Criação de um tradutor XML para a linguagem de marcações sobre emoção EmotionML

Marcelo Nichele ¹ Patricia A. Jaques ¹

Resumo: Este artigo visa a apresentar e descrever a criação de um tradutor XML para a linguagem EmotionML 1.0 A EmotionML é uma linguagem de marcação criada para padronizar a representação das emoções em computadores. O tradutor deve ser capaz de: (i) identificar os elementos da EmotionML, em um documento XML, e retorná-los como objetos instanciados; (ii) gerar dinamicamente classes a partir da gramática definida para a linguagem EmotionML para a instanciação de objetos; (iii) gerar arquivos EmotionML a partir de objetos EmotionML instanciados. Dessa forma, o tradutor proposto pode ser usado nos mais variados tipos de aplicações em Computação Afetiva que envolvam inferência, expressão ou síntese de emoções. A utilização do tradutor permite a um sistema computacional afetivo recuperar informações mantidas em arquivos, modificar os dados do arquivo em tempo real, assim como armazenar novamente as informações no formato EmotionML para futuros acessos ou modificações.

Palavras-chave: Computação afetiva. EmotionML. Tradutor.

Abstract: This article aims at presenting and describing a translator for the language EmotionML 1.0. EmotionML is a XML-based markup language designed to standardize the representation of emotions in computers. The translator must be able to: (i) identify the elements of EmotionML in an XML document, and return them as instantiated objects, (ii) dynamically generate classes from the grammar defined for the language EmotionML in order to instantiate objects; (iii) generate files from EmotionML instantiated objects. The proposed translator can be used in various types of Affective Computing applications that involve inference, expression or synthesis of emotions. An affective computing system endowed with the proposed translator is able to retrieve affective information stored in a file, modify the file data in real time, as well as store information in EmotionML format for future access or modifications.

Keywords: Affective computing. EmotionML. Translator.

1 Introdução

Atualmente, as interfaces de sistemas computacionais têm sido dotadas de habilidades afetivas, a fim de interagir de maneira mais natural e antropomórfica com o usuário. Na área de informática na educação, por exemplo, os ambientes inteligentes de aprendizagem empregam agentes pedagógicos animados ou robôs que inferem as emoções do usuário e expressam emoções, com vistas a prover um professor ou companheiro mais empático [15, 23]. Esses trabalhos têm sido o foco de pesquisa da área de Computação Afetiva (CA).

Essa área estuda como desenvolver artefatos computacionais (robôs, equipamentos ou software) que possam responder aos estados afetivos do usuário. Uma das motivações para dotar computadores de capacidades

http://dx.doi.org/10.5335/rbca.2013.2814

¹ PIPCA/UNISINOS – Av. Unisinos, 950 Bairro Cristo Rei - CEP 93.022-000 São Leopoldo, Brasil { marcelo.nichele@gmail.com, pjaques@unisinos.br }

afetivas, tais como reconhecer e expressar emoções, é a melhoria na interação homem-máquina. Um computador ciente dos estados afetivos do usuário está mais apto a identificar o melhor momento a fazer uma intervenção e ao modo como fazê-la. Além disso, quando provido de mecanismos para expressar seus estados afetivos, ele pode se comunicar de maneira mais antropomórfica e social [26]. Devido a isso, interfaces sensíveis a emoções têm sido desenvolvidas para os mais diferentes domínios, tais como educação, saúde, jogo, entre outros [14].

As pesquisas em CA usam, ainda, bases de vídeos, fotos ou textos anotados para treinamento e extração de informações. Por exemplo, as aplicações para detecção de emoções por face, geralmente, empregam um classificador que é treinado por uma base de fotos (ou vídeos) anotadas (com informações de emoções ou unidades faciais, do inglês Action Units, [7, 8] expressas). Nesses casos, os pesquisadores precisam de uma representação computacional que lhes permita descrever as emoções inferidas ou expressas pelo computador ou aquelas anotadas pelos pesquisadores nas bases. Essa representação precisa ser universalmente aceita para interoperabilidade das aplicações ou de seus componentes. Por exemplo, um ambiente educacional poderia empregar um detector de emoções por face de uma empresa e um robô de outra.

Com o objetivo de discutir e propor uma linguagem para a representação computacional das emoções, surgiu o World Wide Web Consortium (W3C) Emotion Incubator Group². O W3C é uma comunidade internacional que visa a desenvolver padrões para a web. O grupo publicou o documento que consolida os estudos na criação de uma linguagem de marcação para emoções: Emotion Markup Language (EmotionML 1.0) [9, 29]. Essa linguagem permite descrever vários atributos importantes da emoção, tais como elemento de *appraisal* [19], dimensões (entre os quais, valência, excitação e dominância) etc. Para que essa linguagem possa ser efetivamente usada pelos sistemas computacionais afetivos, eles devem dispor de um tradutor EmotionML.

Essa tradução é muito importante para a área de CA, pois, embora a EmotionML seja uma forma estabelecida para armazenamento e compartilhamento de informações sobre emoções e estados afetivos, sistemas computacionais precisam de um formato inteligível por máquina para manipulação interna em tempo de execução. Esse tradutor seria responsável por converter as informações emocionais de texto para objetos que o programa possa manipular em memória e vice-versa. Por exemplo, um editor de anotação de emoções em vídeos poderia usar o EmotionML para guardar as informações anotadas permanentemente em arquivo como uma espécie de base de dados. No entanto, durante a execução do programa, para manipulação interna, ele precisa guardar os dados em estruturas de dados na linguagem de programação empregada. O tradutor seria responsável por realizar essa tradução de arquivos EmotionML para objetos Java para a manipulação pelos programas em tempo de execução. A tradução para classes e objetos é realizada na linguagem Java, uma vez que essa é uma das linguagens de programação mais utilizadas atualmente. Além disso, como Java é uma linguagem orientada a objetos, a tradução das informações em *tags* XML para objetos Java é intuitiva. Outra vantagem é que Java possui diversas bibliotecas de interface com programas em outras linguagens, tais como Java Native Interface, Java Remote Invocation etc.

