



Uma ferramenta de geração de Workflows para a plataforma ERP5

Vanderson Mota dos Santos*
Rogério Atem de Carvalho**

Palavras-chave: Geração de código. Workflows. ERP5. UML. ERP. Diagrama de Máquina de Estados.

Na última década, os sistemas integrados de gestão ou Enterprise Resource Planning (ERP) vêm sendo largamente utilizados pelas empresas, com o objetivo de estas obterem um maior controle dos recursos, desde os funcionários até os insumos de produção. Entretanto, na maioria dos casos, somente empresas de grande porte conseguem adquirir um sistema desse tipo, devido a o seu custo geralmente ser elevado.

Para que, além das grandes corporações, as pequenas e médias empresas possuam este tipo de vantagem competitiva, foi criado o ERP5, que é um ERP de código livre que tem como objetivo prover a solução de gerenciamento de todas os setores de uma empresa. Desenvolvido e mantido pela empresa francesa NEXEDI, o ERP5 utiliza um framework de desenvolvimento de aplicações web chamado Zope, que é desenvolvido com base na linguagem de script Python, e algumas partes mais críticas escritas em C. Esse framework oferece recursos como um banco de objetos (ZODB), sistema de criação de interfaces baseadas em XML (ZPT) e um sistema de gerenciamento de conteúdo (CMF), que possui diversos serviços como de catalogação de informação (Zcatalog), e de máquina de estados (DCWorkflow).

Segundo Carvalho e Campos:

Adicionalmente, o ERP5 implementa sincronização entre diferentes bases de objetos, através do protocolo SyncML XML, e mapeamento Objeto-Relacional que armazena atributos de indexação num banco de dados MySQL, proporcionando uma busca visivelmente mais rápida em relação à funcionalidade de catalogação padrão do ZOPE (CARVALHO; CAMPOS, 2006).

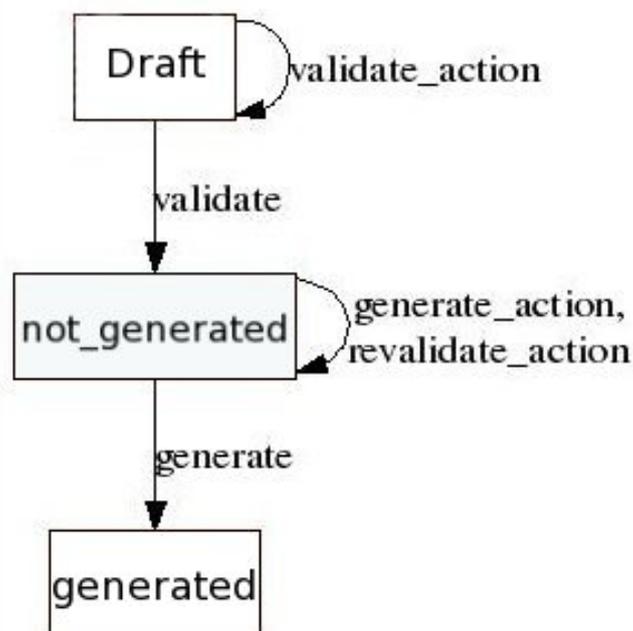
* Bolsista do CNPq. Desenvolvimento de software.

** Doutor em Ciências da Engenharia. Coordenador de Pesquisa do CEFET Campos.

O ERP5 foi assim batizado devido a cinco conceitos-base que fazem parte do seu modelo abstrato, segundo Smets-Solanes:

Recurso: representa um recurso abstrato em um processo de negócio. (ex. a habilidade de um indivíduo, uma moeda, uma matéria-prima, um produto). Nó: o lugar que pode enviar e receber uma quantidade de recursos. Os Nós podem enviar recursos para entidades físicas (ex. um workshop que recebe matéria-prima, processa-a e a envia) ou abstratas (ex. uma conta de banco que pode receber dinheiro). Estoque são um exemplo de nó. Meta-Nós são nós que contêm outros nós. Uma companhia é um meta-nó. Um projeto é um recurso e um nó. Movimento: um movimento descreve o ato do movimento de um grupo de recursos entre dois nós em um dado tempo e uma dada duração. Por exemplo, um movimento pode enviar material de um estoque para um workshop; outro movimento pode enviar dinheiro de uma conta bancária para outra conta. Caminho: o caminho permite definir uma forma de um nó acessar um recurso de que eventualmente, possa precisar. Preços e perfis comerciais podem ser agrupados em um caminho, de forma a definir um preço base para um determinado recurso obtido em um fabricante. Logo, eles podem definir uma maneira de o workshop receber um recurso de um estoque. Um caminho tem data de início e de fim. Ele pode ser usado para representar a atribuição de um indivíduo a um projeto provisório. Item: um item descreve uma instância física de um recurso. Um movimento pode ser expandido em uma série de movimentos buscáveis por meio dos itens. Itens, então, permitem definir como um dado recurso foi realmente enviado (ex. pacotes, números de série de itens em cada container, etc.) (SMETS-SOLANES, 2002).

