

# Ensaio Individual: Flexibilidade em Sistemas Colaborativos

Lucas Cyulik<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Tecnológica Federal do Paraná (UTFPR)  
CEP 80.230-901 – Curitiba – PR – Brasil

lcyulik@alunos.utfpr.edu.br

**Resumo.** *O presente documento tem como objetivo apresentar um ensaio individual sobre o seminário apresentado em sala, cujo o tema foi “Flexibilidade em Sistemas Colaborativos”. Para tanto será feita uma introdução ao tema, uma resenha da literatura consultada, e ao final será apresentada a visão do autor sobre o futuro da área.*

## 1. Introdução à Flexibilidade em Sistemas Colaborativos

Flexibilidade é a capacidade do *groupware* ser adaptável a mudanças, ajustável. O primeiro tipo de mudança existente é a customização que é o ajuste ou criação de algo de acordo com as preferências ou necessidade de alguém. O segundo tipo é a extensibilidade, que é a forma como novas funções podem ser adicionadas facilmente ao sistema, sem interferência com as existentes.

Por fim o terceiro tipo é chamado de *tailoring*, que são mudanças realizadas em funcionalidades do sistema pelo próprio usuário. Quando selecionada explicitamente pelo usuário, a mudança é chamada de adaptável. Quando feita automaticamente pelo sistema, são denominadas adaptativas. Ou ainda mistas, caso o usuário precise aprovar as mudanças sugeridas.

Entre os benefícios encontrados decorrentes do uso de flexibilidade em *groupware* podemos citar:

- Aumento da eficiência de utilização;
- Aumento da satisfação de uso;
- Diminuição da curva de aprendizado.

O uso da flexibilidade entrega ao usuário uma certa liberdade, que lhe permite coordenar como vai ser o uso da ferramenta, o que muitas vezes acaba em uma funcionalidade que os desenvolvedores não haviam pensado para o *groupware* que estavam desenvolvendo. A tecnologia de *groupware* deve providenciar flexibilidade suficiente para ser adaptada às necessidades de cada grupo e à evolução dos processos de trabalho.

## 2. Resenha da Literatura Consultada

O seminário realizado em sala transcorreu sobre exemplos de flexibilidade em jogos e sobre um modelo de arquitetura de software para *groupware* chamado “SLICE”. Dentre

os jogos citados podemos destacar o Left4Dead, um jogo *multiplayer* cooperativo de *survival horror* e tiro em primeira pessoa.

Esse jogo se destaca por sua inteligência artificial voltada a *tailoring* adaptativo, onde dependo das características do grupo de jogadores, o jogo se modifica para proporcionar uma experiência mais divertida aos usuários.

Essa é uma tendência que vem sendo tomada por muitos jogos atualmente, decorrente da grande quantidade de jogos disponíveis no mercado, os estúdios desenvolvedores necessitam entreter o usuário o maior tempo possível, logo características que são adaptadas automaticamente no jogo tornam-se interessantes para tornar os jogos mais dinâmicos.

O outro ponto abrangido pela apresentação foi o modelo de arquitetura SLICE (*Sharing Layers In Cooperative Editing*), destinado a criação de editores de texto cooperativos. O modelo é constituído por um conjunto de camadas, onde cada uma das quais contém objetos. As camadas são empilhadas com transparência entre si, sendo assim, cada camada mostra as camadas abaixo dela filtrando e/ou transformando-a.

Cada camada pode ser compartilhada ou cortada em pedaços. Se a camada é compartilhada, ela é a mesma para cada usuário e cada usuário tem acesso igual a seus objetos. Se a camada é cortada há uma instância (ou *slice*) da camada para cada usuário e cada usuário só tem acesso aos objetos de sua fatia.

