RODRIGO DE OLIVEIRA

ANÁLISE DA UTILIZAÇÃO DO *FRAMEWORK* NO PROCESSO DE ETL PARA UM SISTEMA DE *DATA WAREHOUSE* BASEADO EM ONTOLOGIA E SOA

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

São Paulo

2015

RODRIGO DE OLIVEIRA

ANÁLISE DA UTILIZAÇÃO DO *FRAMEWORK* NO PROCESSO DE ETL PARA UM SISTEMA DE *DATA WAREHOUSE* BASEADO EM ONTOLOGIA E SOA

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof. Me. Marcel Luiz Garcia de Miranda

São Paulo

2015

Catalogação-na-publicação

Oliveira, Rodrigo
ANÁLISE DA UTILIZAÇÃO DO FRAMEWORK NO PROCESSO DE ETL PARA UM SISTEMA DE DATA WAREHOUSE BASEADO EM ONTOLOGIA E SOA / R. Oliveira -- São Paulo, 2015.
76 p.

Monografia (MBA em Tecnologia de Software) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1.BANCO DE DADOS 2.ARQUITETURA ORIENTADA A SERVIÇOS I.Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

DEDICATÓRIA

Dedico este trabalho aos meus pais, João de Oliveira (*in memoriam*) e Vera Lucia de Oliveira, à minha filha Nicolly Andrade de Oliveira e a minha esposa Gisele Lyrio Brandão de Oliveira, que tiveram uma presença importante nas conquistas como nas dificuldades encontradas no decorrer desse curso.

AGRADECIMENTOS

À minha esposa que Gisele Lyrio Brandão de Oliveira, por acreditar no meu potencial, por ter tido muita paciência e que sempre esteve ao meu lado me dando apoio para chegar até o final.

À minha mãe Vera Lucia de Oliveira por ter me apoiado para que eu continuasse em frente no curso sem desistir.

À Universidade de São Paulo por abrir o espaço para esse curso.

Ao PECE – Programa de Educação Continuada em Engenharia, pela qualidade do ensinamento e por ter contribuído para o meu aprendizado.

Ao meu orientador Prof. Me. Marcel Luiz Garcia de Miranda pela paciência, confiança e apoio durante a elaboração desse trabalho.

Aos meus companheiros de aula que contribuíram para que eu continuasse caminhando até o final deste curso.

Ao amigo Alexandre Vieira por colaborar com esse trabalho, tornando ele mais próximo de sua realidade.

RESUMO

Para manterem-se competitivas no mercado, as empresas necessitam de agilidade nas suas respostas e nas tomadas de decisões. Por isso, deve-se dar importância aos dados armazenados nos sistemas de *data warehouse*. Esse trabalho faz um apanhado da literatura e apresenta uma análise para solucionar problemas de heterogeneidade de dados e distribuição dos componentes de ETL (*Extract*, *Transform* e *Load*). São apresentados dois *frameworks* baseado em ontologia e SOA (S*ervice-Oriented Architecture)* para melhorar o processo de ETL e relatos de experiência da implantação de ambos os *frameworks* em diferentes situações. Baseado nesses estudos, é proposta uma estratégia para a aplicação desses *frameworks* em um processo de ETL.

Palavras chaves: ETL, Ontologia, SOA, *Business Intelligence*, *Data Warehouse*

ABSTRACT

To remain competitive in the market, companies need agility in their responses and in decision-making. Therefore, one should give importance to the data stored in the data warehouse systems. This work provides an overview of the literature and presents an analysis to solve heterogeneity of data and distribution problems of ETL components (Extract, Transform and Load). Two are presented frameworks based on ontology and SOA (Service-Oriented Architecture) to improve the ETL process and implementation experience reports of both frameworks in different situations. Based on these studies, it proposes a strategy for the implementation of these frameworks in a process of ETL.

Keyboards: ETL, Ontology, SOA, Business Intelligence, Data Warehouse

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CSV	*Comma-separated values*
BI	*Business Intelligence*
BSC	*Balenced ScoreCard*
DW	*Data warehouse*
EDW	*Enterprise Data Warehouse*
EJB	*Enterprise JavaBeans*
ERP	*Enterprise Resource Planning*
ETL	*Extract Transform Load*
ETL	Extração Transformação Carga
FVG	Fundação Getúlio Vargas
GAV	*Global as View*
LAV	*Local as View*
LUCT	*Limkokwing* universidade de tecnologia criativa
MW	*Megawatt*
OLAP	*Online Analytical Processing*
OLTP	*Online Transaction Processing*
OWL	*Web Ontology Language*
PEC	*Palestine Electric Company*
PIT	Profissionais da tecnologia de informação
RDF	*Resource Description Framework*
SIG	Sistema de Informação Gerencial
SOA	*Service-oriented architecture*
SSIS	*Microsoft SQL Server Integration Services*
TXT	Arquivo de texto
W3C	*Word Wide Web Consortium*
XML	*eXtensible Markup Language*
XLS	Arquivo do formato padrão do Excel

SUMÁRIO

1. INTRODUÇÃO

Esse trabalho propõe-se a apresentar os princípios fundamentais dos processos de ETL (*Extract*, *Transform* e *Load*) em um sistema de *data warehouse*. Além disso, desenvolve a atual importância da utilização da tecnologia de *business intelligence* de maneira estratégica nas organizações. Para isso, o trabalho foi conduzido por meio de argumentos teóricos e estudos de caso, verificados na literatura, aplicados em ambientes reais. Através das referências bibliográficas, são apresentadas duas abordagens baseado em *frameworks*, mostrando suas principais características para apoiar no desenvolvimento dos processos de ETL.

1.1. Motivações

A tecnologia da informação tem um papel extremamente fundamental nas organizações. Diante disso, os sistemas de *Business Intelligence* utilizam os dados disponíveis nas organizações para disponibilizar informações relevantes para a tomada de decisão. Tradicionalmente, os sistemas de *business intelligence* são compostos por quatro importantes tecnologias, são elas: *data warehouse*, análise de negócio, gestão de desempenho e interface com o usuário (TURBAN, SHARDA, *et al.*, 2009).

O principal benefício de um sistema de *business intelligence* em uma empresa é a capacidade de fornecer informações precisas quando necessário, incluindo uma visão em tempo real do desempenho corporativo geral e de suas partes individuais. Essas informações são necessárias para todos os tipos de decisão, para o planejamento estratégico e mesmo para a sobrevivência da organização (TURBAN, SHARDA, *et al.*, 2009).

Os resultados de uma pesquisa em 510 corporações indicam os benefícios de um sistema de *business intelligence* conforme a visão dos participantes (ECKERSON, 2003):

- Economia em tempo (61%)
- Versão única da verdade (59%)
- Melhores estratégicas e planos (57%)
- Melhores decisões táticas (56%)

- Processos mais eficientes (55%)
- Economia dos custos (37%)

Thompson (2004) relatou, a partir de um estudo, que os maiores benefícios são:

- Geração de relatórios mais precisos e rápidos (81%)
- Melhor tomada de decisão (78%)
- Melhor serviço ao cliente (56%)
- Maior receita (49%)

O *data warehouse* é o lugar onde os dados são armazenados para oferecer suporte a tomada de decisões gerenciais. Nele há uma variedade de dados que representam as reais condições da organização em um determinado ponto no tempo. O objetivo do conceito era a criação de um ambiente de banco de dados que estivesse sempre disponível para consultas e contivesse todas as informações dos sistemas transacionais, incluindo os dados históricos. No entanto, esse ambiente seria reorganizado e estruturado de forma a oferecer rapidez e eficiência nas consultas, análises e apoio a decisão (TURBAN, SHARDA, *et al.*, 2009).

Um dos elementos fundamentais para a construção do ambiente de *data warehouse* é o processo de ETL (*Extract*, *Transform* e *Load*). Um correto sistema de ETL extrai os dados de diversas origens de dados, impõe padrões de qualidade e consistência dos dados, de forma que as fontes separadas possam ser utilizadas em conjunto, e finalmente fornece os dados em um formato de apresentação para que os desenvolvedores de aplicação possam criar aplicativos para os usuários finais tomarem as decisões (KIMBALL e CASERTA, 2004).

Um sistema de ETL acrescenta valor significativo aos dados, ou seja, é muito mais do que um “duto” para a obtenção de dados de sistemas de origem para o *data warehouse*. Especificamente, o sistema de ETL (KIMBALL e CASERTA, 2004):

- Remove erros e corrige os dados;
- Fornece medidas de confiança nos dados;
- Captura o fluxo dos dados transacionais;
- Ajusta os dados de múltiplas fontes para o uso em conjunto;
- Estrutura os dados a serem utilizados por ferramentas de usuário final.

1.2. Objetivo

O objetivo desse trabalho é apresentar o conceito de um processo de ETL no ciclo de vida de um sistema de *data warehouse*. Para isso, foi analisado a criação de um modelo de ETL com a utilização de dois *frameworks* baseado em ontologia e *service-oriented architecture*, para identificar a viabilidade do desenvolvimento de um sistema de *data warehouse*.

1.3. Justificativas

Tendo em vista o cenário atual das áreas de negócios, percebemos que é um diferencial as empresas disporem de um sistema de *data warehouse* para a análise da informação para tomada de decisão. Sabendo que é constante o tráfego de informações que circulam nas empresas, é imprescindível que essas informações sejam extraídas, padronizadas e disponibilizadas no sistema em um menor tempo possível. O processo de ETL atende essas características, sendo aplicado em uma necessidade específica de um sistema de *data warehouse*.

A proposta é apresentar a contribuição do *framework* quando utilizado para a construção de um processo de ETL, esclarecendo o conceito da tecnologia aplicando em um sistema de *data warehouse*.

1.4. Estrutura do Trabalho

Este trabalho está estruturado em 5 capítulos, conforme segue:

O Capítulo 1 – INTRODUÇÃO apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

O Capítulo 2 – BUSINESS INTELLIGENCE (BI) apresenta brevemente a introdução da tecnologia de BI e seus principais componentes para a aplicação do conceito.

O Capítulo 3 – DATA WAREHOUSE apresenta a importância e características sobre o ambiente de *data warehouse*, bem como sua arquitetura e ciclo de vida do sistema.

O Capítulo 4 – PROCESSOS DE ETL apresenta a importância da fase de ETL em um projeto de *data warehouse*, bem como a análise de dois *frameworks* que são utilizados para solucionar problemas no processo de ETL.

O Capítulo 5 – CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS – Este capítulo apresenta as considerações finais do trabalho e sugestões de trabalhos futuros.

2. *BUSINESS INTELLIGENCE*

Este capítulo apresenta brevemente a importância da tecnologia de *business intelligence* e os seus componentes nas organizações para tomarem ações estratégicas. O livro *Business Intelligence* - Um enfoque gerencial para a inteligência do negócio foi utilizado como base (TURBAN, SHARDA, *et al.*, 2009).

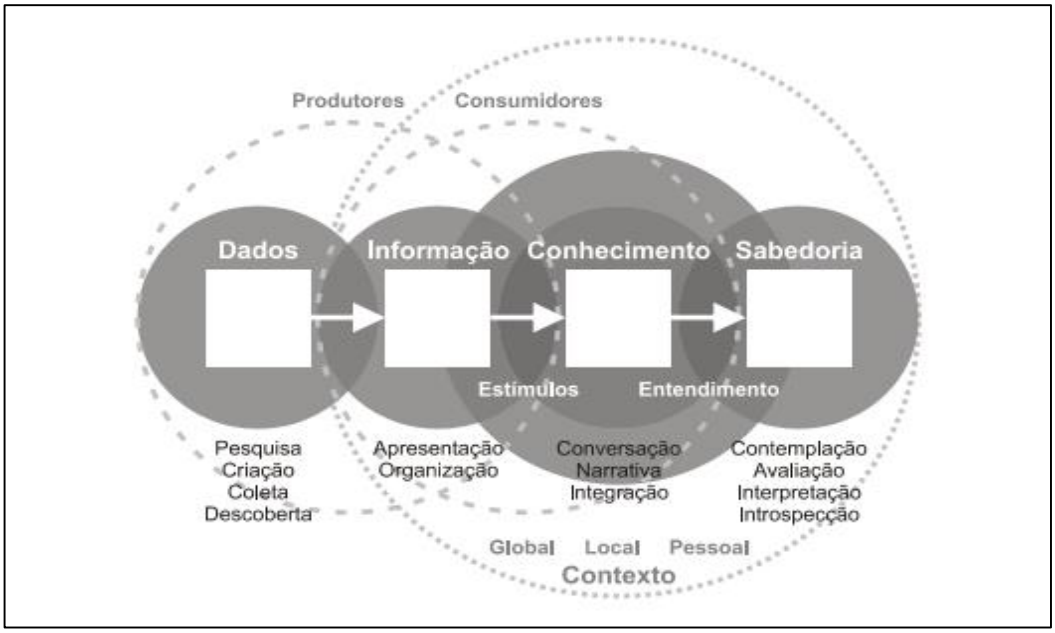
2.1. A Tecnologia de *Business Intelligence*

Para manterem-se competitivas no mercado, as empresas necessitam de agilidade nas suas respostas e nas tomadas de decisões, sejam elas: estratégicas, táticas ou operacionais (TURBAN, SHARDA, *et al.*, 2009). Para isso, a informação é a chave fundamental para essas empresas se anteciparem às tendências de mercado, a inovação, a transformação das estratégias em ações e orientar seus recursos para permanecerem à frente de sua competição (BALLARD, FARRELL, *et al.*, 2006).

Deve-se dar importância ao volume de dados oportunos e relevantes armazenados pela empresa (TURBAN, SHARDA, *et al.*, 2009). Com um sistema computacional, os dados passarão por um processo de captura, consolidação, organização e distribuição para serem analisados e explorados, de forma rápida e fácil. Isso é o objetivo da tecnologia que é conhecida como *Business Intelligence* (BI) (BALLARD, FARRELL, *et al.*, 2006). O termo *Business Intelligence* foi consumado pelo *Gartner Group* (consultoria fundada em 1979 que desenvolve tecnologias relacionadas à introspecção necessária para seus clientes tomarem suas decisões diárias) na década de 90, porém, o conceito originou-se nos anos 70, nos sistemas de geração de relatórios de Sistema de Informação Gerencial - SIG (TURBAN, SHARDA, *et al.*, 2009).

O processo de BI baseia-se na transformação dos dados em informações e posteriormente em conhecimento (TURBAN, SHARDA, *et al.*, 2009). Como mostra a figura 1.

Figura 1. Representação do processo continuo de entendimento



Fonte: (SHEDROFF, 1999)

O BI pode ser utilizado como um sistema computacional para ajudar as empresas à tomarem suas decisões gerenciais (TURBAN, SHARDA, *et al.*, 2009), bem como apoiar nos momentos críticos, tais como encontrar melhores oportunidades de crescimento e descobrir as principais áreas de lucros e perdas (BALLARD, FARRELL, *et al.*, 2006).

2.2. Aplicação e componentes de um sistema de *Business Intelligence*

Para ilustrar como o BI pode auxiliar a gestão de uma empresa, apresenta-se o seguinte exemplo:

No congresso internacional de qualidade em serviços e sistemas de saúde (Qualihosp) promovido pela Fundação Getúlio Vargas - FGV, foi publicado em 2013 uma pesquisa relacionada a análise de custo das reinternações hospitalares. Percebeu-se que as reinternações hospitalares que aconteciam nos estados norte-americanos eram de 20% dos pacientes internados, que correspondiam a um custo aproximado de 17 bilhões de dólares para o sistema de saúde americano. A pesquisa mostra que muitos desses casos poderiam ter sido evitados e estão relacionados à qualidade da assistência prestada (VIEIRA, GUIMARÃES e CAETANO, 2013).

Através de um sistema de *business intelligence*, foi realizado um estudo com base no banco de dados de diferentes operadoras de planos de saúde. Essas informações estavam distribuídas na utilização de beneficiários titulares e dependentes de empresas privadas que oferecem o benefício aos seus colaboradores. Entre essa população, foram avaliadas 54.368 internações que estavam classificadas como clínica ou cirúrgica e considerado como reintenações hospitalares aquelas que ocorreram em um período inferior a 30 dias (VIEIRA, GUIMARÃES e CAETANO, 2013).

Em relação a distribuição das 54.368 internações avaliadas, houve uma pequena predominância associada ao sexo masculino contemplando 52.4% das reinternações, comparado com o sexo feminino que ficou com 47.6%. Além disso, a distribuição realizada no tipo de internação foi de 53.8% para os procedimentos cirúrgicos, comparado com os 46.2% dos procedimentos clínicos. No entanto, ocorreu uma elevação de 10% do custo médio das internações de procedimentos cirúrgicos em relação aos procedimentos clínicos (VIEIRA, GUIMARÃES e CAETANO, 2013).

Foram identificados 6.832 reinternação que corresponde à 12.6% do total de internações, em que 65.2% eram relacionadas a procedimentos clínicos e 34.8% para procedimentos cirúrgicos. A frequência das reinternações clínicas (17.7%) foram maiores que as cirúrgicas (8.1%) em todas as idades. O aumento da frequência das reinternações cirúrgicas foi proporcional ao aumento das idades que variam em torno de 4.6% nas faixas até 18 anos a 14.0% para os pacientes acima de 59 anos. As reinternações clínicas tiveram um comportamento semelhante, porém a faixa etária com maior taxa está entre 49 a 53 anos com 27.0%, a menor taxa de reinternações clínicas ocorreram em pacientes com 18 anos ou menos (12.0%) (VIEIRA, GUIMARÃES e CAETANO, 2013).

Do custo total com internações, 16.4% foram considerados relacionados a alguma reinternação. Quando comparado ao custo médio das internações, as reinternações foram 39.0% mais caras. As reinternações clínicas, em média, são 60.0% mais caras que as internações e as cirúrgicas, em média, também têm o seu custo elevado, que chega a 11% em relação a própria internação (VIEIRA, GUIMARÃES e CAETANO, 2013).

