# SelfBI: Tomada de decisão sob demanda do usuário utilizando dados da Web*

**Manoela Camila Barbosa da Silva** [1], **Sahudy Montenegro González** [1]

[1]Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
Rod. SP-264 KM 110 – 18.052-780 – Sorocaba – SP – Brazil

manoelacamila.silva@gmail.com,sahudy@ufscar.br

***Abstract.*** *Data from Web sources, such as social media, tend to be volatile to be stored in the DW, making them a good option for situational data. Situational data are useful for decision-making queries at a particular time and situation, and can be discarded after analysis. This article describes an architecture that aims to integrate situational data from social media to user queries at the right time; this is, when the user needs them for decision making. The focus of the work is (1) the ETL process for obtaining situational data in real time; and (2) to propose an OLAP operator capable of integrating these data into user queries results.*

***Resumo.*** *Os dados provenientes de fontes Web, como mídias sociais, tendem a ser voláteis para serem armazenados no DW, tornando-se uma boa opção para dados transitórios. Os dados transitórios são úteis para consultas de tomada de decisão em um determinado momento e cenário e podem ser descartados após a análise. Este artigo descreve uma arquitetura que visa integrar dados transitórios, provenientes de mídias sociais, às consultas de usuários no momento certo em que o usuário necessita deles para a tomada de decisão. O foco do trabalho se encontra (1) no processo de ETL para obtenção de dados transitórios em tempo real; e (2) na proposta de um operador OLAP que integra esses dados aos resultados das consultas dos usuários.*

## 1. Introdução

O crescimento exponencial e contínuo do volume de informações sendo disponibilizadas por meio da Web tem causado a necessidade de evolução dos mecanismos de *Business Intelligence* (*BI*) tradicionais. Essa evolução diz respeito, principalmente, à obtenção de dados em tempo real e à utilização da riqueza e diversidade de informações provenientes da Web e suas mídias sociais como fonte de dados [Abello et al. 2015][Mansmann et al. 2014].

Dentre as abordagens propostas para obtenção de dados em tempo real se encontram os *Right-Time Data Warehouses*, que se baseiam na ideia de que os dados devem ser buscados na fonte e atualizados no *Data Warehouse* (*DW*) conforme forem necessários [Thomsen et al. 2008]. O princípio é que os dados não precisam ser constantemente atualizados, mas sim estarem atualizados no momento em que o usuário necessita deles para a tomada de decisão. A utilização de dados Web traz alguns desafios aos sistemas de suporte à decisão tradicionais, devido, principalmente, à natureza não estruturada dos dados. Um deles é a necessidade de adaptação dos operadores *OLAP*, uma vez que os operadores tradicionais não foram projetados para lidar

*Este trabalho é resultado do Auxílio Regular à Pesquisa FAPESP Processo No. 2011/12115-1

com esse tipo de dado [González and Berbel 2014]. Além disso, como exposto por [Etcheverry and Vaisman 2012], os dados provenientes de fontes como mídias sociais tendem a ser muito voláteis para serem armazenados permanentemente no *DW*, tornando-se uma boa opção como dados transitórios. Dados transitórios são úteis para a tomada de decisão em um determinado momento e cenário, mas após tal análise podem ser descartados, pois não enriquecem o *DW* em relação às demais consultas.

Este artigo descreve uma arquitetura que visa integrar os conceitos de *Right-Time DW* e dados transitórios, permitindo ao usuário consultar dados armazenados no modelo multidimensional tradicional (chamados de estacionários) com dados provenientes de mídias sociais. O foco do trabalho se encontra no processo de *ETL* dos dados transitórios e na proposta de um novo operador *OLAP*. O objetivo do operador é integrar, ao cubo da consulta, dados extraídos de fontes Web, de maneira a apresentar nova informação útil ao usuário.

O restante deste documento está organizado em seções. A Seção 2 expõe os trabalhos relacionados. A Seção 3 apresenta os detalhes da arquitetura proposta. O desenvolvimento do estudo de caso é descrito na Seção 4. Por fim, os resultados obtidos são discutidos na Seção 5 e as considerações finais são apresentadas na Seção 6.

