AWMo: Accessible Web Modeler

Filipe D. N. Grillo
Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo
Av. Trabalhador São-carlense 400 - Centro
São Carlos/SP, Brasil
grillo@icmc.usp.br

Renata P. M. Fortes
Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo
Av. Trabalhador São-carlense 400 - Centro
São Carlos/SP, Brasil
renata@icmc.usp.br

RESUMO

Com o aumento do uso das atividades de modelagem em processos de desenvolvimento de software, a participação de pessoas com deficiência visual em tais processos fica comprometida. Os modelos são em sua maioria visuais e, portanto, seu processo de construção requer o posicionamento de elementos no espaço do documento por meio de um dispositivo de apontar com o mouse e sua leitura requer o uso da visão. Neste contexto foi desenvolvida ferramenta Web que possibilita a edição de modelos de software por meio de duas visões equivalentes: uma visão gráfica, tradicional e outra textual, totalmente compatível com leitores de tela. Um estudo de caso foi conduzido com o objetivo de avaliar a efetividade da visão textual para possibilitar que deficientes visuais construam e alterem modelos de software de maneira independente e os resultados indicam que a abordagem é funcional e possui potencial para ser utilizada profissionalmente por engenheiros de software, quer sejam ou não deficientes visuais.

1. INTRODUÇÃO

Atualmente no Brasil, o número de pessoas que possuem algum tipo de deficiência é de 45,62 milhões. De acordo com o Censo 2010, este número representa cerca de 23,92% de toda a população país [4].

Com o aumento de serviços de utilidade pública sendo oferecidos pelo governo por meio da Internet e outros meios eletrônicos, tendência chamada de *e-gov* (sigla para governo eletrônico), aumenta-se a preocupação com relação ao acesso desses cidadãos. Por esse motivo, existem algumas iniciativas que buscam melhorar a acessibilidade de tais serviços tais como a portaria e-MAG (Modelo de Acessibilidade em Governo Eletrônico) [1] entre outras leis e decretos que abordam desde meios eletrônicos como o e-MAG quanto o acesso à ambientes físicos, como a obrigatoriedade da construção de rampas para pessoas que utilizem cadeiras de rodas.

Mais especificamente no âmbito da computação, os deficientes visuais têm historicamente uma participação mais ativa, pois os códigos de programas de computadores são essencialmente texto e, portanto, mais facilmente acessíveis por meio de tecnologias assistivas, tais como leitores de tela e mostradores de braille, por exemplo.

No entanto, alguns tipos de dados ainda permanecem intrinsecamente dependentes de sentidos específicos, como a visão. Os modelos de software por exemplo, são geralmente representados na forma de diagramas visuais, usualmente compostos por formas geométricas e conectores ligando essas formas geométricas. Na maioria das vezes também existe conteúdo escrito nestes diagramas, mas o significado das notações visuais é tão grande que por meio da leitura do texto apenas, o entendimento do conteúdo fica bastante prejudicado. Além disso, com o crescimento da disciplina de engenharia de software e linguagens visuais como a UML (*Unified Modeling Language*) e consequente aumento de atividades de modelagem que utilizam essas linguagens visuais [5], o acesso de deficientes visuais fica prejudicado, sendo necessário o auxílio de outra pessoas tanto para a leitura quando para a edição desses modelos.

Neste contexto, foi desenvolvida a AWMo (*Accessible Web Modeler*), uma ferramenta Web desenhada para permitir que os usuários modelem diagramas de classe da UML em duas visões distintas: a *visão gráfica* onde desenvolvedores de software com visão podem desenhar os diagramas de maneira tradicional, arrastando os elementos e conectando-os visualmente. E a *visão textual*, onde desenvolvedores de software com ou sem deficiência visual podem acessar e desenhar os mesmos diagramas utilizando uma gramática textual desenvolvida especialmente para a AWMo.

2. TRABALHOS RELACIONADOS

Foram encontrados na literatura, outros trabalhos que se dedicam a levar ferramentas para desenvolvimento orientado a modelos para a Web tais como o GEMSjax [2], que é uma implementação Web da ferramenta GEMS (*Generic Eclipse Modeling System*). O GEMS é um projeto que surgiu para unir as experiências sobre ferramentas de metamodelagem visual da comunidade GME, na universidade de Valderbilt e as comunidades Eclipse tais como EMF e GMF. A GEMSjax utiliza o *Google Web Toolkit* para criar uma interface Web capaz de ser utilizada para modelagem e metamodelagem.

