3
Apoio à Tomada de Decisões Sobre Adaptação e Extensão para Design de Interfaces

Este capítulo faz uma ponte entre a análise e o design das aplicações extensíveis. Primeiro caracterizamos adaptação com base nos trabalhos de Dieterich e co-autores (1993) e Assis (2005), que foram revisados para contemplar as técnicas e mecanismos de extensão apresentados no capítulo anterior. Nessa caracterização, propomos uma classificação unificada na qual enquadramos essas técnicas e mecanismos. Finalmente, apresentamos a proposta principal deste trabalho, com base no trabalho de Hackos & Redish (1998), propondo e organizando perguntas a serem feitas durante a análise para orientar o designer a considerar soluções alternativas de adaptação a serem incorporadas no sistema.

3.1.
Caracterização da Adaptação

Como dito anteriormente, o ponto inicial desde estudo é investigar o que nos leva a propor soluções de extensibilidade ou customização para um sistema. Para isso, é necessário caracterizar o que consideramos adaptação.

Quando um sistema é concebido, os designers definem quais os problemas que ele se dedica a resolver. Esta decisão normalmente é tomada de acordo com os objetivos dos usuários. Porém, os usuários podem ter diferentes objetivos, diferentes graus de conhecimento sobre um determinado assunto e, até mesmo, podem mudar de idéia com o tempo.

Sendo assim, torna-se difícil, senão impossível, propor soluções que atendam a todas as necessidades e preferências de todos os perfis usuários em todas as situações, ao longo do tempo.

Para que um sistema possa servir a diferentes tipos de usuários, ou mesmo servir a um usuário ao longo do tempo, torna-se importante propor soluções de customização e extensão, de forma que os usuários possam escolher dentre

algumas alternativas a que mais lhe agrada ou a que satisfaz suas necessidades. As alternativas estão relacionadas a desde **o que** pode ser adaptado até **como** o usuário deve interagir com o sistema para usufruir de tal adaptação.

Inspirados no trabalho de Assis (2005) e levando em consideração a classificação de Dieterich e co-autores (1993), realizamos um estudo sobre os mecanismos de adaptação do *Mozilla Thunderbird* a partir do qual propusemos o seguinte conjunto de perguntas para caracterizar a adaptação (Silva et al. 2008).

Iniciativa

Quem tem a iniciativa de adaptar?

A iniciativa será tomada em função de quê?

Quando tomar a iniciativa de adaptação?

Proposta

Quem propõe as opções de adaptação para cada iniciativa?

Quais são as possíveis propostas de adaptação?

Por que executar cada proposta de adaptação?

As propostas de adaptação serão em função de quê?

Quando as opções de adaptação devem ser propostas?

Decisão

Quem decide que adaptação deve ser executada?

A decisão de que adaptação deve ser executada será em função de quê?

Quando tomar a decisão de qual adaptação deve ser executada?

Execução

Quem executa a adaptação?

Quando a adaptação é executada?

Como executar a adaptação?

Como desfazer a adaptação?

Até quando persiste a adaptação realizada?

Ao tentar caracterizar os mecanismos de adaptação apresentados no capítulo 2, no entanto, houve a necessidade de rever essas perguntas como é apresentado nas próximas seções.

3.1.1. Quem realiza a adaptação?

Seguindo o modelo apresentado por Dieterich e co-autores (1993), podemos classificar uma adaptação segundo suas etapas e os agentes que a realizam, sejam o usuário, o sistema ou ambos: quem **inicia** a adaptação, quem **propõe** as alternativas de adaptação, quem **decide** qual adaptação deve ser escolhida e quem **executa** a adaptação (Figura 2).

Existem pelo menos quatro possíveis agentes que realizam adaptações:

1. O sistema, no caso de sistemas adaptativos. Nesse caso, costuma-se embutir no sistema modelos (de domínio, aplicação, tarefas, e usuários, entre outros) e regras que realizam a adaptação conforme o conteúdo dos modelos e que atualiza os próprios modelos, conforme o histórico de interação do usuário.
2. Qualquer usuário, conforme permitido pelo designer. Em sistemas cujo perfil de usuário não restringe sua capacidade de adaptá-los, geralmente os mecanismos de adaptação ficam disponíveis a todos, sem distinção.
3. Usuários específicos, conforme permitido pelo designer. Em alguns sistemas, os mecanismos de adaptação ficam restritos a um único perfil, em geral de administrador, sem que este possa delegar poderes a outros usuários.
4. Usuários específicos, conforme permitido por um administrador ou um superior seu. Em sistemas multi-usuário, com um modelo de usuários ou de grupos de usuários bem definido, muitas vezes é desejável restringir não apenas o conjunto de funcionalidades disponíveis a cada usuário, mas também os mecanismos de adaptação.

