessas questões bem mais exploradas.

Além disso, a estratégia de realização textual em tempo de execução também pode se mostrar inviável em cenários reais envolvendo grandes volumes de dados. Para esses casos, uma estratégia de processamento em *batch*, fazendo a pré-realização de sentenças envolvendo diferentes contextos de entrada e as armazenando em uma estrutura de *cache* pode se mostrar interessante.

Por fim, do ponto de vista tecnológico, uma das possibilidades de trabalho futuro é uma questão importante que também poderia ser explorada: a possibilidade de utilização de uma pergunta feita em linguagem natural como contexto de entrada do serviço de geração textual, que proveria uma resposta trazendo informações analíticas relacionadas à pergunta realizada. Essa questão refere-se a uma abordagem de *questioning-answering* completa, sendo uma forma de complementar o trabalho de Silva (2011), conforme comentado no Capítulo 1. Conforme também já exposto no Capítulo 1, segundo Fenn (2011), sistemas baseados em perguntas e respostas em linguagem natural (*natural-language question-answering*) irão suportar a tomada de decisão de forma radicalmente mais rápida. De acordo com Fenn e Lehong (2011), a expectativa é de que, em longo prazo, além do horizonte de cinco anos, serviços enriquecidos com contexto

(semântica) e sistemas baseados em perguntas e respostas em linguagem natural estejam entre as principais forças tecnológicas.

REFERÊNCIAS BIBLIOGRÁFICAS

AUSTIN, J. L. **How to do things with words**. Clarendon: Oxford, 1962.

BERNERS-LEE, T.; HANDLER, J.; LASSILA, O. **The Semantic Web**: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. **Scientific American**, v.2, n.1, p.1-12, maio 2001. Disponível em: <<http://www.sciam.com/article.cfm?id=the-semantic-web>>. Acesso em: 10 abr. 2010.

BONTCHEVA, K. Generating Tailored Textual Summaries from Ontologies. In: EUROPEAN SEMANTICWEB CONFERENCE, 2., 2005, Greece. **Proceedings...** [S.l.]:Springerpp, 2005. p. 531-545.

BURRELL, G.; MORGAN, G. **Sociological paradigms and organisational analysis**: elements of the sociology of corporate life. Inglaterra: Ashgate Publishing Limited, 1979. 427 p.

CALHOUN, M. A.; STARBUCK, W. H. Barriers do creating knowledge. In: EASTERBY-SMITH, M.; LYLES, M. **Handbook of organizational learning and knowledge management**. Malden: Blackwell, 2005.

CANHAM, M.; Hegarty M. **Effects of knowledge and display design on comprehension of complex graphics**: learning and instruction, v.20, abr. 2010. p. 155-166.

CHEN, M. et al. Data, information, and knowledge in visualization. **IEEE Computer Graphics and Applications**, v. 29, n. 1, p. 12-19, jan/fev. 2009.

CHOMSKY, N. **Topics in the Theory of Generative Grammar**. 5. ed. 1978. 95 p.

COOK, M. P. Visual representations in science education: the influence of prior knowledge and cognitive load theory on instructional design principles. **Science Education**, v. 90, n. 6, p. 1073-1091, 2006.

CULLEN, C.; O'NEILL, I.; HANNA, P. Flexible Natural Language Generation in Multiple Contexts. **Lecture Notes in Artificial Intelligence**, v. 5603, p. 142–153. [S.l.]: Springer, 2009.

DANLOS, L.; MEUNIER, F.; COMBET, V. EasyText: an Operational NLG System. EUROPEAN WORKSHOP ON NATURAL LANGUAGE GENERATION, 13., 2011, França. **Manuscript...** Disponível em: <<http://hal.archives-ouvertes.fr/docs/00/61/47/60/PDF/easyVanessaFredo.pdf>>. Acesso em: 12 nov. 2011.

DAVENPORT, T. H.; MARCHAND, D. A.; DICKSON, T. **Dominando a Gestão da Informação**. Porto Alegre: Bookman, 2004. 408 p.

EGC. **Portal do Departamento de Engenharia e Gestão do Conhecimento da UFSC**. Disponível em: <<http://www.egc.ufsc.br/>>. Acesso em: 15 dez. 2012.

EVERS, H. **Towards a Malaysian Knowledge Society**. Bangi: [s.n.], 2001.

FENN, J. Trends that matter: 84 technology, societal and business trends. In: _____. **Trends that matter: top trends and their business impact**. [S.l.]: Gartner, 2011.

FENN, J.; LEHONG, H. Hype Cycle for Emerging Technologies. In: FENN, J. **Gartner's hype cycle special report for 2011**. [S.l.]: Gartner, 2011.

FERRES, L. et al. Helping people with visual impairments gain access to graphical information through natural language: the iGraph System. In: INTERNATIONAL CONFERENCE ON COMPUTERS HELPING PEOPLE WITH SPECIAL NEEDS, 10., 2006, Austria. **Lecture notes...** [S.l.]:[s.n.], 2006. p. 1122-1130.

