ConBRepro
X CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO
EVENTO ON-LINE
02 a 04 de dezembro 2020

Um framework para a integração de dados heterogêneos em ambientes industriais

Cristian R. Pastro
Joceleide D. C. Mumbelli
Gustavo W. Denardin
Marcelo Teixeira
PPGEE - Universidade Tecnológica Federal do Paraná, câmpus Pato Branco
Luiz Fernando Puttow Southier
PPGIa – Pontifícia Universidade Católica do Paraná

Resumo: O poder de decisão e de informação é determinante em ambientes industriais. Na atual conjuntura tecnológica, as diferentes fases de um processo produtivo geram um grande volume de dados que, quando bem geridos, podem servir a políticas de processamento de informação alinhadas tanto com a engenharia de processos quanto com a própria gestão organizacional. Um dos empecilhos, porém, é que dados industriais são por essência heterogêneos, gerados, armazenados e processados distintamente. Isso limita a convergência desses dados para um formato tal que possam ser processados conjuntamente. Este trabalho propõe uma infraestrutura computacional que coleta, processa e integra dados heterogêneos provenientes de ambientes industriais e retorna, como saída, um conjunto de dados harmonizados morfologicamente e aptos a servirem a procedimentos de inteligência computacional e corporativa. A abordagem foi testada sobre dados heterogêneos extraídos do processo de pintura em uma montadora de automóveis. Resultados sugerem que, no contexto da aplicação proposta, a arquitetura se mostrou mais eficiente do que as abordagens existentes.

Palavras-chave: ETL, Dados heterogêneos, processamento de informações, Inteligência empresarial.

A framework for the integration of heterogeneous data in industrial environments

Abstract: Decision and information are decisive in industrial environments. In the current technological conjuncture, the different steps of a production process generate a large volume of data that, when well-managed, can serve information processing policies in line with both process engineering and organizational management itself. One of the obstacles, however, is that industrial data is essentially heterogeneous, distinctly generated, stored and processed. This limits the convergence of these data to a format such that they can be processed together. This work proposes a computational infrastructure that collects, processes, and integrates heterogeneous data from industrial environments and returns, as an output, a set of data that is morphologically harmonized and able to serve computational and corporate intelligence procedures. The approach was tested on heterogeneous data extracted from the painting process in an automobile manufacturer. Results

suggest that, in the context of the proposed application, the architecture proved to be more efficient than the existing approaches.

Keywords: ETL, Heterogeneous Data, Information Processing, Business Intelligence.

1. Introdução

Diante uma crescente e acirrada competitividade de mercado, os ambientes de produção industrial têm gradativamente experimentado e se adaptado a cenários de operação auxiliados por *Business Inteligence* (BI) (TRIEU, 2017). Em essência, tais cenários podem servir tanto à gestão de negócios quanto a subsídios para tomadas de decisões em sistemas de produção, alavancando políticas de engenharia de controle e automação. Para isso, no entanto, depende-se, sobretudo, da disponibilidade de dados, de ferramentas para processar esses dados, e de esquemas de visualização que ultrapassem a dimensão do óbvio, que geralmente resulta de planilhas eletrônicas convencionais.

Atualmente, na esfera industrial, ferramentas como Tableau, *TIBCO Spotifire* e *Microsoft Power BI* (SHUKLA; DHIR, 2016), vêm sendo amplamente utilizadas, não só para a gestão de negócios, mas também como para subsídio a tomadas de decisões em chão de fábrica. Elas permitem que padrões sejam identificados por qualquer pessoa ligada a produção, por meio de interfaces intuitivas e concisas (FOLLEY; GUILLEMETTE, 2010). Entretanto, mesmo que essas ferramentas diversifiquem a transformação e visualização de dados, elas limitam a entrada a certos domínios, como planilhas, arquivos CSV (*Comma-separated values*), JSON (*JavaScript Object Notation*) e bancos de dados relacionais.

Em contrapartida, dados coletados em ambientes industriais podem emergir de múltiplos domínios, como de sistemas SCADA (*Supervisory Control and Data Acquisition*), sensores, *logs* de equipamentos de hardware e arquivos de formatos proprietários, requerendo, portanto, tratamento computacional prévio, antes de serem aceitos por ferramentas de BI (SHUKLA; DHIR, 2016).

Uma possibilidade é a utilização de políticas *Extract-Transform-Load* (ETL) para adequar dados heterogêneos em bases padronizadas. Aplicações ETL funcionam bem para fontes estruturadas de dados, e nesse caso muitas tecnologias contam nativamente com recursos predefinidos para processamento ETL, como por exemplo tecnologias de *Data Wareshouse* (DW) e *Data Mart* (DM) (VASSILIADIS, 2009). Porém, DWs e DMs não são sensíveis à morfologia ou natureza dos dados, além de envolverem um alto custo de implementação e complexidade proporcional à variedade e ao volume de dados.

Juntos, esses fatores justificam a construção de *frameworks* mais gerais para recursos ETL, com foco na indústria. Uma alternativa surge por meio dos chamados Data *Lakes* (DLs). Um DL é um repositório que inclui grandes volumes de dados, tanto estruturados quando não estruturados, em geral de fácil acesso. A principal vantagem em relação a um DW ou DM, é que um DL não promove a eliminação precoce de atributos nem a agregação estática. Ele retém todos os atributos e os disponibiliza para múltiplos usuários e aplicações, o que viabiliza a personalização do conhecimento a ser extraído do DL. Como desvantagem, um DL não gera, por si só, conhecimento, o que depende de políticas adicionais de agregações que resultem em *insights*, associados ao conhecimento (MATHIS, 2017).

