

Uma ferramenta de software livre para apoiar a construção colaborativa de diagramas UML

Mauro C. Pichiliani¹, Celso M. Hirata¹

¹Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica (ITA)
Caixa Postal 12.228-900 - São José dos Campos - SP

{pichilia, hirata}@ita.br

***Abstract.** Physical meetings are required to elaborate a detailed software design, in which the construction and the verification of software models are performed. These physical meetings have costs and affect the project schedule. Collaborative tools can be used during the development of software to mitigate these effects. Aiming to achieve this goal, this paper describes an extension of free software to support the collaborative elaboration of UML (Unified Modeling Language) diagrams through the Internet.*

***Resumo.** Reuniões presenciais são necessárias para elaborar um projeto detalhado de software, na qual a construção e verificação de modelos de software são realizadas. Essas reuniões presenciais têm custos e impactam os prazos de projeto. Uma abordagem para mitigar esses efeitos é usar ferramentas colaborativas no desenvolvimento de software. Com este intuito, descreve-se neste artigo uma extensão de um software livre para apoiar a elaboração colaborativa de diagramas UML (Unified Modeling Language) através da Internet.*

1. Introdução

Em desenvolvimento de software, as atividades de análise e projeto requerem um esforço considerável dos desenvolvedores para se obter os modelos de forma eficiente, em termos de custo e prazo. Muito desse esforço baseia-se na elaboração, apresentação, discussão e verificações de artefatos produzidos. A maioria dos artefatos é produzida utilizando-se de ferramentas de desenvolvimento. Contudo, a maioria dessas ferramentas não é colaborativa. Ferramentas que apóiem a colaboração remota permitem reduzir o custo decorrente de reuniões presenciais, além de desempenharem o papel de agente facilitador do trabalho em conjunto. Desta forma, a modificação de aplicações não colaborativas, para que elas apóiem a colaboração síncrona, é apresentada como uma alternativa para ajudar na comunicação, coordenação e colaboração entre equipes distribuídas e conectadas pela Internet.

Atualmente, pouco esforço tem sido feito no sentido de modificar softwares livres existentes para que eles apóiem a colaboração síncrona. Isto se deve à complexidade inerente à modificação de aplicações de código livre e aos detalhes técnicos envolvidos na implementação dos requisitos colaborativos.

O objetivo deste artigo é descrever como um software livre foi estendido para suportar requisitos colaborativos. A extensão do software livre segue a abordagem do Mapeamento de Componentes, proposto por Pichiliani et al. [Pichiliani et al. 2006]. De acordo com esta abordagem, os desenvolvedores devem mapear os componentes de uma

aplicação não colaborativa, que segue o estilo arquitetural MVC (*Model, View, Controller*), para os componentes de uma aplicação colaborativa, de acordo com um conjunto mínimo de requisitos colaborativos.

O software livre escolhido para receber os requisitos colaborativos foi a ferramenta CASE (*Computer Aided Software Engeneering*) de modelagem ArgoUML [ArgoUML 2007]. Após a extensão, a ArgoUML se tornou uma ferramenta que fornece apoio à elaboração colaborativa síncrona de diagramas UML através da Internet, oferecendo uma alternativa para a redução dos custos relacionados às reuniões presenciais durante a modelagem de diagramas UML.

Este artigo está dividido em cinco seções. A Seção 2 descreve os aspectos funcionais de uma aplicação colaborativa síncrona. Na Seção 3 apresentam-se algumas características da ArgoUML e os requisitos colaborativos especificados. Na Seção 4 descreve-se a estrutura interna da ArgoUML e como a extensão foi implementada. Por fim, na Seção 5 são apresentadas as conclusões.

2. Aspectos Funcionais de Ferramentas Colaborativas

Uma ferramenta colaborativa deve levar em consideração alguns aspectos funcionais para poder apoiar a colaboração. Santoro [Santoro et al. 2002] cita como principais aspectos funcionais a comunicação, a coordenação, a percepção (*awareness*) e o compartilhamento de informações.

A comunicação é importante em trabalho de grupo, seja ela utilizada para trocar idéias, discutir, aprender, negociar ou para tomar decisões. Diferentes canais de comunicação podem ser utilizados para tornar a comunicação mais eficiente e produtiva como, por exemplo, *chat* ou áudio-conferência.

A coordenação do grupo que trabalha com uma ferramenta colaborativa síncrona pode exigir um sofisticado mecanismo de controle das ações para garantir o sucesso da colaboração. Mecanismos de controle de concorrência, como travas, são utilizados para garantir a consistência dos elementos que estão sendo manipulados pelos participantes na elaboração do documento compartilhado.

