Geração Automática de Esqueletos para Sistemas ETL

Miguel Guimarães e Orlando Belo

Centro de I&D ALGORITMI
Departamento de Informática, Escola de Engenharia, Universidade do Minho
Campus de Gualtar, 4710-057 Braga, PORTUGAL

Resumo. O projeto de um sistema de povoamento obedece a regras de concepção e implementação bastante específicas, bem como integra um variadíssimo leque de regras de trabalho abrangendo um grande número de tarefas de extração, transformação e carregamento de dados. Com base em experiências práticas de aplicação, hoje sabemos que uma abordagem orientada por padrões de ETL facilita o seu projeto, desenvolvimento e exploração, além de facilitar muito a sua compreensão. Neste trabalho, abordamos esse tipo de aproximação e apresentamos uma forma de produzir automaticamente "esqueletos" para sistemas de ETL, a partir da sua especificação em redes de Petri coloridas. Além de expormos a base de definição e construção de padrões de ETL, bem como a sua especificação em redes de Petri coloridas, apresentamos e discutimos também a forma como podemos providenciar de forma automática a implementação física dos referidos esqueletos.

Palavras-Chave: *Data Warehousing Systems*, Modelação Conceptual, Lógica e Física de Sistemas de ETL, Redes de Petri Coloridas e Kettle.

1 Introdução

Há muito que os profissionais de *data warehousing* reconhecem a importância e a criticidade da implementação de um sistema de povoamento de um *data warehouse*, vulgarmente reconhecido como sistema de ETL (*Extract-Transform-Load*), no ciclo de desenvolvimento de um *sistema de data warehousing* (SDW). O sistema de ETL é apenas um dos componentes do sistema global, mas mesmo assim é o responsável pelo consumo de cerca de 70% de todos os recursos envolvidos na implementação de um SDW [4]. Por natureza, é um sistema bastante complexo, que levanta imensas dificuldades e problemas ao longo da realização de cada uma das fases do tradicional ciclo de desenvolvimento de um SDW. Como tal, é clara a importância da realização de uma implementação adequada de qualquer sistema de povoamento para a garantia do sucesso de um SDW. Uma implementação pobre ou ineficiente pode resultar em algo que não garanta a qualidade da informação que usa para fazer o povoamento e, como tal, coloca em causa a validade e a utilidade do próprio SDW [2]. Como sabemos, um sistema de ETL pode ser especificado, e em alguns casos desenvolvido, através da utilização de uma linguagem de *workflowing*, que nos permita expressar tarefas capazes de serem integradas nos processos macro de um sistema de ETL. Esses processos assentam, normalmente, em três grandes classes de componentes, nomeadamente: angariação, transformação e carregamento de dados. Na prática, isto

significa que qualquer linguagem para ETL deverá ter mecanismos e componentes que nos permitam realizar os mais diversos tipos de operações dentro de cada uma das classes referidas, bem como permitir definir adequadamente a forma como tais componentes coordenam as atividades que desenvolvem entre si. Neste trabalho, enveredámos por uma vertente de desenvolvimento orientada especificamente para a redução da "distância" que existe frequentemente entre fase de modelação conceptual e a fase de apresentação de um modelo físico, mesmo sabendo que não conseguimos, à partida, num modelo conceptual, refletir todos os aspetos físicos que uma verdadeira implementação de um sistema de ETL considera. Assim, para demonstrarmos uma contribuição prática para essa "eliminação", decidimos produzir a partir de uma especificação em *Redes de Petri Coloridas* (RPC) uma primeira versão física do sistema modelado, algo que posteriormente sabemos precisar de ser complementado e ajustado aos requisitos operacionais do sistema de ETL. A esse primeiro esquema físico atribuímos a designação de "esqueleto" ETL, com o intuito de nos referirmos apenas a uma estrutura de suporte elementar. Após uma breve apresentação das RPC e da sua aplicação, utilizando um dos padrões de ETL que desenhámos e modelámos previamente (secção 2), apresentamos o processo de geração dos esqueletos para sistemas ETL (secção 3), revelando a forma como estes podem ser importados e utilizados numa ferramenta de desenvolvimento de sistemas de ETL. Terminamos, depois, este artigo, com algumas breves conclusões.