Esse trabalho propõe um *framework*, implementado em software, que explora o conceito de DL para estender e melhor acomodar recursos ETL na indústria. Inicialmente, assume-se a ideia de que dados industriais são expostos como DLs e estão aptos a serem acessados, examinados e amostrados. Então, propõe-se uma formalização matemática para o conceito de ETL, o que permite a agregação e tratamento não estático de

subconjuntos de dados com características afins. Essa teoria é implementada em software por meio de uma arquitetura denominada E-ETL.

Por meio de manipulação essencialmente algébrica do DL, interfaceada com o usuário via recursos intuitivos de software, é possível construir diferentes *insights* para os dados do DL via fusão multissensorial, decomposição, ou qualquer outra agregação semântica a que se tenha interesse (HALL; LLINAS, 2017). Isso resulta em agregações mais ricas semanticamente, de pouca complexidade, e rapidamente personalizáveis, ampliando o impacto de ferramentas de BI na indústria. Além disso, a abordagem permite que *insights* sejam reaproveitados ou combinados em novos *insights*, diminuindo o tempo de projeto e potencializando o poder informacional. A adição de novas fontes de dados no DL, bem como a posterior extração e tratamento, também são tarefas bem definidas a partir da E-ETL.

A abordagem foi testada no DL de uma montadora de automóveis com fábrica no Paraná, relativo ao seu processo de pintura. Resultados sugerem que a abordagem proposta é de baixo custo, fácil implementação e uso, além de extensível a outras malhas industriais já concebidas ou outros tipos de dados a serem considerados.

Estruturalmente, a Seção 2 expõe os conceitos relacionados; a Seção 3 as principais contribuições; um teste prático na fábrica de automóveis é apresentado na Seção 4; e a Seção 5 discute as conclusões e perspectivas.

2. Referencial Teórico

Apesar da importância, a implementação prática de BI em ambientes de produção pode falhar devido a problemas no gerenciamento de dados, muitas vezes heterogêneos e variados (FOLLEY; GUILLEMETTE, 2010).

Um dos pilares do gerenciamento de dados é o ETL, intimamente ligado ao BI, pois por meio dele informações podem ser extraídas de sua fonte original para sustentar técnicas de BI. Em ambientes industriais, o ETL pode ser aplicado no desenvolvimento de Data *Warehouses* (DWs) (JINDAL; TANEJA, 2012). DWs são grandes armazéns de dados, contendo informações de diversos setores de toda a corporação. Uma versão local de um DW é um *Data Mart* (DM), útil para mapear dados de setores corporativos específicos (KOUR, 2015).

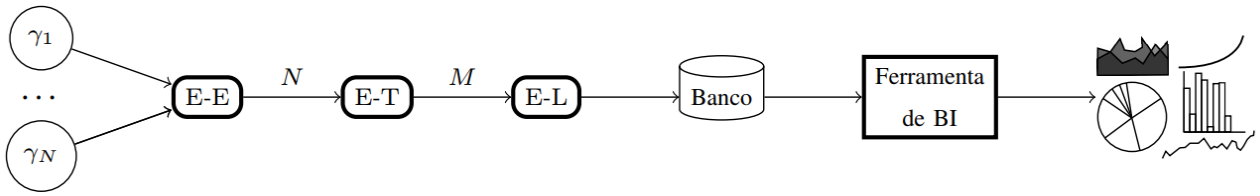
Além dos tradicionais DMs e DWs, um conceito que está se popularizando é o de *Data Lake* (DL). Diferente dos DMs e DWs, um DL não impõe a necessidade de filtrar ou ajustar dados previamente. Eles são depositados no lake e acessados de diferentes formas, por múltiplos usuários, para a construção de informação. Ainda que o conceito de DL sirva diretamente à aplicações de processamento de informações por cientistas de dados, ele também pode disponibilizar dados para usuários não técnicos ou para aplicações corporativas, podendo ser utilizado para apoiar tomadas de decisões por meio do BI. A grande crítica ao DL é que devido a heterogeneidade dos dados armazenados pode levar a um complexo e difícil gerenciamento, transformando o *Data Lake* em um *Data Swamp*, um complexo armazém de dados com pouca ou nenhuma utilidade (MATHIS, 2017).

Em suma, BI pode ajudar na tomada de decisões em meios industriais. Porém, seu sucesso requer políticas eficientes de gerenciamento dos dados por meio de DW ou DL. Os resultados deste artigo se alinham a essas políticas e passam, sobretudo, pela extensão ETL descrita na sequência.

3. Arquitetura para extensão do ETL

A arquitetura E-ETL proposta nesta seção é composta por 3 componentes principais, E-E, E-T, E-L, cada um estendendo a respectiva fase (E, T e L) do processo ETL clássico. O esquema geral do E-ETL é mostrado na Figura 1.