## 2. Trabalhos relacionados

O trabalho descrito em [Abello et al. 2015] apresenta o conceito de *OLAP* exploratório, em que novas fontes de dados e novas maneiras de estruturar, integrar e consultar dados são exploradas. O uso de novas fontes de dados, provenientes da Web, tem o intuito de enriquecer a tomada de decisão. Os autores de [Mansmann et al. 2014], por exemplo, usam o conteúdo do Twitter como fonte de dados para fazer a descoberta de métricas e dimensões para cubos OLAP.

Algumas pesquisas abordam a integração de dados Web em sistemas tradicionais. Os autores de [Etcheverry and Vaisman 2012] propõem uma maneira de executar operações *OLAP* sobre cubos RDFs, criados a partir de dados Web, sem integrá-los ao *DW*. O trabalho em [Benker 2013] propõe uma arquitetura que integra características de sistemas *OLTP* e *OLAP*, visando prover dados para a tomada de decisão em *right-time*. A arquitetura é composta de três componentes: operacional, analítico e de monitoramento. Este último é responsável por reagir a eventos em tempo de execução no sistema OLTP e entregar a informação em *right-time* para o componente analítico. No trabalho realizado por [Abelló et al. 2013], é proposto um novo operador *OLAP*, *drill-beyond*, para estender cubos multidimensionais com dados transitórios e permitir consultas de usuários sem o auxílio de um especialista. Contudo, nenhuma implementação foi encontrada.

Uma das lacunas das propostas acima é a falta de automatização das operações de integração de dados às consultas dos usuários. Esse e outros desafios, como a automatização dos processos de *ETL* e da captura de semântica de fontes de dados complexos, foram citados em [Abello et al. 2015]. Este trabalho tem intuito de contribuir mediante o desenvolvimento de um operador que, de maneira automática, integra dados extraídos de fontes Web com dados existentes no *DW* para enriquecer o resultado das consultas formuladas pelo usuário.

## 3. SelfBI: a arquitetura proposta

A Figura 1 apresenta a arquitetura proposta em cinco camadas. Ela tem o intuito de ser uma arquitetura genérica, de integração de dados estacionários com dados transitórios.