Ainda que essa ferramenta seja importante para os trabalhos em CA, não se tem conhecimento de um tradutor EmotionML. Dessa forma, este artigo apresenta um tradutor para a linguagem EmotionML. Este trabalho engloba, também, a geração dinâmica das classes que representam os elementos e atributos contidos no documento XML (eXtensive Markup Language), assim como a instanciação de objetos a partir das classes criadas e dos dados representados em EmotionML. Da mesma forma, a ferramenta realiza o fluxo inverso, ou seja, gera documentos XML a partir dos objetos instanciados. Neste artigo, além de se apresentar a ferramenta proposta, são descritos alguns cenários de uso da ferramenta como exemplos.

Este artigo encontra-se organizado como segue. A seção 2 apresenta uma introdução sobre emoções. A seção 3 descreve a linguagem de marcação EmotionML. A seção 4 explica o trabalho proposto e o detalhamento de todo o processo de tradução. A seção 5 expõe o cenário utilizado como exemplo de execução da aplicação, e, finalmente, a seção 6 apresenta as conclusões do trabalho.

2 Entendendo as emoções

Embora o termo "emoção" seja usado popularmente para muitos fenômenos de ordem afetiva, estes devem ser denominados pelo termo genérico "estado afetivo". Um estado afetivo pode ser visto como um termo mais abrangente, o qual inclui outros estados além das emoções, como, por exemplo, o humor [12, 19, 27].

² http://www.w3.org/2005/Incubator/emotion

A emoção diferencia-se dos outros estados afetivos principalmente por ser mais intensa, rápida e por ser o resultado de uma avaliação de uma situação. De acordo com Scherer [28], emoção é um episódio relativamente breve de resposta sincronizada para a maioria dos ou todos os seres vivos, para a avaliação de um evento interno ou externo, como sendo de maior importância. Raiva, tristeza, alegria, medo, desespero e vergonha são alguns exemplos de emoções. Ortony, Clore e Collins [22] propõem uma definição similar, porém mais precisa, para emoções. Conforme os autores, emoções são reações com valência a eventos, agentes ou objetos, cuja natureza particular é determinada pela maneira como a situação disparadora é construída. De acordo com essa definição, surpresa não é uma emoção, pelo fato de não ter uma valência. Fridja [12] considera que uma emoção é um estado mental intencional, porque é direcionada a um objeto, seu objeto intencional. Por exemplo: estou com raiva de John, mas admiro Nicolas. John e Nicolas são objetos intencionais das emoções sentidas.

Existem várias teorias aceitas sobre emoções, e cada teoria as representa por meio de informações específicas. Atualmente, o modelo de componentes (componential model) tem recebido considerável atenção dos pesquisadores em emoções [1, 17, 24]. Segundo esse modelo, as emoções em humanos são caracterizadas pela presença de quatro componentes principais [28]: (i) componente motivacional-comportamental, que diz respeito às inclinações de um indivíduo para agir de acordo com essas interpretações; (ii) componente subjetivo: responsável pela parte de "sentimento subjetivo", é mais elaborado em seres humanos que estão habituados a rotular as emoções que sentem; (iii) componente somático: envolve a ativação dos sistemas nervosos central e automático e sua manifestação corporal, e o (iv) componente cognitivo: engloba os processos cognitivos que avaliam as situações e disparam as emoções.

O componente cognitivo das emoções tem recebido, igualmente, especial atenção pelos pesquisadores em psicologia cognitiva [19, 27]. As emoções são disparadas por uma avaliação cognitiva de situações e eventos antecedentes, chamada de *appraisal*, sem tradução para o português [27]. O padrão de reação nos diferentes domínios de resposta (fisiologia, sensações e tendência à ação) é determinado pelo resultado dessa avaliação. Os psicólogos cognitivistas estão interessados na avaliação cognitiva e nos eventos que disparam as emoções. Essa abordagem, conhecida como abordagem cognitiva das emoções, tem influenciado vários trabalhos em computação afetiva [1, 15, 17].

3 EmotionML

A EmotionML é uma linguagem de marcação, baseada em XML, que serve para representar emoções e outros estados afetivos relacionados, tais como humor e traços de personalidade. Ela foi concebida para ser utilizada como uma linguagem *plugin*, juntamente com outras linguagens de marcação necessárias pela aplicação. A linguagem, definida pelo W3C Emotion Incubator Group³, tem seu emprego previsto em três tipos principais de aplicações: (*i*) anotação manual de dados [18]; (*ii*) detecção automática de estados afetivos [2, 4, 5, 30]; e (*iii*) geração e expressão de comportamento emocional [3, 15, 23, 30].

Na anotação manual de dados, os pesquisadores usam ferramentas específicas para anotar expressões emocionais observadas por indivíduos em gravações audiovisuais. Essas anotações são utilizadas futuramente para fins de pesquisa e avaliação, como, por exemplo, para confrontar as observações de vários pesquisadores sobre uma determinada emoção a fim de verificar a taxa de acerto humano na inferência de emoções por expressões faciais ou voz. Outro exemplo de aplicação é fornecer dados para treinamento de classificadores para inferência de emoções por expressões faciais. Após treinados, os classificadores estão aptos a reconhecer automaticamente as emoções, precisando, para tanto, de uma linguagem que lhes permita codificar a informação inferida. Por fim, um robô que expressa emoções durante a interação com usuário utiliza-se de uma linguagem para comunicar internamente entre seus componentes (mente e corpo, por exemplo) a emoção que deve ser expressada. Para todos esses tipos de aplicações, a linguagem EmotionML foi concebida.

