

Model2gether: uma ferramenta de apoio ao ensino e a aprendizagem de modelos por cegos

**Leandro Luque^{1,2}, Christoffer L. F. Santos¹, Davi O. Cruz², Leônidas O. Brandão³,
Anarosa A. F. Brandão¹**

¹Escola Politécnica - Universidade de São Paulo (USP)
Av. Prof. Luciano Gualberto, 158 – 05.508-010 – São Paulo – SP – Brasil

²Departamento de Análise e Desenvolvimento de Sistemas
Faculdade de Tecnologia do Estado de São Paulo (Fatec) – Mogi das Cruzes, SP – Brasil

³Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
São Paulo, SP – Brasil

leandro.luque@usp.br, leo@ime.usp.br, anarosa.brandao@usp.br

Resumo. *O uso frequente na área da computação de modelos que possuem uma forte dependência em relação a representações gráficas cria obstáculos para a inclusão de pessoas cegas em cursos da área. No que diz respeito à educação presencial, existem várias medidas que podem contribuir para reduzir tais obstáculos, como o uso de objetos físicos para representar elementos de diagramas. No entanto, estas medidas não são efetivas em atividades de ensino eletrônico (EaD via Internet), devido à distância física entre os participantes. Neste contexto, este artigo apresenta o Model2gether, um protótipo web de apoio à inclusão de pessoas cegas em atividades de ensino eletrônico de modelos com forte vinculação a representações gráficas.*

Endereço do vídeo :<https://www.youtube.com/watch?v=2OrkJ8F7uw>

1. Introdução

De acordo com a Organização Mundial da Saúde[1], o número de pessoas com deficiência visual no mundo é estimado em 285 milhões, das quais 39 milhões são totalmente cegas. No Brasil, esses números são estimados em 35 milhões e 730 mil, respectivamente [2].

Tecnologias da informação e comunicação desempenham um papel importante na inclusão social de pessoas cegas. No entanto, muitas ferramentas utilizadas nas áreas da Ciência da Computação e Engenharia da Computação ainda são inacessíveis para este público [3, 4].

Ao considerar a inclusão de pessoas cegas em programas de educação em computação e na indústria da área, o uso frequente de modelos com forte vinculação a representações gráficas (modelos gráficos) cria barreiras de acessibilidade. Exemplos desses modelos incluem diagramas de fluxo de dados [5], modelos entidade-relacionamento [6] e modelos da Linguagem de Modelagem Unificada (UML) [7].

1.1. Justificativa Pedagógica

Em programas de educação em computação, muitos instrutores usam modelos para ensinar conceitos e geralmente não é permitido aos alunos cegos expressar seus conhecimentos por meio de tarefas alternativas [8]. Portanto, a fim de transmitir informação gráfica

em sala de aula, os professores têm de usar ferramentas, tais como canetas, papéis ou impressoras para desenhos táteis, visores táteis, objetos físicos com semânticas específicas, entre outros [9].

Algumas dessas ferramentas, tais como canetas para desenhos táteis e objetos físicos, permitem que instrutores conduzam algumas atividades incluindo alunos cegos e com visão normal em atividades cooperativas (por exemplo, a criação de um diagrama ou a identificação de erros em um diagrama criado anteriormente). Apesar da utilidade destas ferramentas em salas de aula tradicionais (presenciais), a sua aplicação em atividades de ensino eletrônico é difícil. Esta dificuldade decorre principalmente da falta de interação física entre os participantes, considerando sua distância espacial, bem como da limitada disponibilidade de dispositivos táteis.

Viabilizar a inclusão de alunos cegos em atividades de ensino eletrônico é importante porque, além de serem utilizadas em programas tradicionais de ensino, essas atividades são essenciais em programas de aprendizagem híbrida e a distância, que vêm crescendo em todo o mundo [10, 11].