Em situações onde a possibilidade de conflitos entre os usuários é pequena ou existir um moderador para coordenar as ações dos usuários, o controle de concorrência pode ser dispensado. Nestas situações, a coordenação das ações pode ficar a cargo do chamado protocolo social, caracterizado pela ausência de mecanismos de controle explícitos entre as atividades e pela confiança nas habilidades dos participantes de mediar por si as interações por meio dos um canais de comunicação disponíveis.

Nas ferramentas colaborativas síncronas, a percepção faz com que um participante fisicamente separado esteja ciente da presença e das ações dos demais participantes da colaboração. Ela relaciona-se com coordenação, comunicação e cooperação, permitindo também socializar virtualmente o ambiente, além de indicar o esforço dos participantes em interagir com a ferramenta e com os demais participantes.

A percepção é implementada nas ferramentas colaborativas síncronas por meio de elementos de percepção. Estes elementos fornecem informações imediatas a respeito do estado dos participantes na colaboração e de suas interações com a ferramenta. Exemplos destes elementos incluem a visão radar, uma visão em miniatura do espaço de

trabalho compartilhado, e os *telepointers*, que são representações remotas da posição do mouse dos usuários.

A colaboração necessita que os participantes compartilhem informações. Este aspecto é essencial para os grupos devido à necessidade de prevenir esforços repetitivos e assegurar que todos os participantes do grupo estejam utilizando a mesma informação, de forma que não haja inconsistências. As ferramentas colaborativas geralmente fazem uso de documentos compartilhados como mecanismo de compartilhamento de informação e de armazenamento da memória do grupo que é responsável por registrar todo o processo de interação, a comunicação realizada, passos desencadeados, e os produtos gerados durante a colaboração.

3. A ArgoUML

Após avaliar as ferramentas CASE de modelagem de código livre ArgoUML, Violet, EclipseUML, JUDE (*Java and UML Developers Environment*), Dia (*Diagram Editor*) e Umbrello UML Modeler, pôde-se verificar que nenhuma delas permite a edição colaborativa síncrona de diagramas UML. A ferramenta que mais se destacou na avaliação, e que se enquadra nos requisitos do Mapeamento de Componentes, foi a ArgoUML. Dentre os principais fatores que levaram à escolha da ArgoUML como ferramenta a ser estendida, destacam-se os seguintes:

- A ArgoUML é uma ferramenta CASE de modelagem madura, possui o código livre, foi escrito em Java e contém seus componentes estruturados de acordo com o estilo arquitetural MVC;
- Como toda ferramenta CASE de modelagem, a ArgoUML permite a edição dos principais diagramas UML, contém mecanismos de geração automática de código, a partir do modelo, auxilia a tarefa de engenharia reversa e possui um sistema de críticas dos modelos gerados;
- A ArgoUML atende a vários padrões, a saber: UML (especificação 1.3), XMI (*XML Metadata Interchange*), SVG (*Scalable Vectorial Graphics*) e OCL (*Object Constraint Language*);
- A ArgoUML recebeu elogios da crítica especializada, sendo recomendada por várias publicações como uma ferramenta ideal para editar diagramas UML.

Os principais requisitos colaborativos funcionais, apresentados a seguir, foram implementados na ArgoUML por meio da aplicação da técnica Mapeamento de Componentes. Eles servem como base para apoiar a colaboração síncrona durante a modelagem de diagramas UML em uma ferramenta CASE tradicional. Os requisitos especificados foram os seguintes:

- Os usuários devem poder iniciar, entrar ou sair de sessões de edição de diagramas colaborativas novas ou que já estejam em andamento;
- A ferramenta deve permitir o uso simultâneo de vários usuários de forma colaborativa, durante a edição de diagramas UML em um espaço de trabalho compartilhado;

- As ações sobre os elementos do diagrama devem ser coordenadas por um mecanismo de controle de concorrência que mantenha consistente o estado global do diagrama entre os usuários;
- Interações dos usuários que modifiquem o estado do diagrama devem ser propagadas para os demais usuários através de dispositivos de percepção; e
- A ferramenta deve permitir a comunicação dos usuários entre si durante a edição do diagrama, permitindo a coordenação de suas ações.

4. Estendendo a ArgoUML

4.1 Estrutura da ArgoUML

A ArgoUML possui duas grandes dependências externas, que são o NSUML e o GEF. Apenas o GEF é relevante para o contexto deste artigo. O GEF (*Graphical Edition Framework*) [GEF 2007] é um *framework* genérico utilizado na representação e no armazenamento de grafos. Este *framework* permite a construção de aplicações para a edição de grafos utilizando um modelo composto por nós e arestas. Este *framework* por si só é implementado no estilo arquitetural MVC e utiliza a biblioteca gráfica Swing do Java, permitindo que ele seja uma interface gráfica para a manipulação de estruturas de dados.