Figura 1 – Fluxo operacional da arquitetura E-ETL



As N fontes de dados, denotadas γ_n, para $1 \leq n \leq N$, são processadas pela arquitetura E-ETL na seguinte forma:

- E-E: diferentemente do ETL tradicional, já na fase de extração o E-ETL formaliza matematicamente a ideia de *tags*, ideia similar aos metadados de um datalake. Uma etiqueta é um rótulo associado a um dado extraído, facilitando as etapas subsequentes e generalizando o conceito para outras possíveis extensões e fontes de dados.
- E-T: os formalismos matemáticos, criados na etapa anterior, são aqui explorados para o tratamento mais eficiente de dados se comparado ao ETL clássico. Um dos diferenciais é, por exemplo, o suporte nativo à incorporação de modelos semânticos como a transformação, de modo que o resultado seja um conjunto de dados minimamente sensíveis ao contexto da aplicação. Como essa etapa pode associar ou desmembrar fontes de dados, o número de fontes de saídas em geral difere do número de fontes de entrada, provendo M fontes para a etapa seguinte;
- E-L: essa fase envolve a transposição dos dados transformados para um formato armazenável, como por exemplo um banco de dados *relacional* baseado em operações SQL, um novo *Data Lake*, *Data Mart*, ou *Data Warehouse*. Na E-ETL, essa fase é significativamente mais padronizada do que na ETL clássica, inclui um conjunto mais representativo de dados, além de que esses dados tendem a incorporar um maior teor descritivo e relevância semântica, frutos dos modelos complementares associados ao processo de transformação.

As etapas posteriores são análogas às de um processo BI tradicional. Ou seja, uma ferramenta de BI explora os dados gravados na base para fins de visualização por gráficos e relatórios voltados à gestão. A diferença é que, agora, espera-se que os dados sejam mais consistentes e mais bem alinhados à descoberta de conhecimento de cunho industrial, por meio de ferramentas de BI. A seguir, apresentam-se os pormenores e a formalização dos três componentes da arquitetura E-ETL.

3.1. Formalização da Extração dos Dados (E-E)

Os conceitos de *fenômeno*, *instância*, e *tag* estruturam a fase de extração de dados da E-ETL. A relação entre eles é exposta na Figura 2, e em seguida discutida e exemplificada.

Figura 2 – Arquitetura E-ETL para a fase de extração de dados

$$\mathcal{P}^h = \left\{ \mathcal{S}^h = \{\mathcal{S}_1^h, \ldots, \mathcal{S}_y^h\}, \quad \mathcal{J}^h = \left\{ \mathcal{J}_1^h = \left\{ \mathcal{T}_1^h = \{\mathcal{T}_{1,1}^h, \ldots, \mathcal{T}_{1,y}^h\}, \; \mathcal{D}_1^h = \{\mathcal{D}_{1,1}^h, \ldots, \mathcal{D}_{1,z}^h\} \right. \right. \ldots \left. \left. \mathcal{J}_x^h = \left\{ \mathcal{T}_x^h = \{\mathcal{T}_{x,1}^h, \ldots, \mathcal{T}_{x,y}^h\}, \; \mathcal{D}_x^h = \{\mathcal{D}_{x,1}^h, \ldots, \mathcal{D}_{x,z'}^h\} \right. \right. \right\}$$

3.1.1. Fenômeno

Um fenômeno pode ser entendido como sendo uma representação lógica de uma fonte de dados. São exemplos de fenômenos, conjuntos de dados de sensores, os dados extraídos de uma classe de planilhas, os dados advindos de um sistema SCADA ou de um CLP, conjunto de medições de qualidade de produto etc.

Formalmente, $\mathcal{P}^h \in \mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_v\}$ representa um fenômeno dentre um conjunto finito $\mathcal{P}$ de possíveis fenômenos. Assume-se que, para cada fenômeno $\mathcal{P}^h$, exista um conjunto de instâncias e um conjunto de configurações, ou *setups*, associados. As instâncias representam subconjuntos de dados com características afins, nos quais se tenha interesse; já os *setups* identificam as características morfológicas assumidas pela porção de dados de interesse. Os elementos $\mathcal{I}_i^h \in \mathcal{I}^h = \{\mathcal{I}_1^h, \cdots, \mathcal{I}_x^h\}$ e $\mathcal{S}_j^h \in \mathcal{S}^h = \{\mathcal{S}_1^h, \cdots, \mathcal{S}_y^h\}$, formalizam, respectivamente, uma instância $\mathcal{I}_i^h$, em um conjunto $\mathcal{I}^h$ de possíveis instâncias, e um *setup* $\mathcal{S}_j^h$, em um conjunto $\mathcal{S}^h$ de possíveis *setups*, ambos relativos a um fenômeno $\mathcal{P}_h$.

Um fenômeno, portanto, pode ser identificado pela suas instâncias e seus *setups*, o que é exposto na forma de uma dupla ordenada $\mathcal{P}^h = (\mathcal{I}^h, \mathcal{S}^h)$, para $|\mathcal{I}^h| \geq 0$ e $|\mathcal{S}^h| > 0$.

3.1.2. Instanciação

Internamente, cada *setup* $\mathcal{S}_i^h$ em $\mathcal{S}^h$ é exposto na forma de um par ordenado (id_j^h, dom_j^h), em que id_j^h é o *identificador* e dom_j^h é o *domínio* esperado para cada configuração de dados do *setup*. Similarmente, cada instância $\mathcal{I}_i^h$ em $\mathcal{I}^h$ é formada por um par ordenado $(\mathcal{T}_i^h, \mathcal{D}_i^h)$ contendo: um conjunto de *tags* $\mathcal{T}_i^h$; e um conjunto de *dados* $\mathcal{D}_i^h$, associados à instância.