Uma abordagem possível para reduzir esta dificuldade é definir e preparar, antecipadamente, materiais acessíveis para as atividades que possuem conteúdo gráfico, permitindo que os instrutores entreguem materiais (p.ex.: diagramas táteis) para os alunos antes do início da atividade. Isto, no entanto, impossibilita a modificação dos materiais acessíveis durante a atividade, resultando numa maior rigidez do conteúdo. Alguns autores [12, 13] defendem que quanto menor a flexibilidade do curso, menores serão as oportunidades de diálogo e interferências durante o seu desenvolvimento. Considerando a distância transacional, essa inflexibilidade pode aumentar a sensação de distância entre os participantes [13].

Para os fins deste trabalho, distância transacional pode ser entendida como o espaço psicológico e de comunicação a ser atravessado durante as interações entre instrutores e alunos [13], sendo ela aumentada pela ocorrência da distância espacial ou temporal. Assim, restringir o conteúdo relacionado à informação gráfica pode inviabilizar o uso de diagramas criados com base em situações levantadas durante a aula, o que poderia reduzir a motivação dos alunos.

Apesar da extensa literatura sobre a acessibilidade de modelos do ponto de vista de atividades individuais [14, 15, 16, 17, 18, 19, 20, 21], muitas das atividades que envolvem modelos gráficos utilizados para o ensino e a aprendizagem são cooperativas. A literatura sobre atividades cooperativas de modelagem envolvendo pessoas cegas é escassa [22] e encontramos apenas uma ferramenta de software que apoia parcialmente tais atividades [23, 22].

Neste contexto, este trabalho apresenta o protótipo de uma ferramenta desenvolvida para preencher esta lacuna, seguindo o modelo de *software* livre. Esta ferramenta é chamada Model2gether e está disponível em www.model2gether.com (ou www.model2gether.com:6223).

Os requisitos para esta ferramenta consideram tanto a acessibilidade quanto mecanismos de apoio à cooperação. Segundo Preece ([24]), atividades cooperativas envolvem três mecanismos sociais:

- Conversacionais: que facilitam o fluxo da comunicação e auxiliam a superar dificuldades;
- Coordenação: que permitem que pessoas trabalhem e interajam em conjunto;
- Consciência (*awareness*): que permitem descobrir o que está acontecendo, bem como o que os outros participantes da atividades estão fazendo.

Como exemplo, por meio da ferramenta, é possível que um instrutor com visão normal clique sobre um elemento de um diagrama e os alunos cegos sejam imediatamente notificados sobre o elemento selecionado, bem como possam navegar pelos elementos relacionados. A Figura 1 apresenta os requisitos implementados nesta ferramenta, adaptados de Luque et al. [9, 25]. Atualmente, ela oferece apoio à modelagem de modelos de casos de uso da UML. No entanto, o apoio a modelos de classes e entidade-relacionamento está sendo implementado.

Requisito		Grupo
Como participante com visão normal, eu desejo	1: criar e editar diagramas usando notações gráficas	Acessibilidade
	2: acessar diagramas criados por outros participantes com visão normal por meio de notações gráficas	
	3: acessar diagramas criados por participantes cegos por meio de notações gráficas	
	4: acompanhar mudanças nos diagramas durante atividades para compreender e colaborar	Consciência
	5: acompanhar destaques feitos por outros participantes durante atividades por meio de notações gráficas. Isto permitirá acompanhar explicações e colaborar	Conversacional e Acessibilidade
Como participante cego, eu desejo	6: criar e editar modelos em formato acessível	Acessibilidade
	7: acessar modelos criados por participantes com visão normal em formato acessível	
	8: acessar modelos criados por outros participantes cegos em formato acessível	
	9: acompanhar mudanças nos modelos durante atividades para entender e colaborar	Consciência
	10: acompanhar destaques feitos por outros participantes durante atividades em formato acessível. Isto permitirá acompanhar explicações e colaborar	Conversacional e Acessibilidade
Como participante, eu desejo	11: destacar elementos em um modelo que estou apresentando, criando ou editando, de tal forma que outros participantes possam acompanhar explicações e colaborar	Conversacional

Figura 1. Requisitos da ferramenta.

