

MANUTENÇÃO DINÂMICA DE MODELOS EM COMPUTAÇÃO SENSÍVEL AO CONTEXTO

Rodrigo Hernandez SOARES¹; Ricardo Couto Antunes da ROCHA²

PALAVRAS-CHAVE: CEP, Esper, Computação Sensível ao Contexto, SBE.

1 - INTRODUÇÃO

Em cenários distribuídos e heterogêneos, a interpretação e inferência de informações contextuais pode variar dinamicamente devido a fatores como, por exemplo, a mudança do sistema sensível ao contexto com o qual a aplicação está interagindo.

Um modelo de contexto especifica como as informações contextuais são caracterizadas em um ambiente sensível ao contexto, em termos de aspectos como tipos e inter-relacionamento entre eles, estruturas e propriedades, primitivas de manipulação das informações e as condições ou regras responsáveis pela inferência das informações contextuais. Como o modelo contextual está diretamente relacionado ao funcionamento dos sistemas baseados em eventos (SBE) nos quais as aplicações sensíveis ao contexto são fundamentadas, esse funcionamento, juntamente com as linguagens de assinatura/notificação de interesses, exercem papel fundamental em definir como os modelos de contexto são gerenciados em tempo de execução. Nas arquiteturas em geral, este gerenciamento é estático.

O objetivo deste trabalho é propor uma metodologia de avaliação do suporte dinâmico a atualização de modelos de contexto em *middleware* para computação sensível ao contexto, que permita estabelecer critérios de implementação e impactos na sua arquitetura. A principal contribuição é a metodologia proposta, que pode ser aplicada em sistemas de *middleware* sensíveis ao contexto para avaliar quais os cenários contextuais dinâmicos podem ser implementados e os impactos na refatoração da arquitetura para suportar tais requisitos.

2 - MATERIAL E MÉTODOS

A metodologia utilizada para avaliação do suporte dinâmico a atualização de modelo

¹ Instituto de Informática - rodrigosoares@inf.ufg.br

² Instituto de Informática - ricardo@inf.ufg.br

de contexto propõe a identificação e classificação dos objetos que compõem um modelo, dos atores que interagem com esse modelo de forma dinâmica (criação, atualização e remoção de objetos de modelo em tempo de execução) e das consequências que essas interações podem causar sobre diversos aspectos da máquina de processamento de eventos e do sistema sensível ao contexto como um todo. Além disso, um estudo de caso utilizando a máquina de processamento de eventos CEP³ *Esper* (EsperTech) é proposto, com o intuito de materializar os objetos de modelo no contexto dos SBEs e de demonstrar como a atuação dos atores sobre esses objetos pode impactar o funcionamento desse tipo de sistema.

3 - RESULTADOS E DISCUSSÃO

Essa seção é dividida em duas partes, sendo que a primeira parte detalha a proposta de classificação citada na seção anterior enquanto a segunda descreve o estudo de caso.

3.1 - Classificação

Objetos de modelos são elementos conceituais que compõem um modelo de contexto e que podem sofrer uma atualização dinâmica. Neste trabalho, classificamos os objetos de modelo em:

- **Eventos:** representam alterações em situações contextuais de interesse de uma ou mais aplicações ou serviços.
- **Entidades:** representam os objetos que compõem o cenário de computação sensível ao contexto e cujo estado é compreendido como informação contextual (ex: entidade Professor no contexto de uma universidade).
- **Regras:** representam condições, expressões e consultas, descritas segundo uma linguagem definida pela máquina de eventos, e que permite tanto a aplicações como ao restante do sistema indicar situações ou condições que podem produzir novos eventos contextuais e/ou associá-los a entidades.
- **Operações:** representam primitivas específicas de um tipo de contexto e implementadas na máquina de eventos. Por exemplo, um contexto de localização geográfica pode oferecer a operação de *distância*.

³ Complex Event Processing - é uma tecnologia emergente que permite definir, gerenciar e prever eventos, situações, condições excepcionais, oportunidades e ameaças em redes complexas e heterogêneas (Bass, 2007).

Atores são os elementos que compõem um cenário de computação sensível ao contexto e que podem atuar no ambiente de maneira a modificar os objetos de modelos dinamicamente. Neste trabalho, classificamos os atores em:

- **Aplicação:** é a aplicação sensível ao contexto e usuário final dos eventos contextuais produzidos pelo ambiente.
- **Sistema:** é um sistema sensível ao contexto, independente, e capaz de receber, processar e disseminar eventos contextuais para aplicações. Um ambiente distribuído pode ser composto por diversos sistemas.
- **Localização:** é a localização física dos sistemas. A localização é um ator de atualização nos modelos quando o sistema sensível ao contexto é distribuído e a localização física dos sistemas define um escopo particular de uso das informações.
- **Usuário:** são os usuários finais das aplicações, que podem atuar na modificação dos modelos.
- **Contexto:** são as informações contextuais, que podem levar a uma mudança de comportamento em qualquer outro ator do ambiente.

As *consequências*, que descrevem as relações de interferência entre *atores* e *objetos de modelo*, são descritas na Figura 1. Nela, é possível perceber quais *objetos de modelos* podem ser alterados por quais *atores* e quais são as *consequências* dessas alterações sobre a máquina de eventos. Nesse trabalho, as consequências foram classificadas em:

Reconhecimento de tipos: alterações em tempo de execução nos objetos *Evento* e *Entidade* podem ocasionar o problema do reconhecimento de tipos. Esse problema pode surgir, por exemplo, se o *Sistema* alterar os objetos de *Eventos* e *Entidades*, o que pode exigir que essas alterações sejam refletidas na *Aplicação*, que deve, de alguma forma, perceber e incorporar essas alterações realizadas no modelo.