A utilização de uma ferramenta para analisar os dados é de grande importância para apoiar na avaliação de qualidade dos sistemas de saúde. Nessa pesquisa foram analisadas informações de um período de doze anos (2001 a 2012) em uma população ativa. Além disso, o grande volume de internações hospitalares permite concluir que as reinternações são mais frequentes e mais prevalentes para tratamentos clínicos e cirúrgicos e estão totalmente relacionadas à qualidade do cuidado prestado no período de internação no primeiro mês após a alta hospitalar (VIEIRA, GUIMARÃES e CAETANO, 2013).

Um sistema de BI tem como seus principais objetivos fornecer rapidamente o acesso a informações, proporcionar a manipulação dos dados e promover aos gestores a capacidade de exploração e análise para apoia-los na tomada de decisão, ou seja, os gestores precisam das informações certas na hora certa e no lugar certo (TURBAN, SHARDA, *et al.*, 2009).

Para planejar e executar o desenvolvimento de um sistema de BI de maneira organizada é proposta uma arquitetura do projeto que é composta por quatro grandes componentes:

- ***Data warehouse*** – é um repositório de dados que tem o objetivo de fornecer as informações para a tomada de decisão (TURBAN, SHARDA, *et al.*, 2009).

- **Análise de negócio** – corresponde a um conjunto de ferramentas utilizadas para manipular, minerar e analisar de maneira avançada os dados de um *data warehouse*, incluindo nessa etapa o processamento analítico on-line OLAP (TURBAN, SHARDA, *et al.*, 2009).

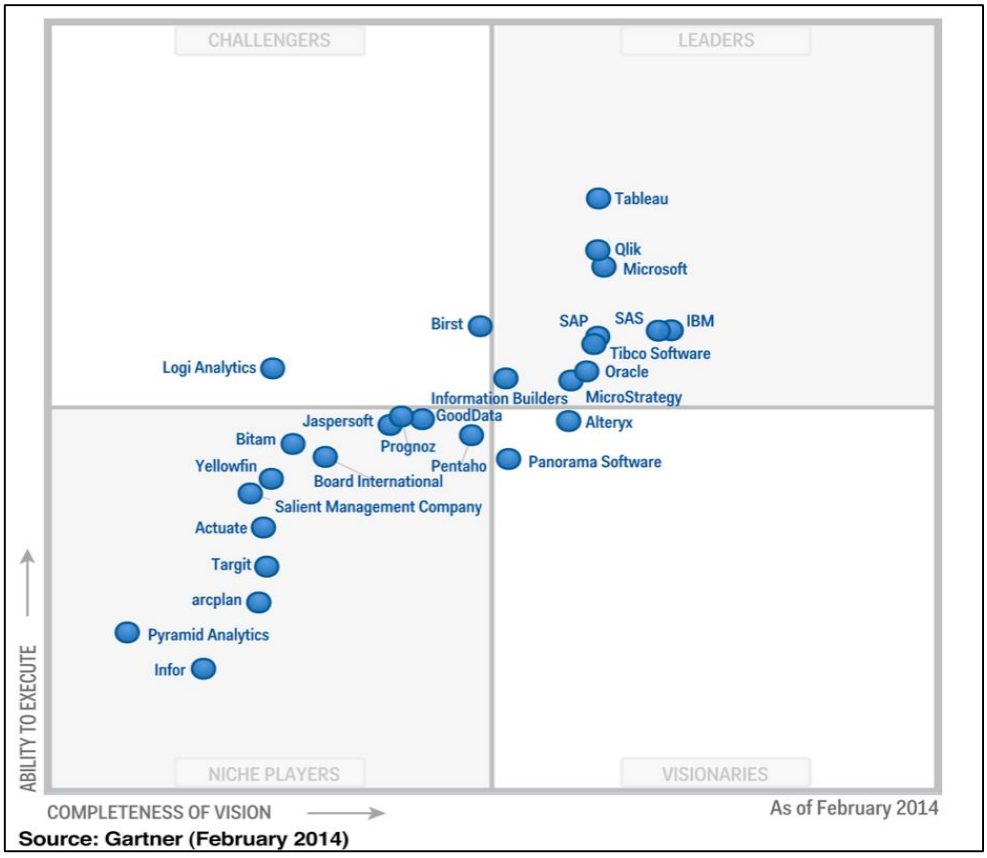
- **Gestão de desempenho do negócio** – conjunto integrado de processos para aperfeiçoar o desempenho dos negócios, direcionado com o uso do *Balanced Scorecard* (BSC – ferramenta de planejamento estratégico desenvolvida pelos professores Robert Kaplan e David Norton) (TURBAN, SHARDA, *et al.*, 2009).

- **Interface com os usuários** – são exibições visuais usadas para monitorar o desempenho do negócio, essas visões proporcionam as informações

consolidadas e organizadas em uma ou mais telas para serem compreendidas rapidamente pelos usuários (TURBAN, SHARDA, *et al.*, 2009).

Existem diversas ferramentas que se propõem a atender a plataforma analítica e de *business intelligence*. A consultoria *Gartner Group* realiza uma pesquisa dessas ferramentas e analisa com o propósito de identificar as inovações que orientam o mercado e também fornece o quadrante mágico que mostra a relação dessas ferramentas (GARTNER, 2014), conforme a figura 2.

Figura 2. Quadrante mágico de *Gartner* para plataforma analítica e *business intelligence*



Fonte: (GARTNER, 2014)

2.3. Considerações do capítulo

A tecnologia de *Business Intelligence* foi criada para contribuir desde a tomada de decisões operacionais até as estratégicas nas organizações. Tendo em vista que a informação é peça fundamental para o sucesso da implantação da tecnologia.

O *data warehouse*, análise do negócio, gestão de desempenho e a interface do usuário são componentes fundamentais para o sucesso da implantação da tecnologia de *business intelligence* nas organizações. Além disso, existem sofisticadas ferramentas analíticas para dar suporte na análise de dados.

3. DATA WAREHOUSE

Este capítulo apresenta de forma breve a importância e as características de uma aplicação de *data warehouse*, bem como sua arquitetura e ciclo de vida do sistema. O livro *Building the Data Warehouse* oferece a base para a construção desse capitulo (INMON, 2005).

3.1. Características e elementos que compõem um sistema de *data warehouse*

O *data warehouse* ou armazém de dados como também é conhecido, é um sistema de repositório de dados utilizado para armazenar e organizar informações de múltiplas fontes de dados de forma consolidada, para facilitar as consultas, a criação de relatórios de tendência e possibilitar a análise estratégica de grandes volumes de dados (DATE, 2004). Esse sistema deve ter condições de extrair, limpar, padronizar e entregar os dados de origem para um armazenamento de dados dimensional, e em seguida, suportar e implementar as consultas e análises para proporcionar a tomada de decisão (KIMBALL e CASERTA, 2004). O *data warehouse* possui um conjunto de características, tais como: dados orientados a assuntos, variação de tempo, integração, acessíveis aos usuários, dados não voláteis (DATE, 2004).

- **Dados orientados a assunto:** é a organização dos dados de maneira que poderá enfatizar um determinado negócio da empresa;

- **Variação de tempo:** permite realizar uma comparação de análises de dados empresariais com relação a vários períodos de tempo;

- **Integração:** é o processo de organização das fontes de dados heterogêneas para serem armazenadas em uma única base de dados;

- **Acessíveis aos usuários:** consiste na entrega da informação através de painéis de monitoramento ou *dashboard* para que possa ser analisado;

- **Dados não voláteis:** são dados que não devem sofrer alteração, a não ser que seja para realizar alguma correção de carga de dados.

Certamente a origem dos sistemas de apoio a decisão e do *data warehouse* iniciou-se nos anos 60 e vem tendo uma grande evolução por causa do amadurecimento dos sistemas empresariais que armazenam diariamente grandes volumes de dados e que precisam ser estudados de maneira rápida (INMON, 2005).

Obviamente existem diversos caminhos para a implantação de um sistema de *data warehouse*, mas a indicação é que deve ser de maneira interativa, através de etapas, de modo ordenado, pois será mais factível o projeto chegar ao sucesso e terá um custo menor (INMON, 2005).

Um fator crítico na implantação de um sistema de *data warehouse* é o nível de granularidade do dado. Quanto mais dados detalhados menor será o nível de granularidade e menos versátil será a consulta, além de aumentar o volume de dados que será armazenado no *data warehouse* (INMON, 2005). O nível de granularidade depende do tipo de negócio que está sendo aplicado o sistema. Pode-se levar em consideração que se deve construir os objetos com o menor o nível de granularidade possível, lembrando que um menor nível de granularidade não significa encontrar maior quantidade de dados detalhados (KIMBALL e CASERTA, 2004). Com isso, é importante um estudo aprofundado sobre a necessidade de visualização dos dados da empresa para encontrar um nível adequado de granularidade.

A seguir apresentam-se os principais elementos para a construção de um sistema de *data warehouse*:

- **Fonte de dados:** é um conjunto de dados heterogêneos que podem estar armazenados nos sistemas transacionais da empresa e serem integrados em um *data warehouse***,** esses dados estão em subdivisões chamadas de áreas temáticas (INMON, 2005). A integração desses dados heterogêneos é um grande desafio para os desenvolvedores, pois para incluir novos assuntos, deve-se confrontar com o escopo do *data warehouse* (KIMBALL e CASERTA, 2004).

- **Área do *Data Stage*:** é onde ficam armazenados todos os processos relacionados à extração de dados heterogêneos, transformação, padronização e carga de dados no repositório do *data warehouse*. Esses dados não são apresentados ao usuário final (KIMBALL e CASERTA, 2004).

- **Apresentação:** é a área onde os dados são armazenados, organizados e disponibilizados para consultas pelo usuário final. Essa apresentação pode ser modelada em banco de dados relacional, que se refere, por exemplo, no modelo estrela, ou pode ser apresentada baseada no banco de dados multidimensional ou processamento analítico *on-line* (OLAP), sendo assim, são armazenadas em CUBOS (KIMBALL e CASERTA, 2004).

- ***Data Mart*:** são seguimentos ou partes de um *data warehouse*, que normalmente são separados por negócio ou assuntos determinados pelo usuário (TURBAN, SHARDA, *et al.*, 2009).

- ***Data Mining*:** através de ferramentas, a mineração de dados consiste na busca no *data warehourse* por padrões, afirmações, hipóteses e regras pré-definidas que dificilmente seriam encontradas por consultas feitas por usuários (INMON, 2005).

- **Ferramentas de acesso a dados:** são as ferramentas de permitem o acesso a camada de apresentação do *data warehouse* pelos usuários de negócios para realizarem suas consultas e qualquer outro tipo de análise (KIMBALL e CASERTA, 2004).

3.2. Arquitetura e ciclo de vida de um sistema de *data warehouse*

Nessa seção apresenta-se a importância da arquitetura escolhida para o ambiente de *data warehouse* e o planejamento do ciclo de vida que compõe o sistema de *data warehouse*.

3.2.1. Arquitetura

A definição de uma arquitetura dará suporte na estrutura da modelagem do dado e no gerenciamento do *data warehouse* e dos *data marts* (BALLARD, FARRELL, *et al.*, 2006). Existem três abordagens básicas para definir a arquitetura do *data warehouse*, quais sejam: *enterprise data warehouse* (EDW), *independent data marts*, *dependent data marts.*

- ***Enterprise data warehouse***: é um ambiente em que todos os dados do *data warehouse* estão logicamente centralizados, essa arquitetura irá apoiar uma grande parte da exigência de negócio por tratar-se de um armazenamento de dado mais integrado. Uma das vantagens desse cenário é o gerenciamento de um único objeto integrado (BALLARD, FARRELL, *et al.*, 2006).

- ***Independent data marts****:* são subconjuntos de um *data warehouse* controlados de maneira independente, que normalmente tratam assuntos de uma determinada unidade de negócio e que não compartilham informações entre eles (BALLARD, FARRELL, *et al.*, 2006).

- ***Dependent data marts****:* diferente do *data marts* independente, os *data marts* dependentes podem compartilhar e confrontar dados entre si, pois eles são interligados para fornecer uma visão global dos dados (BALLARD, FARRELL, *et al.*, 2006).

Esses modelos de arquiteturas também permitem trabalhar em combinações, ou seja, podendo utilizar o *enterprise data warehouse* (EDW) com os *data marts* dependentes, assim o *data marts* recebe o dado do EDW, não precisando ir busca-lo no *data stage* (BALLARD, FARRELL, *et al.*, 2006).

3.2.2. Modelo Dimensional

O modelo dimensional é muito popular no *data warehouse* e também é conhecido como *start schema*. O objetivo dele é melhorar o desempenho de grandes consultas (BALLARD, FARRELL, *et al.*, 2006). Basicamente esse tipo de modelo é separado em dois tipos de tabelas com diferentes características: fato e dimensão.

Tabela Fato: é onde ficam armazenadas todas as medidas (quantidade de produtos, valores de produtos e etc.) resultantes de um processo de negócio, é a principal tabela no modelo *start schema* que fica rodeado pelas tabelas de dimensões (KIMBALL e CASERTA, 2004).

Tabela Dimensão: são tabelas que contém descrições textuais ou variações de período do negócio e servem para segmentar ou sumarizar os relatórios por um

determinado assunto. São objetos integrados fundamentais para completarem uma tabela fato (KIMBALL e CASERTA, 2004).

3.2.3. O ciclo de vida do projeto de *data warehouse*

O ciclo de vida do desenvolvimento de um sistema de *data warehouse* é muito diferente do que uma aplicação no ambiente operacional, que por padrão é voltado por requisitos. Quando a finalidade é a criação de sistemas transacionais, deve-se primeiro entender os requisitos, por padrão, essa técnica é conhecida como a abordagem cascata. Pois as diferentes atividades são especificadas e uma atividade após a sua conclusão direciona para a próxima atividade e aciona o seu início. No caso de sistema de *data warehouse* inicia-se pelos dados, uma vez que eles estão disponíveis, são integrados e em seguida devidamente testados para verificar se existe algum tipo de polarização nos dados. Caso exista, são escritos programas para avaliar os dados, os resultados são analisados e finalmente os requisitos do sistema são compreendidos e caso haja a necessidade, são feitos ajustes para a concepção do sistema e o ciclo começa novamente para um novo conjunto de dados. Devido aos constantes reajustes no ciclo de vida de desenvolvimento de diferentes tipos de dados a abordagem da metodologia mais indicada é a espiral (INMON, 2005).

O modelo de ciclo de vida do desenvolvimento do sistema no ambiente do *data warehouse* é quase o oposto de um modelo clássico (INMON, 2005), conforme tabela 1:

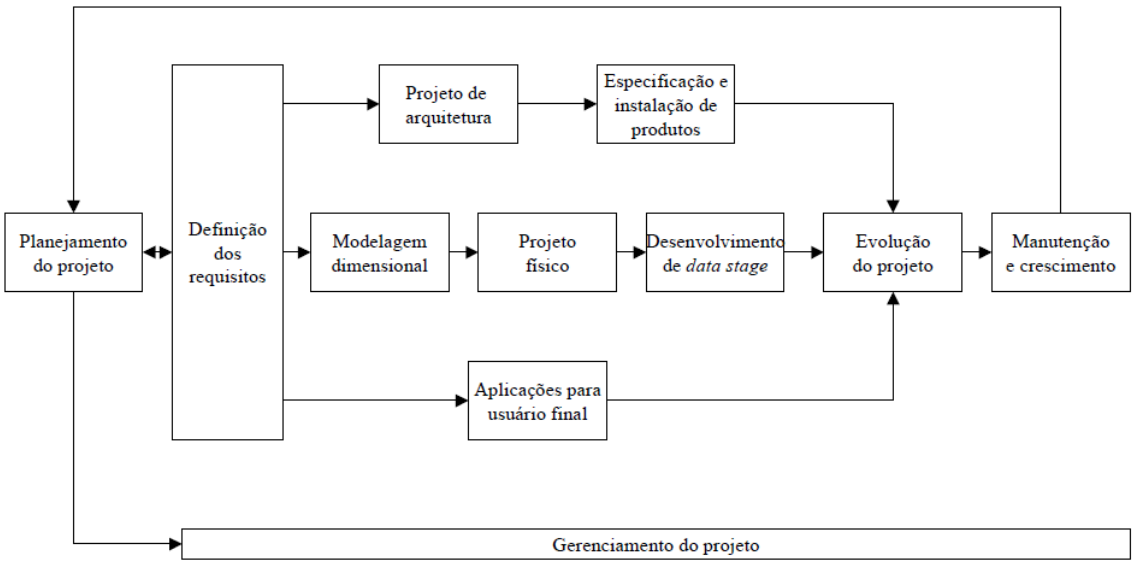
Tabela 1. O ciclo de vida de desenvolvimento do sistema para o ambiente de *data warehouse* é quase exatamente o oposto do SDLC clássico

Modelo Clássico	Modelo do *data warehouse*
• Levantamento de requisitos • Análise • Projeto • Desenvolvimento • Teste • Integração • Implementação	• Implementação do *data warehouse* • Integração dos dados • Teste de polarização dos dados • Desenvolvimento de programas • Desenho do sistema de suporte a decisão • Análise de resultado • Entendimento dos requisitos

Fonte: (INMON, 2005)

Segundo Kimball (2008), a abordagem global do ciclo de vida da implementação do *data warehouse* é composta pelas seguintes tarefas:

Figura 3. O ciclo de vida do negócio dimensional



Fonte: (KIMBALL, REEVES, *et al.*, 2008)

- **Planejamento do projeto** – É o início do ciclo de vida do projeto, pois aborda a definição e delimitação do âmbito do projeto de *data warehouse*, incluindo a avaliação de prontidão e justificativa comercial. Essas tarefas são críticas devido à alta visibilidade e custo associados com a maioria dos projetos de *data warehouse*. O resultado do planejamento é a identificação de todas as tarefas associadas com o ciclo de vida da implementação do *data warehouse* e a observação de todas as partes envolvidas. Os arquitetos do *data warehouse* devem entender os fatores chaves que impulsionam o negócio para determinar de forma eficaz os requisitos e negócio e traduzi-los em considerações de técnicas para a implementação (KIMBALL, REEVES, *et al.*, 2008).