2 RPC e Padrões ETL

A aplicação de RPC [3] à especificação de padrões de sistemas de ETL, em particular, é algo que tem vindo a ser desenvolvido com algum sucesso. Na terminologia dos SDW, um padrão ETL é um elemento de trabalho que representa uma ou mais tarefas consideradas comuns na implementação de um sistema de ETL. Estes elementos são formalizados usualmente através de um conjunto de processos ou tarefas, recorrendo a um conjunto de atividades pré-estabelecidas [5]. Dois dos padrões mais utilizados em sistemas de ETL - *surrogate key pipelining* e *change data capture* - foram modelados e validados anteriormente em [6], respetivamente, recorrendo a modelos especificados com RPC. O uso desta linguagem para a modelação de padrões de sistemas de ETL permitiu avaliar e validar se um dado modelo reflete *a priori* o tipo de comportamento que se espera que o sistema que representa tenha no futuro. Acreditamos que, tal permite evitar intervenções mais sérias durante a fase de implementação e consequentemente reduzir gastos com recursos computacionais não previstos ou extraordinários. De forma a podermos explicar como a modelação de padrões pode ser executada com RPC, selecionámos um outro padrão de ETL: a dimensão de variação lenta - *slowly changing dimension* (SCD) - com manutenção de história [7]. Este padrão foi projetado e modelado previamente com RPC (fig. 1) de forma a ser capaz de anotar as diversas alterações que vão ocorrendo ao longo do tempo nos dados contidos nos sistemas de informação alvo e que são responsáveis pela alimentação das tabelas de dimensão do SDW. No modelo RPC do padrão ETL SCD, o primeiro passo que se realiza tem a ver com a obtenção de todos os registos que estão na tabela de auditoria relativa à tabela dimensão envolvida - *Audit Records*. Em seguida, os registos recolhidos são

submetidos a um processo de validação assegurado pela transição *Audit Data Verification*. Esta transição é um apontador para a página *Data Verification*, que é a página responsável por verificar, registo a registo, se os dados recolhidos estão de acordo com as regras de negócio estabelecidas na regulação do funcionamento do SDW.

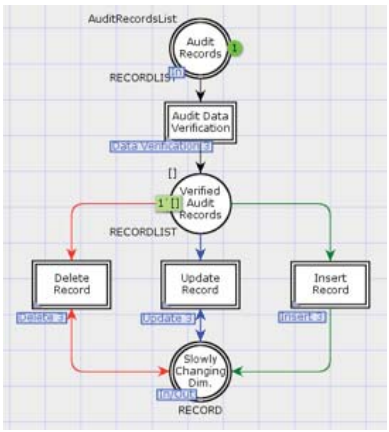

Fig. 1. Modelo RPC para o padrão ETL SCD – figura extraída de [7].

Caso esses dados não apresentem qualquer problema para o processo em causa, eles serão encaminhados para o lugar *Verified Audit Records*. Caso contrário, os registos serão colocadas em tabelas de erro específicas. Mas tarde, os registos que não passaram, por algum motivo, o filtro de validação, serão analisados e tratados por alguém responsável por essa tarefa. Passada esta primeira fase de tratamento, temos um conjunto de registos devidamente verificados no lugar *Verified Audit Records*.

3 O Processo de Geração de Esqueletos ETL

A exportação do modelo RPC apresentado pode ser realizada através do ambiente de trabalho da ferramenta CPN Tools [1]. Este é processo bastante simples que pode ser realizado a partir da área índice da CPN Tools, que nos permite, basicamente, guardar o modelo desenvolvido nos formatos *Petri Net Markup Language* (PNML) ou *Extensible Markup Language* (XML). No nosso caso, optámos pelo segundo formato, uma vez que nos permite descrever parcialmente o comportamento de programas que são capazes de os processar e porque segue a norma ISO/IEC 15909. Na fig.2 podemos ver um pequeno excerto do ficheiro XML que foi gerado para o nosso exemplo de trabalho. O processo relativo à criação e implementação de um esqueleto ETL apenas terminará quando este for importado para o ambiente de trabalho de uma ferramenta de construção de sistemas de ETL. Para o nosso caso de demonstração e acolhimento dos esqueletos modelados escolhemos: o *Pentaho Data Integration* (PDI), uma ferramenta *open-source* que não requer a aquisição de uma licença para a sua utilização, podendo ser instalada em várias plataformas computacionais, como o Windows, o Linux, ou Mac OSX. No caso do padrão ETL SCD que foi modelado, o