Funcionamento da máquina de eventos: a atualização de um objeto *Operação* é capaz de fazer com que uma mesma regra gere resultados diferentes dependendo da forma com que essa operação é implementada. Assim sendo, temos que as operações estão diretamente relacionadas com o funcionamento da máquina de eventos, pois são capazes de alterar a forma com que os eventos são processados independente das regras registradas por clientes.

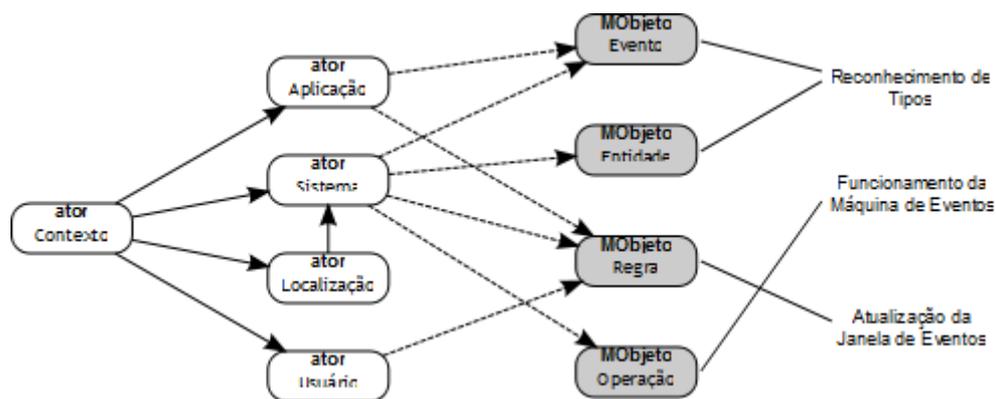


Figura 1: Relação de causalidade entre atores, modificações nos objetos de modelos e impacto nos sistemas.

Atualização da janela de eventos: máquinas que provêm o conceito de estado entre eventos definem janelas em tempo de execução que restringem os eventos que podem ser envolvidos em um casamento com interesses de clientes. A alteração de uma regra pode, por exemplo, fazer com que todos os eventos que compõem essa janela se tornem inválidos (ou seja, já não são de interesse do cliente).

3.2 - Estudo de Caso

O estudo de caso apresentado nessa seção se baseia em um exemplo de como o estado "ocupado" pode ser inferido através do uso da *engine CEP Esper*. A Figura 2 mostra um exemplo de regra descrita na linguagem EPL capaz de inferir esse estado. Na figura, os objetos de modelo identificados são: *Ocupado*, *LocalizacaoPessoa*, *AtividadeComputador* (*Evento*), *localizacao*, *pessoa*, *computador* (*Entidade*), *groupwin*, *lastevent* (*Operação*) e toda a expressão descrita na figura (*Regra*).

```

insert into Ocupado
select fluxo1.pessoa as pessoa,fluxo1.localizacao as localizacao
from
LocalizacaoPessoa.std:groupwin(pessoa.nome).std:lastevent() as fluxo1,
AtividadeComputador.std:groupwin(computador.dono.nome).std:lastevent()
as fluxo2
where
fluxo1.pessoa.nome = fluxo2.computador.dono.nome and
fluxo1.localizacao.coordnada = fluxo2.localizacao.coordnada and
fluxo1.localizacao.coordnada = fluxo1.pessoa.saladetrabalho

```

Figura 2: consulta EPL que descreve o estado "ocupado".

De acordo com a regra descrita, uma pessoa será considerada "ocupada" se houver atividade de seu computador e a localização do computador e da pessoa forem iguais, sendo que essa localização deve ser a sala de trabalho da pessoa. Essa regra gera como resultado o evento *Ocupado*. Se a modelagem desse evento for alterada pelo sistema sensível ao contexto, então os possíveis clientes que registraram interesse nesse tipo de evento devem incorporar essa atualização para que não haja incompatibilidade de modelos entre o sistema e a aplicação cliente.

Outro impacto em modelagem de entidades pode ser percebido na forma com que a entidade *localizacao* é modelada. Para que a regra funcione da forma esperada, o campo *coordenada* deve ser modelado como um atributo textual. Porém, se esse campo for modelado como coordenada geométrica, então a regra pode não funcionar como esperado. Isso porque se a definição da relação de igualdade entre duas coordenadas for feita com base na comparação dos valores numéricos associados às mesmas, é possível que a *engine* não gere um evento *Ocupado* mesmo que uma pessoa esteja em sua sala de trabalho utilizando o computador. Uma possível solução para esse problema seria criar uma operação, como por exemplo a operação *proximidade*, que definiria o nível de proximidade entre duas entidades de acordo com algum critério e, com base nisso, retornaria como resultado se essas entidades estariam em um mesmo local.

4 - CONCLUSÕES

Este artigo apresentou uma discussão sobre a atualização dinâmica de modelos de contexto em computação sensível ao contexto e o seu impacto nas máquinas de processamento de eventos. Os trabalhos futuros desta pesquisa envolverão a generalização e a modelagem teórica do problema de atualização dinâmica de modelos, assim como uma proposta arquitetural e a implementação de um protótipo de *middleware* para computação sensível ao contexto com suporte para adaptação dinâmica dos modelos contextuais.

Bibliografia

(Bass, 2007) Bass, T. (Abril de 2007). Acesso em Junho de 2011, disponível em What is Complex Event Processing?: <http://tibcoblogs.com/cep/2007/04/23/what-is-complex-event-processing-part-1/>

EsperTech. (s.d.). Acesso em Junho de 2011, disponível em <http://esper.codehaus.org/>