1.2. Contexto de Uso

A ferramenta pode ser utilizada como apoio a atividades de qualquer disciplina que envolva o uso de modelos gráficos, como diagramas, para o ensino de conceitos. Isto inclui disciplinas de Engenharia de Software, Análise de Sistemas, Projeto de Sistemas, Estruturas de Dados, entre outras. Embora contribua principalmente para atividades de ensino eletrônico, ela pode ser empregada em atividades presenciais.

2. Desenvolvimento

Model2gether é uma ferramenta gratuita, distribuída sob a licença GNU General Public License (GPL). Ela foi desenvolvida seguindo o Processo Unificado. Os seguintes artefatos foram produzidos: Documento de Visão, Especificação de Requisitos de Software e Documento de Arquitetura de Software. O Model2gether funciona no modelo cliente-servidor, devendo ser ele hospedado em um servidor Web.

Ela está organizada em 4 módulos lógicos. O módulo *Domínio* contém *Plain Old Java Objects* (POJO) que implementam abstrações de domínio, tais como Usuário e Compartilhamento. Ademais, este módulo contém um metamodelo que pode representar modelos baseados em grafos (com grafos aninhados em nós). Todos os elementos do metamodelo implementam o padrão de projeto Observador (*Observer*). Desta forma, qualquer mudança no metamodelo é imediatamente notificada aos observadores.

Os serviços independentes da web estão localizados no módulo de *serviços genéricos*. Além de outros serviços, este módulo gerencia a persistência das classes de domínio por meio de objetos de acesso aos dados (DAO) e uma implementação baseada em Java Persistence API (JPA). Além disso, este módulo contém classes responsáveis por produzir modificações no metamodelo e por gerenciar o conjunto de regras que devem ser seguidas para produzir uma instância válida de modelo. Os padrões *Factory Method* e *Strategy* são utilizados neste processo. Todos os serviços deste módulo podem ser acessados por meio de uma fachada (*Facade*).

O módulo de *serviço Web* é responsável por receber requisições e controlar o fluxo da aplicação. Ele foi implementado por meio de controladoras do *Spring MVC* e modelos da visão (*view-models*). Com exceção da interface de acesso, registro e das páginas de erro, todos os outros recursos são filtrados por filtros de interceptação.

Por fim, o módulo de *apresentação* é responsável por apresentar e controlar as interações com a interface do usuário. Ele foi escrito usando *JavaServer Pages* (JSP), HTML 5, CSS 3 e Javascript. Neste módulo, uma instância do metamodelo em Javascript é usada para representar o modelo em edição. Um dos observadores do metamodelo é um controlador Javascript que envia mensagens para o módulo de serviço Web por meio de *websockets*. Ademais, este código é responsável por receber mensagens do websocket e requisitar mudanças na representação gráfica ou textual do modelo. Atualmente, a ferramenta usa a biblioteca JSUML2 para desenhar diagramas de casos de uso. No entanto, uma biblioteca Javascript com melhor usabilidade está sendo desenvolvida.

3. Estudo de Caso

Em abril de 2016, foi realizado um estudo com 4 pessoas cegas e uma pessoa com visão normal. Os participantes cegos foram convidados da lista Cegos Programadores.

Primeiramente, foi proposto aos participantes cegos testar a ferramenta a fim de validar sua acessibilidade. Para isso, eles foram instruídos a realizar as seguintes atividades: (i) abrir a ferramenta e criar uma conta; (ii) acessar a ferramenta; (iii) localizar o menu de ajuda, abrir e ler o conteúdo; (iv) abrir um modelo de casos de uso baseado numa descrição textual; e (vi) compartilhar esse modelo com um outro usuário no sistema.