A ferramenta ArgoUML possui sua estrutura dividida em subsistemas, organizados de acordo com o estilo arquitetural MVC. Cada subsistema possui um nome e é implementado como um único diretório/pacote Java. As partes do Modelo, do Controle e da Visão são implementadas na ArgoUML por meio de seus subsistemas.

4.2 Modificando a ArgoUML

O uso do mapeamento na ArgoUML envolveu a modificação dos componentes dos subsistemas. Em particular, os componentes dos subsistemas das partes Visão e Controle receberam uma atenção especial por serem altamente acoplados. Além disso, identificar qual é a responsabilidade e qual é a função de alguns componentes das partes Visão e Controle representou um grande desafio, consumindo esforços consideráveis dos autores.

Antes do início da implementação, um estudo foi conduzido no código da ferramenta e na documentação disponível. Este estudo teve como objetivo a compreensão da estrutura, arquitetura, funcionamento e organização do código fonte, tanto do GEF como da ArgoUML, visando identificar como o mapeamento seria implementado nesta ferramenta. Este estudo levou quatro meses, contando com o esforço de um desenvolvedor, de nível pleno, trabalhando, em média, seis horas por dia.

A modificação de uma versão específica da ArgoUML levantou uma questão relevante sobre a continuidade dos requisitos implementados. Deste modo, as versões do GEF e da ArgoUML foram congeladas e modificadas em paralelo ao desenvolvimento oficial da ferramenta. Entretanto, os principais desenvolvedores da lista oficial da ArgoUML se mostraram interessados no resultado final, a ponto de incluírem no cronograma de modificações do projeto possíveis alterações, a serem implementados a longo prazo, que incorporem resultados deste artigo.

Devido ao forte acoplamento entre os componentes do GEF e os componentes da ArgoUML, percebe-se que, para implementar o compartilhamento dos elementos,

uma grande quantidade de modificações deve ser implementada diretamente nas classes do GEF, pois apenas especializar as classes do GEF por meio da herança não é suficiente para implementar a mudança de comportamento das classes.

Outro fator que influenciou a modificação do GEF está relacionado com a abstração do tipo de elemento manipulado pelo *framework*. Para o GEF não importa o tipo de elemento utilizado no diagrama, desde que os elementos do modelo possam ser armazenados em um grafo. Esta característica simplificou o uso da ferramenta, pois em vez de se concentrar a implementação de requisitos colaborativos em apenas um tipo de diagrama, implementar requisitos colaborativos no GEF faz com que todos os diagramas UML com que a ArgoUML trabalha permitam a edição colaborativa.

Além da modificação da ArgoUML e do GEF, um Servidor de Colaboração foi desenvolvido para auxiliar a troca de dados entre os participantes durante a sessão colaborativa, replicar os elementos do diagrama para todos os usuários, encaminhar as mensagens de comunicação e armazenar a versão mais atualizada do modelo.

A aplicação completa do mapeamento demandou a modificação de 40 classes na ArgoUML e 19 classes no GEF. Foram criadas oito novas classes para a ArgoUML e quatro novas classes para o GEF. Para incluir acesso às novas funcionalidades, três arquivos de configuração foram modificados. O Servidor de Colaboração demandou a criação de duas novas classes. Todo o trabalho de modificação manual do GEF e da ArgoUML, além da criação do Servidor de Colaboração, levou três meses de trabalho, contando com o esforço de um desenvolvedor, de nível pleno, que trabalhou, em média, seis horas por dia.

Obter informações sobre como modificar a ArgoUML evoluiu mais da metade do tempo total de implementação. Apesar de contar com a ajuda dos desenvolvedores oficiais da ferramenta, por meio da troca de mensagens na lista de discussão oficial da ArgoUML, a documentação pobre aliada à falta de padronização do código fonte representou um grande dificuldade para a implementação do mapeamento.

A Figura 1 apresenta a ArgoUML estendida, destacando os novos elementos da interface gráfica. Na Figura 1, os usuários *hirata* e *mauro* modelam um diagrama de classes durante uma sessão colaborativa.

O item A da Figura 1 mostra a aba que contém a ferramenta de *chat*, utilizada na comunicação entre os usuários durante a modelagem. O item B apresenta o *telepointer* do usuário *hirata*, indicando que o usuário *hirata* está com o seu ponteiro perto da classe sem nome, cuja cor é mais escura devido à trava do usuário *hirata*. O item C da Figura 1 mostra a janela de travas, indicando que o usuário atual é o usuário *mauro*. Pode-se notar na Figura 1 que existe um elemento do diagrama, identificado por *FigClass*, cuja trava está atribuída ao usuário *mauro*, além do elemento identificado por *FigInterface*, cuja trava está atribuída ao usuário *hirata*. O item D da Figura apresenta a interface chamada <<interface>>, preenchida com uma cor mais clara, pois o usuário *mauro* é proprietário de uma trava neste elemento do diagrama.