FERRES, L.; LINDGAARD, G.; SUMEGI, L. Evaluating a tool for improving accessibility to charts and graphs. INTERNATIONAL ACM SIGACCESS CONFERENCE ON COMPUTERS AND ACCESSIBILITY, 12., 2010, Orlando. **Proceedings...** Nova Iorque: ACM, 2010. p. 83-90.

FREITAS, F. L. G. **Ontologias e Web semântica**. Santos: Universidade Católica de Santos, 2003.

GHISI, F. B.; CECI, F.; SELL, D. Aspectos relacionados com a eficácia do processo de aquisição de conhecimento a partir da apresentação de informações numéricas: sumários textuais podem ser mais adequados que representações gráficas? In: INFORMATION DESIGN INTERNATIONAL CONFERENCE, 5., 2011, Brasil. **Proceedings...** Brasil: [s.n], 2011.

GOLFARELLI, M.; RIZZI, S.; CELLA, I. What`s next in business intelligence. In: INTERNATIONAL WORKSHOP ON DATA WAREHOUSING AND OLAP, 7., 2004, Washington. **Proceedings...** Nova Iorque: ACM, 2004. p. 1-6

GRUBER, T. R. **A translation approach to portable ontology specifications**. Knowledge System Laboratory. Stanford: Stanford University, 1993. Disponível em: <<http://tomgruber.org/writing/ontolingua-kaj-1993.pdf>>. Acesso em: 15 out. 2012.

HAJNYSZ, M. **Next generation business intelligence for small and mid-size enterprises: adoption, preferences and offers in Poland**. 2007. 152 f. Dissertação (Mestrado) – Center For Information And Communication Technologies (CICT), Technical University of Denmark, Kongens Lyngby, 2007. Disponível em: <http://www.imm.dtu.dk/pubdb/views/edoc_download.php/5396/pdf/imm5396.pdf>. Acesso em: 15 jun. 2010.

GUARINO, N. **Formal ontology and information systems**. [S.l.]: IOS, 1998.

HIRSCHMAN, L.; GAIZAUSKAS, R. Natural language question answering: the view from here. **Natural Language Engineering**, Reino Unido, v. 7, n. 4, p.275-300, 2001.

INMON, W. H. **Building the Data Warehouse**. 4. ed. Indianapolis: Wiley Publishing, 2005. 541 p.

- KACPRZYK, J.; ZADROZNY, S. Supporting decision making via verbalization of data analysis results using linguistic data summaries. **Recent Advances in Decision Making**, Berlin, v. 222, p. 121-143, 2009.
- KHAN, R. Business Intelligence: an integrated approach. **Business Intelligence Journal**, v. 5, n. 1, p. 64-70, maio 2012. Disponível em: <http://www.saycocorporativo.com/saycouk/bij/journal/Vol5No1/Article_7.pdf>. Acesso em: 15 set. 2010.
- KIMBALL, R.; ROSS, M. **The Data Warehouse toolkit**: the complete guide to dimensional Modeling. 2 ed. Nova Iorque: Wiley Computer Publishing, 2002. 436 p.
- KIMBALL, R. et al. **The Data Warehouse lifecycle toolkit**: practical techniques for building dimensional data warehouses. 2 ed., Indianápolis: Wiley Publishing, 2008.
- LAW, A. S. E et al. A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. **Journal of clinical monitoring and computing**, v. 19, n. 3, p. 183-94, 2005.
- MACEDO, H. Model driven development approach to natural language generation systems. **ACM SIGSOFT Software Engineering Notes**, v. 35, n. 4, p. 1, 2010.
- MELLISH, C. et al. A Reference Architecture for Natural Language Generation Systems. **Natural Language Engineering**, v. 12, n. 1, p. 1, 2006.
- NEUBAUER FILHO, A.; ASSAD, R. Business Process Management: monitorando a estratégia do negócio em tempos de governança corporativa. **Gestão**, Curitiba, v. 8, n. 8, p.7-26, jan. 2007.
- OLIVERA, F. Memory systems in organizations: an empirical investigation of mechanisms for knowledge collection, storage and access. **Journal of Management Studies**, v. 37, n.6, p. 811-832, 2000.

PINHEIRO, C. A. R. **Inteligência analítica**: mineração de dados e descoberta de conhecimento. Rio de Janeiro: Ciência Moderna, 2008. 397 p.

POLI, R.; OBRST, L. The Interplay Between Ontology as Categorical Analysis and Ontology as Technology. In: POLI, R.; HEALY, M.; KAMEAS, A. (Org.). **Theory and applications of ontology**: computer applications, 1. ed. [S.l.]: Springer, 2010. 576 p.