Uma *tag* pertencente ao conjunto $\mathcal{T}_i^h$ é denotada por $T_{i,k}^h \in T_i^h = \{T_{i,1}^h, \cdots, T_{i,y}^h\}$. Analogamente, um dado $\mathcal{D}_{i,l}^h$ pertencente ao conjunto de dados $\mathcal{D}_i^h$, é denotado por $\mathcal{D}_{i,l}^h \in \mathcal{D}_i^h = \{\mathcal{D}_{i,1}^h, \cdots, \mathcal{D}_{i,z}^h\}$.

3.1.3. Rotulação

Internamente, o conjunto de *tags* $\mathcal{T}_i^h$ pertencente a uma instância $\mathcal{I}_i^h$ tem a função de rotular o conjunto de dados $\mathcal{D}_i^h$ pertencente a mesma instância $\mathcal{I}_i^h$, com um valor e um tipo. O valor estaria associado ao contexto em que o dado foi extraído, enquanto o tipo é associado à sua morfologia.

Formalmente, uma *tag* $\mathcal{T}_{i,k}^h$ é uma dupla ordenada $\mathcal{T}_{i,k}^h = (\mathcal{V}_{i,k}^h, \mathcal{U}_{i,k}^h)$ que define uma visão semântica parcial, construída a partir de um valor $\mathcal{V}_{i,k}^h$ e de um tipo $\mathcal{U}_{i,k}^h$, para um conjunto de dados $\mathcal{D}_i^h$ associado a uma instância $\mathcal{I}_i^h$. Cada *tag* $\mathcal{T}_{i,k}^h$, contida em uma instância $\mathcal{I}_i^h$ de um fenômeno $\mathcal{P}_h$, deve estar associada a um *setup* $\mathcal{S}_j^h$ desse fenômeno. Essa equivalência é verificada quando os indexadores j, de um *setup*, e k, de uma *tag*, tiverem valores iguais. Assim sendo, assume-se que o número de *tags* dentro de uma instância, e de *setups* no fenômeno que contém a instância, são iguais.

A associação entre as *tags* e os dados de uma determinada instância é simbólica, ou seja, é tomada em nível abstrato para fins de identificação do contexto dos dados da instância. Exemplo de *tags* incluem a data na qual os dados foram extraídos, o modelo de um certo sensor utilizado, ou o nome da pessoa que realizou a medição dos dados.

Em resumo, uma instância de um fenômeno inclui um conjunto de *tags* que identificam os dados dessa instância, cada *tag* com uma semântica diferente. Os tipos de dados admissíveis também variam, e são limitados apenas pelo poder computacional de representá-los, sendo *date*, *integer*, *string, boolean* e *float* os mais comuns no ambiente fabril.

Exemplo 1: Suponha que um determinado fenômeno $\mathcal{P}^h$ represente medidas de espessura de camada de tinta aplicadas por uma fábrica automotiva em seu processo de pintura.

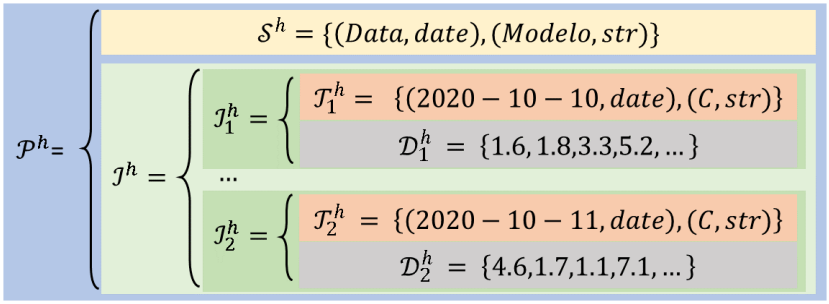
Cada conjunto de medições de interesse pode ser identificado por uma data da medição e uma descrição do modelo do produto que foi medido. Assim, $\mathcal{P}^h$ contém o conjunto de setups $\mathcal{S}^h = \{(Data, date), (Modelo, str)\}$.

Suponha, ainda, que se deseje projetar um determinado insight sobre as medidas de espessura realizadas em um determinado período, sobre um modelo. Para fins ilustrativos, assuma os dias 2020-10-10 e 2020-10-11, e duas unidades de um modelo de carro C.

Nesse caso, o fenômeno $\mathcal{P}^h$contém duas instâncias $\mathcal{I}_1^h$ e $\mathcal{I}_2^h$, i.e., $\mathcal{I}^h = \{\mathcal{I}_1^h, \mathcal{I}_2^h\}$. Cada instância $\mathcal{I}_i^h$ possui um conjunto de dados $\mathcal{D}_i^h$ correspondente às medições de espessura realizadas, além de um conjunto de tags $\mathcal{T}_i^h$, em que $\mathcal{T}_1^h = \{(2020-10-10, date), (C, str)\}$ e $\mathcal{T}_2^h = \{(2020-10-11, date), (C, str)\}$.

Em relação à arquitetura introduzida na seção 3.1, esse exemplo pode exposto como na Figura 3.

Figura 3 – Exemplo de aplicação da arquitetura E-ETL.



3.2. Formalização da Transformação dos Dados (E-T)

Em sua forma bruta, dados podem conter ruídos, valores faltantes, ou mesmo não servir ao propósito que se deseja. Tais problemas prejudicam a utilização dos dados, podendo inclusive fornecer conhecimentos incorretos e prejudicar o processo de tomada de decisão (ZHU et al, 2018).

Além de promover correções de possíveis anomalias, a etapa E-T do E-ETL sugere que fusão multissensorial, decomposição, ou qualquer outra agregação semântica possa enriquecer a base de dados e ampliar as possibilidades nas etapas subsequentes de utilização.