Após o teste, o participante com visão normal, um instrutor com mais de 10 anos de experiência no ensino de modelagem orientada a objetos, ensinou aos participantes conceitos de modelagem de caso de uso em uma aula. Questionamentos por parte dos participantes foram permitidos a qualquer momento.

Todos os participantes cegos ficaram muito entusiasmados com a ferramenta e deram um retorno positivo (p.ex.: “Foi a primeira vez que Eu me senti confortável acompanhando uma aula que continha conteúdo gráfico”). Um dos participantes usou a ferramenta por uma semana para trabalhar com colegas a fim de criar um modelo de casos de

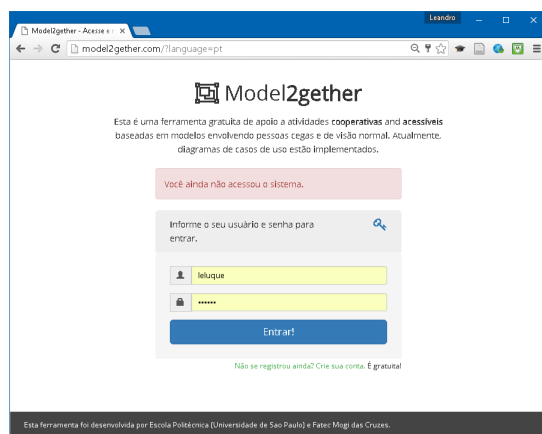


Figura 2. Acesso ao sistema

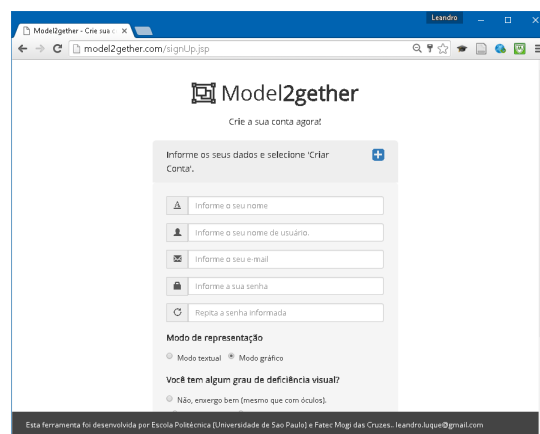


Figura 3. Criação de conta

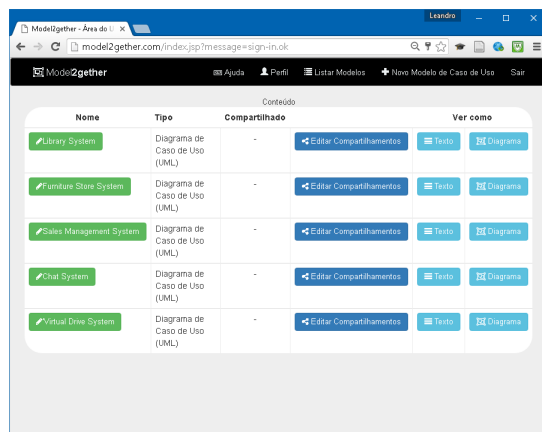


Figura 4. Listagem de modelos

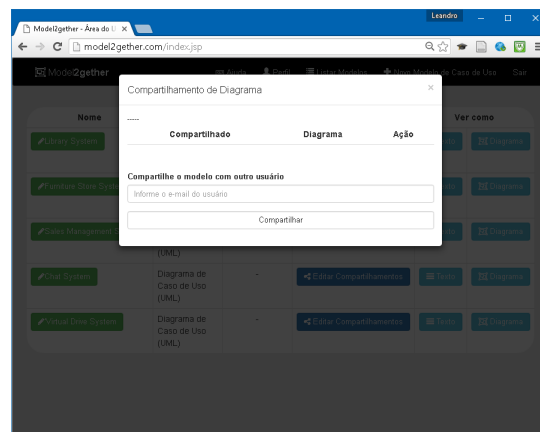


Figura 5. Compartilhamento

uso para um curso que estava fazendo.