- **Modelagem dimensional**: para o desenvolvimento do modelo dimensional é indicado a criação de uma matriz que representa os principais processos de negócio e suas dimensionalidades, a matriz serve como um modelo para garantir que o *data warehouse* seja extensível em toda a organização ao longo do tempo. Juntando essa análise com o entendimento do negócio, é possível iniciar o desenvolvimento do modelo dimensional. Esse modelo tem como objetivo identificar a granularidade das tabelas fatos, dimensões associadas, atributos e caminhos de pesquisas hierárquicas e em tabelas fatos. O projeto do banco de dados lógico é contemplado com estruturas de tabelas adequadas e relacionadas com chaves primárias e estrangeiras. Esse conjunto de atividades termina com o desenvolvimento do mapeamento de dados da fonte de dados do destino (KIMBALL, REEVES, *et al.*, 2008).

- **Projeto físico**: o projeto do banco de dado físico concentra-se na definição das estruturas físicas necessárias para apoiar o projeto lógico da base de dados. Elementos principais desse processo incluem os padrões de nomenclatura, configuração do ambiente de banco de dados, indexação e particionamento de tabelas do banco de dados. Essas são as estratégias preliminares que contemplam esse processo (KIMBALL, REEVES, *et al.*, 2008).

- **Projeto e desenvolvimento da área de *staging*:** Tipicamente o processo de projeto e desenvolvimento da área de *staging* é a tarefa do projeto de *data warehouse* mais subestimada. Existem três etapas principais na fase de preparação de dados: extração, transformação e carga. O processo de extração expõe questões de qualidade de dados que foram gravadas dentro dos sistemas transacionais, isso impacta significativamente na credibilidade do projeto de *data warehouse*. Para solucionar esses problemas de qualidade de dados é desenvolvida a área de *staging* (KIMBALL, REEVES, *et al.*, 2008).

- **Projeto de arquitetura*:*** Ambientes de *data warehouse* requerem a integração com diversas tecnologias, o desenho técnico de arquitetura estabelece uma visão geral do *framework*. Para isso, deve-se considerar três principais fatores: seus requisitos de negócio, ambiente técnico atual e as instruções técnicas planejadas (KIMBALL, REEVES, *et al.*, 2008).

- **Especificação e instalação de produtos:** Após a utilização do desenho técnico de arquitetura como um *framework*, componentes específicos, tais como: a plataforma de *hardware*, sistema de gerenciamento de banco de dados e as ferramentas de teste de dados, terão que ser selecionadas a avaliadas. Um processo de avaliação técnica padrão é definido juntamente com fatores específicos de cada componente, em seguida, eles são instalados e exaustivamente testados para garantir uma integração adequada dentro do ambiente do *data warehouse* (KIMBALL, REEVES, *et al.*, 2008).

- **Aplicações para o usuário final:** a recomendação é definir um conjunto de aplicativos padrões para os usuários finais, mesmo porque nem todos os usuários de negócio precisam de acesso *ad-hoc* para o *data warehouse.* Especificação dos aplicativos precisam descrever o modelo dos relatórios, os parâmetros para orientar a utilização e os cálculos necessários. Esse documento garante que a equipe de desenvolvimento e os usuários de negócio tenham um entendimento comum sobre as aplicações a serem entregues.

Com a especificação da aplicação, o desenvolvimento de aplicativos de usuário final envolve a configuração da ferramenta de metadados e construção dos

relatórios especificados. Idealmente esses aplicativos são construídos usando uma ferramenta de acesso de dados avançada, que proporciona ganhos significativos de produtividade para a equipe de desenvolvimento do aplicativo. Além disso, ele oferece um mecanismo poderoso para os usuários de negócio modificarem facilmente seus relatórios existentes (KIMBALL, REEVES, *et al.*, 2008).

- **Evolução do projeto**: O processo de evolução representa a conversão da tecnologia, dos dados e dos aplicativos para a utilização dos usuários de negócio, é necessário um extenso planejamento para garantir que todas essas peças se encaixam corretamente. Além disso, deve ser estabelecido um suporte aos usuários e as estratégias de comunicação ou *feedback* antes de qualquer usuário de negócio ter acesso ao *data warehouse* (KIMBALL, REEVES, *et al.*, 2008)

- **Manutenção e Crescimento:** O excesso de trabalho permanece após a implantação inicial do *data warehouse*, os desenvolvedores precisam continuar focados em seus usuários de negócio, fornecendo-lhes apoio em dúvidas que surgem com a utilização do *data warehouse*. O *data warehouse* deve evoluir e crescer, ao contrário de iniciativas de desenvolvimento de sistemas tradicionais, a mudança deve ser vista como um sinal de sucesso e não de falha. Processos de priorização devem ser estabelecidos para lidar com essas demandas dos usuários de negócio para a contínua evolução e crescimento, após a identificação das prioridades do projeto, volta-se para o início do ciclo de vida, alavancando e desenvolvendo o que já foi estabelecido no ambiente do *data warehouse*, com foco nas novas exigências (KIMBALL, REEVES, *et al.*, 2008).

- **Gerenciamento de Projeto:** O gerenciamento de projeto garante que todas as atividades do ciclo de vida do projeto permanecem no rastro e em sincronia. Como mostrado na figura 3, as atividades de gerenciamento de projeto ocorrem durante todo o ciclo de vida. Essas atividades concentram-se nos seguintes *status:* monitoramento das atividades, acompanhamento de

problemas e controle de mudanças para preservar o limite do escopo. Finalmente, o gerenciamento de projeto inclui o desenvolvimento de um plano de comunicação abrangente que aborda tanto a organização de sistemas como a área de negócio. Essa comunicação é extremamente crítica para as expectativas de gestão e o gerenciamento das expectativas é absolutamente fundamental para alcançar os objetivos do *data warehouse* (KIMBALL, REEVES, *et al.*, 2008).

3.3. Considerações do capítulo

Este capítulo apresentou a importância da construção com planejamento do *data warehouse*. Foi discutido que o *data warehouse* possibilita a análise de grandes volumes de dados coletados de diversas fontes de dados, oferece apoio nas tomadas de decisões presentes e na previsão de eventos futuros. Com o acesso rápido às informações organizacionais, o *data warehouse* proporciona uma vantagem competitiva das empresas no mercado.

O próximo capítulo apresenta com detalhes o processo de integração que ocorre na construção de um ambiente de *data warehouse.*

4. PROCESSOS DE ETL

Esse capítulo apresenta a importância do processo de ETL em um sistema de *data warehouse*. Além disso, mostra dois *frameworks* utilizados para desenvolver a tarefa de ETL. Como principais bases para esse capitulo, foram utilizados: O livro de *The Data Warehouse* ETL *Toolkit* (KIMBALL e CASERTA, 2004) , o artigo *A Framework for Interoperable Distributed* ETL *Components Based on* SOA (AWAD e ABDULLAH, 2010) e o artigo *A Framework Model Study for Ontology-driven* ETL *Processes* (ZHANG e WANG, 2008).

4.1. Integração de dados

O processo de extração, transformação e carga de dados (ETL) é parte fundamental do sistema de *data warehouse* (DW), consome cerca de 70% dos recursos necessários para a implantação e manutenção do DW, mas também existem outros componentes importantes para a conclusão do projeto (KIMBALL e CASERTA, 2004). O sistema de *data warehouse* consolida dados de diversas fontes. Um sistema de ETL deve ser capaz de realizar essa atividade o que aumenta os elementos de complexidade, tais como: baixa qualidade e ausência dos dados adequados, falta de documentação de sistemas transacionais, fonte de dados redundantes e conflitantes (TURBAN, SHARDA, *et al.*, 2009).

Devido à complexidade e a alta curva de aprendizagem ao longo desses processos de ETL, muitas organizações preferem dirigir-se para o desenvolvimento *in-house* para executar tarefas de carga, limpeza e padronização de dados (VASSILIADIS, SIMITSIS e SKIADOPOULOS, 2002). O fluxo do processo de ETL destina-se à realização da extração de dados de uma ou mais fontes heterogêneas, limpa-las, padroniza-las e entrega-las em um sistema de *data warehouse*.

- **Extração:** é a primeira etapa do processo de integração, pois são extraídos os dados de diversas bases e formatos. Normalmente essas bases são de sistemas transacionais (OLTP – Processamento de transação em tempo real), arquivos em sistemas de diferentes plataformas de *hardwares* e protocolos de comunicação, sistemas legados em diversos formatos, como: TXT, CSV, XLS, etc. Antes de iniciar qualquer etapa de extração, é importante ter um documento que ilustra o mapeamento entre a relação dos campos da

base de dados de origem para os campos de destino (KIMBALL e CASERTA, 2004).

- **Transformação:** é a principal etapa no fluxo do processo de ETL, pois realiza todas as mudanças e padronizações necessárias nos dados que serão analisados. As outras etapas de extração e entrega são necessárias, mas são apenas seleção e carga de dados nos seus respectivos repositórios (KIMBALL e CASERTA, 2004).

- **Carga:** após ter os dados devidamente tratados, a fase de entrega, consiste na realização da carga nas tabelas fato e nas dimensões, que é a estrutura básica de um sistema de *data warehouse* (KIMBALL e CASERTA, 2004).

A qualidade dos dados que serão integrados é outro fator importante na implementação de um sistema de *data warehouse*, pois, seus valores e descrições precisam representar exatamente, fazerem sentido e estarem consistentes aos seus objetos associados (KIMBALL e CASERTA, 2004).

Os metadados são dados sobre outros dados, contendo informações relevantes sobre a criação, gerenciamento e o uso do *data warehouse* (DATE, 2004). Todos os metadados devem ser coletados durante a limpeza e padronização das etapas operacionais do ETL, também devem ter a capacidade de fazer a interligação entre os usuários do *data warehouse* e os dados neles contidos (KIMBALL e CASERTA, 2004).

O metadado também pode ser utilizado para descrever o dado que está sendo envolvido no processo de ETL, em diferentes ambientes, ele é o nome de um tipo de dado diferente. Os metadados no processo de ETL também podem ser envolvidos no contexto das fontes de dados, regras de transformação, regras de extração e no fluxo de trabalho. Respectivamente, esse modelo de serviço implica em diferentes áreas: (WANG e YE, 2010)

- **Metadado em *data warehouse*:** no modelo de ETL, esse metadado é utilizado principalmente para descrever a informação relevante do processo de

ETL para o *data warehouse,* especificamente descreve informações de fonte de dados e o destino dos dados, tais como: o modelo utilizado no *data warehouse*, nível de granularidade dos dados, tabelas relacionais, atribuídos, restrição de integridade, visualizações, procedimentos e as regras de integração de dados (WANG e YE, 2010).

- **Metadado em fonte de dados:** os metadados de fonte de dados descrevem principalmente as informações da origem dos dados que está preste a ser extraída, incluindo o formato dos dados, endereço de IP, portas de acesso, o esquema do banco de dados da origem, estrutura das tabelas relacionais, conjunto de atributos, índices, visualizações, procedimentos, restrições de integridade referencial e etc. Adicionalmente, envolve descrição de medidas, tais como: média, valores máximo ou mínimo de um determinado campo, número total de registros e outras informações estatísticas (WANG e YE, 2010).

- **Metadado em regras de extração:** metadados de extração descreve, principalmente, a relação de mapeamento entre a fonte de dados e o *data warehouse*, incluindo as informações dos campos de origem, informações dos campos de destino e as informações de regras de transformação. Pode conter diversos relacionamentos entre os campos de origem e os campos de destino, e para cada uma dessas situações, pode-se utilizar diferentes regras de extração (WANG e YE, 2010).

- **Metadado em regras de transformação:** normalmente, as regras de transformação são implementadas através de *triggers*, procedimentos e funções armazenadas no banco de dados do *data warehouse*. E a definição e chamada de métodos dessas regras de transformação são preservadas através dos metadados, tais como: o nome dos procedimentos, descrição das funções e os tipos de valores de retorno (WANG e YE, 2010).

- **Metadado em regras de processo:** metadado em regras de processo é utilizado principalmente para descrever a informação do processo de

> extração de dados, tais como: a sequência de execução dos processos, o manuseamento do detalhe de exceção. Na sequência de execução do processo, é permitido incluir outros processos diferentes de granularidade existentes no ETL, e o método de descrição e o mesmo utilizado nos outros modelos, o que torna todo o processo de ETL reutilizável (WANG e YE, 2010).

Os modelos servem para ilustrar de forma abstrata algo que ainda não podemos perceber, através de notações de fácil compreensão. Podemos utilizar os modelos para certificar que todos os objetos exigidos pela empresa estão plenamente representados (BALLARD, FARRELL, *et al.*, 2006). Para isso, é importante que, mesmo sendo um modelo conceitual simples, deve ser aplicado no processo de ETL, afim de: (I) facilitar a redefinição e revisão dos esforços e (II) servir como meio de comunicação entre todos os *stakeholders* envolvidos (VASSILIADIS, SIMITSIS e SKIADOPOULOS, 2002).

Os projetos de *data warehouse* sofrem com alguns problemas no processo de ETL, tais como a integridade entre suas fontes de dados heterogêneas e a falta de distribuição e interoperabilidade dos componentes.

A integridade entre fontes de dados heterogêneas pode dividir-se em duas partes, sendo que, a primeira é a estrutura de diferentes fornecedores de banco de dados (SQL Server e Oracle, por exemplo), são diversos tipos de dados e esquemas diferentes um do outro. A segunda inclui a semântica do dado, são diferentes formatos e representação do mesmo dado (unidades de medidas e tipos de moedas) (ZHANG e WANG, 2008). Para esse cenário de fontes de dados heterogêneas, pode utilizar como solução a tecnologia *web* semântica de metadados para o esquema, compartilhando, reutilizando e estruturando com o uso de ontologia (ZHANG e WANG, 2008).

A falta de distribuição e interoperabilidade dos componentes são considerados problemas no processo de ETL porque são firmemente acoplados e normalmente esses componentes são instalados em cada local da origem do dado (AWAD e ABDULLAH, 2010). Gerenciar esses componentes exige muito esforço de administração e manutenção, porque esses administradores podem estar em lugares diferentes, por isso, se os componentes de ETL forem distribuídos em componentes

interoperáveis como as aplicações de serviço *web* (SOA) e hospedados em servidores, a administração deles será centralizada e padronizada entre todos os administradores de ETL (AWAD e ABDULLAH, 2010).

A monitoração do uso dos dados no ambiente do *data warehouse* é fundamental para gerir de forma eficaz o sistema. Esse processo é muito importante para manter o sucesso do projeto após o sistema de *data warehouse* ser construído, o principal componente para a manutenção do *data warehouse* é o gerenciamento de desempenho (INMON, 2005). Segundo Jack Olson, o *data profiling* é percurso necessário para projetar qualquer tipo de sistema que utiliza dados (KIMBALL e CASERTA, 2004).

O *data profiling* que pode ser criado durante o processo de monitoração dos dados (INMON, 2005), inclue:

- Catálogo de todas as tabelas do *data warehouse*;
- Perfil de conteúdo dessas tabelas;
- Perfil de crescimento de tabelas no *data warehouse*;
- Catálogo dos índices disponíveis para a entrada nas tabelas;
- Catálogo das tabelas sumarizadas e as origens desses resumos.

O *data profiling* é um exame sistemático de qualidade, escopo e contexto de fonte de dados para permitir que um sistema de ETL possa ser construído. Essa perspectiva é especialmente importante para a equipe de ETL, porque pode revelar que os dados de origem são profundamente falhos e não podem apoiar nos objetivos do negócio e o esforço do *data warehouse* pode ser cancelado (KIMBALL e CASERTA, 2004).

A integração de dados deve ter lugar entre os sistemas de transação da organização antes que qualquer dado chegue ao *data warehouse*, mas raramente isso ocorre, a menos que a organização tenha estabelecido em um recurso de planejamento único no sistema de *Enterprise Resource Planning* - ERP, e mesmo assim, é provável que outros recursos sistêmicos de processamentos importantes existam fora do sistema principal de ERP (KIMBALL e CASERTA, 2004). Esse é um dos motivos pelos quais os dados integrados em um sistema de *data warehouse* são heterogêneos, mesmo parecendo ser homogêneos quando já acessado pelo repositório, e esses dados são divididos em subdivisões orientadas por assuntos (INMON, 2005).

A primeira divisão de dados dentro do *data warehouse* é com base nos principais assuntos da corporação e dentro de cada assunto existem as subdivisões que são organizados em tabelas. O dado dentro do *data warehouse* é subdividido pelo o seguinte critério (INMON, 2005):

- Orientado por assunto;
- Tabelas;
- Ocorrências dos dados dentro das tabelas.

A capacidade de desenvolver processos de ETL eficientes em parte é ser capaz de determinar o justo equilíbrio entre o armazenamento dos dados em uma área de *stage* ou o processamento em *in-memory*. Essa decisão do modo de processamento dos dados de origem para o destino em última análise é tomada pelo arquiteto de ETL (KIMBALL e CASERTA, 2004). O desafio de alcançar esse equilíbrio delicado entre área de *stage* ou mantê-lo em memória durante o processo de ETL é uma tarefa que deve ser levada em conta, a fim de criar processos mais eficientes. O problema de escolher trabalhar com os dados em uma área de *stage* ou não depende de dois objetivos conflitantes, que é conseguir os dados a partir da origem para o destino de forma rápida e a capacidade de recuperação de falhas sem reiniciar o processo pelo início (KIMBALL e CASERTA, 2004).

