

CRIAÇÃO DE BIBLIOTECA DE METADADOS PARA FRAMEWORK DE GAMIFICAÇÃO

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA

(PIBIC/CNPq/INPE)

Bolsista: Jeremias Ponciano da Silva
E-mail: jeremiason2911@gmail.com

Orientador: Eduardo Martins Guerra
E-mail: guerraem@gmail.com

Julho de 2016
São José dos Campos

1 – Introdução

A gamificação (do original do inglês: Gamification) corresponde ao uso de mecanismos de jogos orientados ao objetivo de resolver problemas práticos ou de despertar engajamento entre um público específico. Com frequência cada vez maior, esse conjunto de técnicas tem sido aplicado por empresas e entidades de diversos segmentos como alternativas às abordagens tradicionais, sobretudo no que se refere a encorajar pessoas a adotarem determinados comportamentos ou a familiarizarem-se com as funcionalidades da aplicação.

O conceito de gamificação está cada vez mais frequente entre aplicações científicas que buscam o apoio da sociedade, o que é conhecido como citizen science. Dessa forma, isso não é só aplicado como estratégia de marketing pelo mercado, como também em aplicações que buscam o maior aproveitamento de perspectiva subjetiva em aplicações científica. Um grande exemplo de aplicação dessa técnica é para identificação e classificação de imagens.

Para simplificar a construção desse tipo de aplicação, existem alguns componentes reutilizáveis que provêm algumas funcionalidade de gamificação. Normalmente, esses componentes são implantados em servidores na internet que podem ser utilizados por diversas aplicações. Exemplos de funcionalidades desses componentes são: registro de pontos, registros de conquistas e criação de rankings. O Framework Esfinge Gamification provê uma solução baseada em metadados para a implementação mais flexível e desacoplada desse tipo de solução. Porém, apesar de extensível, os metadados ainda não refletem todas as possibilidades em relação aos mecanismos da gamificação.

Desta forma, esse projeto de iniciação científica tem como o objetivo evoluir o Framework baseado em metadados Esfinge Gamification, adicionando metadados sobre os diferentes tipos de mecanismos que podem estar presentes em uma aplicação com gamificação. Adicionalmente serão aplicadas soluções que vão aumentar a flexibilidade na forma de definição dos metadados e realizar a validação das anotações.

2 – O Framework Esfinge Gamification

O Framework Esfinge Gamification provê uma solução baseada em metadados, que permite a definição da lógica de Gamificação de uma aplicação, com pontos, troféus, etc, possa ser feita de forma declarativa e separa da funcionalidade principal.

2.1 - Características

Este Framework tem o principal objetivo de dissociar as preocupações Gamification da aplicação. Sua ideia principal é permitir que os desenvolvedores se concentrem na lógica da aplicação, apenas a adição de informações sobre como o Gamification devem trabalhar em cada funcionalidade invocada. O Framework intercepta estas invocações e executar a lógica Gamification apropriado. Fornece pontos de extensão que permitem a introdução de comportamento Gamification específica da aplicação.

O Framework define vários tipos de realizações e diferentes implementações para

armazenar Gamification informações sobre os usuários. Para registrar como o Gamification devem ser manuseados na aplicação, Esfinge Gamification usa anotações que devem ser adicionados nos métodos de negócio nas classes de aplicação. O Framework proposto pode ser facilmente integrado em qualquer aplicação Java, sendo responsável por registrar realizações do usuário e não dependem do em qualquer informação relacionada com o domínio específico da aplicação. Ele também fornece uma API que pode ser acessado diretamente pelo aplicativo para recuperar e mostrar informações relacionadas com as realizações do usuário, tais quantos pontos um usuário tem ou sua posição no ranking.

3 – Reflexão e Anotações

Existem situações em que somente a orientação a objetos não é suficiente pra resolver todos os problemas do desenvolvimento de software, mas em alguns casos pode ser utilizada a reflexão.

A reflexão e um recurso em que um software pode se observar e mudar seu comportamento, através da introspecção, onde o software obtém informações sobre seu próprio código. Este conceito também pode ser chamado de metaprogramação, onde pode-se dizer que o código manipula o próprio código. O objetivo primário é manipular instâncias de objetos através do uso de anotações. Dessa maneira pode-se interagir com instâncias de classes que não possuem estrutura conhecida.

Na linguagem JAVA utiliza-se a API Reflection, do pacote `java.lang.reflect`, quando se faz necessário o uso da reflexão. A alteração de classes manipuladas pela API Reflection deve ser feita com o auxílio da manipulação de bytecode de classes, pois em linguagem JAVA não é possível alterar classes que tenham sido carregadas na máquina virtual (JVM – Java Virtual Machine).

4 – Metadados

Os metadados são estruturas de informação que descrevem e ajudam a recuperar ou manipular fontes de informação. Metadados são muitas vezes referidos como “dados sobre dados” (GUENTHER e RADEBAUGH, 2004).

No contexto de linguagens de programação orientadas a objetos, os metadados são comumente informações a respeito das classes. Usando uma combinação entre metadados e a reflexão consegue-se informações a respeito das classes em um software, sem necessariamente conhecer a estrutura destas classes previamente. Além dos metadados fornecidos pelas próprias classes, o que se obtém com a API Reflection na linguagem Java, existem fontes externas de metadados. Metadados podem ser marcados na estrutura das classes, através do uso de anotações. Os metadados assim dizem respeito às informações sobre o código das classes.