O modelo é composto de sete partes descritas a seguir:

- *Abstract document layer*: esta camada é a representação interna do documento. É uma camada especial, uma vez que não tem qualquer representação gráfica. Para além da informação que representa o estado atual do documento, esta camada contém dados específicos para suportar as tarefas comuns, por exemplo, edição de um histórico de comandos dos autores.
- *Document representation layer*: esta camada mantém a representação exibida do documento. O mesmo documento lógico pode ser representado de uma variedade de maneiras. Por exemplo, um texto formatado pode ser representado de uma forma WYSIWYG, como em MacWrite ou como um texto com comandos de formatação incorporados, como um arquivo de origem LaTeX, por exemplo. Mesmo uma representação visual pode ter variações, tais como o tamanho da página, o número de colunas, as fontes, etc.
- *Direct Manipulation (DM) layer*: os objetos da camada de representação são manipulados utilizando uma interface de manipulação direta. A camada *Direct Manipulation* detém os objetos que são frequentemente utilizados em sistemas de manipulação direta para fornecer *feedback* adequado do estado da interação com o utilizador. Isso inclui alças de seleção, linhas de construção, etc.
- *View representation layer*: esta camada define janelas que exibem partes do documento (também chamados de pontos de vista do documento).

- *View manipulation layer*: as *view* são manipulados através de objetos exibidos nesta camada. Objetos típicos são barras de rolagem e vários ícones que permitem mover, redimensionar, etc.
- *Indirect Manipulation (IM) layer*: muitos comandos em chamada de interfaces diretas de manipulação são realmente emitidos por meio de menus e/ou caixas de diálogo. Isso ocorre porque a manipulação direta é inadequada para apoiar as operações precisas ou abstratas. Menus, caixas de diálogo, paletas de ferramentas e outros artefatos de manipulação indireta são realizadas nesta camada.
- *Cursor layer*: esta camada que contém o cursor segue o movimento do *mouse*. Se vários dispositivos de entrada de localização estão disponíveis, cada um deles pode ser representado por um cursor correspondente nesta camada.

O Interessante desse modelo é que hoje as ferramentas de edição de texto colaborativas são comuns no dia a dia de estudantes e pessoas em geral, e este modelo é de 1994, o que demonstra que CSCW é uma área que atrai as pessoas a muito muito tempo, e a tendência é de evoluir cada vez mais.

### 3. Visão Sobre o Futuro da Área

A flexibilidade é um dos temas mais importantes em *groupware*, pois permite ao usuário guiar seu uso dentro de uma ferramenta, com isso o desenvolvedor não entrega uma ferramenta fechada, mas sim uma “tela em branco” onde o usuário faz o que quiser, na medida do possível.

Um ponto importante é que cada pessoa pensa e faz coisas de maneira diferente, e isso não seria diferente em *groupware*, o modo que cada pessoa usa uma ferramenta é distinto, sendo assim é importante dar liberdade ao usuário para usar o programa do seu jeito, e deixar ele adicionar aquilo que acha importante para o seu uso.

Quando uma pessoa pode fazer algo do jeito que ela quer ou gosta, isso traz maior satisfação, pois nada está sendo imposto, a pessoa age de acordo com sua vontade. Decorrente disso a uma maior eficiência, pois o uso do programa se torna mais natural.

Futuramente, acredito que que chegaremos em um nível de flexibilidade onde o usuário leigo conseguirá mudar as regras de negócios do programa, sendo possível assim criar novas soluções dentro da ferramenta. O contato constante das novas gerações com a tecnologia vai tornar muito natural essas formas de mudança, sendo assim desnecessário um programador para adicionar ou modificar uma funcionalidade de um programa.

### Referências

BEAUDOUIN-LAFON, Michel. **Logical models and software architectures for CSCW**. In: **CSCW'94 workshop on distributed systems, multimedia, and infrastructure support in CSCW: delegate copy**. 1994

Booth, Michael. **The AI Systems of Left4Dead**. Disponível em:  
<[http://www.valvesoftware.com/publications/2009/ai\\_systems\\_of\\_l4d\\_mike\\_booth.p  
df](http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf)>. Acesso em: 02 nov. 2015