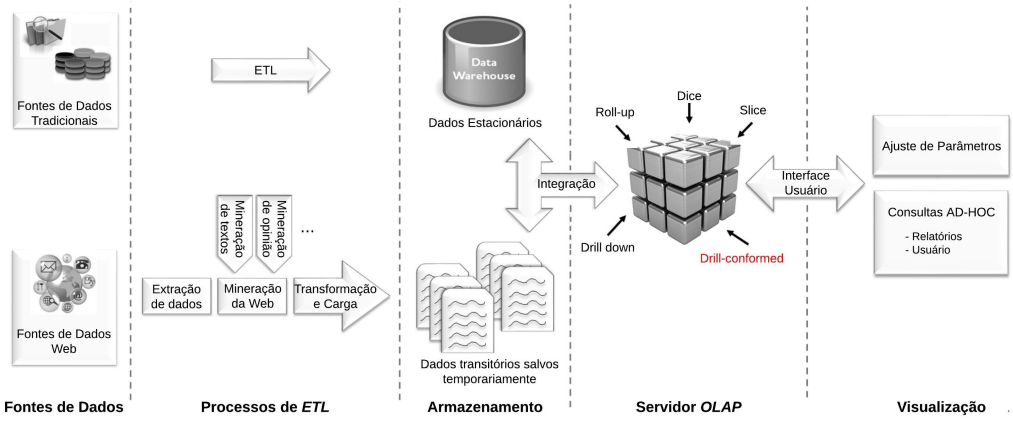


**Figura 1. Arquitetura SelfBI**

A arquitetura tem como eixo principal as consultas centradas no usuário, a partir da demanda personalizada e ajustes de parâmetros. As consultas podem requisitar tanto dados estacionários quanto transitórios. A seleção das fontes Web é feita de forma manual pelo usuário, de acordo com o cenário ao qual a arquitetura será aplicada. Para os dados estacionários, a arquitetura segue a estrutura tradicional de um *DW*. Para os dados transitórios, a arquitetura mostra quatro processos: Detecção e extração, Mineração, Transformação e Carga de dados (DeMTC). Os dados são armazenados de maneira temporária e separados dos dados tradicionais. Os processos de dados transitórios são contínuos e procuram ser executados em tempo real, conforme a demanda do usuário. O processo de detecção e extração é executado quando o usuário realiza uma consulta. Este é responsável por elaborar e realizar uma requisição que obtém dados relacionados à busca do usuário, a partir fontes Web indicadas. Seu objetivo é o princípio do *right-time DW*, que visa disponibilizar a informação certa no momento certo para o usuário. Após a extração, tais dados são processados pela etapa de mineração, a fim de extrair padrões e características que poderão ser apresentados ao usuário. O principal foco nesta etapa é a descoberta de conhecimento, a partir da exploração das redes sociais e da análise de sentimento (mineração de opinião). Após o processo de mineração, os dados podem ser transformados antes da execução da etapa de carga, que armazena os dados temporariamente.

O quinto processo, a integração de dados transitórios e estacionários em consultas, é executado utilizando o operador de integração, chamado de *drill-conformed*. O operador é baseado na proposta descrita em [Abelló et al. 2013]. O nome atribuído representa a integração de dados transitórios e estacionários nas consultas, que tem de ser projetada de maneira coerente, de forma a compartilhar uma estrutura uniforme que permita a fusão desses dados como um todo integrado. O operador de integração trabalha com os domínios dos atributos de fatos e dimensões no modelo multidimensional. Cada atributo do modelo estacionário, denominado de $Ae_i$, pertence a um domínio, definido pelo seu conjunto de valores e indicado por $dom(Ae_i)$. Assim, cada conjunto de valores extraído para os dados transitórios é considerado um domínio, indicado por $dom(At_j)$.

Visando a integração de dados, foram definidas duas classes de atributos provenientes dos dados transitórios: (1) atributos de intersecção, aqueles que servem para fazer o cruzamento entre dados transitórios e estacionários; (2) atributos relevantes, aqueles atributos transitórios que podem enriquecer a tomada de decisão, e não fazem parte da primeira classe. A técnica utilizada para determinar a qual dessas classes pertence o atributo é a *Probabilistic Latent Semantic Analysis (PLSA)*, técnica estatística para análise de dados co-ocorrentes, considerando, neste caso, a co-ocorrência entre $dom(Ae_i)$ e $dom(At_j)$. A proposta do operador está sendo desenvolvida, considerando a integração semântica de dados e a integração a servidores OLAP. Com o propósito de ser centrada no usuário, a integração vai levar em consideração o histórico de consultas dos usuários, adaptando a proposta descrita em [Thollot et al. 2012].

## 4. Estudo de caso

Para mostrar a aplicação da arquitetura, o estudo de caso consiste de um sistema de disponibilização de *streamings*, no qual, o gestor, para disponibilizar uma nova *streaming*, necessita selecionar conteúdos que tenham uma boa aceitação, de forma a gerar lucro. Além disso, as *streaming* são disponibilizadas por regiões, isto é, com base em diversos fatores *streamings* podem estar disponíveis apenas em algumas regiões e em outras não. O gestor deseja poder analisar o *feedback* dos usuários nas redes sociais acerca das diversas *streamings* em um dado momento e em diversas regiões, a fim de poder planejar suas aquisições naquele período. Além disso, ele deseja manter dados permanentes no *DW* acerca das *streamings*, como o número de acessos e o número de sinalizações como favorita, a fim de ajudar na decisão da disponibilização de uma determinada *streaming*.

**Seleção das Fontes de Dados:** para os dados estacionários, as fontes selecionadas foram o banco de séries[1] e o *movie db*[2], que consistem de bases de dados colaborativas que fornecem informações acerca das diversas séries e filmes existentes, respectivamente. Para os dados transitórios, o Twitter foi selecionado como a fonte de dados Web. O IMDb[3] e o Netflix foram utilizados para complementar tanto os dados tradicionais quanto os transitórios. As *APIs* oficiais do Twitter[4] e IMDb[5] foram utilizadas, enquanto que para o Netflix foi utilizada a Netflix Roulette API[6].