A decisão para carregar os dados no *data warehouse* deve considerar:

- **A capacidade de recuperação:** Na maioria dos ambientes corporativos, é uma boa prática armazenar os dados na área de *stage* logo depois que eles tenham sido extraídos de sua fonte, e em seguida, logo após cada grande etapa de transformação. Essas tabelas de *staging* (no banco de dados ou em arquivos textos) servem como pontos de recuperação, uma vez implementados esses objetos, o processo tende a não se intrometer com os sistemas de origem caso a transformação venha falhar (KIMBALL e CASERTA, 2004). Esse é um recurso simples e importante que compõe as características do ambiente de *data warehouse*, a capacidade de restaurar rapidamente as tabelas em seu ambiente, principalmente quando a restauração pode ser feita a partir do *storage*, enormes problemas podem ser poupados (INMON, 2005).
- ***Backup*:** O volume de dados impede o seu armazenamento em um *backup* confiável. Muitas vezes, ocorrem problemas que podem ser evitados

se apenas os arquivos de carga forem salvos, compactados e arquivados, pois, caso haja a necessidade da utilização, pode-se simplesmente descompactar e carregar os arquivos no *data warehouse* (KIMBALL e CASERTA, 2004).

- **Auditoria**: Quando chega o momento, a auditoria é realizada entre diferentes partes dos processos de ETL. Os auditores podem simplesmente comparar os arquivos originais de entrada com as regras de transformações lógicas com os arquivos de saída, nesse cenário, o *stage* é extremamente importante quanto as questões de integridade de informação. (KIMBALL e CASERTA, 2004). Apesar da auditoria poder ser realizada a partir do *data warehouse*, existem algumas razões para não faze-la desse ambiente: (INMON, 2005):
    - Os dados que de outra maneira não iriam encontrar-se no *data warehouse*, de repente devem estar lá.
    - O momento de entrada de dados, restrição para *backup* e a recuperação dos dados para o *data warehouse* muda drasticamente quando um recurso de auditoria é necessário.
    - Dados de auditoria no repositório obriga a granularidade dos dados *do data warehouse* estarem no seu nível mais baixo.

O passo final para o sistema de ETL é a entrega para os aplicativos do usuário, acredita-se que a equipe de ETL, trabalhando em estreita colaboração com a equipe de modelagem, deve assumir a responsabilidade pelo conteúdo e estrutura dos dados, fazendo com o que os aplicativos de usuário final sejam simples e rápido. Cada ferramenta para usuário final tem certas sensibilidades que devem ser evitadas e certas características que podem ser exploradas, por isso, os dados físicos devem estar no formato correto. (KIMBALL e CASERTA, 2004).

Um dos problemas enfrentados por profissionais de *data warehouse* é o modelo básico para seu projeto, existem modelos que são amplamente considerados: modelo relacional com a abordagem de Inmon e o modelo multidimensional que é considerado na abordagem de Kimball. A abordagem do modelo relacional inicia-se organizando os dados dentro de tabelas, diferentes colunas estão em cada linha da tabela. A tabela relacional pode ter diferentes propriedades, as colunas de dados têm diferentes indexadores e características físicas, podem ser nulos e são todas definidas em

termos de uma declaração de linguagem de definição de dados – DDL (INMON, 2005). A abordagem multidimensional foi definida pelo Dr. Raph Kimball para projetos de *data warehouse*, apoiado basicamente pelo modelo de *start schema* que tem esse nome porque sua representação retrata uma estrela e a estrutura *snowflake*. Esses dois modelos são formados por tabelas de dimensões e fatos, existem muito mais ocorrências em tabelas fato do que nas tabelas de dimensões (INMON, 2005).

Existem muitas diferenças entre os modelos dimensional e multidimensional, porém a diferença mais importante é o termo de flexibilidade e desempenho. O modelo relacional é altamente flexível, mas não é otimizado para desempenho para qualquer usuário. O modelo multidimensional é altamente eficiente no serviço às necessidades de uma comunidade de usuários, mas não é flexível. Outra diferença importante nas duas abordagens para o desenvolvimento do banco de dados está no escopo do projeto. Por necessidade, o projeto multidimensional tem um alcance limitado, desde os requisitos de processamento que são usados para realizar a modelagem dos dados. O projeto começa a quebrar quando muitos requisitos de processamento estão reunidos, ou seja, o projeto de banco de dados pode ser otimizado para um conjunto de requisitos de processamento. Quando um conjunto completamente diferente de requisito é adicionado ao projeto, a otimização torna-se discutível (INMON, 2005).

O principal objetivo do processo de ETL é a facilidade de movimentar e transformar os dados de seus diversos sistemas de origem para serem carregados em tabelas do *data warehouse* ou *data marts* (DUPOR e JOVANOVI, 2014). Esses processos necessitam de modelagem, *design* e fundamentos metodológicos. Na figura 4 é exibido um *framework* para o processo de ETL que lida com as fases de um projeto de *data warehouse*, durante esse período o projeto é concentrado em duas tarefas que são executadas praticamente em paralelo: a primeira tarefa envolve o levantamento de requisito da parte dos usuários, a segunda tarefa envolve a análise da estrutura e conteúdo que existe nos dados de origem e o seu mapeamento para o modelo do *data warehouse* (VASSILIADIS, SIMITSIS e SKIADOPOULOS, 2002). O ETL é um conjunto de programas que deve agregar valor significativo aos dados. Especificamente, o sistema de ETL:

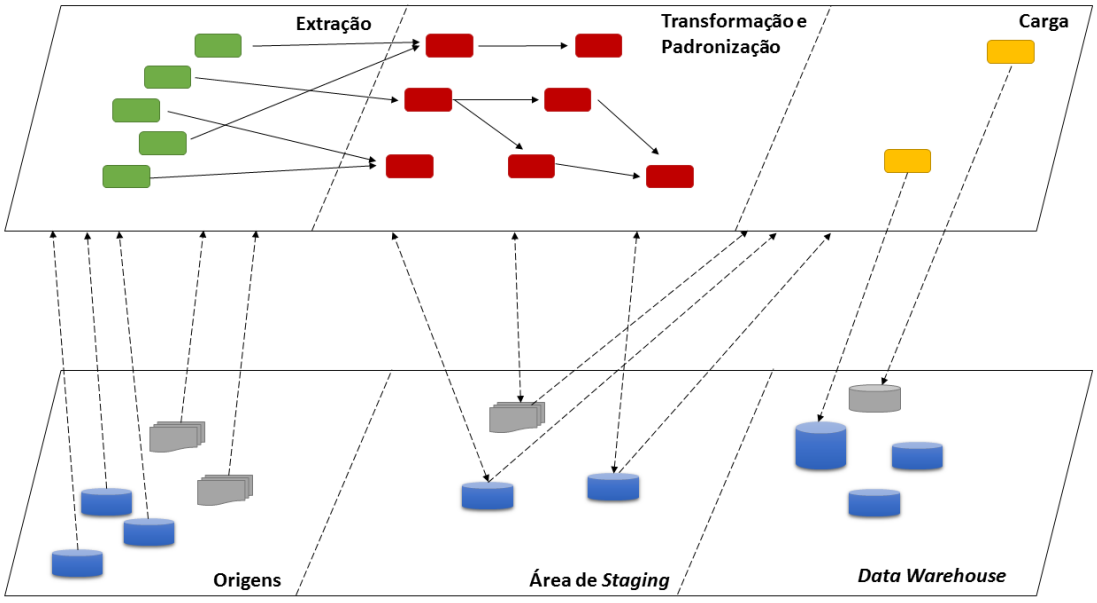
- Remove e corrige dados;
- Fornece medidas confiáveis em dados;

- Ajusta dados de diversas fontes para ser utilizado em conjunto;
- Reestrutura os dados para serem utilizados em ferramentas de visualização para o usuário final.

4.2. *Framework* de ETL baseado em ontologia

O *framework* referente ao ambiente do processo de ETL é composto por duas camadas, sendo a inferior responsável pelo armazenamento de todos os dados que são envolvidos no processo. No lado esquerdo da figura 4, pode-se observar os dados de origem, fornecidos por sistemas transacionais ou arquivos de texto, por exemplo. Esses dados são extraídos por rotinas que produzem *snapshots* completos ou parciais, conforme ilustrado na parte superior da figura. Em seguida, os dados são propagados para a área de *staging*, onde eles são transformados e limpados antes de serem carregados para o *data warehouse*. O *data warehouse* é mostrado no lado direito da figura 4, que compreende os dados no ambiente alvo, que normalmente são tabelas fatos e suas tabelas de dimensões (VASSILIADIS, SIMITSIS e SKIADOPOULOS, 2002).

Figura 4. O ambiente do processo de ETL



Fonte: (VASSILIADIS, SIMITSIS e SKIADOPOULOS, 2002)

Apesar do fato de o ETL ocupar uma grande quantidade de recursos e ser um dos componentes mais relevantes no desenvolvimento e manutenção do *data warehouse*, ainda há limitações nos projetos que cobrem a arquitetura, técnicas de desenvolvimento e metodologias de integração de dados. Isso é uma motivação para a criação do modelo conceitual do *data warehouse* para apoiar o time de desenvolvimento do projeto de acordo com as necessidades dos utilizadores e no que diz respeito às fontes de dados disponíveis (DUPOR e JOVANOVI, 2014).

É uma expectativa que durante o ciclo de vida do sistema de *data warehouse* os usuários de negócio solicitem alterações no processo de carregamento que pode ser causado por necessidade de negócio, conformidade, mudança de requisitos ou até mesmo por alterações nos sistemas de transação, adicionando ou removendo funcionalidades. Por outro lado, também é realista esperar que os desenvolvedores realizam as manutenções e alterações nos sistemas de ETL. No entanto, eles podem mudar para outros projetos ou mudar de emprego (DUPOR e JOVANOVI, 2014).

Em seguida, apresentam-se problemas específicos que podem ocorrer durante o desenvolvimento e manutenção dos sistemas de ETL, conforme DUPOR e JOVANOVI (2014):

- Dificuldade de comunicação entre os usuários de negócio e arquitetos de ETL causada pela falta de modelo padronizado;
- Tempo perdido mantendo o processo de ETL em que o desenvolvedor deve entender e explorar todo o processo, a fim de entender o que está acontecendo com os dados;
- Duplicação de processamento de dados e transformações causadas por desconhecimento de todas as etapas do processo, resultando em tempo de execução mais longo e na complexidade dos mapeamentos dos dados, tornando a manutenção mais exigente;
- Falta de documentação necessária que torna o processo de ETL mais difícil de ler e entender.

Dupor e Jovanovi (2014), comentam que existem trabalhos que apresentam uma abordagem baseada em tecnologia da *web* semântica para facilitar o processo de seleção de informações relevantes a partir das fontes de dados disponíveis,

transformando para preencher o *data warehouse*. Esses trabalhos utilizam ontologias para especificar a semântica de esquemas do armazenamento de dados, e apresenta possibilidade de automatizar o processo de fluxo de trabalho do ETL. (DUPOR e JOVANOVI, 2014).

A princípio não existem critérios de organização e padronização nas informações disponibilizadas na *web*, nesse cenário surge a *web* semântica que através do uso intensivo de metadados realiza a criação de uma rede capaz de reconhecer o significado de documentos, pelo meio de processamento de computadores. A arquitetura ilustrada na figura 5 é dividida em três grandes camadas: A camada de estrutura é responsável por estruturar os dados e definir seus significados, ela é implementada utilizando as tecnologias de XML (*eXtensible Markup Language*), RDF (*Resource Description Framework*) e RDF *Schema*. A camada de esquema é composta pelas ontologias que possibilitam a criação de um conjunto comum de termos que são utilizados para descrever e representar um domínio. A camada lógica é responsável por fazer a conclusão sobre os dados apresentados (LIMA e CARVALHO, 2005).

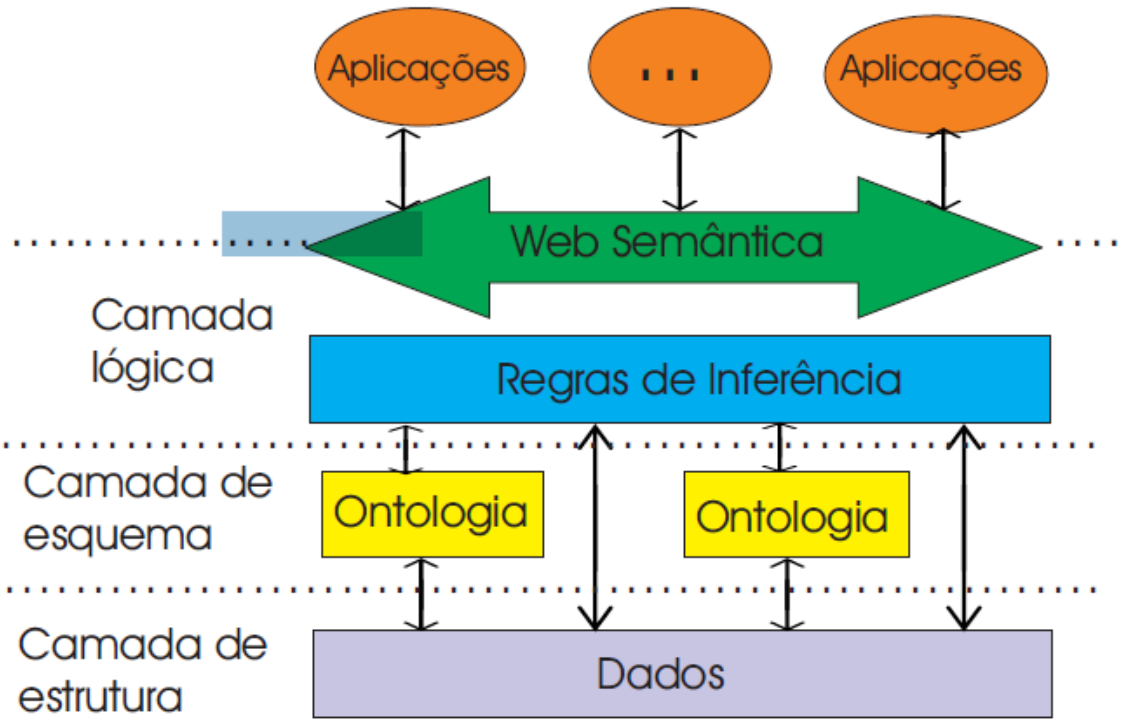


Figura 5. Arquitetura da Web Semântica

Fonte: (ZHANG e WANG, 2008)

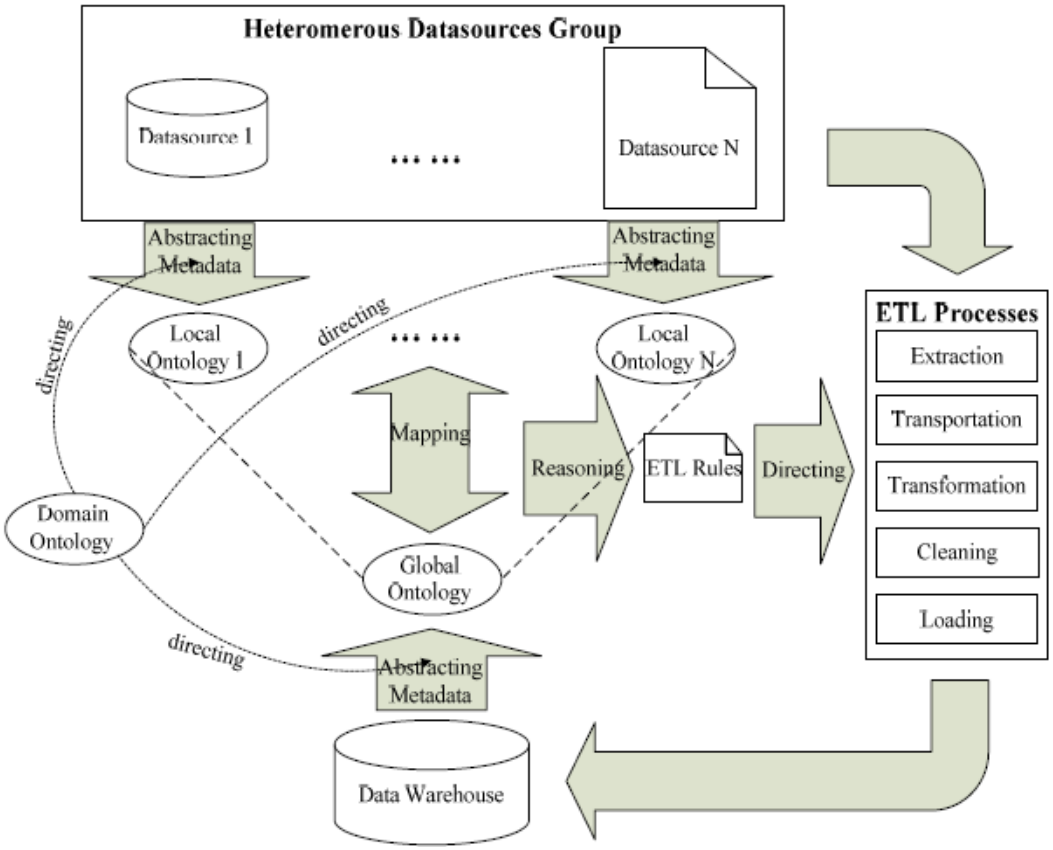
Uma ontologia descreve um modelo abstrato de um fenômeno do mundo em um conhecimento ou domínio, por exemplo: engenharia, medicina, loja e etc. Através disso, os conceitos, as propriedades, as funções, os relacionamentos e compartilhamento devem ser explicitamente especificados e manipulados por computadores. A importância da ontologia para a *web* trata-se em três aspectos: Identificação do contexto, Fornecimento de definições compartilhadas e Reuso das ontologias. Quando as ontologias são construídas levando em consideração esses itens, elas permitem o compartilhamento de informações em diferentes agentes de software, com isso, as buscas são baseadas em semântica, melhorando a integração de dados em banco de dados heterogêneos (LIMA e CARVALHO, 2005).