A sintaxe dos principais elementos da linguagem EmotionML é descrita a seguir.

O elemento <emotionml> é a raiz do documento. Deve conter um ou mais elementos <emotion> e pode conter um elemento <metadata>.

³ http://www.w3.org/2005/Incubator/emotion/XGR-emotion/

O elemento <emotion> representa uma simples anotação relativa a uma emoção e possui os seguintes elementos filhos, que podem ocorrer uma única vez: <category>, <dimensions>, <appraisals>, <action-tendencies>, <itensity> e <meta-data>. Os elementos <link> e <modality> podem ocorrer uma ou mais vezes. Os atributos, opcionais, são: (1) date - data que registra a ocorrência do evento; (2) timeRefURI - indica a URI usada na definição de tempo inicial e final da ocorrência do evento; (3) timeRefAnchor - indica se deve ser medido o tempo inicial ou final do intervalo indicado em timeRefURI e (4) offsetToStart - especifica o *offset* para o início do input do ponto indicado em timeRefURI e timeRefAnchor.

O elemento <category> especifica a categoria do elemento <emotion> por meio do atributo *name*, que indica o nome da categoria e do atributo *set*, o qual indica o conjunto de valores válidos para o atributo *name*. A categoria serve para prover uma identificação rápida do tipo de emoção que ocorreu e, por isso, deve ser um nome único, simples, eleito entre aqueles definidos no vocabulário da EmotionML [9, 29].

A Figura 1 traz um exemplo de uso desses elementos para a representação de emoções de um robô. O trecho em EmotionML ilustra a intensidade das dimensões afetivas de *arousal*, valência e potência para o estado emocional corrente do robô. Nesse exemplo, não é nomeada uma emoção específica para o robô, apenas as dimensões a ela relacionadas.

Figura 1: EmotionML para representar emoções de um robô [29]

Devido às restrições de espaço, os demais elementos existentes serão mostrados pelo diagrama de classes da Figura 3. Uma descrição completa desses elementos pode ser encontrada na especificação da linguagem em [9].

4 O tradutor EmotionML

Este trabalho tem como objetivo geral criar um tradutor para a linguagem EmotionML. O tradutor deve ser capaz de: (i) ler um documento EmotionML; (ii) gerar dinamicamente classes para os elementos EmotionML detectados; (iii) instanciar os dados do documento EmotionML em objetos Java usando as classes geradas no passo anterior; (iii) permitir manipulação e alteração dos dados por qualquer aplicação; (iv) registrar os dados atualizados em um novo documento EmotionML. Para tanto, o tradutor é composto pelos seguintes módulos do sistema: Parser, Gerador de Objetos, Gerador de Classes e Gerador XML.

A linguagem de programação Java foi utilizada em dois propósitos no tradutor: como linguagem objeto do tradutor e como linguagem de programação do próprio tradutor. No primeiro caso, quando lê um documento XML, o tradutor gera dinamicamente classes Java para os elementos e instancia os dados representados no arquivo em objetos Java. A linguagem Java foi escolhida pelas características de portabilidade (independência de plataforma) e orientação a objetos. Essa última permite mais facilmente tratar os elementos EmotionML, criando dinamicamente classes para cada um dos elementos definidos na linguagem. As mesmas vantagens oferecidas pelo Java, juntamente com o fato de esta ser uma das linguagens orientada a objetos mais populares na atualidade, levaram à sua adoção também na programação do tradutor.

4.1 Funcionamento do tradutor

Conforme pode ser observado na Figura 2, o processo de tradução tem início pela entrada do documento EmotionML no módulo do tradutor que é responsável pelo *parser*. Esse módulo efetua as análises léxicas e sintáticas definidas na gramática da linguagem EmotionML e tem como saída a árvore de nodos que representam os elementos descritos no documento, a partir da qual o tradutor pode gerar dinamicamente as classes Java que representam os elementos EmotionML, sendo essa definição centralizada na gramática da linguagem. Além disso, o tradutor instancia essas classes recém-criadas em tempo de execução com objetos EmotionML que representam os elementos do documento. Esse procedimento será visto em detalhes nas seções *4.3 e 4.4*. Nesse ponto, os objetos instanciados estão disponíveis para uma aplicação qualquer usar ou alterar seus dados, podendo ser transformados novamente em documentos XML por meio do módulo Gerador XML. O processo de geração do documento XML será visto em detalhes na seção *4.5*.

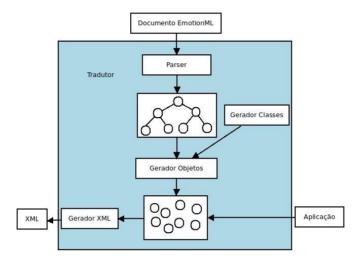


Figura 2: Representação gráfica do tradutor EmotionML

Todos os módulos estão incorporados ao tradutor, ou seja, estão definidos na gramática. Porém, os módulos *Gerador de Classes*, *Gerador de Objetos* e *Gerador XML* foram implementados em código Java pelas classes que foram construídas durante este trabalho para atingir esses propósitos específicos. Essas classes podem ser invocadas separadamente, porém algumas precisam seguir uma ordem de execução para o seu correto funcionamento. Por exemplo, se há a necessidade de gerar as classes, o módulo *Gerador de Classes* deve ser executado antes dos demais. O módulo *Gerador de Objetos* deve ser executado após o *Parser*, pois recebe como entrada a árvore de nodos que representam os elementos EmotionML no documento XML. A fase final, que é realizada pelo *Gerador* XML, recebe como entrada a lista de objetos EmotionML e deve ser executada após o *Gerador de Objetos*.