Todo novo módulo, agregado ao ERP5, é desenvolvido tendo como base esses cinco conceitos, de forma que o comportamento dessas novas estruturas permaneçam de acordo com os workflows, que visam definir todo o trajeto da informação.



Exemplo 1: Workflow da ferramenta no ERP5

Para agilizar a construção de novos módulos, minimizando os custos, tempo de desenvolvimento e testes, faz-se necessária a criação de uma ferramenta que determine todo o fluxo de informação destes, baseando-se em diagramas da Unified Modeling Language (UML), linguagem para modelar sistemas orientados a objeto, possibilitando a produção de módulos adicionais por desenvolvedores iniciantes nesta plataforma, bastando que estes possuam um conhecimento razoável de análise e projeto de sistemas, bem como sobre o aspecto funcional da plataforma em questão.

O desenvolvedor projeta o Diagrama de Classes (responsável por construir todos os elementos estruturais (dados e interface) dos módulos, sendo que esta parte não é tratada pela ferramenta em discussão) e o Diagrama de Máquina de Estados numa ferramenta CASE-UML (Computer Aided Software Engineering com notação UML) e logo depois exporta esse modelo para um arquivo no formato XML Metadata Interchange (XMI) que, por sua vez, usa a XML como linguagem de notação, como o próprio nome sugere.



Exemplo 2: Diagrama de Máquina de Estados

Devido à flexibilidade da XML, existem ferramentas CASE que geram um arquivo XMI de maneiras e formatações diferentes. Portanto, o desenvolvimento da ferramenta apresentada neste artigo baseou-se no XMI exportado pelo software ArgoUML(ferramenta CASE-UML), pois além de ser totalmente open-source, possui uma comunidade de desenvolvedores bastante ativa.

A geração do workflow dar-se-á após o *parsing do XMI*, ou seja, um algoritmo irá analisar toda a informação contida neste arquivo e transformar um Diagrama de Máquina de Estados (que por sua vez define o fluxo de mensagens entre objetos) em um workflow referente à(s) classe(s) especificadas anteriormente no modelo, já dentro de uma instância da plataforma ERP5.



Exemplo 3: Máquina de estados anterior convertida para workflow no ERP5

Para se adequar aos padrões desta plataforma, todos os estados iniciais de um Diagrama de Máquina de Estados serão tratados como Draft, enquanto que os estados finais serão ignorados, ou seja, o estado anterior, ao fim, será tratado como último. Qualquer elemento dentro do modelo não deverá apresentar palavras acentuadas, a fim de seguir a codificação do python, que é a linguagem de desenvolvimento desta ferramenta.

É considerável a redução do tempo de desenvolvimento e teste de novos módulos para esta plataforma, pois o tempo em que os desenvolvedores estariam ocupados lidando diretamente com o código para definir como, quando e por quem a informação irá passar, eles se concentrarão em resolver problemas referentes à lógica do negócio, e, conseqüentemente, produzir módulos de melhor qualidade, além de atender à crescente demanda do mercado de ERP de maneira mais ágil.

Referências

BROCKMANN, M et al. *Kit de construção de aplicativos de Web*. Rio de Janeiro: Alta Books, 2002. 562p.

BRUECK, D.; TANNER, S.; *Python 2.1 Bible*. New York: Hungry Minds, 2001. 731p.

FOWLER, M; KENDALL, K. *UML Essencial: um breve guia para a linguagem-padrão de modelagem de objetos*. 2.ed. Rio de Janeiro: Bookman, 2000. 169p.

LARMAN, C. *Utilizando UML e padrões*. 2.ed. São Paulo: NOBEL, 2002. 607p.

CARVALHO, R. A; CAMPOS, R. A Development Process Proposal for the ERP5 System. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS, Taipei. Proceedings of the 2006.

SMETS-SOLANES, J-P. ERP5: a Technical Introduction. Disponível em: <http://www.erp5.org/sections/documentation/articles/linuxtag.html/view>. Acesso em: 23 jun. 2002.

_____; CARVALHO, R. A. de. ERP5: A Next-Generation, Open-Source ERP Architecture. *IEEE IT Professional*, v. 5, n. 4, jul., 2003.