4. Apresentação da ferramenta

Ao abrir a ferramenta, no endereço www.model2gether.com (ou www.model2gether.com:6223), uma interface de acesso é apresentada (Figura 2). Caso o usuário não possua uma conta, ele pode criar uma gratuitamente navegando até a opção intitulada “Não se registrou ainda? Crie sua conta”. Para criar uma conta, ele deve informar os seguintes dados: nome, e-mail, nome de usuário, senha, modo de representação padrão, grau de deficiência visual e se já estudou/estuda na área da Computação (Figura 3).

Após informar um nome de usuário e senha válidos, a interface principal da ferramenta é apresentada (Figura 4). Nela, são exibidas opções de acesso a ajuda, modificações de dados do perfil, listagem de modelos e criação de novos modelos. A opção selecionada por padrão é “Listar Modelos”. Nela, são apresentados os modelos criados pelo usuário ou compartilhados com ele. Além do nome e do tipo do modelo, são exibidas opções para compartilhar e abrir modelos. A opção de compartilhamento permite que um modelo seja compartilhado com outros usuários da ferramenta (Figura 5).

As opções de abertura de modelos permitem o acesso de duas formas: uma com-

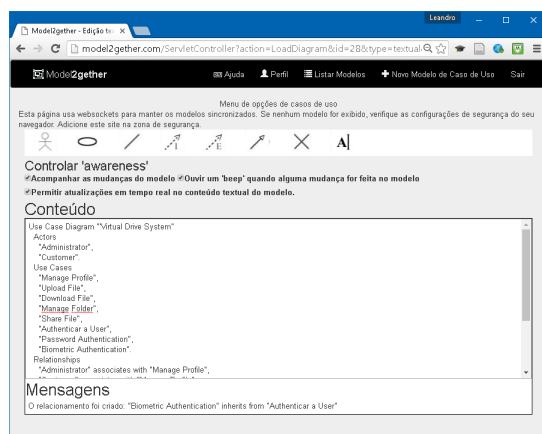


Figura 6. Edição textual

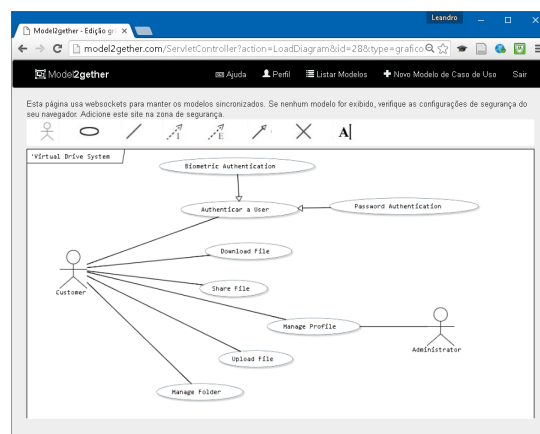


Figura 7. Edição gráfica

patível com leitores de tela, para deficientes visuais e outra gráfica, para pessoas com visão normal. Na primeira, os modelos são especificados por meio de uma Linguagem Específica de Domínio [26] (Figura 6). A interação é possível por meio de atalhos - uma combinação de modificadores e teclas (p.ex.: Ctrl+Shift+A) - e navegação por meio de teclado. Nesta interface, existem opções para controlar mecanismos de comunicação e consciência (awareness). Se o usuário marcar a caixa “Acompanhar as mudanças do modelo”, ele receberá mensagens de texto (p.ex. Um novo caso de uso foi criado: Emprestar material) sempre que algum usuário modificar o modelo (modificações visuais não são notificadas para usuários cegos). Além disso, se o usuário marcar “Ouvir um beep (apito) quando alguma mudança for feita no modelo”, beeps serão produzidos sempre que outro usuário modificar o modelo ou selecionar algum elemento. Outra possibilidade é o usuário especificar que a representação textual do modelo deve ser modificada a cada alteração do modelo (“Permitir atualizações em tempo real do conteúdo textual do modelo”). Adicionalmente, existem atalhos que permitem ao usuário navegar pelo histórico de ações que os usuários realizaram no modelo.