O ambiente SLIM é outro exemplo deste tipo de ferramenta. Ele é um ambiente para modelagem colaborativa síncrona e

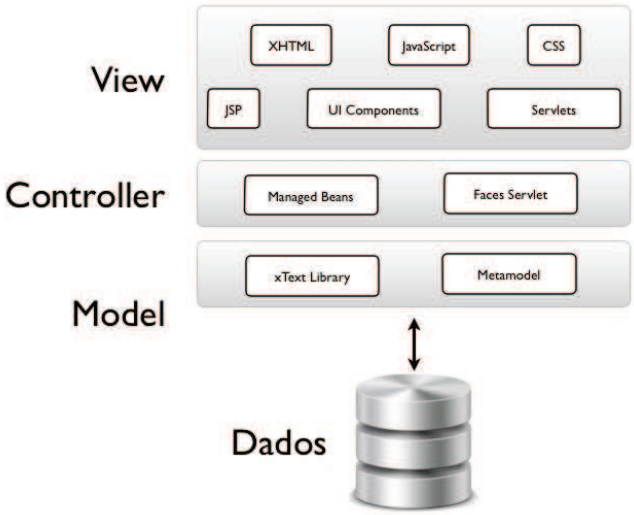

Figura 1: Visão de alto nível da arquitetura utilizada na AWMo com o conceito de MVC (Model-View-Controller) e as tecnologias empregadas em cada uma das camadas.

seus diagramas são desenvolvidos com SVG (*Scalable Vector Graphics*) e técnicas da Web 2.0 como Comet para comunicação com o servidor com pouca latência. O principal objetivo do SLIM é possibilitar a colaboração entre pessoas distantes geograficamente [6].

3. ARQUITETURA

A AWMo foi desenvolvida utilizando a linguagem Java e o *framework* de aplicação JSF (JavaServer Faces) com uma arquitetura MVC (*Model-view-controller*). A Figura 1 ilustra a organização dos componentes entre as camadas de visualização, controle e os modelos.

Na camada de modelos além dos Beans que são utilizados para fazer a comunicação com o servidor por meio de mapeamento utilizando a biblioteca Hibernate existe uma biblioteca contendo toda a infra-estrutura da gramática textual da AWMo gerada pelo Xtext.

O Xtext provê um conjunto de ferramentas que permite a definição de gramáticas com a notação EBNF (Extended Backus–Naur Form) e a geração de recursos para editores textuais, tais como *parser* da linguagem, validadores e geradores de código. O Xtext pode também gerar um metamodelo Ecore e classes EMF (Eclipse Modeling Framework) a partir da gramática da linguagem. Isto torna possível a manipulação programática dos modelos e metamodelos.

A Figura 2 mostra o funcionamento da AWMo em todas as suas etapas de uso. No caso de um usuário abrindo um modelo com a visão gráfica, as informações são buscadas no banco de dados e são criados objetos Java pelo Hibernate e seus mapeamentos, posteriormente os objetos Java são convertidos para JSON com o uso da biblioteca GSON e por sua vez o JSON consumido por um código JavaScript na página que monta o diagrama na tela utilizando a biblioteca jsUML2.

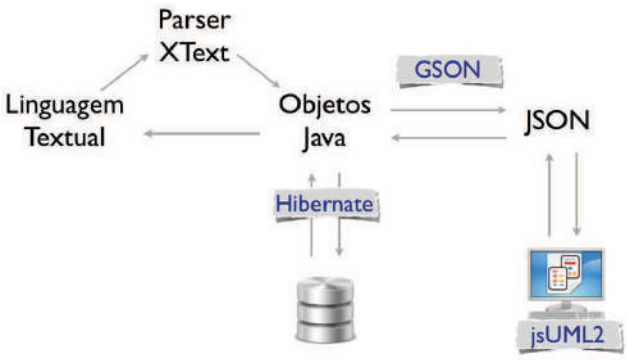

Figura 2: Modelo de funcionamento da AWMo, mostrando os passos, ferramentas e bibliotecas utilizadas para transformação dos modelos para ambas as visões disponíveis.

Ao salvar um modelo a partir da visão gráfica um código JavaScript percorre o modelo como ele está representado pela jsUML2 gerando uma representação em JSON quer é enviada para o servidor. Com o uso da biblioteca GSON a representação em JSON é convertida em objetos Java e por fim, os objetos Java são convertidos em dados relacionais para serem armazenados no banco de dados MySQL.

No caso de um usuário abrindo um diagrama com a visão textual as informações são buscadas no banco de dados e são criados objetos Java pelo Hibernate e seus mapeamentos, posteriormente os objetos Java são convertidos para a notação textual definida pela gramática desenvolvida com o Xtext e então esta notação textual é exibida para o usuário em um editor de texto na página. No sentido oposto, ao salvar um diagrama utilizando a visão textual a representação textual existente na tela é enviada para o servidor, o *parser* gerado pelo Xtext valida se a sintaxe e semântica estão corretas e disponibiliza o acesso por meio de classes EMF. Essas classes EMF são percorridas gerando objetos java para serem armazenados no banco de dados pelo Hibernate.