Apesar dessa independência em relação ao *parser* da linguagem, a assinatura desses métodos está definida na gramática da linguagem EmotionML, a fim de facilitar a manutenção do tradutor.

Nas seções seguintes, são descritos, em maiores detalhes e individualmente, os módulos do tradutor.

4.2 Parser

Os *parsers* são os componentes dos tradutores responsáveis por realizar a análise sintática de uma linguagem, ou seja, por determinar se uma cadeia de símbolos léxicos pode ser gerada a partir da gramática de uma linguagem. No trabalho proposto, mais especificamente, esse software efetua a identificação das *tags* XML com suas propriedades informadas, assim como a validação dos elementos e seus atributos. A saída desse módulo é a árvore de nodos que representam os elementos do documento XML.

O processo inicia com a entrada do nome do arquivo que representa o documento EmotionML. Em seguida, a análise léxica é realizada produzindo os *tokens*, e, na sequência, o método que efetua o *parser* é executado.

O ANTLR foi utilizado para criação do parser. Another Tool for Language Recognition (ANTLR) é um gerador de parser, ou seja, uma ferramenta que automatiza a construção de analisadores sintáticos [25]. Com uma descrição formal de uma linguagem, também chamada gramática, ANTLR gera um programa que reconhece as sentenças e aplica as ações conforme a linguagem. Para isso, deve-se criar uma gramática com as definições da estrutura da linguagem, assim como as ações que devem ser executadas para a tradução da linguagem. Além disso, na gramática EmotionML, estão as definições de todas as classes que representam cada um dos elementos descritos no documento XML, além das propriedades de cada elemento e suas validações. A gramática do EmotionML foi definida no escopo deste trabalho a partir da especificação da linguagem disponível [9].

4.3 Gerador de Classes

Para uma melhor manipulação dos dados a serem processados referentes aos elementos da EmotionML, é necessário criar classes que possam representar cada um dos elementos da linguagem. Dessa forma, ao processar um documento EmotionML, o tradutor pode instanciar objetos Java a partir das classes geradas que representem esses elementos com os valores de seus atributos definidos, possibilitando, ainda, uma melhor manipulação desses objetos pelas aplicações. A definição dessas classes está centralizada na gramática da linguagem, assim como as ações que efetuam o preenchimento dos seus atributos. Dessa forma, se a gramática é modificada devido a alguma extensão da linguagem EmotionML, essa modificação refletir-se-á na geração dinâmica das classes. Além disso, essas classes são compiladas e carregadas na Máquina Virtual Java (*Java Virtual Machine* – MVJ) em tempo de execução.

Um exemplo do diagrama de classes em Unified Modelling Language (UML) [11], definido a partir do estado atual da especificação do EmotionML, pode ser visualizado na Figura 3. Esse esquema foi modelado de acordo com a especificação da EmotionML, em que uma classe *Emotion* é composta pelas classes *Category*, *Modality*, *Link*, *Dimensions*, *Appraisals ActionTendencies* e *Intensity*.

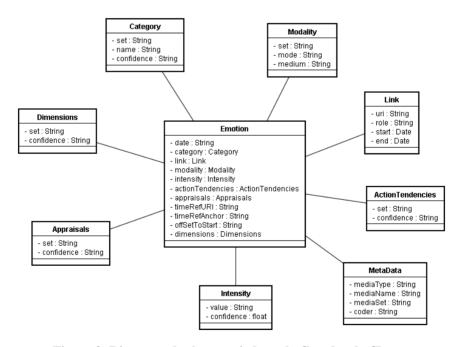


Figura 3: Diagrama de classes criadas pelo Gerador de Classes

Foi desenvolvido um *template* para a geração do código que representa as classes Java geradas pelo tradutor. Nela, estão especificadas a estrutura do código e as declarações dos elementos que compõem uma classe, tais como a *package*, o nome da classe, os atributos e os métodos e seus parâmetros. Um trecho desse *template* pode ser visualizado na Figura 4.

```
1group Java:
 2
 3
 5package(packageName, className, fields, methods, staticFields) ::= <<</pre>
 6package <packageName>;
 8<class(className=className, fields=fields, methods=methods, staticFields=staticFields)>
 9
10>>
11
12 class(className, fields, methods, staticFields) ::= <<
13public class <className> {
   <staticFields:staticField(); separator="\n">
14
16
   <fields:field(); separator="\n">
17
   <methods:method(); separator="\n">
191
20>>
21
22 staticField() ::= <<
23final static <it.type> <it.name> = "<it.staticValue>";
25
26field() ::= "private <it.type> <it.name>;"
28method() ::= <<
```

Figura 4; Template genérico das classes Java

A geração das classes é realizada por um método definido na gramática da linguagem, que foi implementado em Java e recebe como argumentos a lista de classes a serem geradas e o caminho descrevendo onde elas serão geradas. O método que compila as classes recém-criadas recebe como argumentos a lista de classes, o caminho que descreve a localização dessas e o nome do pacote Java. Esses dois métodos são invocados pelo método generateEmotionClasses() que foi definido na gramática da linguagem e gerado automaticamente pelo ANTLR.

4.4 Gerador de Objetos

O principal módulo do sistema, que é o responsável pela geração dos objetos, realiza o mapeamento do documento já processado em objetos instanciados. Para tanto, é utilizada a Application Programming Interface (API) Java Reflection [10], a qual permite que um programa Java acesse a sua própria estrutura interna e suas propriedades. Assim, é possível, por exemplo, obter o nome de todos os membros de uma classe, como atributos e métodos, bem como executar um método usando a introspecção.