A interface para usuários com visão normal apresenta modelos de casos de uso em sua forma diagramática. A interação é possível por meio de mouse ou teclado (Figura 7).

4.1. Comparação com Ferramentas Similares

Até o presente momento, há dois grupos de ferramentas similares a Model2Gether. Um deles permite a especificação de modelos UML por meio de texto. Neste grupo, estão as ferramentas: MetaUML, PlantTextUML Editor, PlantUML, UMLetino e yUML. Para visualizar uma lista mais abrangente de ferramentas com estas características, veja Luque et. al [9]. O outro grupo diz respeito a ferramentas que apoiam a modelagem cooperativa envolvendo pessoas cegas, como CMMI e AWMo.

Para comparar essas ferramentas com Model2Gether, a Tabela 1 apresenta um conjunto de características comuns a atividades realizadas com modelos gráficos no meio acadêmico e na indústria.

5. Considerações Finais

Neste artigo, a ferramenta Model2gether foi apresentada. Trata-se de uma ferramenta de apoio à modelagem cooperativa envolvendo pessoas cegas, desenvolvida sob na forma de

Tabela 1. Comparação entre ferramentas

Funcionalidade	Modelagem Cooperativa			Editores Textuais de UML				
	Model2gether	CCMi	AWMo	MetaUML	PlantTextUML Editor	PlantUML	UMLetino	yUML
1. Permite edição textual	✓		✓	✓	✓	✓	✓	✓
2. Permite edição gráfica	✓	✓	✓				✓	
3. Permite compartilhamento	✓	✓	✓	✓	✓	✓	✓	✓
4. Permite armazenamento de modelos	✓	✓		✓	✓	✓	✓	✓
5. Permite exportar diagramas para imagens	✓	✓			✓	✓	✓	✓
6. Apoia cooperação em Tempo Real (TR)	✓	✓						
7. Implementa mecanismos de conversação em TR	✓							
8. Implementa mecanismos de consciência em TR	✓							
9. Implementa mecanismos de coordenação em TR		✓						

software livre.

Dada a importância de tais atividades na academia e na indústria, bem como a falta de soluções que implementam os requisitos para apoiar a modelagem cooperativa em tempo real, Model2gether pode ser considerado uma ferramenta promissora. Testes de usabilidade realizados com a ferramenta mostraram que ela contribui para atividades de ensino e aprendizagem envolvendo pessoas cegas.

Referências

- [1] World Health Organization et al. Visual impairment and blindness—fact sheet no. 282. 2012. *Accessed: February*, 2015.
- [2] IBGE Censo. Censo demográfico 2010. *Consultado em*, 12, 2010.
- [3] Elaine Gerber. The benefits of and barriers to computer use for individuals who are visually impaired. *Journal of Visual Impairment & Blindness*, 97(0), 2003.
- [4] Christopher Power, André Freire, Helen Petrie, and David Swallow. Guidelines are only half of the story: accessibility problems encountered by blind users on the web. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 433–442. ACM, 2012.
- [5] Yu-Liu Chen et al. Data flow diagram. In *Modeling and Analysis of Enterprise and Information Systems*, pages 85–97. Springer, 2009.
- [6] Peter Pin-Shan Chen. The entity–relationship model: toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1):9–36, 1976.
- [7] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.
- [8] Karin Müller. *How to make unified modeling language diagrams accessible for blind students*. Springer, 2012.
- [9] Leandro Luque, Leônidas de Oliveira Brandão, Romero Tori, and Anarosa Alves Franco Brandão. On the inclusion of blind people in uml e-learning activities. *Brazilian Journal of Informatics in Education*, 23(02):18, 2015.
- [10] Kátia Morosov Alonso. A expansão do ensino superior no brasil e a ead: dinâmicas e lugares. *Educação & Sociedade*, 31(113):1319–1335, 2010.
- [11] M Arrigo. E-learning accessibility for blind students. *Recent Research Developments in Learning Technologies*, 2005.