A estrutura de dados da AWMo combina a informação textual que é utilizada por ambas as visões da ferramenta tais como nome das classes, nome de atributos, métodos e seus tipos com as informações espaciais que são utilizadas apenas pela visão textual da ferramenta que são as coordenadas X e Y de cada classe dentro do diagrama.

4. PRINCIPAIS FUNCIONALIDADES

A principal funcionalidade da AWMo é permitir que tanto deficientes visuais e pessoas com visão possa construir modelos de software de maneira colaborativa e incremental, ou seja, é possível que um modelo seja criado inicialmente por um usuário da visão textual e posteriormente seja editado pela visão gráfica e vice-versa.

O conceito chave de prover duas visões diferentes para a edição do mesmo modelo é que ambas as visões permitem que os usuários realizem as mesmas tarefas e além disso, alcancem os mesmos resultados. Dessa forma, se um usuário deficiente visual criar um novo diagrama de classes na AWMo, editá-

lo e salvá-lo, um usuário com visão poderá abrir o mesmo diagrama na visão gráfica e continuar o trabalho ou realizar alterações nele que, por sua vez, serão novamente acessíveis na visão textual e vice-versa [3].

Isso é possível pois a AWMo utiliza uma estrutura de dados única para os modelos que permitem que eles sejam editados em ambas as visões existentes na ferramenta e posteriormente são convertidos para esse modelo único para ser salvo no banco de dados.

A Figura 3 ilustra a tela do editor textual da AWMo. Nota-se no menu que é possível alternar entre o modo visual e o modo gráfico de edição para o mesmo diagrama que encontra-se aberto.

A Listagem 1 mostra um digrama de classes simples que ilusta a relação de agregação entre as classes Carro e Motor.

```
Classes

classe Motor {
  atributo private ligado : boolean
  atributo private combustivel : string
  atributo public potencia : int

  metodo public ligar : boolean
  metodo public desligar : boolean
  metodo public setCombustivel(↩
      combustivel:string) : boolean
}

classe Carro {
  atributo public portas : int
  atributo private cor : string

  metodo public darPartida : boolean
  metodo public desligar : boolean
}

Relações

relacao composicao Carro 1 Motor 1
```

Listagem 1: Exemplo de um diagrama criado com a linguagem textual da AWMo.

Por meio do uso da visão gráfica o mesmo diagrama seria exibido conforme mostrado na Figura 4.

5. RESULTADOS OBTIDOS

Com o objetivo de avaliar a efetividade da abordagem utilizada pela AWMo para permitir o acesso a modelos de software para deficientes visuais um estudo de caso intitulado "Modelagem de software acessível na Web" foi conduzido pelos autores. Este estudo de caso contou com a participação de duas pessoas e foi composto de uma entrevista inicial para coletar informações pessoais sobre o participante e suas características, observação de utilização da AWMo para a realização de um conjunto pré-definido de objetivos e finalmente uma segunda entrevista para coletar informações sobre a experiência de uso da ferramenta.

O primeiro participante era do sexo masculino, com 41 anos de idade e possui apenas 10% de visão em ambos os olhos

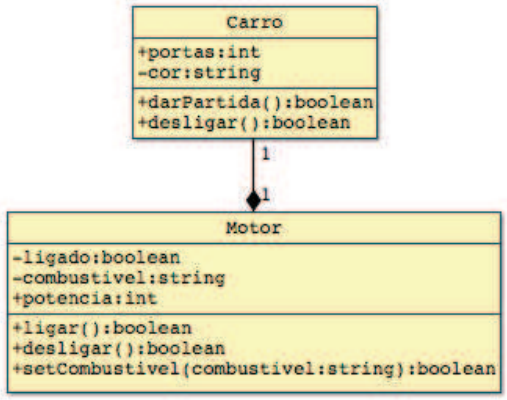

Figura 4: Exemplo do diagrama da listagem 1 como exibido pela visão gráfica da AWMo.

em função de cicatrizes em suas retinas, é formado em matemática e atualmente está no terceiro ano de um curso de bacharelado em sistemas de informação. Ele utiliza computadores desde 1984, com computadores como o CP 500. Em relação à programação de computadores, teve contato aos 18 anos e atualmente utiliza principalmente Java, PHP e SQL.

Em relação à modelos e diagramas da UML, teve contato ao longo do curso de sistemas de informação em disciplinas como banco de dados (modelos entidade-relacionamento) e programação orientada a objetos onde o conteúdo em Java era ilustrado com diagramas de classe da UML e também chegou a construir alguns diagramas utilizando a ferramenta BlueJ.