Segundo Guariano e Guirreta (1995), ontologia é um acordo sobre o conceito compartilhado que inclui *frameworks* para modelar o conhecimento de um domínio e comprometer-se com a representação conceitual do domínio (ZHANG e WANG, 2008). A utilização de ontologia no ambiente da *web* semântica requer o uso de uma linguagem de ontologia compatível com a *web*, a *Word Wide Web Consortium* (W3C – principal organização de padronização de *Word Wide Web*) recomenda o uso da *Web Ontology Language* (OWL), com o objetivo da linguagem tornar-se um padrão para a criação de ontologias (LIMA e CARVALHO, 2005).

Existem praticamente três sub-linguagens projetadas para serem usadas por diferentes comunidades de implementadores e usuários: OWL *Life*, OWL DL e OWL *Full*. Essas linguagens são utilizadas na criação de classes e propriedades de classes para a definição de um domínio de ontologia e compartilhamento da informação na internet. Cada um tipo de linguagem OWL é uma extensão de sua predecessora e a sua escolha depende da característica do desenvolvimento da ontologia (LIMA e CARVALHO, 2005).

O processo de ETL que é fundamentado em diversos grupos de fonte de dados heterógenas pode utilizar um modelo de *framework* baseado em ontologia para estabilizar a construção do projeto de *data warehouse*. O *framework* é dividido em quatro fases: Metadado abstrato, Mapeamento de ontologias, de Regras e Direção do ETL (ZHANG e WANG, 2008). Conforme a figura 6.

Figura 6. Um modelo de framework baseado em ontologia para processo de ETL



Fonte: (ZHANG e WANG, 2008)

As fases do modelo de *framework* são descritas a seguir:

- **Metadado abstrato:** Esta fase é a construção do modelo abstrato do metadado que é composto por três etapas, sendo a primeira um processo de extração do metadado dos arquivos de origem, a segunda, é a criação do modelo de abstração do metadado como uma ontologia local. E por fim, o desenvolvimento da abstração para a ontologia global. Nessa fase inteira, para garantir a qualidade das ontologias, esse processo deve ser reconhecido por especialistas no domínio da ontologia (ZHANG e WANG, 2008).

- **Mapeamento de ontologias:** O objetivo do mapeamento da ontologia é definir a relação da estrutura e semântica entre dois conceitos de ontologias: local e global. Ele nos ajuda a compreensão da heterogeneidade de estrutura e semântica, um dos métodos de mapeamento é a comparação da semelhança

entre as ontologias (ZHANG e WANG, 2008). A principal dificuldade em construir uma ontologia é juntar classes e propriedades de ontologias diferentes, da melhor forma possível para se fazer inferências (LIMA e CARVALHO, 2005).

- **Argumentação de regras:** Essa fase é derivada do processo de mapeamento de ontologias, as descrições fornecidas pela fase de mapeamento para que as fontes de dados estejam mais acessíveis aos processos automatizados, especialmente em processos de ETL. Ele pode ajudar a encontrar conflitos na relação de mapeamentos, o teste de coincidência, teste de subsunção e o teste de equivalência são os três problemas básicos da definição (ZHANG e WANG, 2008).

- **Direção do ETL:** É a última fase, ela nos obriga a fornecer as regras de ETL para o computador, em seguida, operar os processos de ETL que será mais eficaz e eficiente (ZHANG e WANG, 2008).

Um exemplo é ilustrado para fornecer uma demonstração da funcionalidade e a estrutura interna do desenvolvimento da ontologia. Uma parte do projeto de *data warehouse* é listado na tabela 2.

Tabela 2. Exemplo ilustrativo de banco de dados

Data Source	Schema	Database Management System
DS1	Departamento (id, nome, localização)	Microsoft SQL Server 2005
DS2	Departamento (código, nome, cidade, rua, número)	Oracle 9i
DW	Departamento (código departamento, nome, cidade, rua)	Oracle 91

Fonte: (ZHANG e WANG, 2008)

É simples identificar que as duas fontes de dados (DS1 e DS2) são construídas para controlar informações básicas de departamento de uma determinada organização. No entanto, elas estão armazenadas em ambientes totalmente diferentes. A diferença de

esquema e banco de dados faz com que seja difícil a integração, especialmente em projetos de *data warehouse*. É normal que o ambiente do *data warehouse* também tenha sua própria estrutura, agora o passo é a construção das ontologias (ZHANG e WANG, 2008).

Os elementos básicos para a construção de uma ontologia são as classes, as instâncias das classes e os relacionamentos que são as propriedades entre as instâncias. As classes fornecem um mecanismo para agrupar recursos com características similares, são eles que definem um conjunto de indivíduos que compartilham algumas propriedades. Um grupo de indivíduos associados a uma classe é chamado de extensão de classe, onde os indivíduos de uma extensão de classe são chamados de instâncias de classe. Cada indivíduo do OWL é um membro da classe (LIMA e CARVALHO, 2005). A seguir, um exemplo da linguagem OWL DL dos dados:

<owl: Class rdf: ID="DW">
<rdfs: subClassOf>
<owl: Restriction>
<owl: onProperty>
<owl: ObjectProperty rdf: about="#dbmstype"/>
</owl: onProperty>
<owl: hasValue rdf: resource="#Oraclo_9i"/>
</owl: Restriction>
</rdfs: subClassOf>
<rdfs: subClassOf rdf: resource="#DB"/>
</owl: Class>

Atualmente existem diversas métodos de mapeamento de ontologias, por exemplo: GLUE é um sistema que aplica técnicas de aprendizado em máquinas para encontrar mapeamentos semânticos entre conceitos armazenados em diferentes ontologias. Já o HMATCH é um algoritmo para correspondência dinâmica em ontologias distribuídas que considera a avaliação de afinidade linguística como um passo atômico. Segundo Zhang e Wang (2008), pode-se combinar essas abordagens adicionando pesos para cada uma delas, nesse exemplo pode-se facilmente descobrir que:

<owl: Class rdf: ID="ID">

<owl: equivalentProperty rdf: resource="CODIGO"/>
</owl: Class>

O campo ID (DS1) faz a mesma relação que o campo CODIGO (DS2), assim, o que for estabelecido para o campo ID será igual para o campo CODIGO. Existem APIs OWL que são fáceis de serem utilizadas no *data wahouse*, e depois da implantação, pode-se encontrar relacionamentos entre cada fonte de dados (ZHANG e WANG, 2008).

Após a fase de mapeamento e argumentação de regras, uma série de regras de relacionamentos podem ser listadas para atendermos os requisitos da fase de direção do ETL. Então é preciso transformar essas regras para as tarefas de ETL que contêm os processos de armazenamento correspondentes. Utilizando ontologia no ETL faz com que ele fique mais eficiente e flexível em todo o processo que pode ser parcialmente automatizado com as regras do ETL no *framework* (ZHANG e WANG, 2008).

Como já mencionado no decorrer do texto, outro problema que pode ser solucionado com a utilização de um *framework* é a falta de distribuição e a interoperabilidade dos componentes de ETL. Ao desenvolver um projeto de *data warehouse*, muitas fontes de dados estão disponíveis em locais geograficamente diferentes e cada uma delas necessitam de uma ferramenta completa de ETL para executar apenas o processo de Extração. No entanto, as características de transformação e carregamento ficam redundantes na origem, pois, elas só são necessárias no ambiente do *data warehouse* (VASSILIADIS, SIMITSIS e SKIADOPOULOS, 2002).

4.3. *Framework* de ETL baseado em *Service-oriented architecture*

O gerenciamento de ferramentas de ETL geram esforço na administração e manutenção dos processos, principalmente quando essas ferramentas estão distribuídas em locais diferentes. Por outro lado, quando os componentes de ETL são distribuídos em componentes interoperáveis e hospedado no servidor de aplicação como os serviços *web*, a administração dos componentes são centralizadas e padronizadas entre os administradores de ETL do projeto. Além disso, às vezes, devido à complexidade, longa curva de aprendizado das ferramentas de ETL e a dificuldade de conseguir alguma extensibilidade nos termos de funcionalidade,

algumas organizações preferem adotar o desenvolvimento *in-house* para executar esses processos de ETL, o que pode gerar um aumento no tempo e esforço do projeto (AWAD e ABDULLAH, 2010).

Conforme ilustrado na figura 4, do ambiente do processo de ETL, o *framework* tradicional do ETL é fortemente acoplado. Suas funcionalidades, o conceito anterior, a fase atual e o relacionamento entre eles, mostram uma grande dependência entre os seus processos e fases (VASSILIADIS, SIMITSIS, *et al.*, 2005). No entanto, é proposto um novo *framework* baseado em SOA para aprimorar a distribuição e a interoperabilidade do modelo tradicional (AWAD e ABDULLAH, 2010). *Service-oriented architecture* – SOA que pode ser traduzido para serviço orientado a arquitetura é uma abordagem para definir a integração entre arquiteturas, baseado no conceito de serviços. O objetivo do SOA pode ser descrito como levar os benefícios do acoplamento, encapsulamento e a computação distribuída para a integração em um nível corporativo (KEEN, ACHARYA, *et al.*, 2004).

No novo *framework*, mostrado na figura 7, a camada de dados e a de armazenamento são exatamente similares ao *framework* tradicional, a camada de negócio é construída baseada em SOA (AWAD e ABDULLAH, 2010). As quatros principais partes estão descritas a seguir:

- ***Service Orchestration Point:*** Também conhecido como *Directory Service* ou *Service Registry.* Essa parte descreve os serviços que são disponíveis em seu domínio que são: Extração, transformação e carregamento. Esses três serviços são referenciados como fornecedores de serviços (*Service Providers*) e registrados no *Orchestration Point* (AWAD e ABDULLAH, 2010).

- ***Service Providers:*** Cada fornecedor de serviço é um componente executa uma resposta ao pedido de um cliente. O *framework* tem três serviços para fornecer, que são: extração, transformação e carregamento (AWAD e ABDULLAH, 2010).

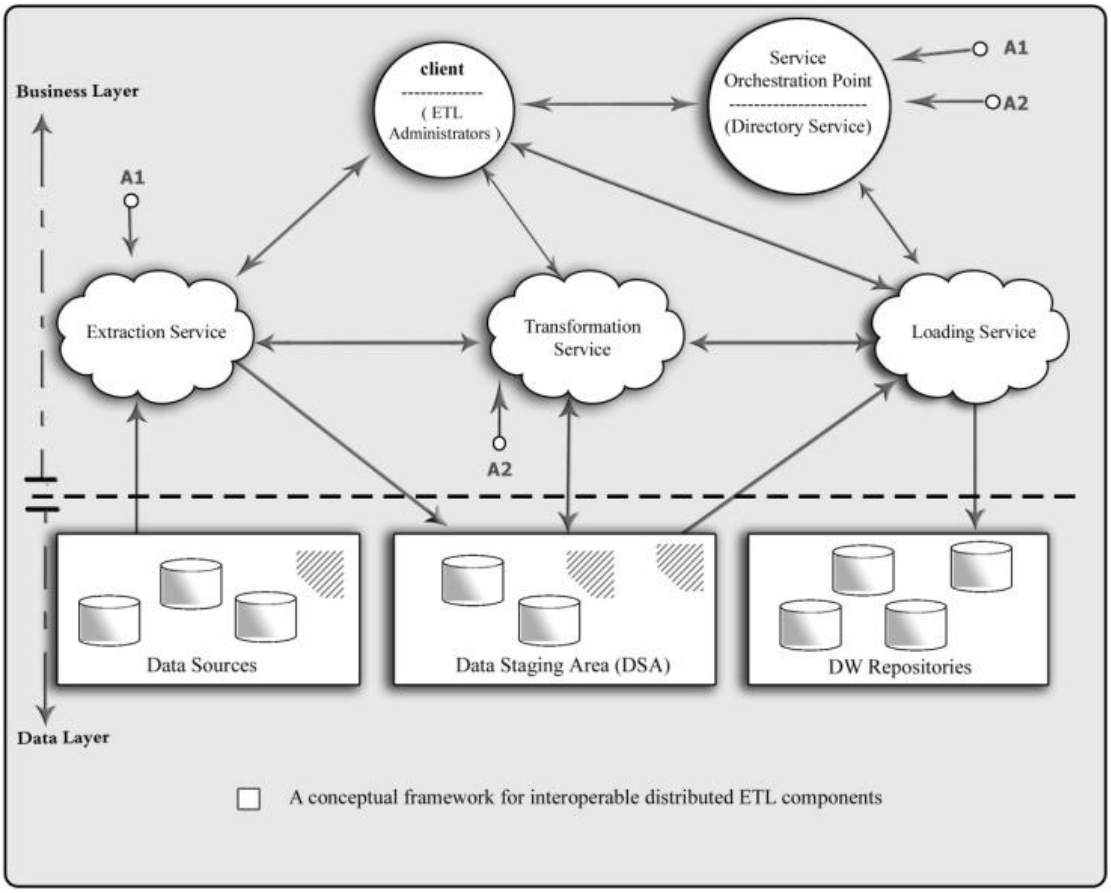
- ***Service Customers*** (Serviço Consumidor): Cada serviço consumidor é um componente que consome o resultado de um serviço prestado por um fornecedor. O principal *Service Customers* no *framework* é o cliente, que é

representado pelos administradores de ETL. Além disso, os três fornecedores de serviço podem ser consumidores de serviço, por exemplo: o serviço de transformação pode executar algumas funções do serviço de extração, no caso em que a transformação e a extração são executadas em um único *patch* (AWAD e ABDULLAH, 2010).

- ***Service Interface:*** interface de serviço define o acesso programático através dos três serviços, estabelece a identificação e as regras para a invocação do serviço. A relação entres os serviços de prestador e consumidor é dinâmico e estabelecido em tempo de execução através de um mecanismo de ligação do *Orchestration Point*. Essa relação dinâmica minimiza as dependências entre os serviços de consumidor para o provedor, portanto as funcionalidades de extração, transformação e carregamento de fato suporta o baixo acoplamento por serem distribuídas em serviços *web* interoperáveis no novo *framework* (AWAD e ABDULLAH, 2010).

A descoberta de um serviço pelo cliente pode ser feita com a utilização do *Orchestration Point*, depois de visualizar o serviço, o cliente continua com a comunicação remota com qualquer outro serviço distribuído. O cliente é representado pelo serviço consumidor e o processo de ETL é representado pelo serviço fornecedor, mas também, o ETL pode consumir qualquer outro serviço (AWAD e ABDULLAH, 2010).

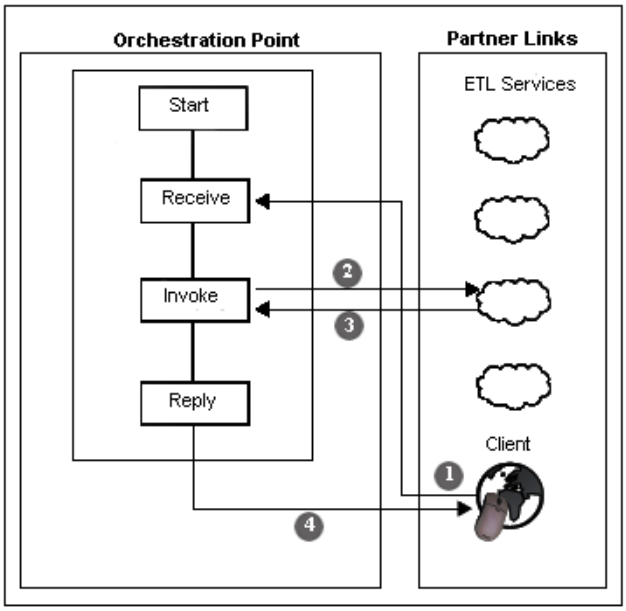
Figura 7. Um framework interoperável para componentes de ETL distribuído



Fonte: (AWAD e ABDULLAH, 2010)

Mostra-se na figura 8 um exemplo do fluxo de ações quando o cliente demanda uma execução das funcionalidades do ETL. No momento em que o cliente deseja consumir um certo serviço de ETL, o *Orchestration Point* começa com a atividade “receber” em que o cliente recebe o pedido. Em seguida, ele invoca o serviço adequado de ETL e responde de volta para o cliente. Tipicamente, o processo de *Orchestration Point* interage com um ou mais *web services* de ETL, esses serviços *web* são chamados de serviços parceiros ou serviços externos (AWAD e ABDULLAH, 2010).

Figura 8. O fluxo dos passos para consumir um serviço de ETL pelo cliente



Fonte: (AWAD e ABDULLAH, 2010)

Após o desenvolvimento de um protótipo baseado na restruturação do *framework*, os resultados foram comparados entre o modelo tradicional e a nova abordagem. A conclusão apresenta algumas vantagens em reestruturar o *framework* em relação ao modelo tradicional. Esses pontos são destacados a seguir:

- **Interoperabilidade:** A interação entre os clientes (administradores de ETL) e os serviços de ETL fracamente acoplados tem a interoperabilidade generalizada. Exige-se que os clientes e os serviços possam se comunicar e entender uns aos outros, não importa em qual plataforma está sendo executado. Esse objetivo é alcançado porque os clientes e serviços de ETL têm uma forma padrão de comunicação, além disso, fornece consistência entre as plataformas, sistemas e idiomas para os componentes do *framework* de ETL (AWAD e ABDULLAH, 2010).

- **Flexibilidade:** Serviços fracamente acoplados no *framework* de ETL são tipicamente mais flexíveis que os fortemente acoplados. Isso torna a aplicação mais fácil de evoluir com mudanças de requisitos, portanto, a nova abordagem do *framework* de ETL pode ter componentes extras sem muita complexidade. Com base nesse recurso, o novo *framework*, futuramente poderá ser estendido

para qualquer componente extra que visa atender novos requisitos de negócio da empresa (AWAD e ABDULLAH, 2010).

- **Reutilização**: os desenvolvedores de ETL em sistema de *data warehouse* da indústria podem reutilizar o código de um componente de ETL existente desenvolvido por qualquer outro provedor de ETL para atender novos requisitos de negócio. Isso permite reutilizar a funcionalidade que já existe dentro ou fora da empresa, em vez de desenvolver o código que reproduz essas funções. Isso pode resultar em uma grande economia de custo e tempo de desenvolvimento das aplicações (AWAD e ABDULLAH, 2010).