resultado final foi uma tradução direta do modelo RPC. Cada uma das quatro transições existentes na página principal do modelo foram traduzidas diretamente num componente ETL do PDI, denominado *Execute Transformation*, que é o componente responsável por executar uma dada transformação. Assim, a geração do esqueleto ETL traduziu-se em quatro *Execute Transformation* componentes, denominados respetivamente por *Audit Data Verification*, *Delete Records*, *Insert Records* e *Update Records*.

```
<page id="ID1422273702">
    <pageattr name="SCD/H 3"/>
    -- Representação XML para o lugar Verified Audit Records.
    <place id="ID1422273978">
      (...)
      <text>Verified Audit Records</text>
      (...)
    </place>
    -- Representação XML para o transição Insert Record.
    <trans explicit="false" id="ID1422273704">
      ( ... )
      <text>Insert Record</text>
      ( ... )
      -- Representação XML para a referência da transição a outra página.
      <subst portsock="(ID1422273769,ID1422273978)(ID1422273765,ID1422273981)"
subpage="ID1422273710">
      ( ... )
      </trans>
      ( ... )
</page>
```

Fig. 2. Fragmentos de um ficheiro XML com a especificação de um modelo RPC.

No formato XML os componentes são identificados pelo elemento XML *step* (fig. 3). Para diferenciar cada um dos diferentes componentes devemos utilizar o elemento XML *type*, que no caso de uma tabela de entrada tem o texto *TableInput* e numa tabela de saída o texto *TableOutput*. No caso do componente de validação, o elemento *type* tem o texto *Validator*. A ligação entre diferentes componentes ETL é realizada através do elemento XML *order*. Uma ligação é um *hop* com elementos *from* e *to*, que correspondem, respetivamente, ao *step* origem e ao *step* destino. No caso das ligações, estas não são identificadas pelo tipo do *step*, mas sim pelo seu nome. Logo não poderá haver nomes de componentes ETL repetidos. Esta é uma regra que qualquer esqueleto ETL tem que necessariamente de obedecer. No caso particular da transformação *Delete Records* existe uma tabela de entrada, que contém os registos verificados durante a transformação *Audit Data Verification*, três tabelas de saída e um componente de seleção. Quanto às tabelas de entrada e de saída a sua representação em XML já foi apresentada. Mas, ainda falta a componente de seleção. Esta é identificada pelo elemento *type* com o texto *SwitchCase*. Para a transformação *Update Records* já definimos todas as representações XML dos seus componentes, não havendo, assim, a necessidade de os detalhar. Quanto ao processo de importação este é bastante simples. Através da opção *File* do menu principal da ferramenta PDI, acedemos à opção *Import from an XML file* e identificamos o ficheiro relativo ao esqueleto ETL que queremos importar. Depois de confirmar, obtemos o resultado que

podemos observar na fig. 4. Nela podemos identificar as quatro tarefas principais anteriormente modeladas, que aqui estão representadas por quatro componentes *Transformation Executer* do PDI.

```
-- Representação XML para o step Audit Data Verification.
<step>
        <name>Audit Data Verification</name>
        <type>TransExecutor</type>
        ( ... )
        <specification_method>filename</specification_method>
        <filename>/Users/hmg/Desktop/auditDataVerification.ktr</filename>
        ( ... )
</step>
-- Representação XML para o step DimCustomer.
<step>
    <name>DimCustomer</name>
    <type>TableOutput</type>
    ( ... )
</step>
```

Fig. 3. Pequeno excerto de um ficheiro XML relativo a alguns *steps* incluídos na modelo físico do esqueleto importado.