Durante a execução do estudo piloto foram encontrados alguns problemas menores de acessibilidade na AWMo e alguns problemas com as instruções das tarefas que os participantes eram solicitados a realizar na AWMo.

Após corrigidos os problemas encontrados tanto com a ferramenta quanto com seu protocolo, o estudo foi conduzido com um segundo participante, com 31 anos e cursando o terceiro ano de um curso de ensino a distância de Bacharelado em Sistemas de Informação oferecido com apoio de uma universidade federal. O participante é completamente cego em função de uma doença craniana e perdeu a visão quando tinha 15 anos de idade. Ele vem utilizando computadores a 7 anos e teve contato com programação de computadores a 5 anos atrás, sendo familiarizado com as linguagens Java, C# e JavaScript e outras tecnologias para desenvolvimento Web tais como HTML e CSS.

O segundo participante foi capaz de realizar todas as tarefas proposta pelo estudo em tempo em geral menor do que o primeiro participante, o que sugere que a ferramenta seja ainda melhor para usuários completamente cegos do que para usuários com baixa visão.

6. CONCLUSÕES

Figura 3: Captura de tela da AWMo com um modelo chamado Carro aberto para edição na visão textual.

Os resultados coletados com a condução do estudo de caso descrito indicam que a abordagem utilizada pela AWMo é bastante promissora para viabilizar a colaboração entre pessoas com e sem deficiências visuais em atividades de modelagem de software. O retorno obtido dos participantes do estudo foi bastante positivo indicando que a AWMo poderia ser utilizada como ferramenta de trabalho.

Foi possível notar que a abordagem utilizada não onera significativamente o processo de construção de modelos, se comparados o processo de descrever os modelos utilizando linguagem natural e o uso da linguagem textual da AWMo. O que pode ser visto pela capacidade dos participantes de construírem modelos razoavelmente complexos, em tempos entre 15 e 18 minutos. Além disso, os modelos criados com a linguagem textual da AWMo ficam automaticamente disponíveis para utilização na visão gráfica.

Como trabalhos futuros, os autores pretendem projetar novos experimentos para avaliar melhor a interface gráfica e o uso da AWMo por usuários com visão.

Do ponto de vista técnico, a AWMo possui uma arquitetura bastante flexível que permite que futuros colaboradores desenvolvam novas formas de visualização e edição de modelos sem afetar os que já existem. Uma dessas possíveis formas de visualização, é uma representação em HTML, utilizando *links* para melhorar a navegação do usuário no conteúdo do modelo durante sua leitura e permitindo que a leitura possa ser realizada de maneira não sequencial.

Com isso em mente, a AWMo será disponibilizada como uma ferramenta *open source* e um manual técnico será publicado para auxiliar potenciais desenvolvedores que se interessem em contribuir com a ferramenta.

7. REFERÊNCIAS

[1] Portaria N^o 3, de 7 de maio de 2007. Institucionaliza o Modelo de Acessibilidade em Governo Eletrônico - e-MAG no âmbito do Sistema de Administração dos Recursos de Informação e Informática - SISP. *Diário Oficial [da] República Federativa do Brasil*, page 103, 5 2007. n. 87, Seção 1, ISSN 1677-7042.

[2] M. Farwick, B. Agreiter, J. White, S. Forster, N. Lanzanasto, and R. Breu. A web-based collaborative metamodeling environment with secure remote model access. In *Proceedings of the 10th international conference on Web engineering*, ICWE'10, pages 278–291, Berlin, Heidelberg, 2010. Springer-Verlag.

[3] F. D. N. Grillo, R. P. M. Fortes, and D. Lucrédio. Towards collaboration between sighted and visually impaired developers in the context of model-driven engineering. In *Workshop GMLD (on Graphical Modeling Language Development). Joint Proceedings*, volume 1, pages 241–251, Copenhagen, 2012. 8th European Conference on Modelling Foundations and Applications (ECMFA 2012), Technical University of Denmark - DTU Informatics.

[4] IBGE. *Censo Demográfico 2010 - Resultados Preliminares da Amostra*. Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro, 2010.

[5] L. A. Ricci and D. Schwabe. An authoring environment for model-driven web applications. In *Proceedings of the 12th Brazilian Symposium on Multimedia and the web*, WebMedia '06, pages 11–19, New York, NY, USA, 2006. ACM.

[6] C. Thum, M. Schwind, and M. Schader. SLIM - A Lightweight Environment for Synchronous Collaborative Modeling. In *Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems*, MODELS '09, pages 137–151, Berlin, Heidelberg, 2009. Springer-Verlag.