- [12] Romero Tori. *Educação sem distância*. Senac, 2010.
- [13] Michael G Moore. 2 theory of transactional distance. *Theoretical principles of distance education*, page 22, 1993.
- [14] Matt Calder, Robert F Cohen, Jessica Lanzoni, Neal Landry, and Joelle Skaff. Teaching data structures to students who are blind. *ACM SIGCSE Bulletin*, 39(3):87–90, 2007.
- [15] Filipe Del Nero Grillo and Renata Pontin de Mattos Fortes. Accessible modeling on the web: a case study. *Procedia Computer Science*, 27:460–470, 2014.
- [16] Filipe Del Nero Grillo, Renata Pontin de Mattos Fortes, and Daniel Lucrédio. Towards collaboration between sighted and visually impaired developers in the context of model-driven engineering, 2014.
- [17] Rubén Iglesias, Sara Casado, T Gutierrez, JI Barbero, CA Avizzano, S Marcheschi, and M Bergamasco. Computer graphics access for blind people through a haptic and audio virtual environment. In *Haptic, Audio and Visual Environments and Their Applications, 2004. HAVE 2004. Proc. of the 3rd IEEE International Workshop on*, pages 13–18. IEEE, 2004.
- [18] Alasdair King, Paul Blenkhorn, David Crombie, Sijo Dijkstra, Gareth Evans, and John Wood. Presenting UML Software Engineering Diagrams to Blind People. In K Miesenberger, J Klaus, W L Zagler, and D Burger, editors, *Computers Helping People with Special Needs*, number 3118 in Lecture Notes in Computer Science, pages 522–529. Springer Berlin Heidelberg, July 2004. DOI: 10.1007/978-3-540-27817-7_76.
- [19] Claudia Loitsch and Gerhard Weber. *Viable haptic UML for blind people*. Springer, 2012.
- [20] Oussama Metatla, Nick Bryan-Kinns, and Tony Stockman. Constructing relational diagrams in audio: the multiple perspective hierarchical approach. In *Proc. of the 10th international ACM SIGACCESS conference on computers and accessibility*, pages 97–104. ACM, 2008.
- [21] Luciano TE Pansanato, Christiane E Silva, and Luzia Rodrigues. Uma experiência de inclusão de estudante cego na educação superior em computação. In *XX Workshop on Computing Education*, 2012.
- [22] Oussama Metatla, Nick Bryan-Kinns, Tony Stockman, and Fiore Martin. Cross-modal collaborative interaction between visually-impaired and sighted users in the workplace. In *Proc. of the Designing Interactive Systems Conference 2012*. Georgia Institute of Technology, 2012.
- [23] Oussama Metatla, Nick Bryan-Kinns, Tony Stockman, and Fiore Martin. Designing for collaborative cross-modal interaction. In *Proc. of Digital Engagement 2011 the 2nd RCUK Digital Economy All Hands Meeting*, 2011.
- [24] Jenny Preece, Helen Sharp, and Yvonne Rogers. *Interaction Design-beyond human-computer interaction*. John Wiley & Sons, 2015.
- [25] Leandro Luque, Leônidas O Brandão, Romero Tori, and Anarosa AF Brandão. Are you seeing this? what is available and how can we include blind students in virtual uml learning activities. In *Brazilian Conference on Informatics in Education*, volume 25, pages 204–213, 2014.
- [26] Martin Fowler. *Domain Specific Languages*. Addison-Wesley Professional, 2010.