Para a criação dos objetos, o módulo deverá identificar a classe que representa determinado nodo, gerada pelo Gerador de Classes, carregá-la e instanciá-la conferindo valores aos seus atributos. Todo o processamento de criação dos arquivos que representam as classes é efetuado pelo *StringTemplate*. Para obter-se os objetos, deve-se invocar o método *generateEmotionObjects()*, que tem a responsabilidade de carregar as classes recémcriadas na JVM em tempo de execução. Esse carregamento das classes para a JVM é feito sob demanda, ou seja, somente são carregadas as classes que estão no documento XML.

4.5 Gerador XML

Uma importante fase para um tradutor é a fase final, quando se emite a saída do processamento. O componente do tradutor responsável pela geração de textos estruturados é chamado de *emitter*. No ANTLR, a ferramenta *StringTemplate* é geralmente utilizada como *emitter*.

O Gerador XML é responsável por gerar novamente o documento EmotionML a partir dos dados alterados pela aplicação. Ele utiliza a ferramenta *StringTemplate* para criar o documento XML e a API Java Reflection [10] para obter a estrutura interna de cada objeto, assim como os seus atributos. Para tanto, o módulo

deve ter como entrada os objetos já instanciados e, para cada um, criar um elemento no documento com suas propriedades.

A utilização do *StringTemplate* como *emitter* possibilita uma maior gerência sobre a camada de saída do tradutor, tornando sua manutenção mais flexível, separando o processamento de saída da lógica do *parser*. Assim, quando ocorrer alguma mudança, ou na medida em que vão surgindo novas necessidades de saída do tradutor, basta alterar os *templates* apropriados.

5 Exemplo de casos de uso e cenário

Essa seção tem como objetivo descrever alguns casos de uso da EmotionML, a fim de exemplificar como o tradutor poderia ser empregado em uma aplicação de CA. Também, será apresentado, na seção 5.1, o funcionamento mais detalhado do tradutor em um cenário de anotação manual de emoções.

Como explicado na seção 3, para criação do EmotionML, foram identificados 39 casos de uso para a linguagem, os quais foram divididos em 3 principais grupos: (1) anotação manual de dados, (2) reconhecimento automático de emoções e (3) geração de emoções.

Os casos de uso referentes ao grupo "anotação manual de dados" referem-se à anotação manual de vídeos, gravações faciais, gravações de falas etc., ou seja, são aquelas anotações geradas manualmente por pesquisadores, que usam essa informação para estudar e observar a expressão de emoções no homem, a fim de imitar em robôs e personagens animados. Um exemplo dos casos de uso referentes a esse grupo pode ser visto na Figura 5.

```
<emotionml xmlns="http://www.w3.org/2008/11/emotionml">
    <metadata>
        <media-type>video</media-type>
        <media-name>ed1_4</media-name
        <media-set>humaine database</media-set>
        <coder-set>JM-AB-UH</coder-set>
    </metadata>
        <category set="humaineDatabaseLabels" name="Amusement"/>
        <intensity value="0.52"/>
    </emotion>
    <emotion>
        <category set="humaineDatabaseLabels" name="Irritation"/>
        <intensity value="0.63"/>
    </emotion>
    <emotion>
       <category set="humaineDatabaseLabels" name="Relaxed"/>
        <intensity value="0.02"/>
    </emotion>
    <emotion>
        <category set="humaineDatabaseLabels" name="Frustration"/>
        <intensity value="0.87"/>
    </emotion>
    <emotion>
        <category set="humaineDatabaseLabels" name="Calm"/>
        <intensity value="0.21"/>
    </emotion>
    <emotion>
        <category set="humaineDatabaseLabels" name="Friendliness"/>
        <intensity value="0.28"/>
```

Figura 5: Anotação manual de vídeo [9]

A Figura 5 ilustra um trecho de um arquivo EmotionML gravado por um Editor manual de anotações. Nesse arquivo, são descritas várias emoções observadas por um pesquisador no vídeo de nome "ed1_4" proveniente da base de dados "humaine". Cada *tag* <emotion> descreve uma emoção identificada no vídeo e sua intensidade. Por exemplo, a primeira *tag* descreve que foi observada a emoção "amusement" (divertimento em português) com intensidade de 0,52 (entre -1 e 1). Nesse caso, o Editor mantém um registro das informações afetivas anotadas em arquivos EmotionML entre as sessões de anotações. Isso permitiria que o pesquisador começasse as suas anotações em uma determinada sessão, fechasse o Editor e recomeçasse em uma outra data e/ou horário. Nesse caso, este utilizaria o tradutor para registrar as emoções no arquivo EmotionML na primeira sessão, assim como para ler em memória as informações do arquivo EmotionML de registro na sessão seguinte. O tradutor,

também, poderia ser empregado por outras aplicações que utilizassem o arquivo de anotações, tais como um programa de treinamento de um classificador (ex. rede neural) durante a fase de aprendizagem.