5. Conclusões

Este artigo apresentou uma alternativa para a redução de custos relacionados à elaboração de reuniões presenciais, descrevendo a modificação de um software livre para que ele forneça apoio à elaboração colaborativa síncrona de diagramas UML,

através da Internet. Deste modo, têm-se uma alternativa que permite a redução dos custos decorrente de reuniões presenciais durante a modelagem de diagramas UML.

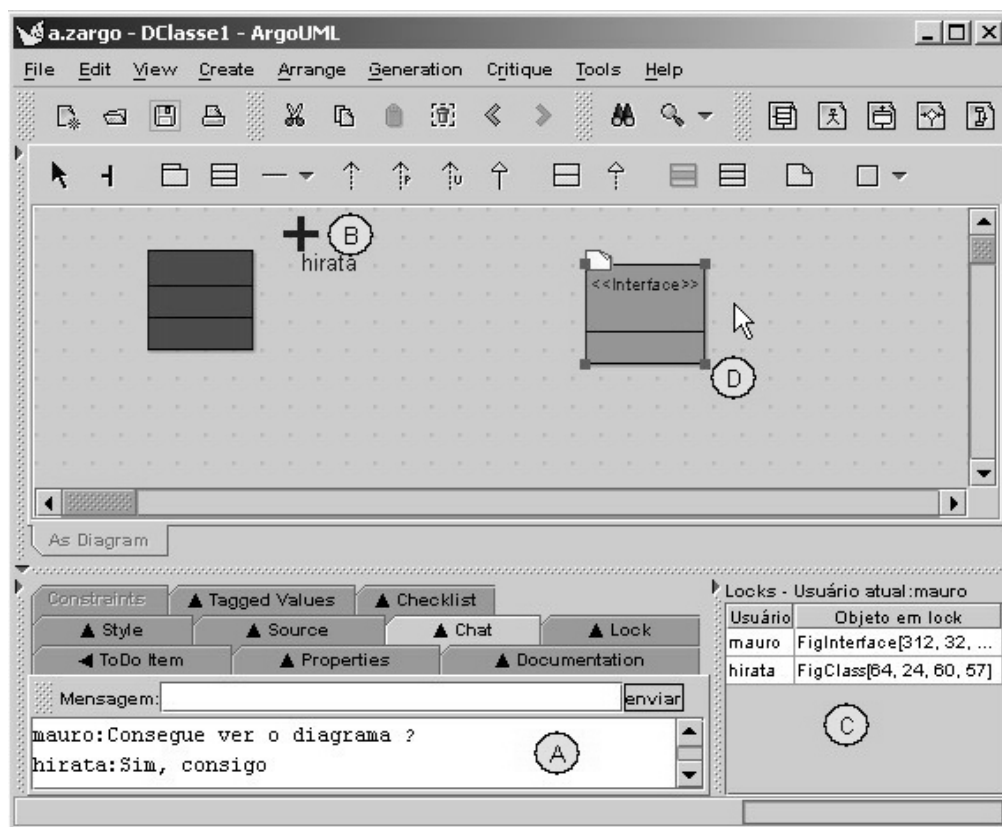


Figura 1. Interface gráfica da ArgoUML estendida.

A experiência adquirida pelo autor durante a modificação de uma ferramenta complexa como a ArgoUML não só pode ser reutilizada em outras aplicações, mas também representa um esforço da comunidade acadêmica em modificar aplicações existentes para suportar funcionalidades colaborativas. Além disso, o suporte à colaboração síncrona em uma ferramenta CASE de código livre representa um diferencial, permitindo que a ArgoUML se destaque em relação às seus concorrentes.

A ArgoUML estendida para permitir a edição colaborativa síncrona está disponível para *download* no endereço: <http://www.comp.ita.br/~pichilia/argo.htm>.

6. Referências

- ArgoUML. “ArgoUML”, web site acessado em fev./2007: <http://argouml.tigris.org/>.
- GEF. “Graphical Edition Framework”, web site acessado em fev./2007: <http://gef.tigris.org/>.
- Santoro, F. M., Borges, M. R. S. & Santos, N. (2002) “Ambientes de aprendizagem do futuro: teoria e tecnologia para cooperação”, mini-curso do XIII Simpósio Brasileiro de Informática na Educação, São Leopoldo, Rio Grande do Sul, 2002.
- Pichiliani, Mauro. C., Hirata, Celso M. (2006) “A Guide to map application components to support multi-user real-time collaboration”, em: Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2006), Georgia, E.U.A., 2006.