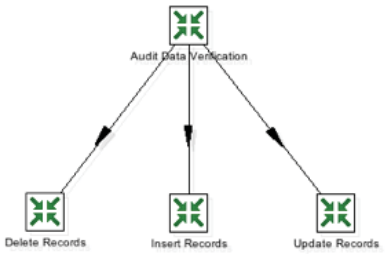

Fig. 4. O esquema físico geral do padrão ETL SCD em PDI.

De seguida, na fig. 5 podemos ver duas outras representações mais detalhadas, cada uma delas correspondendo a dois *Transformation Executers* integrados no esquema físico geral do esqueleto ETL em SCD, representando, por sua vez, duas das tarefas que foram previamente modeladas também com RPC - *Audit Data Verification* e *Insert Record*, respetivamente. De referir que, por falta de espaço, os modelos RPC destas duas tarefas não foram incluídos neste trabalho.

6 Conclusões e Trabalho Futuro

Depois da realização deste trabalho, não hesitamos em dizer que a geração de esqueletos ETL a partir de uma dada ferramenta de especificação (como as RPC) para posterior importação numa ferramenta de implementação de sistemas de ETL é uma

clara mais valia, sobretudo porque dá um outro significado e importância ao modelo e ao próprio processo de modelação de um sistema de ETL, além de adiantar algum serviço de implementação de uma forma praticamente automática. Diríamos mesmo que, com a geração adequada e validação de um modelo RPC para um sistema ETL (ou de um padrão ETL como foi o nosso caso), a estrutura base do modelo físico correspondente à sua implementação praticamente está assegurada. Neste artigo, utilizámos apenas como caso de estudo um único padrão ETL, mas isso foi suficiente para demonstrar como uma especificação conceptual (ou lógica) pode ser diretamente traduzida num modelo físico passível de ser executado.

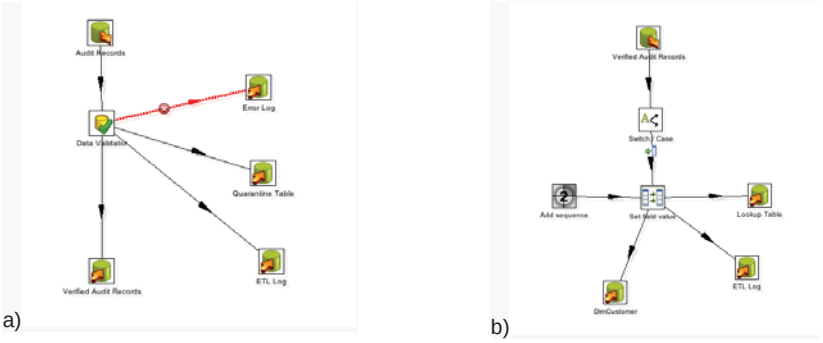

Fig. 5. Esquemas físicos das atividades Audit Data Verification (a) e Insert Record (b) em PDI.

Referências

[1] CPN Tools, 2014. [Online] Availale at <http://cpntools.org/> [Accessed on 28 April 2014].

[2] English, L. P., Improving data warehouse and business information quality: methods for reducing costs and increasing profits. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[3] Jensen, K., Kristensen, M., Wells, L., Coloured petri nets and cpn tools for modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transf. 9, 213–254, 2007.

[4] R. Kimball and J. Caserta, The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. 2004, p. 528.

[5] Oliveira, B. & Belo, O., ETL Standard Processes Modelling: A Novel BPMN Approach. In 15th International Conference on Enterprise Information Systems (ICEIS), 2013.

[6] Silva, D., Fernandes, J.M., Belo, O., "Assisting Data Warehousing Populating Processes Design Through Simulation Using Coloured Petri Nets", In Proceedings of 3rd Industrial Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH' 2013), Reykjavik, Iceland, July 29-31, 2013.

[7] Silva, D., Fernandes, J.M., Belo, O., "Approaching ETL Processes Modelling with Coloured Petri Nets", Technical Report, Department of Informatics, University of Minho, 2013.