Os casos de uso referentes ao grupo "reconhecimento automático de emoções relacionadas ao comportamento dos estados afetivos do usuário" referem-se àquelas anotações relacionadas ao reconhecimento automático de emoções por meio de sensores. Portanto, são os casos de uso que descrevem o reconhecimento automático das emoções do usuário. Um sistema computacional afetivo geralmente reconhece as emoções por sensores que identificam os estados afetivos do usuário. Por exemplo, o sistema poderia reconhecer as emoções do usuário por suas expressões faciais registradas por uma câmera de vídeo [21] e armazenar essas informações em um arquivo XML. Um exemplo de arquivo EmotionML referente a esse grupo pode ser visto na Figura 6.

```
<emotionml xmlns="http://www.w3.org/2008/11/emotionml">
<emotion date="2001-11-23T14:36Z">
  <!--the first modality detects excitement.
      It is a camera observing the face. An URI to the database
      (a dedicated port at the server) is provided to access the
      wideo stream -->
  <category set="everyday" name="excited"/>
  <modality medium="visual" mode="face"/>
  k uri="http://192.168.1.101:456" start="26s" end="98s"/>
<emotion date="2001-11-23T14:36Z">
  <!--the second modality detects anger. It is an IR camera
observing the face. An URI to the database (a dedicated port
      at the server) is provided to access the video stream. -->
  <category set="everyday" name="angry"/>
  <modality medium="infrared" mode="face"/>
  <link uri="http://192.168.1.101:457" start="23s" end="108s"/>
</emotion>
<emotion date="2001-11-23T14:36Z">
  <!--the third modality detects excitement again. It is a
      wearable device monitoring physiological changes in the
      body. An URI to the database (a dedicated port at the
 server) is provided to access the data stream.-->
<category set="everyday" name="excited"/>
  <modality medium="physiological" mode="body"/>
  k uri="http://192.168.1.101:458 start="19s" end="101s"/>
<emotion date="2001-11-23T14:42Z">
 <category set="everyday" name="angry"/>
<modality medium="physiological" mode="body"/>
  <link uri="http://192.168.1.101:458 start="2s" end="6s"/>
</emotion>
</emotionml>
```

Figura 6: EmotionML para reconhecimento automático de emoções [9]

Na Figura 6, cada *tag* <emotion> guarda uma emoção detectada por um programa multimodal que realiza detecção de emoções por imagens de vídeo e por um dispositivo *wearable* que monitora mudanças fisiológicas no corpo. Toda vez que detecta uma emoção, o programa registra a emoção reconhecida e o instante de tempo em que ela aconteceu. Assim, como no caso de uso citado na Figura 5, o tradutor poderia ser empregado para leitura e registro permanente dos dados em um arquivo de EmotionML.

Os casos de uso referentes ao grupo *Geração de emoções relacionadas ao comportamento do sistema* dizem respeito às anotações decorrentes da geração de respostas, via sistema, decorrentes de processamento relacionado às emoções. Como exemplo, citam-se as emoções que um robô deve expressar em uma determinada situação. Portanto, são aquelas em que a geração das emoções parte do sistema. Um exemplo de caso de uso referente a esse grupo pode ser visto na Figura 1, explicada na seção 3.

5.1 Exemplo de funcionamento do tradutor

O cenário descrito nessa seção pertence ao grupo *anotação manual de dados* e refere-se a uma anotação manual de vídeo. O documento utilizado na execução do tradutor no cenário proposto pode ser visto na Figura 7.

Trata-se da descrição de emoções contidas em trechos de um determinado vídeo. No elemento *metadata*, encontram-se especificadas informações sobre a mídia. A *tag media-type* identifica que a mídia é um vídeo. O nome do arquivo que contém a mídia é ext-o3 (*tag media-name*), e o formato de codificação do vídeo é 4 (*tag coder*). Em cada elemento *emotion*, há a descrição das emoções que podem ser visualizadas em um determinado trecho do vídeo. Por exemplo, na primeira descrição, observa-se que, no trecho que começa após 3,24 segundos de vídeo e finaliza após 15,4 segundos, foi observada a emoção irritação com intensidade 0,46. Pode-se notar que o elemento "*metadata*" está num formato diferente dos demais elementos. Esse formato é uma das questões em aberto na especificação da linguagem, pois viola uma premissa que diz que um dos objetivos do projeto *EmotionML* era que não seria permitido texto livre. No entanto, este trabalho reconhece esse formato com os valores dos atributos em texto livre e o formato padrão dos demais elementos. Contudo, a geração dos objetos e do documento *XML* assume que o elemento <*metadata*> é uma propriedade de um elemento <*emotion*>, e isso vale também para os objetos, onde o elemento <*metadata*> é representado pela classe *MetaData*.

```
<emotionml xmlns="http://www.w3.org/2008/11/emotionml">
    <metadata>
       <media-type>video</media-type>
       <media-name>ext-03</media-name>
        <media-set>EmoTV</media-set>
       <coder>4</coder>
    </metadata>
    <emotion>
       <category set="emoTV-labels" name="irritation"/>
       <intensity value="0.46"/>
       k uri="ext03.avi" start="3.24s" end="15.4s">
    </emotion>
    <emotion>
       <category set="emoTV-labels" name="despair"/>
       <intensity value="0.48"/>
       k uri="ext03.avi" start="3.24s" end="15.4s"/>
    </emotion>
</emotionml>
```

Figura 7: EmotionML para anotação manual de emoções [9]

A execução do tradutor pode-se realizar de duas formas: por linha de comando ou por chamadas aos métodos, específicos de cada fase, em um programa. Um exemplo da execução do tradutor em um programa pode ser visto na Figura 8.

```
CharStream input = new ANTLRFileStream(args[0]);
 // Efetua a primeira fase da tradução a análise léxica
EmotionMLLexer lex = new EmotionMLLexer(input);
              os Tokens processado
 CommonTokenStream tokens = new CommonTokenStream(lex):
 // Instancia o parser da EmotionML
 EmotionMLParser parser = new EmotionMLParser(tokens);
 // Executa o gerador de classes, esta fase é opciona
 EmotionMLParser.generateEmotionClasses(path, packageName);
  // Efetua o parser da linguagem sobre o documento EmotionMI
 EmotionFileInfo emotionML = parser.parserGrammar().emotionFile;
 // Executa o gerador de objetos da EmotionML
 List<Object> emotionObjects = parser.generateEmotionObjects(path).emotionObjects;
  / Gera o EmotionML.xml referente aos objetos EmotionML especificado:
 EmotionMLParser.generateEmotionXMLFile(emotionObjects.xmlFullFileName):
 for (EmotionElement emotion : emotionML.getEmotions()) {
      System.out.println("emotion");
      for (String element : emotion.getElements().keySet()) {
          System.out.println("element " + element);
          for (String attribute : emotion.getElements().get(element).keySet()) {
    System.out.println( " attribute " + attribute + " => " + emotion.get
                                                                           + emotion.getElements().get(element).get(attribute) );
 for (Object o : emotionObjects) {
      System.out.println(o.toString());
      Field[] fields = o.getClass().getDeclaredFields();
     for (Field field: fields) {
    final Object v = Inspector.getValue(o, field.getName(), true);
         System.out.println(" fieldName " + field.getName() + " value " + v);
}
```