Alternativamente, é sugerido pelo E-ETL que, após os dados serem separados em formalismos como conjuntos e subconjuntos que reúnem características afins, operações algébricas possam ser definidas sobre os conjuntos de dados a fim de gerarem transformações morfológicas que resultem em novos conjuntos, possivelmente com maior riqueza semântica e potencial de personalização.

Na arquitetura E-ETL, uma transformação depende da criação de uma *regra* que nada mais é do que um mapa que aplica manipulações algébricas sobre conjuntos de dados e resulta em novos conjuntos, formatados conforme a regra em questão. Regras podem ser utilizadas para a implantar técnicas como limpeza, associação, fusão de dados etc. Em geral, espera-se que uma regra tenha potencial para agregar informações semânticas aos dados originais.

Definição 1 (Regra): Seja $\mathcal{P} = \{\mathcal{P}^1, \cdots, \mathcal{P}^v\}$ um conjunto de fenômenos definidos como em 3.1.1. Sejam $P^i \subseteq \mathcal{P}, i = 1, \ldots p$, subconjuntos de fenômenos a serem manipulados morfologicamente. Seja $\mathcal{R}$ um conjunto de regras a serem aplicadas sobre os fenômenos

para a sua transformação morfológica. Uma regra $\mathcal{R}^m \in \mathcal{R}$ é definida como uma tripla ordenada:

$$\mathcal{R}^m = (\mathcal{Y}^m, \mathcal{J}^m, \mathcal{V}^m),$$

em que:

- $\mathcal{Y}^m$: é um valor inteiro indicando a prioridade de execução da regra $(0 \preccurlyeq p \prec \infty)$, sendo a proximidade de 0 indicando maiores prioridades.
- $\mathcal{J}^m$: é um mapa

$$\mathcal{J}^m: P^i \rightarrow \{P^{v+1}, \dots, P^{v'}\}$$

que associa um conjunto P^i de fenômenos de entrada a um conjunto de novos fenômenos.

- $\mathcal{V}^m$: é o conjunto finito de variáveis envolvidas nas regras de transformação.

Uma regra na arquitetura E-ETL é de fato executada quando associada a um conjunto de fenômenos de entrada por meio da função Ψ, definida a seguir.

Definição 2 (Execução de Regra): Seja $P^i \subseteq \mathcal{P}$ um subconjunto de fenômenos em $\mathcal{P}$, e seja $\mathcal{R}^m$ definida como na Definição 1. Então:

$$\mathcal{P}_o = \Psi(P^i, \mathcal{R}^m)$$

é uma função que aplica a regra de transformação $\mathcal{R}^m$ sobre P^i e retorna um conjunto de fenômenos $\mathcal{P}_o$.

A execução da função Ψ deve ser ordenada por meio da prioridade da regra de execução passada como parâmetro. Assim sendo, as regras de transformação de maior prioridade devem ser as primeiras a serem executadas.

Exemplo 2: Considere o exemplo da Figura 3, no qual o fenômeno $\mathcal{P}^h$ representa medidas de espessura em camadas de tinta. Suponha que, em alguns casos, por erros de digitação ou outro evento, uma tag $\mathcal{T}_{i,j}^h$ associada ao setup $\mathcal{S}_j^h = (Modelo; str)$ possua um valor que não corresponda a nenhum modelo de veículo fabricado pela empresa. Suponha que $\mathcal{A}$ seja o conjunto contendo o nome de todos os modelos de veículos fabricados pela empresa.

Assuma, também, que uma função $\lambda(\mathcal{A}, \omega)$ recebe o conjunto $\mathcal{A}$ e um termo léxico ω e retorna a palavra contida em $\mathcal{A}$ que é mais próxima lexicalmente de ω.

Para esse exemplo, uma regra $\mathcal{R}^m$ poderia ser criada para que, toda vez que o valor de uma tag $\mathcal{T}_{i,j}^h$ não represente um nome de um modelo válido, tal valor seja transformado para o nome de um modelo mais próximo lexicalmente de um nome válido.

Para isso, usando o E-ETL, pode-se criar uma regra $\mathcal{R}^m = (\mathcal{Y}^m, \mathcal{J}^m, \mathcal{V}^m)$, em que:

- $\mathcal{Y}^m = 1$;
- $\mathcal{J}^m$: $\mathcal{S}_j^h = (Modelo, str) \wedge \mathcal{T}_{i,j}^h = (model\_val, str) \wedge model\_val \notin \mathcal{A} \rightarrow model\_val = \lambda(\mathcal{A}, model\_val)$
- $\mathcal{V}^m = \{model\_val\}$.

O fato de $\mathcal{Y}^m$ ter o valor 1 significa que a execução desta regra ocorrerá depois de qualquer outra regra de prioridade 0 possivelmente criada. Regras de prioridade 0, geralmente são utilizadas para limpeza de dados.

O mapa $\mathcal{J}^m$ busca no subconjunto de fenômenos P^i,, todas as *tags* $\mathcal{T}_{i,j}^h = (model\_val, str)$, correspondentes ao setup $\mathcal{S}_j^h = (Modelo, str)$. Caso $model\_val$ não esteja no conjunto $\mathcal{A}$, tal variável passa a receber o valor da função λ, ajustando o nome do carro por similaridade

léxica. Nota-se que as *tags* e os *setups* estão indexados com o mesmo valor j, indicando a correspondência da *tag* e *setup*.