- **Escalabilidade:** os serviços de ETL fracamente acoplados que respeitam as normas de *framework* podem ser escalados facilmente, pelo menos, é mais fácil do que aplicações que seguem os *frameworks* mais firmemente acoplados. A escalabilidade é alcançada porque há menos dependência entre as aplicações requerentes e os serviços que são utilizados. Isso permite o serviço lidar com mais pedidos devido ao fato de não precisar lidar com sobrecarga de comunicação, como os componentes fortemente acoplados (AWAD e ABDULLAH, 2010).

- **Eficiência de Custo:** um *framework* de ETL baseado em SOA resulta em soluções mais econômicas porque a integração de clientes e serviços não requer grandes análises e um único código de soluções personalizadas. Além disso, como os serviços do *framework* são fracamente acoplados, os aplicativos que os utilizam são mais baratos de manter e mais fácil de estender a soluções personalizadas (AWAD e ABDULLAH, 2010).

Em 2012 foi publicado um artigo com o objetivo de discutir a validação de um protótipo do framework de ETL baseado na arquitetura SOA. Nesse artigo, apresentou-se três estudos de caso que foram conduzidos com o objetivo de avaliar o protótipo desta pesquisa que permite validar o processo de ETL em ambientes reais de *data warehouse*. Os estudos de caso foram realizados utilizando três diferentes

organizações que adotam soluções de *data warehouse* para a finalidade de gerarem relatórios estatísticos (AWAD, ABDULAHB, *et al.*, 2012).

Existem pesquisas com fundamentos teóricos que estruturam componentes de ETL distribuídos e interoperáveis com base na arquitetura SOA. Por outro lado, não fornecem testes e validações suficientes para validar o novo *framework* de ETL, para esse efeito, um protótipo de ETL baseado na arquitetura SOA foi desenvolvido com o propósito de implementar todas as especificações teóricas (AWAD, ABDULAHB, *et al.*, 2012).

Esse protótipo foi testado a avaliado com a finalidade de validar o *framework* de ETL baseado em SOA, os testes são necessários para fins de validação e verificação de que o protótipo cumpre com suas especificações. O protótipo passou por testes unitários, testes de serviços web, testes de compatibilidade, testes de velocidade e de escalabilidade (AWAD, ABDULAHB, *et al.*, 2012).

4.4. Estudo de caso baseado em *Service-oriented architecture*

Nesta seção são apresentados três estudos de casos com o objetivo de comparar as funcionalidades no processo de ETL entre as ferramentas tradicionais e o processo baseado na arquitetura orientada a serviço. O principal artigo utilizado como base foi o *Service oriented architecture based* ETL *framework vaidation* (AWAD, ABDULAHB, *et al.*, 2012)

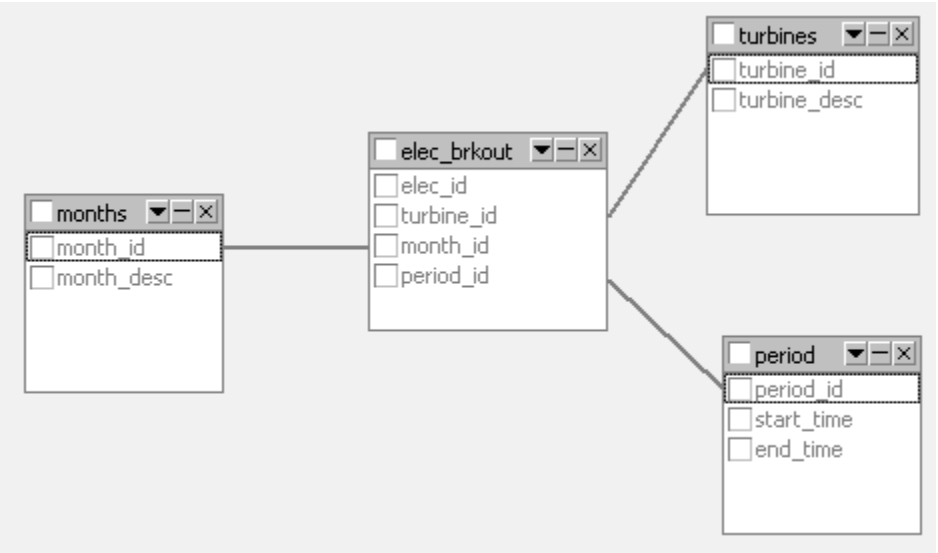
4.4.1. Estudo de caso: Aplicando funcionalidades de ETL na empresa *Palestine Electric Company* usando ferramentas tradicionais e novas de ETL.

Palestine Electric Company (PEC) é uma companhia que opera uma usina de geração de energias na Palestina. A PEC foi formada em gaza sob as leis da Palestina para desenvolver, possuir e gerenciar usinas nos territórios palestinos. A usina é formada por quatro turbinas a gás e duas turbinas de vapor em uma configuração de dois blocos. As turbinas são projetadas para permitir que o sistema de bicombustível queime tanto o gás natural como os combustíveis fósseis. O combustível utilizado pela usina é distribuído em dois grandes tanques de armazenamento, com capacidade de 20 milhões de litros, e o consumo diário de combustível é de aproximadamente 700.000 de litros em plena operação (AWAD, ABDULAHB, *et al.*, 2012).

A empresa PEC conta com muitos relatórios estatísticos com os seguintes objetivos: apoiar as decisões financeiras, levantar os números de horas do funcionamento de cada turbina, quantidade de energia gerada em *Megawatt* (MW) ao longo do tempo, estatística de apagões devido aos impedimentos de ocupação em Israel e devido aos problemas técnicos para o consumo de combustível em cada turbina para realizarem notificações relativas à quantidade necessária de combustível. A geração desses tipos de relatórios estatísticos requer a extração de determinados campos de dados do principal banco de dados da empresa, às vezes, é necessária a realização de algumas transformações nos dados extraídos para serem carregados no *data warehouse*, para fins de geração de relatórios estatísticos (AWAD, ABDULAHB, *et al.*, 2012).

As funcionalidades de ETL são aplicadas no esquema de tabelas utilizando a ferramenta de ETL tradicional, que é usado pela companhia PEC (*Microsoft SQL Server Integration Services* - (SSIS)) e o protótipo baseado na arquitetura SOA (AWAD, ABDULAHB, *et al.*, 2012).

Figura 9. Star Schema das tabelas fato e dimensão extraída do banco de dados PEC



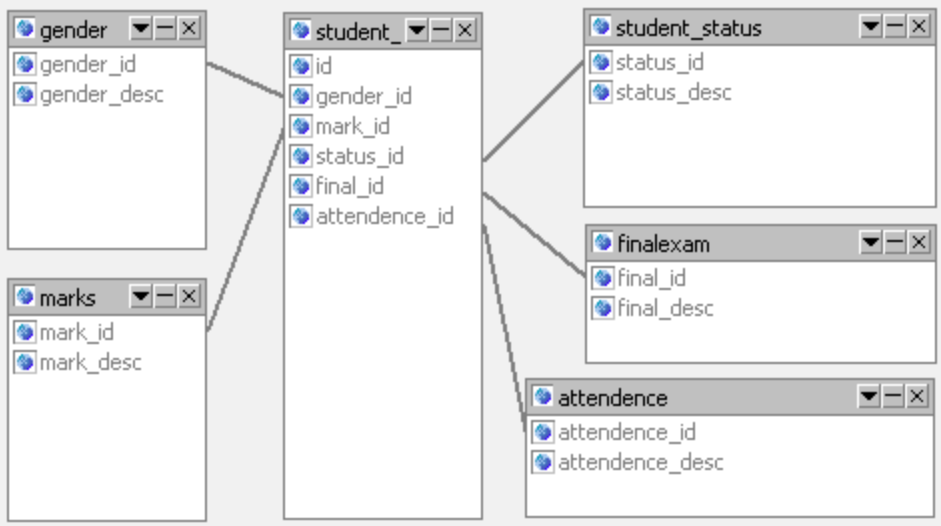
Fonte: (AWAD, ABDULAHB, *et al.*, 2012)

4.4.2. Estudo de caso: *Limkokwing* Universidade de Tecnologia Criativa utilizando ferramentas novas e tradicionais de ETL

Limkokwing Universidade de Tecnologia Criativa (LUCT) é uma universidade da Malásia que tem os seguintes departamentos: Faculdade de Arquitetura e Ambiente; Faculdade de Gestão e Negócios Globalizados; Faculdade de Comunicação, Mídia e Transmissão; Faculdade de Design e Inovação; Faculdade de Tecnologia de Informática e Comunicação; Faculdade de Criatividade e Multimídia. A LUCT realiza muitos relatórios estatísticos para efeitos de tomada de decisão, tais como: relatório de desempenho, relatório financeiro e o relatório de rastreamento de visto dos alunos estrangeiros (AWAD, ABDULAHB, *et al.*, 2012).

Foram utilizadas as mesmas ferramentas do primeiro estudo de caso para a aplicação do processo de ETL na universidade. Para gerar e extrair os dados nos principais bancos de dados, pode ocorrer de precisar transforma-los antes de carrega-los no ambiente do *data warehouse* (AWAD, ABDULAHB, *et al.*, 2012).

Figura 10. Star Schema das tabelas fato e dimensão extraída do banco de dados LUCT



Fonte: (AWAD, ABDULAHB, *et al.*, 2012)

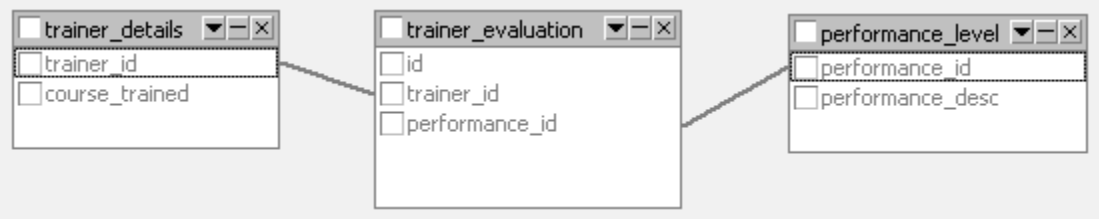
4.4.3. Estudo de caso: Aplicando funcionalidades de ETL na sociedade de profissionais de tecnologia usando ferramentas novas e tradicionais de ETL.

A sociedade de profissionais de tecnologia de informação (PIT) é uma premiada consultoria de treinamentos no centro da faixa de Gaza, na Palestina. A PIT tem sido um bom negócio desde 1997 e tem apresentado um crescimento maior que 100% em cada ano, contando com sete filiais no ano de 2011. Além disso, a sociedade coloca muita ênfase nos treinamentos para que seus estagiários possam participar da força de trabalho da TI com experiência e confiança. Os consultores da PIT prestam serviços especializados para seus clientes que são desde projetos de sistemas de pequenas empresas, como a integração de gerenciamento de grandes contratos do governo (AWAD, ABDULAHB, *et al.*, 2012).

A PIT fornece serviços de treinamentos técnicos e não-técnicos para os setores públicos e privados. Além disso, seus instrutores não são apenas certificados, mas também possuem experiências práticas nos assuntos que ensinam. Na época da pesquisa, a PIT contava com 40 formadores, alguns eram empregados pela PIT, enquanto outros eram parceiros (AWAD, ABDULAHB, *et al.*, 2012).

A sociedade necessitava de relatórios estatísticos com o objetivo de tomada de decisão, tais como: relatório de desempenho e relatório para medir a preparação dos estagiários. Como os estudos de caso anteriores, a extração ocorre na base principal da PIT e quando necessário, ocorrem algumas transformações antes de enviar os dados para o *data warehouse*. A companhia PIT utiliza o *Oracle Warehouse Builder* (OWB) e o protótipo de ETL baseado em SOA como ferramentas de solução para o desenvolvimento do ambiente de *data warehouse* (AWAD, ABDULAHB, *et al.*, 2012).

Figura 11. Star Schema das tabelas fato e dimensão extraída do banco de dados PIT



Fonte: (AWAD, ABDULAHB, *et al.*, 2012)

As ferramentas tradicionais de ETL foram utilizadas nos três estudos de casos, simulando a situação atual das organizações. As tabelas de dados são criadas e gerenciadas através das ferramentas tradicionais de ETL. Cada uma dessas ferramentas é instalada em um servidor com o propósito de extrair determinados campos do banco de dados de origem e carrega-los em um repositório do *data warehouse*. A ferramenta tradicional de ETL é usada para extrair as informações de contexto necessárias para a realização do processo de negócio que permite a geração dos relatórios estatísticos. Um servidor é usado para executar as funcionalidades de ETL de cada estudo de caso e a ferramenta de ETL tradicional é instalado como um aplicativo de *desktop* do servidor (AWAD, ABDULAHB, *et al.*, 2012).

O protótipo de ETL baseado em SOA é implantado ao longo de quatro servidores diferentes em cada estudo de caso, dois servidores com o sistema operacional *Linux RedHat* e os outros dois com os sistemas operacionais *Windows Server* 2003. Os servidores de aplicação JBOSS são usados como ambiente de execução e são instalados nos servidores com os sistemas operacionais *RedHat Linux*, o servidor de aplicação *GlassFish* de código aberto é instalado nos servidores com o sistema operacional Windows Server 2003. Cada um desses quatros servidores de aplicação tem o seu próprio recipiente de componentes distribuídos, que no caso é o EJB (AWAD, ABDULAHB, *et al.*, 2012).

O protótipo de ETL baseado em SOA é composto por três componentes, sendo que o componente de extração é implantado separadamente em um dos servidores de

aplicação JBOSS, o componente de transformação é implantado separadamente em outro servidor de aplicação JBOSS (AWAD, ABDULAHB, *et al.*, 2012).

O componente de extração de ETL do protótipo baseado em SOA é implantado separadamente em um dos servidores de aplicação JBOSS, o componente de transformação do protótipo de ETL e o componente de carga é implementado separadamente em um dos servidores de aplicação *GlassFish*. O *Orchestration Point* é implantado no último servidor de aplicação *GlassFish* disponível. Além disso, o componente de administrador de ETL (cliente) é implantado separadamente no mesmo servidor de aplicativos onde está o *Orchestration Point*. Em seguida, a execução das funcionalidades de ETL é feito utilizando navegadores de computadores padrões de clientes, portanto, o protótipo de ETL baseado com SOA é utilizado para extrair, transformar e carregar os campos necessários para a criação do processo de negócio que permite a geração de relatórios estatísticos (AWAD, ABDULAHB, *et al.*, 2012).

Com base em fundamentos teóricos aplicado no novo *framework* de ETL baseado em SOA, existem dois objetivos que precisam ser alcançados. O primeiro objetivo é relacionado com a principal funcionalidade de ETL em geral, enquanto o segundo objetivo está relacionado com a utilização da distribuição e interoperabilidade dos componentes da estrutura de ETL (AWAD, ABDULAHB, *et al.*, 2012).

Como definição da primeira meta: a extração, transformação e a carga de dados em seu formato final para o repositório do *data warehouse* para ser adequado na camada de apresentação das atividades de inteligência de negócio. Este objetivo foi alcançado, tanto na ferramenta de ETL tradicional como no protótipo de ETL baseado em SOA. Os dados extraídos, as transformações e os dados carregados que são necessários para a geração de relatórios estatísticos, são obtidos exatamente semelhante em ambas ferramentas (AWAD, ABDULAHB, *et al.*, 2012).

Ao comparar as duas formas de utilizar a ferramenta de ETL tradicional e o protótipo de ETL baseado em SOA, observa-se que a segunda meta é atingida através do desenvolvimento do novo framework de ETL baseado em SOA, pois, ao utilizar a

ferramenta tradicional de ETL percebe-se que não atinge a maiorias das etapas da meta estabelecida. Ao utilizar a ferramenta tradicional de ETL, conclui-se que ela é uma ferramenta fortemente acoplada e os seus componentes não podem ser distribuídos. Além disso, toda a ferramenta está instalada em um único servidor por causa da impossibilidade da distribuição dos seus próprios componentes. Por consequência, a distribuição e a interoperabilidade entre os componentes de ETL não poderiam ser obtidos quando a ferramenta tradicional de ETL é utilizada. Por outro lado, seguindo o protótipo de ETL baseado em SOA, cada componente do protótipo de ETL é distribuído e implantado ao longo de um recipiente EJB separados em diferentes servidores de aplicação. E para cada um desses componentes existem o processo de interoperabilidade, porque todos os componentes de extração, transformação e carga podem se comunicar de forma mais transparente possível. Portanto, a distribuição e a interoperabilidade são atingidas entre os componentes de ETL (AWAD, ABDULAHB, *et al.*, 2012).

4.5. Estudo de caso baseado em ontologia

Existem três tipos de heterogeneidade em fontes de dados, tais como: sintaxe, esquema e semântica. Assim, a interoperação de dados torna-se uma tarefa difícil na integração de dados. A heterogeneidade sintática é causada pelo uso de diferentes modelos ou linguagens. Heterogeneidade de esquema resulta em diferentes formas estruturais. A heterogeneidade semântica é causada por diferentes significados ou interpretações de dados em vários contextos. Para alcançar a interoperabilidade nos dados, essas questões colocadas de heterogeneidades precisam ser eliminadas (CRUZ e XIAO, 2005).

Com a chegada do XML tem se criado uma plataforma sintática para a alteração e padronização de dados na *Web.* No entanto, a heterogeneidade de esquema pode persistir dependendo dos esquemas de XML que são utilizados. Do mesmo modo que a heterogeneidade semântica pode persistir se ambas heterogeneidades de sintaxe e esquema não ocorram (CRUZ e XIAO, 2005).

A integração de dados semântico é o processo de utilizar uma representação conceitual dos dados e suas relações para eliminar possíveis problemas de heterogeneidade. Na raiz da integração de dados semântico, está concentrada a importância das ontologias, que é uma especificação explícita dos conceitos compartilhados. As ontologias foram criadas pela comunidade da Inteligência Artificial com o objetivo de facilitar o compartilhamento de conhecimento e a reutilização. Levando a semântica para domínios específicos, ontologias são amplamente utilizadas para a representação do conhecimento de um domínio (CRUZ e XIAO, 2005).