Figura 8: Execução do tradutor no cenário proposto

Como observado na Figura 8, o primeiro comando "for" efetua uma varredura nos elementos processados pelo parser, mostrando os seus atributos e seus respectivos valores. A saída desse comando pode ser vista na Figura 9.

```
emotion
element category
 attribute set => "emoTV-labels"
 attribute name => "irritation"
element intensity
 attribute value => "0.46"
element link
 attribute start => "3.24s"
 attribute uri => "ext03.avi"
 attribute end => "15.4s"
element metaData
 attribute media-set => EmoTV
 attribute coder => 4
 attribute media-type => video
 attribute media-name => ext-03
emotion
element category
 attribute set => "emoTV-labels"
 attribute name => "despair"
element intensity
attribute value => "0.48"
element link
 attribute start => "3.24s"
 attribute uri => "ext03.avi"
 attribute end => "15.4s"
element metaData
 attribute media-set => EmoTV
 attribute coder => 4
 attribute media-type => video
 attribute media-name => ext-03
```

Figura 9: Trecho da saída da execução do exemplo da Figura 8

O segundo comando "for", também, percorre todos os elementos do documento, mas a lista é constituída de objetos *EmotionML*. A saída dessa lista pode ser vista na Figura 10.

```
emotion.element.Emotion@1d15445
fieldName intensity value emotion.element.Intensity@1f3aa07
fieldName category value emotion.element.Category@1fc2fb
fieldName modality value null
fieldName appraisals value null
fieldName link value emotion.element.Link@139eeda
fieldName metaData value emotion.element.MetaData@704baa
fieldName actionTendencies value null
emotion.element.Emotion@77a7f9
fieldName intensity value emotion.element.Intensity@1b4fad5
fieldName category value emotion.element.Category@bfea1d
fieldName modality value null
fieldName appraisals value null
fieldName link value emotion.element.Link@1a33d48
fieldName metaData value emotion.element.MetaData@c68c3
fieldName actionTendencies value null
```

Figura 10: Saída dos objetos EmotionML

Após se processar os dados e, se necessário, fazer alguma modificação de valor de algum atributo, o sistema pode, de igual forma, gerar novamente o arquivo XML a partir dos objetos e das mudanças realizadas nos seus atributos.

6 Conclusões

Neste trabalho, foi apresentado todo o processo de desenvolvimento do tradutor para a linguagem EmotionML. O tradutor foi desenvolvido na linguagem de programação Java, e foram utilizadas as ferramentas *ANTLR* como *parser generator* e *StringTemplate* como *template engine*.

Atualmente, os autores não estão cientes de nenhum tradutor criado para a linguagem EmotionML para propósito geral. Courgeon et al. [6] criaram um *parser* para EmotionML específico para ser utilizado no agente conversacional animado MARC com vistas a registrar as emoções expressas pelo agente durante a interação com o usuário. No entanto, esse *parser* tem a restrição de apenas identificar os valores da *tag* <appraisal> que foi necessária naquele trabalho. O trabalho mais semelhante ao proposto é o de Nunes [20], no qual apresenta o PersonalityML, uma linguagem baseada em XML para descrever traços de personalidade segundo o modelo Big Five de personalidade [13]. Nunes [20] propõe, igualmente, um tradutor de PersonalityML para objetos; no entanto, o trabalho não descreve as emoções.

O tradutor desenvolvido será utilizado nos projetos de CA do grupo para armazenar informações afetivas em aplicações de inferência de emoções por face [21], assim como em aplicações educacionais aptas a inferir as emoções do usuário [17]. Ele se encontra disponível para download em http://professor.unisinos.br/pjaques/material/emotionml-translate.tar.gz.

Como trabalho futuro, os autores pretendem reunir esforços para desenvolver uma extensão das linguagens EmotionML e PersonalityML que represente as relações entre emoções e personalidades, assim como um tradutor para essa linguagem. Emoções e personalidade estão fortemente interligadas. A personalidade do usuário determina seu *apprasail* e como ele irá reagir às situações emocionais [16]. Dessa forma, uma extensão unificada dessas linguagens permitirá representar essas relações entre traços de personalidade e emoções, o que ainda não é previsto pelas duas linguagens.

Referências

[1] ADAM, C.; HERZIG, A.; LONGIN, D. 2009. A logical formalization of the OCC theory of emotions. *Synthese*, v. 168, n. 2, p. 201-248, may. 2009.