O conjunto $\mathcal{V}^m$ contém a única variável que é utilizada para o processamento do mapa $\mathcal{J}^m$.

3.3. Formalização do Armazenamento dos Dados (E-L)

A última fase do E-ETL antes da utilização dos dados é o armazenamento. O E-ETL não especifica o formato em que os dados devem ser salvos e nem a fonte de destino e, assim, pode servir à diversas estruturas de saída, como DM, DW, DL, banco de dados baseadas em operações SQL ou até mesmo formatos de armazenamento mais simples, como planilhas, arquivos XML ou JSON. O Algoritmo 1 mostrado na Figura 4 generaliza a ideia de carregamento de um fenômeno $\mathcal{P}^h$ no E-ETL.

Figura 4 – Algoritmo de carregamento dos dados para uma base.

Algoritmo 1: CARREGAMENTO PARA A BASE

Entrada: Fenômeno $\mathcal{P}^h$;
Saída: database $\mathcal{B}$;

início
  $\mathcal{B} \leftarrow \emptyset$;
  para *Cada* $\mathcal{I}_i^h$ **faça**
    parcial $\leftarrow \emptyset$;
    para *cada* $\mathcal{T}_{i,k}^h$ **faça**
      $parcial \leftarrow parcial \cup \{\mathcal{V}_{i,k}^h\}$;
    fim
    $parcial \leftarrow parcial \cup \mathcal{D}_i^h$;
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{parcial\}$;
  fim
  retorna $\mathcal{B}$;
fim

O Algoritmo 1 define bases de dados parciais, sendo que uma parcial é um conjunto contendo o valor das *tags*, para identificar os dados, e o conjunto de dados de cada instância do fenômeno. A base de dados final para o fenômeno $\mathcal{P}^h$, é um conjunto contendo todas as parciais deste fenômeno. Assim sendo, a base $\mathcal{B}$ é um conjunto contendo conjuntos, sendo que cada conjunto interno contém informações de uma instância.

Ao final das três etapas principais do E-ETL, obtêm-se uma base de dados apta para sustentar etapas subsequentes de análise de dados e inteligência computacional. Neste trabalho, para fins de ilustração, os dados são utilizados para alimentar uma ferramenta que permite visualizações interativas de dados, auxiliando no processo de tomada de decisões em ambientes de produção por meio do BI.

4. Aplicação do E-ETL em uma fábrica de automóveis

Essa seção apresenta um exemplo de aplicação da abordagem proposta na fabricação de automóveis, com ênfase particular no processo de pintura cujos dados originais são de natureza heterogênea e de complexo relacionamento e amostragem. Nesse domínio de aplicação, os dados são provenientes de fontes como *logs* de dispositivos de hardware ou sinais de sensores e relatórios SCADA. Ainda outros tantos, essenciais em chão de fábrica, advém de planilhas eletrônicas, atualizadas manualmente pelo próprio usuário. Cada fonte impõe à engenharia sua própria morfologia, política de acesso, opções transformação, tratamento e, consequentemente, de armazenamento eletrônico.

Uma solução buscada pela empresa em questão foi a implantação de ferramentas comerciais de DW, como a *Oracle Warehouse* (ORACLE, 2020), e de DL, como a *Google Cloud* (GOOGLE, 2020). Entretanto, o alto custo e a inflexibilidade dessas ferramentas

inviabilizaram o uso sobre sistemas de produção pré-concebidos e de limitadas opções de reconfiguração. A literatura também foi explorada como complementação do ETL clássico. Abordagens de tratamento (TALEB et al, 2015), de transformação (ZHU et al, 2018), e de pós ETL, como o tratamento por modelos semânticos para a captura de percurso e reação a eventos adversos (BANSAL; KAGEMANN, 2015; WILLIAMS et al; 2015), foram cogitadas. Porém, esses trabalhos falham em: não explorar em paralelo as fases ETL; não personalizar modelos semântico-formais a dados industriais heterogêneos; não reagir adaptativamente à geração contínua de dados fabris.

Uma tentativa de implementação de uma base de dados padronizada para o processo de pintura consistiu em tratar cada dado de maneira individual, tentando criar um DL clássico. Porém o aspecto -heterogêneo dos dados e a variedade de fontes limitou substancialmente a abordagem, requerendo um complexo gerenciamento e monstrando-se ineficiente à medida em que a base de dados crescia.

O projeto de concepção do E-ETL foi então proposto como forma de abreviar tais limitações. Ele foi desenvolvido e implementado em linguagem *Python 3.7.3*, em uma arquitetura utilizando Ubuntu Linux versão 16.04.

O primeiro passo para se criar um projeto de análise de dados de pintura automotiva utilizando o E-ETL é extrair fontes representativas dedados, dentre as quais foram consideradas neste artigo:

- quatro procedimentos de medição de qualidade de veículo pintado;
- análises de inspeção visual de veículo pintado;
- características de cores das tintas;
- histórico de modificação dos parâmetros de configuração do robô aplicador de tinta;
- manual de realização de um procedimento de medição de qualidade.

Para cada fonte de dado foi criado um fenômeno. Em cada fenômeno foram configurados *setups* de acordo com as especificações de cada dado. Entre os *setups* configurados estão: data de medição de um parâmetro de qualidade; modelo de um veículo analisado; cor de um veículo; e viscosidade da tinta utilizada em um processo. Após configurar os *setups* e as fontes de dados, o próprio E-ETL separa as informações em fenômenos, instâncias, *tags* e dados. Assim sendo, a tarefa de estruturação dos dados passa a não depender mais do projetista.