RADEN, N. **Business Intelligence 2.0**: simpler, more accessible, inevitable. **Information week software**, 2007. Disponível em: <<http://www.intelligententerprise.com/showArticle.jhtml;jsessionId=KAK15UEUQBOGWQSNDLRCKH0CJUNN2JV N?articleID=19700261>>. Acesso em: 17 out. 2008.

RADFORD, A. et al. **Linguistics**: an introduction. 2. ed. Nova Iorque: Cambridge University Press, 2009. 450 p.

REITER, E.; DALE, R. Building applied natural language generation systems. **Natural Language Engineering**, Nova Iorque, v. 1, n. 3, p. 57-87, mar. 1997.

_____. Building natural-language generation systems. Nova Iorque: Cambridge University Press, 2000.

REITER, E.; SRIPADA, S. Acquiring correct knowledge for natural language generation. **Journal of Artificial Intelligence**, v. 18, n. 1, p. 491-516, 2003. Disponível em: <<http://www.aaai.org/Papers/JAIR/Vol18/JAIR-1813.pdf>>. Acesso em: 27 out. 2011.

SCAIFE, M. External cognition: how do graphical representations work? *International Journal of Human-Computer Studies*, Nova Iorque, v. 45, n. 2, p. 185-213, 1996.

SCHREIBER, G. et al. Knowledge engineering and management: the CommonKADS methodology. [S.l.]: The Mit Press, 2000. 455 p.

SELL, D. **Uma Arquitetura para Business Intelligence Baseada em Tecnologias Semânticas para Suporte a Aplicações Analíticas**. 2006.

210 f. Tese (Doutorado) – Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 2006.

SELL, D. et al. SBI: a semantic framework to support business intelligence. In: INTERNATIONAL WORKSHOP ON ONTOLOGY-SUPPORTED BUSINESS INTELLIGENCE, 1., 2008, Germany. **Proceedings...** Karlsruhe: ACM Digital Library, 2008. p. 1-11.

SELL, D. et al. Adding semantics to business intelligence: towards a smarter generation of analytical tools. In: MIRCEA, M. **Business Intelligence - Solution for Business Development**, 1. ed. Romania: Intech, 2012. p. 33-54.

SILVA, D. C. D. **Uma arquitetura de business intelligence para processamento analítico baseado em tecnologias semânticas e em linguagem natural**. 2011. 161 p. Dissertação (Mestrado) – Programa de Pós-graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 2011.

SILVA, E. L. da; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 3. ed. Florianópolis: Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina, 2001.

SIMPLENLG. **Java API for Natural Language Generation**. Disponível em: <<http://code.google.com/p/simplenlg/>>. Acesso em: 20 out. 2012.

SMITH, M. K.; WELTY, C.; MCGUINNESS, D. L. (Ed.). **OWL Web Ontology language guide**: W3C Recommendation, fev. 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-guide-20040210/>>. Acesso em: 20 ago. 2012.

STANFORD CENTER FOR BIOMEDICAL INFORMATICS RESEARCH. **Protégé**. Califórnia. Disponível em: <<http://protege.stanford.edu>>. Acesso em: 21 ago. 2012.

STERN, E.; APREA, C.; EBNER, H. G. Improving cross-content transfer in text processing by means of active graphical representation. **Learning and Instruction**, v. 13, n. 2, p. 191-203, 2003.

STEIL, A. V. **Estado da arte das definições de gestão do conhecimento e seus subsistemas**. Florianópolis: Instituto Stela, 2007.

STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge engineering: principles and methods. **Data & Knowledge Engineering**, v. 25, n. 1-2, p. 161-197, mar. 1998 [S.l.]:ACM, 1998.

TAKEUCHI, H.; NONAKA, I. Criação e dialética do conhecimento. In: _____. **Gestão do conhecimento**. Porto Alegre: Bookman, 2008. p. 19.

THE APACHE VELOCITY PROJECT. Disponível em: <<http://velocity.apache.org/engine/>>. Acesso em: 20 out. 2012.

TRAFTON, J. G. et al. Turning pictures into numbers: extracting and generating information from complex visualizations. *International Journal of Human-Computer Studies*, v. 53, n. 5, p. 827-850, 2000.

TURBAN, E. et al. Business intelligence: um enfoque gerencial para a inteligência do negócio. Porto Alegre: Bookman, 2009. 256 p.

W3C OWL Working Group. **OWL 2 Web Ontology language document overview (second edition)**: W3C recommendation, dez. 2012. Disponível em: < <http://www.w3.org/TR/owl2-overview/>>. Acesso em: 21 dez. 2012.

WALSH, J.P.; UNGSON, G. R. Organizational memory. **The Academy of Management Review**, v. 16, n. 1, p. 57-91, 1991.

WIIG, K. M. Knowledge management: an introduction and perspective. **The Journal of Knowledge Management**, v. 1, n. 1, p. 6-14, 1997.