5 – Atividades realizadas e Resultados

Segue um resumo das atividades realizadas e uma lista das atividades realizadas durante o projeto e resultados alcançados

5.1 – Atividades Realizadas

Na primeira etapa realizada foi o estudo sobre alguns conceitos fundamentais sobre Gamificação, técnicas, estratégias, mecanismos visto dos jogos que podem ser aplicadas nos mais diversos cenários, e estudos de técnicas para criação de metadados.

Na segunda foi dada continuidade ao estudo relação a abordagens de desenvolvimento de software que são utilizadas. Foram estudados padrões de projeto de software e também o uso de Anotações e Reflexões, que são usadas para ampliar a eficiência da linguagem Java no design de sistemas onde a orientação a objetos não é suficiente.

Na terceira etapa foi dada a continuidade na evoluir o Framework, adicionando metadados sobre os diferentes tipos de mecanismos que podem estar presentes em uma aplicação com Gamificação, e feito um exemplo simples para prova de conceito.

5.2 – Resultados

Foi iniciado um estudo de caso de uso, com base em um outro software já existente, modificando para testar as novas annotations desenvolvidas durante o IC.

Foram feitas algumas implementações deste caso de uso, como pode se ver nas figuras 1 e figura 2.

```
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

import com.esfinge.gamification.annotation.GamificationProcessor;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
@GamificationProcessor(IncrementScoreProcessor.class)
public @interface IncrementScore
{
    String vote();
}
```

Figura 1 – Implementação de Esfinge Gamification.

```

import java.lang.annotation.Annotation;
import java.lang.reflect.Method;

import com.esfinge.gamification.achievement.Point;
import com.esfinge.gamification.mechanics.Game;
import com.esfinge.gamification.processors.AchievementProcessor;
import com.esfinge.gamification.utils.ReflectionUtils;
import com.leona.bean.VoteBean;

public class IncrementScoreProcessor implements AchievementProcessor
{
    private String param;

    @Override
    public void receiveAnnotation(Annotation arg0)
    {
        IncrementScore sc = (IncrementScore) arg0;

        param = sc.vote();
    }

    @Override
    public void process(Game game, Object encapsulated, Method method, Object[] args)
    {
        // Find user
        int index = ReflectionUtils.findParameterNamed(method, param);
        VoteBean vote = (VoteBean) args[index];

        // Score Logic
        int score = calculateScore(vote);

        Point p = new Point((int)score, "score");
        game.addAchievement(vote.getUserId(), p);
    }
}

```

Figura 2 – Implementação de Esfinge Gamification.

Porem foi encontra algumas dificuldades de implementação do caso de uso, por falda de conhecimento sobre o sistema, e também por dificuldade de conhecimento das linguagens de programação para web, foi verificado que a necessidade de mais estudo sobre outras linhagens de programação específicas para websites, e também um pouco mais de tempo para a implementação do caso de uso proposto.

6 – Conclusão

Esta solução é a ideia central da Esfinge Gamification. Pode ser aplicável aos sistemas que precisa de uma lógica Gamification, independentemente do seu domínio. A estrutura usa anotações para configurar o que é a lógica Gamification que deve ser executado para cada método de aplicação. Ele permite a configuração de vários tipos de realizações e a extensão de como essas realizações são armazenados, e facilitando programação, não interferindo diretamente na lógica do programa em que vai ser aplicado, podendo o programador focar somente nas partes em que via ser colocado algum tipo de mecanismo de Gameficacao.

Infelizmente não pode ser concluído um caso de uso para se analisar com mais precisão o desenvolvimento de annotations que foi desenvolvida no decorrer deste programa de Iniciação Científica.

7 – Referências

E. Guerra. A Conceptual Model for Metadata-based Frameworks. Tese de Doutorado, Instituto Tecnológico de Aeronáutica, Brasil, 2010.

E. Guerra, M. Cardoso, F. Silveira e C. Fernandes. Idioms for Code Annotations in Java Language. In SugarLoafPLOP'2010 - VIII Latin American Conference on Pattern Languages of Programming, Salvador, Brasil, 2010.

E. Guerra, J. Souza and C. Fernandes. A Pattern Language for Metadata-based Frameworks. In 16th Conference on Pattern Languages of Programs, Chicago, EUA, 2009.

Y. Vianna, M. Vianna, B. Medina, S. Tanaka. Gamification, INC., disponível em <http://livrogamification.com.br>, 2012.

K. Werbach, D. Hunter. How Game Thinking Can Revolutionize Your Business, Wharton Digital Press, 2012.

S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and Dan Dixon. 2011. Gamification. using game-design elements in non-gaming contexts. In CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11). ACM, New York, NY, USA, 2425-2428. DOI=10.1145/1979742.1979575
<http://doi.acm.org/10.1145/1979742.1979575>

I. Iacovides, C. Jennett, C. Cornish-Trestrail, and A. L. Cox. 2013. Do games attract or sustain engagement in citizen science?: a study of volunteer motivations. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, NY, USA, 1101-1106. DOI=10.1145/2468356.2468553
<http://doi.acm.org/10.1145/2468356.2468553>.