Algumas regras de transformação foram criadas para teste, sendo as regras de limpeza de dados as mais prioritárias. A necessidade de uma regra se evidencia quando, por exemplo, ao extrair uma data de uma planilha a extração obtém um formato *Datatime*, ao invés de uma data no formato dd/mm/aa. *Datetime* é um número real não negativo, em que sua parte inteira representa a quantidade de dias passados desde 01 de Janeiro de 1900. Neste exemplo, o mapa $\mathcal{J}^m$ de uma regra $\mathcal{R}^m$ criada poderia ser igual a

$$\mathcal{S}_j^h: (*, date) \wedge \mathcal{T}_{i,j}^h = (n\_dat, float) \wedge n\_dat = [0-9]\{5\}.00 \rightarrow n\_dat = 1900 - 01 - 01 + (n\_dat + 2)\ days.$$

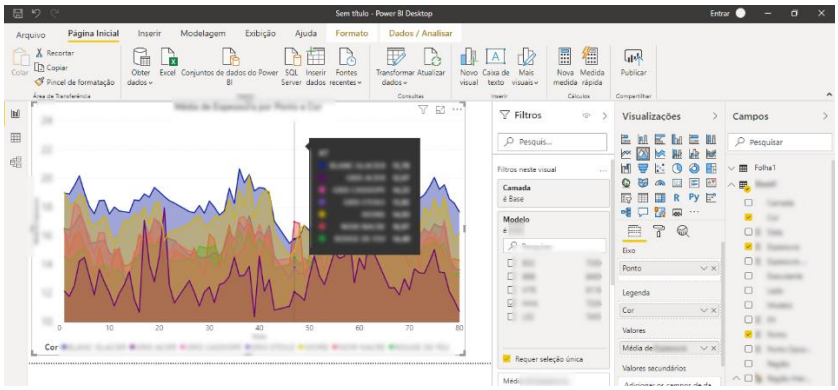
Tal mapa identifica um valor n_data de uma *tag* qualquer que possua o tipo *float* mas que possua um *setup* correspondente com o tipo *date*. Caso, além disso, o valor de n_dat tenha a forma de um *Datetime* exposto pela expressão *regular* $n\_dat = [0-9]\backslash\{5\backslash\}.00$, o valor da *tag* passa por uma fórmula de transformação que ajusta o seu valor.

Além de regras de limpeza, foram criadas regras de fusão de dados. Por exemplo, suponha um fenômeno contendo medições de uma característica de qualidade que é realizada em pontos determinados de um veículo. Suponha também um mapa que indica a localização de tais pontos medidos, contendo características como lado, região medida, ponto oposto,

etc. Uma regra de fusão foi criada para mapear cada medida com a região exata ao qual a medição foi realizada. A criação de regras similares e, portanto, a generalização desse método para outros casos, passa apenas pelo processo de adição de novos mapas e de medições de novas características desejadas de qualidade.

Com base no Algoritmo 1, uma base de dados fundamentada por arquivos CSV foi gerada para o processo de pintura da fábrica de automóveis. A título de experimento, a base de dados foi carregada no software *Microsoft Power BI*, no qual uma captura de tela dados referentes a medições de uma característica de qualidade para várias cores de tinta diferentes é mostrada na Figura 5.

Figura 5 – Gráfico gerado por meio de dados reais da fábrica de automóveis.



O gráfico foi gerado por meio de dados oriundos de uma regra de fusão de dados. Tal regra mostrou-se apta a sustentar um processo que identificou padrões de comportamento em tintas de cores diferentes. Com isso, foi possível montar grupos de tinta que apresentam padrões similares, considerando características químicas e após uma carroceria ser pintada, características práticas. A criação de grupos de tinta facilita e diminui o tempo de desenvolvimento da pintura para um novo modelo de carro, pois o procedimento de pintura para uma tinta do mesmo grupo pode ser reaproveitado. Além disso, ao desenvolver uma nova pintura, alguns veículos precisam ser pintados para testes e depois, descartados. Trabalhos futuros com o E-ETL podem diminuir o número de veículos pintados para testes, promovendo uma diminuição nos custos de projeto.

Outro *insight* obtido por meio do BI, apoiado pela arquitetura E-ETL, indica que há um desvio padrão médio acima do esperado para uma determinada característica de qualidade considerando regiões opostas do mesmo veículo. Isso vinha prejudicando o controle de qualidade, pois medições dentro dos padrões eram consideradas errôneas. Essa descoberta foi possível por meio de uma fusão de dados possibilitada pelo E-ETL.

Também foram constatados padrões de espessura de camada que em geral, se repetem para todas as unidades de um mesmo veículo, indicando boa conformidade de qualidade para os veículos vendidos pela marca. Porém esse padrão também indica que, exceto para veículos de configuração SUV (*Sport Utility Vehicle*), mesmo havendo pouca diferença, as portas ou para-lamas dianteiros em geral possuem maior espessura de camadas. A partir dessas descobertas, testes práticos em linha de produção serão realizados para melhorar a homogeneização da pintura dos veículos.