As ontologias têm sido usadas em sistemas de integração de dados, porque elas fornecem uma conceptualização explícita e compreensível por máquina de um determinado domínio. Normalmente elas são utilizadas em uma das três seguintes maneiras, conforme Cruz e Xiao (2005):

- **Abordagem individual de ontologia:** todos os esquemas de origens são diretamente relacionados a uma ontologia global compartilhada que fornece uma interface uniforme para o usuário. No entanto, esta abordagem requer que todas as origens tenham o mesmo ponto de vista de um domínio, com o mesmo nível de granularidade;
- **Abordagem múltipla de ontologia:** cada fonte de dados é descrita pela sua própria ontologia local separadamente. Em vez de utilizar uma ontologia em comum, as ontologias locais são mapeadas entre si. Para esse efeito, é adicionado uma representação formal para definir o mapeamento inter-ontologia;
- **Abordagem hibrida de ontologia:** A combinação das duas abordagens anteriores é usada. Em primeiro lugar, uma ontologia local é construída para cada esquema de origem, que, no entanto, não é mapeada para outra ontologia local, mas a uma ontologia global compartilhada. Novas fontes de dados podem ser facilmente adicionadas sem ter a necessidade de modificar os mapeamentos existentes.

A abordagem individual e a híbrida são adequadas para a construção de um sistema central de integração de dados, sendo a primeira mais indicada para os

sistemas *Global as View* – GAV e a última para sistemas *Local as View* – LAV. A abordagem múltipla de ontologia pode ser melhor utilizada para construir sistemas limpos ponto-a-ponto de integração de dados.

Em seguida, ainda segundo Cruz e Xiao (2005) são apresentadas cinco utilizações de ontologias em sistema de integração de dados. E para cada uma delas será representado por um estudo de caso.

- **Representação de Metadados:** Os metadados (por exemplo, esquema de origem) podem ser representados através de ontologias locais, utilizando uma única linguagem;
- **Conceptualização Global:** a ontologia global fornece uma visão conceitual sobre os esquemas de origem esquematicamente heterogêneos;
- **Suporte para consulta de alto nível:** dada uma visão e alto nível das fontes, como previsto por uma ontologia local, o usuário pode formular uma consulta sem conhecimentos específicos das diferentes fontes de dados. Então, a consulta e rescrita sobre as fontes com base nos mapeamentos semânticos entre as ontologias globais e locais;
- **Mediação declarativa**: processamento de consulta em sistemas híbridos ponto-a-ponto utilizam ontologia global como um mediador declarativo para regravação de consultas;
- **Suporte de mapeamento:** A enciclopédia, formalizado em termos de ontologias, podem ser utilizadas para o processo de mapeamento para facilitar a automatização.

4.5.1. Estudo de caso: Representação de Metadados

Na sequência serão apresentados três estudos de caso de ontologias no contexto de integração de dados central. Será usado um exemplo envolvendo duas fontes de dados XML para demonstrar como habilitar a semântica e a interoperabilidade entre eles.

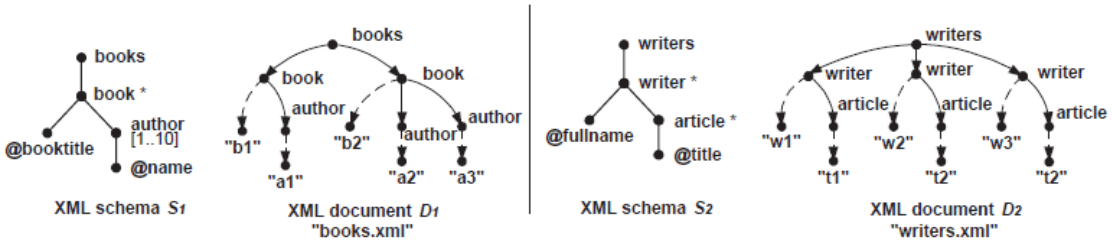
Exemplo 1: A figura 12 mostra dois esquemas XML (S1 e S2) e seus respectivos documentos (D1 e D2), que são representados como árvores. Os dois documentos

XML estão em conformidade com diferentes esquemas, mas representam dados com semelhança semântica. Em particular, ambos esquemas representam um relacionamento de muitos-para-muitos entre dois conceitos: o livro e o autor em S1 (equivalente ao artigo e ao escritor em S2). No entanto, estruturalmente eles são diferentes: o S1 tem o elemento autor alinhado sob o elemento livro, enquanto que o S2 tem o elemento artigo alinhado sob o elemento escritor. Elementos de dados semanticamente equivalentes, como os autores da publicação “b2”, podem ser selecionados usando diferentes padrões de caminhos XML, respectivamente para os esquemas S1 e o S2 (CRUZ e XIAO, 2005):

/books/book[@booktitle="b2"]/author/@name
e
/writers/writer[article/@title="b2"]/@fullname

Onde a restrição para especificar a pesquisa está no conteúdo entre os colchetes.

Figura 12. Duas origens XML com heterogeneidade de esquemas

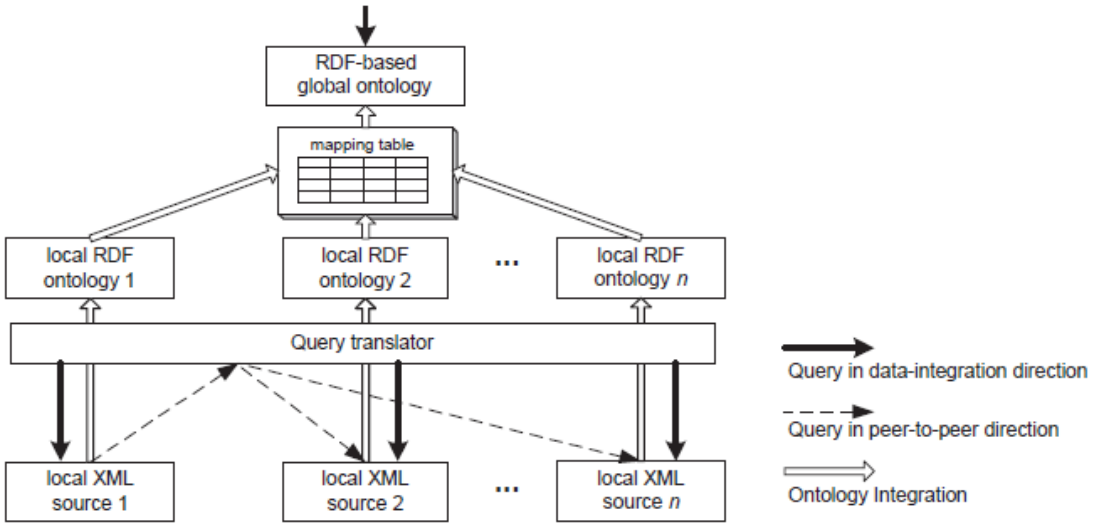


Fonte: (CRUZ e XIAO, 2005)

Esse exemplo demostra que vários esquemas ou estruturas XML podem existir para um único modelo conceitual. Em comparação, o esquema ou a linguagem de ontologia (por exemplo: RDF, DAML + OIL e OWL) que opera em um nível conceitual é estruturalmente equivalente, de modo que os usuários possam formular uma consulta a partir do modelo conceitual, sem considerar a estrutura da fonte de dados (CRUZ e XIAO, 2005).

A figura 13 mostra a arquitetura de um sistema que interage entre as fontes de dados esquematicamente. Os próximos 3 casos estudam em detalhes os princípios dessa arquitetura.

Figura 13. Uma arquitetura de XML para a integração de dados



Fonte: (CRUZ e XIAO, 2005)

O primeiro passo para construir uma ponte entre a heterogeneidade de diversas fontes de dados locais, uma ontologia local deve ser gerada a partir de cada fonte de esquema do banco de dados (por exemplo: relacional, XML ou RDF). A ontologia local é uma conceituação dos elementos e as relações entre os elementos de cada esquema de origem. Para facilitar a interoperabilidade, essas ontologias devem utilizar o mesmo modelo para serem representadas. Além disso, por uma questão de processamento correto de consultas, a estrutura de esquemas de origem e as restrições de integridade (por exemplo: chaves estrangeiras) representadas nos esquemas devem ser preservadas na ontologia local. Foi escolhido o RDFS para representar cada ontologia local (CRUZ e XIAO, 2005).

A geração de ontologias a partir de esquemas de origem é efetuada pela transformação de esquemas baseado em modelos. Em particular, as seguintes abordagens são levadas pelas transformações dos esquemas relacionais e XML (CRUZ e XIAO, 2005).

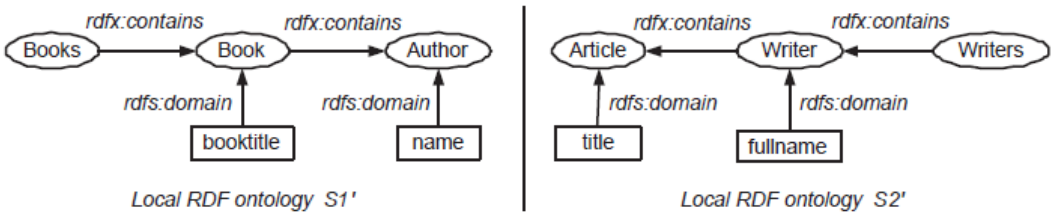
- **Esquema relacional**: relações são convertidas em classes RDF e atributos em propriedades RDF, que são ligadas as suas classes correspondentes. Dependência entre chaves estrangeiras entre duas relações são representados

por duas propriedades (correspondentes as duas relações) compartilhando o mesmo valor no alvo da ontologia local (DATE, 2004).

- **Elementos do esquema XML**: elementos do tipo complexo são convertidos em classes RDF e elementos do tipo simples e atributos são convertidos em atributos RDF. Esse processo de transformação codifica as informações de mapeamento entre cada conceito na ontologia local RDF e o caminho para o elemento correspondente na fonte XML. Relações entre elementos XML são representadas usando um meta-propriedade: *rdfx:contains*, o *rdfx* representa o *namespace* onde *contains* é definido. Esta meta-propriedade permite o RDF representar o exato lugar da estrutura do XML, conectando as duas classes RDF que representam os dois elementos XML (CRUZ e XIAO, 2005).

Exemplo 2 – Seguindo o exemplo 1, a figura 14 mostra a ontologia local RDF (S1 e S2), que são geradas, respectivamente, entre os esquemas de origem XML (S1 e S2) (CRUZ e XIAO, 2005).

Figura 14. RDF baseado em ontologias locais geradas a partir de esquemas XML



Fonte: (CRUZ e XIAO, 2005)

4.5.2. Estudo de caso: Conceptualização Global

Para tornar o sistema de integração acessível através de uma interface uniforme da ontologia global, mapeamentos semânticos são estabelecidos entre a ontologia global e as ontologias locais. O processo de mapeamento é realizado durante a construção da ontologia global, que é gerenciado pela fusão das ontologias locais, como por exemplo a utilização da abordagem GAV (CRUZ e XIAO, 2005). É considerado que cada ontologia local é incorporada pela ontologia global até a ontologia alvo, esse

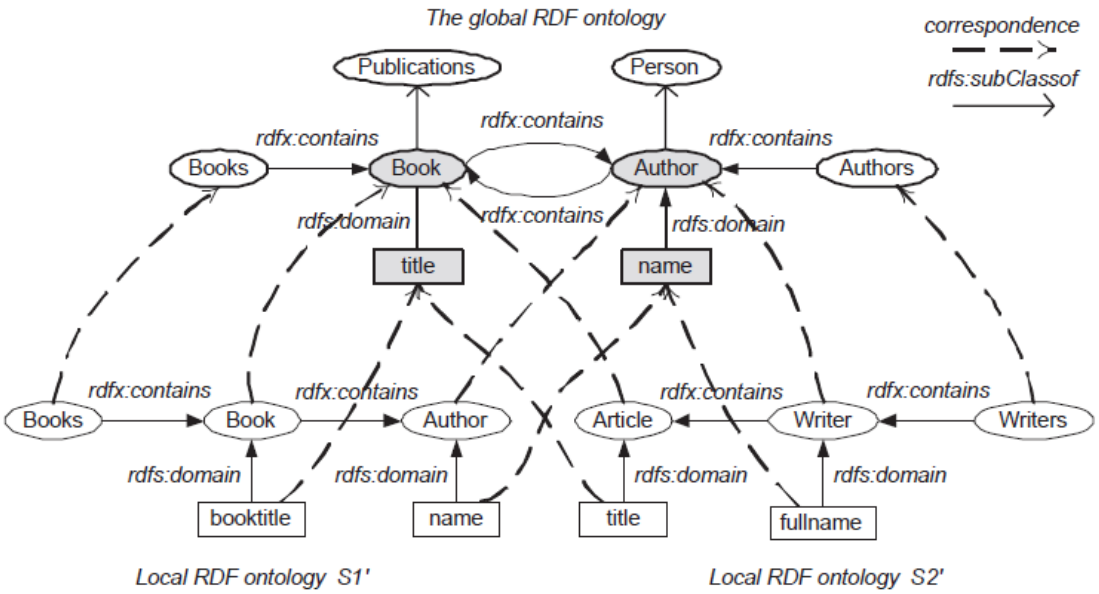
processo de incorporação de ontologias consiste em diversas operações, conforme destacado por Cruz e Xiao (2005):

- **Cópia das classes e/ou suas propriedades**: são copiados para a ontologia alvo todas as classes e/ou propriedades que não existem nela;
- **Incorporação de classes**: classes conceitualmente equivalentes de uma classe na ontologia local e de destino são combinadas em uma classe na ontologia alvo;
- **Incorporação de propriedades**: propriedades conceitualmente equivalentes de uma propriedade na ontologia local e de destino são combinadas em uma propriedade na ontologia alvo;
- **Incorporação de relacionado**: relações conceitualmente equivalentes a partir da classe C1 para a outra classe C2 entre a ontologia local e de destino são combinadas em uma única relação na ontologia de destino;
- **Generalização de classes**: classes relacionadas nas ontologias locais e de destino podem ser generalizadas para uma superclasse. A superclasse pode ser obtida pesquisando um conhecimento existente do domínio.

Através das operações acima as correspondências semânticas são estabelecidas. Por exemplo, para cada elemento de PL em uma ontologia local, caso exista um elemento equivalente em PG na ontologia global, os dois elementos são fundidos e uma correspondência entre PL e PG é gerada (CRUZ e XIAO, 2005).

Exemplo 3 – A figura 15 mostra as ontologias RDF Globais geradas pela fusão entre as ontologias locais S1 e S2 do exemplo 2. Note que as classes (propriedades) representadas em cinzas são classes incorporadas, e a classe **Livro** e **Autor** também são estendidas, por meio da **Publicação** e **Pessoa** que são suas respectivas superclasses (CRUZ e XIAO, 2005).

Figura 15. Uma visão conceitual sobre as fontes originais



Fonte: (CRUZ e XIAO, 2005)

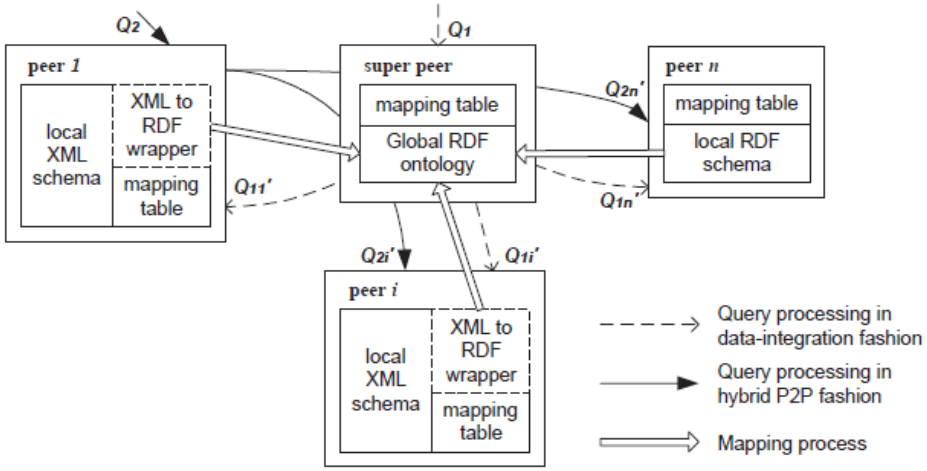
4.5.3. Estudo de caso: Suporte para consultas de alto nível

Dada uma visão conceitual das fontes de informações disponíveis, o usuário pode representar uma consulta em termos da ontologia global. A consulta é considerada de alto nível quando sua formulação não requer identificação de um determinado esquema de origem. A consulta é reformulada por um algoritmo em uma subconsulta para cada origem. As subconsultas através das fontes estão sujeitas a estrutura de esquemas de origens que são expressadas em diferentes linguagens desde que seja utilizada uma consulta de alto nível. Um mecanismo de inferência pode ser necessário para reescrever uma consulta, por exemplo, quando um envolvimento de conceito na consulta possui superconceitos ou subconceitos (CRUZ e XIAO, 2005).

Além de lidar com consultas de alto nível sobre a ontologia global, uma consulta de algoritmo de tradução bidirecional também é suportada. Neste caso, pode-se traduzir uma consulta colocada contra uma fonte de XML para uma consulta equivalente contra qualquer outra fonte XML (CRUZ e XIAO, 2005).

Exemplo 4 – suponha que o usuário solicita a consulta “Encontrar as pessoas que tem publicação escrita b2”. Essa consulta será expressa em uma linguagem de consulta RDF como a RDQL. Primeiro, Pessoa tem subconceito de Autor, que correspondem a dois conceitos diferentes (Autor e Escritor) em dois diferentes bancos de dados locais RDF. Portanto, a consulta inicial será reescrita em duas subconsultas para esses bancos de dados. Por sua vez, as consultas podem ainda ser reescritas usando uma linguagem de consulta XML incorporando as expressões do caminho do Exemplo 1 (a menos que os dados foram materializados sob as ontologias locais RDF). Usando o mecanismo de consulta transacional bidirecional, uma consulta que envolve os conceitos Livro e Autor em uma fonte será traduzida em uma consulta envolvendo Artigo e Escritor de outra fonte de dados, utilizando as correspondências estabelecidas pela ontologia global (CRUZ e XIAO, 2005).