**Processo de Extração, Transformação e Carga:** para a etapa de extração dos dados estacionários, um *web crawler* percorre as páginas das fontes e obtém os dados acerca de seriados e filmes. As informações extraídas a partir desse mecanismo foram nome, gênero e tipo. Foram extraídos dados de 10.611 seriados e 19.624 filmes. Após a extração, foram removidos caracteres especiais dos nomes das *streamings* e os gêneros foram padronizados.

**Processo de Detecção e extração, Mineração, Transformação e Carga:** a extração dos dados transitórios é feita por um *Web Service RESTful*, que recebe o nome da *streaming* e realiza a busca. A busca nas *APIs* do Netflix e do IMDB é responsável por obter a nota média atribuída à *streaming*. Para a busca no Twitter, um filtro

[1]http://bancodeseries.com.br/index.php
[2]https://www.themoviedb.org/movie/upcoming
[3]http://www.imdb.com/
[4]https://dev.twitter.com/rest/public
[5]http://www.omdbapi.com/
[6]http://netflixroulette.net/api/

com palavras-chave é criado, buscando obter resultados relacionados com a *streaming*. Visando o enriquecimento semântico dos dados transitórios, os *tweets* são classificados como *feedback* positivo, negativo ou neutro[7]. A experimentação desta etapa é exposta na próxima seção.

**Armazenamento:** o armazenamento dos dados estacionários é realizado usando um esquema estrela no SQL Server. As dimensões do modelo são: `DLocalidade`, `DData`, `DStreaming`, `DEpisodio` e `DUsuario`. A tabela fato armazena o número de acessos de uma *streaming* e quantas vezes a mesma foi sinalizada como favorita. Os dados transitórios estão armazenados no MongoDB. Os dados são compostos da nota média atribuída à *streaming* pelos usuários e um conjunto de *tweets*, cada um com sua classificação de opinião. A localidade de origem, a data e o usuário do *tweet*, também, estão armazenados, com o propósito de fornecer uma visão útil dos dados para o relacionamento destes com os estacionários.

## 5. Experimentos: processo de DeMTC

Esta seção apresenta os primeiros resultados experimentais da proposta, que são relacionados ao processo de DeMTC. Dois tipos de experimentos foram executados. O primeiro é o tempo de resposta da *API* de análise de sentimento, a fim de verificar o impacto na obtenção dos dados em tempo real. O segundo é a relevância dos resultados retornados pelo processo de extração. A relevância mensura se o *tweet* obtido pelo processo de detecção e extração se relaciona com a *streaming* da busca. Em ambos foram extraídos *tweets* de séries de TV e filmes.

O primeiro experimento obteve um resultado médio de 10000 *tweets* a cada 28 segundos, o que é um resultado positivo quando avalia-se a execução desse processo próximo de tempo real. Para avaliar a relevância, foi utilizado um método de validação manual, de forma que dois usuários classificaram os *tweets* retornados como *relacionados* (à *streaming*), *não relacionados* (assunto não é a *streaming*) e *não identificado* (se existe relação entre o texto do *tweet* e a *streaming*). Ambos usuários classificaram cada um dos *tweets*. Para eliminar discordância, aqueles *tweets* em que a opinião dos usuários divergiu também foram classificados como *não identificado*.

Para facilitar a análise e identificação de pontos críticos do algoritmo, as *streamings* foram manualmente classificadas em sete categorias, de acordo com seus títulos em inglês: (1) contém uma palavra e a mesma é muito comum, de forma que pode ser utilizada em vários contextos e de maneiras diversas (exemplos: *Friends*, *Saw*) ; (2) contém uma única palavra, sendo comum (exemplo: *Castle*); (3) contém uma palavra que não é frequentemente utilizada (exemplo: *Zootopia*); (4) composição de duas palavras (exemplo: *Modern Family*); (5) nome de contexto específico/fictício, como histórias em quadrinhos (exemplo: *X-men: Apocalipse*); (6) nome grande, com três palavras ou mais (exemplo: *How I Met Your Mother*); (7) nome grande que representa expressões do cotidiano e pode ser usado em outros contextos (exemplo: *Friends with benefits*).