4.1. Análise da Aplicação do E-ETL na fábrica de automóveis

No exemplo de aplicação na fábrica de automóveis, constatou-se que os dados separados em conjuntos com morfologia similar, pela fase E-E do E-ETL, constituem um DL virtual intermediário com informações semiestruturadas em *tags* e *dados*. Essa semiestruturação

de dados permitiu, na fase E-T, que as mesmas regras, ou regras similares, fossem utilizadas para transformar fontes de dados diferentes, incorporando informações semânticas à base. A fase E-L reestrutura os dados em uma nova base de dados, agora limpa e com informações semânticas mais ricas que em sua forma original.

A não necessidade de recriar regras de transformação para todas as fontes de dados reduziu significativamente o tempo de desenvolvimento do projeto, em se comparando com o ETL tradicional. Assim sendo, quando uma nova fonte de dados era adicionada ao E-ETL, em geral, as regras de transformação já criadas foram capazes de realizar limpezas e transformações de forma automática.

Além disso, quando as regras já criadas são insuficientes, o fato do E-ETL possuir padrões matemáticos permitiu que regras já criadas possam ser adaptadas para os novos dados. Um exemplo disso ocorreu com a regra já citada, $\mathcal{R}^m$. Um novo sensor acabou por transmitir a data no formato de *string*. A alteração da expressão regular e da fórmula de transformação da regra foi capaz de realizar transformações nesse novo sensor.

Uma das principais vantagens do E-ETL identificadas no experimento da fábrica de automóveis, em relação às formas tradicionais de ETL, é que, como os mapas das regras de transformação indicam a morfologia esperada para os dados que serão transformados, a própria regra é capaz de identificar a necessidade de transformação dos dados. Assim sendo, uma regra irá transformar um dado automaticamente, se for necessário. No ETL tradicional, o usuário é o responsável por selecionar quais transformações devem ser aplicadas a quais dados.

Embora softwares como o *Microsoft Power BI* permitam que transformações sejam realizadas na base, um padrão matemático implementado em linguagem de programação se mostrou mais eficiente em se tratando de dados heterogêneos, permitindo que fontes de dados oriundas de diferentes etapas do processo industrial de pintura automotiva fossem convergidas para uma base de dados única e homogeneizada.

5. Conclusão

Esse trabalho mostrou o desenvolvimento da E-ETL, uma arquitetura baseada no ETL, mas otimizada para fontes de dados industriais heterogêneas. O E-ETL mostrou-se vantajoso no cenário ao qual foi aplicado, pois técnicas matemáticas permitiram que mapas de transformação de dados já criados, fossem reaproveitados para outras fontes de dados com morfologia similar. Esse reaproveitamento permitiu que uma ferramenta de extração-tratamento-carregamento dos dados fosse desenvolvida mais rapidamente em se comparando às abordagens tradicionais citadas nesse trabalho.

A base de dados gerada utilizando E-ETL, criou um DL virtual intermediário e por intermédio dele, mostrou-se apta a sustentar processo de inteligência computacional e corporativa por meio de softwares de visualização de informações, gerando *insights* significativos aos engenheiros da fábrica de automóveis.

Embora o E-ETL tenha se mostrado próspero, ainda não é provado se tal arquitetura pode ser generalizada para outros domínios industriais e de processamento de informações.

Referências

BANSAL, S. K.; KAGEMANN, Sebastian. Integrating big data: A semantic extract-transform-load framework. **Computer**, v. 48, n. 3, p. 42-50, 2015.

FOLEY, E; GUILLEMETTE, M. G. What is business intelligence? **International Journal of Business Intelligence Research (IJBIR)**, v. 1, n. 4, p. 1-28, 2010.

Google LLC. **Google Cloud Plataform.** Disponível em: < https://cloud.google.com/> Acesso em: 22 jun. 2020.

JINDAL, R.; TANEJA, S. Comparative study of data warehouse design approaches: a survey. **International Journal of Database Management Systems**, v. 4, n. 1, p. 33, 2012.

KOUR, A. Data Warehousing, Data Mining, OLAP and OLTP Technologies Are Indispensable Elements to Support Decision-Making Process in Industrial World. **International Journal of Scientific and Research Publications**, v. 5, n. 1, p. 1-7, 2015.

LIGGINS II, M.; HALL, D.; LLINAS, J. (Ed.). **Handbook of multisensor data fusion: theory and practice**. CRC press, 2017.

MATHIS, C. Data lakes. **Datenbank-Spektrum**, v. 17, n. 3, p. 289-293, 2017.

Oracle Corporation. **Oracle warehouse builder.** Disponível em: < https://www.oracle.com/database/technologies> Acesso em: 22 jun. 2020.

SHUKLA, A.; DHIR, S. Tools for data visualization in business intelligence: case study using the tool Qlikview. In: **Information Systems Design and Intelligent Applications**. Springer, New Delhi, 2016. p. 319-326.

TALEB, I.; DSSOULI, Rachida; SERHANI, Mohamed Adel. Big data pre-processing: A quality framework. In: **2015 IEEE international congress on big data**. IEEE, 2015. p. 191-198.

TRIEU, V. Getting value from Business Intelligence systems: A review and research agenda. **Decision Support Systems**, v. 93, p. 111-124, 2017.

VASSILIADIS, P. A survey of extract–transform–load technology. **International Journal of Data Warehousing and Mining (IJDWM)**, v. 5, n. 3, p. 1-27, 2009.

WILLIAMS, J. W. et al. Semantics for big data access & integration: Improving industrial equipment design through increased data usability. In: **2015 IEEE International Conference on Big Data (Big Data)**. IEEE, 2015. p. 1103-1112.

ZHU, J. et al. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. **Annual Reviews in Control**, v. 46, p. 107-133, 2018.