- [2] ARROYO, I. et al. Emotion Sensors Go To School. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 2009, Brighton, UK. *Proceedings*... Amsterdã, Holland: IOS Press, 2009, p. 17-24.
- [3] ARROYO, I. et al. The impact of animated pedagogical agents on girls' and boys' emotions, attitudes, behaviors and learning. *Advanced Learning Technologies (ICALT)*, 2011 11th IEEE International Conference on, p. 506-510, 2011.
- [4] BAKER, R. et al. Better to be frustrated than bored: the incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, v. 68, n. 4, p. 223-241, 2010.
- [5] DU BOULAY, B. et al. Towards Systems That Care: A Conceptual Framework based on Motivation, Metacognition and Affect. *International Journal of Artificial Intelligence in Education*, v. 20, n. 3, p. 197-229, 2010.
- [6] COURGEON, M.; CLAVEL, C.; MARTIN, J.-C. Appraising emotional events during a real-time interactive game. In: INTERNATIONAL WORKSHOP ON AFFECTIVE-AWARE VIRTUAL AGENTS AND SOCIAL ROBOTS, 2009, New York, USA. Proceedings... Cambridge: ACM, 2009. p. 1-5.
- [7] EKMAN, P. Facial expressions. In: DALGLEISH, T.; POWER, T. (Eds.). *The Handbook of Cognition and Emotion*. New York: John Wiley & Sons, 1999. p. 301-320.
- [8] EKMAN, P.; FRIESEN, W. *Investigator's guide to the facial action coding system.* Palo Alto, CA: Consulting Psychologists Press, 1978.
- [9] W3C Incubator Group. Elements of an EmotionML 1.0. 2013. Disponível em http://www.w3.org/2005/Incubator/emotion/XGR-emotionml-20081120/. Acesso em 01 jul. 2013.
- [10] FORMAN, I. R.; FORMAN, N. Java Reflection in Action. New York: Manning Publications , 2004.
- [11] FOWLER, M. UML Essencial: um breve guia para linguagem padrão. Porto Alegre: Bookman, 2005.
- [12] FRIJDA, N. H. Varieties of affect: Emotions and episodes, moods, and sentiments. In: EKMAN, P.; DAVIDSON, R. J. (Eds.). *The nature of emotions*: fundamental questions. Oxford, USA: Oxford University Press, 1994, p. 197-202.
- [13] GOLDBERG, L. R. An alternative "description of personality": The Big-Five factor structure. *Journal of Personality and Social Psychology*, v. 59, p. 203-234, 1990.
- [14] IEEE Transactions on Affective Computing. 2013. Diposnível em http://www.computer.org/portal/web/tac/home>. Acesso em: 30 jul. 2013.
- [15] JAQUES, P. A.; LEHMANN, M.; PESTY, S. Evaluating the affective tactics of an emotional pedagogical agent. In: ACM SYMPOSIUM ON APPLIED COMPUTING-HUMAN COMPUTER INTERACTION TRACK, 2009, Hawaii, USA. *Proceedings*... New York: ACM. 2009. p. 104-109.
- [16] JAQUES, P. A.; NUNES, M. A. S. N. Ambientes Inteligentes de Aprendizagem que inferem, expressam e possuem emoções e personalidade. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA NA EDUCAÇÃO (JAIE), 2012, Natal, Brasil. *Anais...* Porto Alegre: SBC, 2012, p. 31-82.
- [17] JAQUES, P. A. et al. Evaluating a Cognitive-Based Affective Student Model. In: INTERNATIONAL CONFERENCE ON AFFECTIVE COMPUTING AND INTELLIGENT INTERACTION (ACII), 2011, Menphis, USA. *Proceedings*... Berlin: Springer, 2011, p. 599-608.

- [18] MARTIN, J.-C. et al. Manual annotation and automatic image processing of multimodal emotional behaviors: validating the annotation of TV interviews. *Personal Ubiquitous Comput*, v. 13, n. 1, p. 69-76, 2009.
- [19] MOORS, A. et al. Appraisal theories of emotion: state of the art and future development. *Emotion Review*, v. 5, n. 2, p. 119-124, mar. 2013.
- [20] NUNES, M. A. S. N.; BEZERRA, J. S.; OLIVEIRA, A. A. PersonalityML: a markup language to standardize the user personality in Recommender Systems. *Gestão, Inovação e Tecnologias*, São Cristóvão, Brasil, v. 2, p. 255-273, 2012.
- [21] OLIVEIRA, E.; JAQUES, P. A. Inferindo as emoções do usuário pela face através de um sistema psicológico de codificação facial. In: BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, VIII, 2008, Porto Alegre, Brasil. *Proceedings*... Porto Alegre: SBC, 2008, p. 156-165.
- [22] ORTONY, A.; CLORE, G. L.; COLLINS, A. *The cognitive structure of emotions*. New York, USA: Cambridge University Press, 1990.
- [23] PAIVA, A. Empathy in social agents. *International Journal of Virtual Reality*, v. 10, n. 1, p. 65-68, 2011.
- [24] PANG, B.; LEE, L. Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval*, v. 2, n. 2, p. 1-135, 2008.
- [25] PARR, T. *The definitive ANTLR reference*: building domain-specific languages. Pragmatic Bookshelf, 2007.
- [26] PICARD, R.W. Affective computing. Cambridge, USA: University Press Group Limited, 2000.
- [27] SCHERER, K. R. Appraisal theory. In: DALGLEISH, T.; POWER, M. (Eds.). *Handbook of cognition and emotion*. New York, USA: John Wiley & Sons Ltd., 1999, p. 637-663.
- [28] SCHERER, K. R. Psychological models of emotion. In: BOROD, J. (Ed.). *The neuropsychology of emotion*. Oxford, USA: Oxford University Press, 2000, p. 137-162.
- [29] SCHRÖEDER, M. et al. Emotion Markup Language (EmotionML) 1.0. *W3C Working Draft*, v. 29, p. 3-22, 2010.
- [30] WOOLF, B. P. et al. Affect-aware tutors: recognising and responding to student affect. *International Journal of Learning Technology*, v. 4, n. 3/4, p. 129, oct. 2009.