Figura 16. Uma arquitetura ponto-a-ponto do sistema PEPSINT



Fonte: (CRUZ e XIAO, 2005)

Considerando novamente as duas fontes XML da figura 12, no entanto, desta vez estão conectados em uma arquitetura ponto-a-ponto. É considerado uma arquitetura hibrida ponto-a-ponto com dois tipos de *peers*: *super-peers* contendo uma ontologia global RDF, e cada *peers* contendo uma fonte de dados e uma ontologia. Cada *peers* representa um sistema de informação autônomo e se conecta a um *super-peers* através de mapeamentos semânticos. Os frameworks ou sistemas de integração

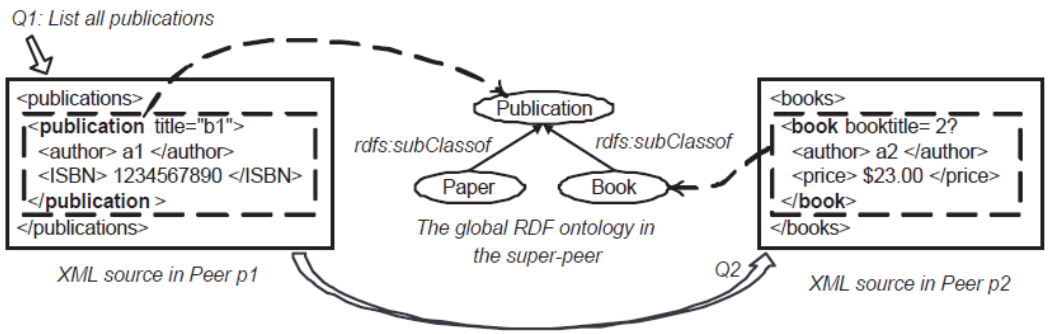
ponto-a-ponto incluem um Modelo Relacional Local (*Local Relational Model* – LRM) (CRUZ e XIAO, 2005).

4.5.4. Estudo de caso: Mediação declarativa

O PEPSINT é um sistema híbrido ponto-a-ponto cuja a arquitetura é mostrada na figura 16. PEPSINT utiliza a abordagem GAV. A ontologia global em uma *super-peer* tem duas funções: (I) Ele fornece ao usuário uma visão de alto nível uniforme das fontes de dados nos pontos distribuídos, e (II) serve como um mediador para a tradução de uma consulta a partir de um ponto para o outro. A primeira função é semelhante à descrita no estudo de caso 3 e esta última função é descrita em detalhes a seguir (CRUZ e XIAO, 2005).

Os usuários podem representar uma consulta contra a fonte XML ou fonte de dados RDF em qualquer ponto. Localmente, a consulta será executada na origem para obter uma resposta local. Enquanto isso, a consulta de origem é reescrita em uma consulta alvo ao longo de cada ponto ligado. A regravação de consulta utiliza a ontologia global, e a composição de mapeamentos a partir de pares originais para o *super-peers* com mapeamento do *super-peers* para os pontos alvos. Ao executar o objetivo da consulta, cada ponto retorna uma resposta para o ponto original, chamado de resposta remota. As respostas locais e remotas são integradas e devolvidas ao utilizador no local onde o ponto foi originado (CRUZ e XIAO, 2005).

Figura 17. Mediação ponto-a-ponto para reescrita de consulta



Fonte: (CRUZ e XIAO, 2005)

Exemplo 5 – Considerando duas fontes XML, um ponto em p1 e o outro ponto em p2, e uma ontologia global expressa em RDF em um *super-peers*. Como mostrado na figura 17, a ontologia global é constituída pela classe **Publicação** e duas subclasses **Artigo** e **Livro**. A classe **Publicação** é mapeada para o elemento **publicação** da fonte XML em p1, enquanto a classe **Livro** corresponde à **livro** da fonte XML em p2. Uma consulta XML Q1 em p1 envolvendo **publicação** será reescrita para uma consulta no alvo Q2 em p2 envolvendo a inclusão de **livro**. Os fragmentos XML no interior das caixas de linhas tracejadas são integrados e devolvidos como respostas (CRUZ e XIAO, 2005).

4.5.5. Estudo de caso: Suporte de mapeamento

Um dicionário pode ser usado para a integração de dados para facilitar a automatização do processo de mapeamento de esquema. Em particular, ele pode ajudar a descobrir a relação semântica entre os conceitos em diferentes esquemas ou ontologias. *WordNet* é um exemplo desse dicionário. É constituído por uma rede e suas relações semânticas, por exemplo: sinônimo, hiperonímia e hipônimo. O termo pode ter múltiplos sentidos, cada um sendo o *sysnset* (CRUZ e XIAO, 2005).

A abordagem do esquema de harmonização baseado em dicionário foi concebida para o sistema de integração de dados ponto-a-ponto. Essa abordagem consiste nos três seguintes passos (Conforme ilustrado na figura 18):

- **Caminho de exploração:** por meio das relações semânticas entre *sysnset* em *WordNet*, foram escolhidos aqueles mais específicos, ou seja, os sinônimos, hiperonímia e hipônimo. E relacionado ao enumerar os caminhos entre dois conceitos arbitrários de diferentes ontologias locais. Como mostrado na figura 18, seis caminhos são encontrados a partir de quantidade para números (CRUZ e XIAO, 2005).
- **Seleção de caminho:** quando vários caminhos são encontrados entre dois conceitos, podemos escolher o caminho ideal, que corresponde a relação semântica mais provável entre dois conceitos. Para este efeito, as semelhanças semânticas são calculadas para todos os caminhos. O cálculo é implementado através da atribuição de diferentes relações semânticas com pesos diferentes, por exemplo: 1.0 para sinônimo e 0.8 para hiperonímia, e em seguida, tendo a média de todos os pesos. O caminho com maior similaridade é então escolhido

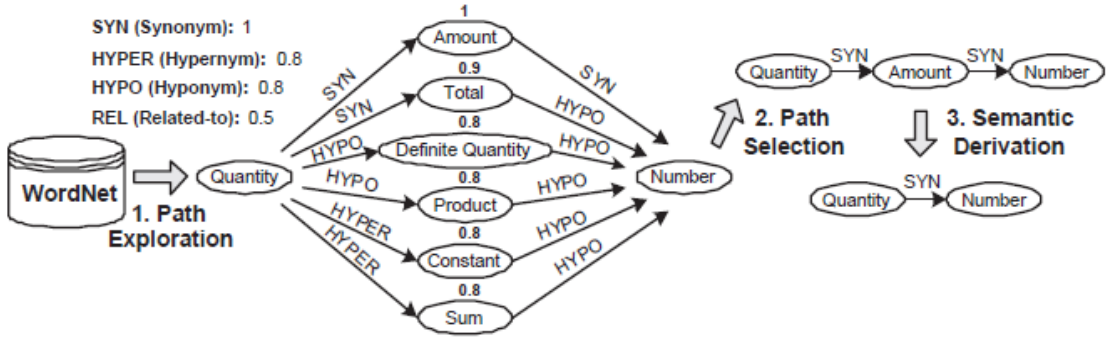
como o caminho o ideal. Se houver mais do que um caminho, então é necessário a intervenção do utilizador (CRUZ e XIAO, 2005).

Derivação semântica: o último passo é derivar a relação semântica, "***Sem***", entre os dois conceitos sobre as relações semânticas ao longo do caminho ideal de *P* entre eles. Mas especificamente, ***Sem***(p) = ***Sem***(pn) é calculado com base no seguinte algoritmo recursivo, onde pn = (R1, R2, ... , Rn) e Ri (1<=i<=n) são as relações semânticas ao longo de *P* (CRUZ e XIAO, 2005).

Sem*(pn)* = ***Sem****(pn-1)* ^ ***Sem****(rn), if n > 1*; (1)

Sem*(pn)* = $\approx, \supseteq, \subseteq ou \sim$ *if n = 1; (2)*

Figura 18. Processo de mapeamento de esquema baseado em dicionário



Fonte: (CRUZ e XIAO, 2005)

Nas formulas anteriores, os símbolos ≈, ⊇, ⊆, e ~, representam respectivamente a relação semântica de sinônimo, hiperonímia e hipônimo. A relação ^ obedece às regras que são mostradas na Tabela 3.

Tabela 3. Regras de interferência para relações semânticas

$\wedge$	$\approx$	$\supseteq$	$\subseteq$	$\sim$
$\approx$	$\approx$	$\supseteq$	$\subseteq$	$\sim$
$\supseteq$	$\supseteq$	$\supseteq$	?	$\sim$
$\subseteq$	$\subseteq$	?	$\subseteq$	$\sim$
$\sim$	$\sim$	$\sim$	$\sim$	$\sim$

Fonte: (CRUZ e XIAO, 2005)

Regras de interferência para relações semânticas: nas células brancas (na interseção de cada par de células amarelas) contêm o resultado da operação sobre as relações de duas células amarelas, e um ponto de interrogação indica a intervenção humana necessária (CRUZ e XIAO, 2005).

Com o surgimento do XML, foi criado uma plataforma sintática para dados padronizados na *web*. No entanto, existem diversos problemas com XML. Em primeiro lugar, os documentos expressam a mesma sintaxe em XML compartilhados, mas podem ter outra representação forma heterogênea, por exemplo: tendo diferentes estruturas e convenções de nomenclatura. Além disso, um documento XML não expressa a semântica dos elementos ou das relações entre os elementos de maneira explicita. Por consequência, não é uma linguagem adequada para a representação de metadados (CRUZ e XIAO, 2005).

Ontologias fornecem uma definição explicita e formal de uma conceptualização compartilhada e são capazes de facilitar o compartilhamento do conhecimento e da reutilização. Foi utilizado as ontologias expressas em RDFS, uma linguagem de esquema semanticamente poderosa, para interligar através de heterogeneidade de sintática, esquemas e semânticas as fontes de dados. Foram apresentados cinco estudos de casos que ilustram o papel de ontologias no processo de integração de dados nas arquiteturas centralizada e ponto-a-ponto (CRUZ e XIAO, 2005).

Investigações relacionadas incluem pesquisas sobre a geração de ontologias, mapeamento e evolução de ontologias. Uma ontologia pode ser gerada manualmente, utilizando uma ferramenta de autoria ou semi-automaticamente a partir de várias fontes de conhecimento, por exemplo: os esquemas de banco de dados. As técnicas

utilizadas para o mapeamento de ontologias, incluindo o alinhamento e a junção de ontologias, se sobrepõem em grande medida com as técnicas de correspondência de esquema. E finalmente, a evolução da ontologia, envolve mudanças na representação, estrutura e semântica das ontologias. Cada passo de uma evolução deve assegurar a coerência entre a versão antiga e nova da ontologia, assim como a evolução de esquema de banco de dados deve garantir a consistência de um novo esquema de banco de dados (CRUZ e XIAO, 2005).

4.6. Considerações do capítulo

Neste capítulo foi apresentada que a importância do processo de ETL consome cerca de 70% do esforço do desenvolvimento e manutenção de um sistema de *data warehouse*. Além disso, o processo concentra-se na extração e padronização de diversos formatos e *layouts* de fonte de dados.

Apesar de existirem diversos assuntos de atenção no processo de ETL, foram identificados e apresentados dois pontos que devem ser considerados no desenvolvimento desta fase no processo de *data warehouse*. O primeiro é a dificuldade do gerenciamento dos componentes de extração, transformação e carga de dados, por não existir distribuição e interoperabilidade entre eles. E o segundo é a existência de heterogeneidade entre as fontes de dados que devem ser integrados.

Para esses dois pontos de atenção, foram apresentados dois *frameworks* para minimizar esses problemas. Sendo que um é baseado em SOA que contribui com a distribuição e interoperabilidade dos componentes de ETL e o outro utilizando técnicas de ontologias tem como objetivo corrigir problemas de heterogeneidade entre fonte de dados.

5. Considerações finais

Este capítulo descreve as considerações finais, contribuições do trabalho e sugestões de trabalhos futuros que poderão ser desenvolvidos a partir deste trabalho.

Embora as organizações encontrem um enigma na implantação e manutenção dos sistemas de *data warehouse*, este trabalho apresentou o processo de extração, transformação e carga de dados, sendo possível conhecer a estrutura e os componentes necessários para o desenvolvimento desta fase em um ambiente de *data warehouse*.

As experiências apresentadas no capítulo 4 mostram que ambos dos *frameworks* podem ser implantados juntos para solucionar problemas diferentes. Apesar de que as ferramentas tradicionais de ETL atingirem seu objetivo principal de extração, transformação e carga de dados, percebe-se que elas são fortemente acopladas, não permitindo a distribuição de seus componentes. Além disso, o *framework* de integração de dados baseado em ontologia mostrou de uma maneira explicita que é capaz de solucionar problemas de heterogeneidade de fonte de dados compartilhando o conhecimento de um determinado domínio ou conceito o que facilita na sua reutilização.

Portanto, essas técnicas podem ser utilizadas por empresas para apoiar no desenvolvimento e manutenção dos processos de ETL desde que sejam diagnosticados possíveis problemas de heterogeneidade na base ou na semântica dos dados e quando essas informações ficam fisicamente separadas em servidores hospedados em lugares geograficamente diferentes.

5.1. Contribuições do Trabalho

Este trabalho apresentou uma análise do processo de ETL de um sistema de *data warehouse* através de um mapeamento baseado nos frameworks de Ontologia e SOA.

Ambos os *frameworks*, através de suas experiências, mostraram-se consistentes em suas propostas. Além disso, faz com que o ETL seja mais eficiente e flexível em todo o seu processo.

Também foram identificadas vantagens quando aplicada a restruturação do processo de ETL utilizando o framework SOA, são elas: Reutilização, Escalabilidade e Eficiência de Custo.

5.2. Trabalhos futuros

Ao finalizar esse trabalho, percebe-se que alguns estudos podem ser realizados, tais como:

- Generalizar os métodos de mapeamento e de raciocínio no *framework;*
- Definir as operações de SQL que correspondem às normas de ETL;
- Projetar um esquema conceitual do *data warehouse* automaticamente.

6. Referências

AWAD, M. et al. Service oriented architecture based ETL framework vaidation. **AwedProcedia Information Technology & Computer Science**, Sintok, n. 2, p. 285 - 289, 2012.

AWAD, M.; ABDULLAH, M. S. A Framework for Interoperable Distributed ETL Components Based on SOA. **IEEE**, Sintok, n. 2, p. 67-70, 2010. ISSN 978-1-4244-8666-3/10.

BALLARD, C. et al. **Dimensional Modelin - In a Business Intelligence Environment.** 1°. ed. [S.l.]: Redbooks, 2006.

CRUZ, I. F.; XIAO, H. The Role of Ontologies in Data Integration. **Engineering intelligent systems for electrical engineering and communications**, Chicago, p. 245, 2005.

DATE, C. J. **Introdução à Sistema de Banco de Dados.** 8°. ed. São Paulo: Campus, 2004.

DUPOR, S.; JOVANOVI, V. An approach to conceptual modelling of ETL processes. **IEEE**, Statesboro, p. 1485-1490, Maio 2014.

ECKERSON, W. **Smart Companies in the 21st Century:** The Secrets of Creating Successful Business Intelligent Solutions. Seattle: WA:The Data Warehousing Institute., 2003.

GARTNER, G. I. Gartner Group. **Site da consultoria Gartner Group**, 02 abril 2014. Disponivel em: <http://www.gartner.com/technology/media-products/newsletters/information_builders/vol3_issue1/gartner.html>. Acesso em: 12 Novembro 2014.

GUARINO, N.; GIARETTA, P. Ontologies and knowledge bases: Towards a terminological clarification. **Towards very large knowledge bases**, Amersterdam, n. 1°, p. 25-32, 1995.

INMON, W. H. **Building the Data Warehouse. Fourth Edition.** 4°. ed. Indianapolis: Wiley, 2005.

KEEN, M. et al. **Patterns:** Implementing an SOA Using an Enterprise Service Bus. 1°. ed. [S.l.]: Redbooks, 2004.

KIMBALL, R. et al. **The Data Warehouse Lifecycle Toolkit**. 1°. ed. Indianapolis: Wiley, 2008.

KIMBALL, R.; CASERTA, J. **The Data Warehouse ETL Toolkit**. 1°. ed. Indianapolis: Wiley, 2004.

LIMA, J. C.; CARVALHO, C. L. Ontologias - OWL (Web Ontology Language), Junho 2005.

SHEDROFF, N. Information Interaction Design: a Unified Field Theory, 1999. Disponivel em: <>. Acesso em: 22 jun. 2015.

THOMPSON, O. **Business Intelligence Success, Lessons Learned**. [S.l.]: TechnologyEvaluation.com, 2004.

TURBAN, E. et al. **Business Intelligence - Um enfoque gerencial para a inteligência do negócio**. São Paulo: Bookman, 2009.

VASSILIADIS, P. et al. Blueprints and Measures for ETL Workflows. **Springer-Verlag Berlin Heidelberg**, Athens, p. 385 – 400, 2005.

VASSILIADIS, P.; SIMITSIS, A.; SKIADOPOULOS, S. Conceptual Modeling for ETL Processes. **IEEE**, Athens, Novembro 2002.

VIEIRA, A.; GUIMARÃES, G.; CAETANO, D. Análise de Custo das Reinternações Hospitalares. **Qualihosp**, São Paulo, p. 206 - 210, 2013.

WANG, H.; YE, Z. An ETL Services Framework Based on Metadata. **IEEE**, Wuhan, 2010.

ZHANG, Z.; WANG, S. A Framework Model Study for Ontology-driven ETL Processes. **IEEE**, Shanghai, 2008.