Ao todo, o experimento coletou 2314 *tweets* relacionados a dezesseis *streamings*. A Tabela 1 mostra os resultados do experimento. Como pode ser observado, o ponto crítico do algoritmo de extração se encontra em *streamings* cujo título é composto de uma única palavra utilizada frequentemente e de muitas maneiras no cenário de *streamings*.

[7]Para isso, foi usada a *API* sentiment140 (http://help.sentiment140.com/api)

Em seguida, encontram-se as *streamings* de nomes grandes, utilizados em diversos contextos. Por outro lado, as *streamings* nas outras categorias obtiveram uma taxa alta de dados relacionados.

**Tabela 1. Resultados dos experimentos do processo de DeMTC**

| | **Relacionado** (%) | **Não Relacionado** (%) | **Não Identificado** (%) |
|---|---|---|---|
| 1- 1 palavra muito comum | 26 | 73.5 | 0.5 |
| 2- 1 palavra comum | 72 | 21.5 | 6.5 |
| 3- 1 palavra | 100 | 0 | 0 |
| 4- 2 palavras | 98 | 1 | 1 |
| 5- Contexto Específico | 96 | 1.5 | 2.5 |
| 6- Nomes Grandes | 86 | 3 | 11 |
| 7- Nomes Grandes c/ muitos contextos | 57 | 35 | 8 |
| Todas as séries | 83.5 | 13 | 3.5 |
| Todos os filmes | 69 | 28 | 3 |
| Todos | 78 | 19 | 3 |

## 6. Considerações Finais

Neste trabalho, foi apresentada uma proposta de arquitetura em cinco fases que permite integrar, em uma consulta de usuário, dados históricos e tradicionais com dados coletados de fontes Web, que têm vida útil curta e são específicos para a tomada de decisão de um domínio. Os resultados do processo de DeMTC continuarão a ser melhorados. A proposta do operador de integração encontra-se, atualmente, em desenvolvimento.

## Referências

Abello, A., Romero, O., Pedersen, T. B., Berlanga, R., Nebot, V., Aramburu, M. J., and Simitsis, A. (2015). Using Semantic Web Technologies for Exploratory OLAP: A Survey. *IEEE Transactions on Knowledge and Data Engineer*, 27:571–588.

Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón, J.-N., Naumann, F., Pedersen, T., Rizzi, S. B., Trujillo, J., Vassiliadis, P., and Vossen, G. (2013). Fusion cubes: Towards Self-Service Business Intelligence. *International Journal of Data Warehousing and Mining*, 9:66–88.

Benker, T. (2013). A Hybrid OLAP & OLTP Architecture Using Non-Relational Data Components. *Enterprise Modelling and Information Systems Architectures*, 222 of Lecture Notes in Informatics:41–57.

Etcheverry, L. and Vaisman, A. A. (2012). Enhancing OLAP Analysis with Web Cubes. *Lecture Notes in Computer Science*, 7295:469–483.

González, S. and Berbel, T. (2014). Considering unstructured data for OLAP: a feasibility study using a systematic review. *Revista de Sistemas de Informação da FSMA*, 14.

Mansmann, S., Rehman, N. U., Weiler, A., and Scholl, M. H. (2014). Discovering OLAP dimensions in semi-structured data. *Information Systems*, 44:120–133.

Thollot, R., Kuchmann-beauger, N., and aude Aufaure, M. (2012). Semantics and Usage Statistics for Multi-Dimensional Query Expansion. In *Proceedings of International Conference of Database Systems for Advanced Applications*, pages 250–260.

Thomsen, C., Pedersen, T. B., and Lehner, W. (2008). RiTE: Providing On-Demand Data for Right-Time Data Warehousing. *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 456–465.