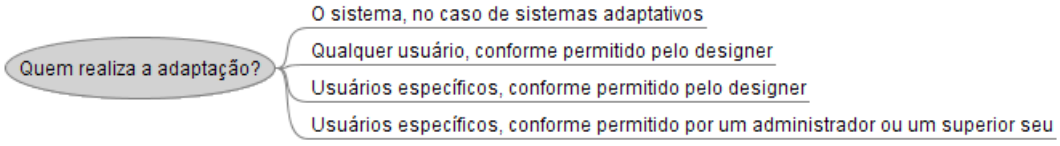


Figura 26: Quem realiza a adaptação?

3.1.2.
O que adaptar?

Como visto anteriormente, Koch (2000 *apud* Assis, 2005) classificou o que pode ser adaptado em: adaptação do **conteúdo**, **apresentação** e **navegação** no contexto de sistemas hipermídia adaptativos.

A **adaptação do conteúdo** permite mostrar ou esconder informações. Isto pode ser feito pelo próprio usuário ou até mesmo pelo sistema. Imagine uma aplicação cujo conteúdo é dividido em abas. O usuário pode abrir ou fechar as abas de acordo com o que deseja ver.

Podemos ter também a seleção da informação de acordo com o nível de conhecimento do usuário. Por exemplo, digamos que o sistema saiba, *a priori*, se o usuário é iniciante, intermediário ou especialista. E para cada tipo de usuário, o sistema possui um conjunto de informações diferentes. Para os especialistas, ele pode mostrar menos coisas, e deixar que ele mesmo faça o restante. Para o iniciante, ele pode fazer algumas perguntas, dar dicas ou exibir ajuda. Isto tudo dependeria da inferência ou conhecimento do sistema sobre quem é o usuário e assim, a escolha do que mostrar estaria à cargo somente do sistema.

Esta adaptação engloba também o uso de perspectivas diferentes sobre um mesmo conteúdo como, por exemplo, mudar uma unidade de medida (Kg X lb).

A **adaptação da apresentação** engloba diferentes aspectos da interface ou de formatação do conteúdo. Podemos ter alterações de cores, tamanho e tipo de fontes, agrupamento, posicionamento e ordenação de elementos de interface, por exemplo. Esta adaptação engloba ainda alterar textos ou imagens, como itens de menu, rótulos e ícones.

A **adaptação da navegação** diz respeito aos caminhos que devem ser percorridos para atingir determinado objetivo. Isto pode incluir os elementos que compõem os menus, a alteração/criação de teclas de atalho etc. Trata de qual conteúdo será visitado e de que forma ele será visitado através da navegação.

Como estamos tratando de sistemas extensíveis em geral, em vez de navegação vamos lidar com **interação**, que diz respeito às formas de o usuário realizar uma tarefa qualquer. Por exemplo, ele pode criar um atalho para uma operação ou seqüência de operações (que englobam a gravação de macros e a

programação por demonstração), um molde a ser aplicado a diferentes objetos, ou configurar a forma como ele realiza uma tarefa.

Além dos tipos de adaptação propostos por Koch, precisamos considerar também mais dois tipos de adaptação: 1) do **comportamento** do sistema, ou seja, a forma como ele realiza uma seqüência de operações.; 2) de **objetivos**, ou seja, a incorporação de novas funcionalidades que ampliem o conjunto de objetivos que o sistema apóia.

Podemos citar como forma de adaptação do comportamento, a reordenação de operações de uma macro ou a configuração de preferências, como por exemplo, a freqüência com que uma aplicação salva um documento.

Como forma de adaptação de objetivos, podemos citar a integração de componentes ou funcionalidades e a criação de novas funcionalidades através da criação de códigos.

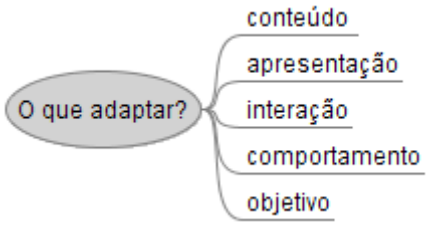


Figura 27: O que adaptar?

***

Sob uma perspectiva semiótica, podemos identificar as dimensões dos sistemas que são afetadas por uma adaptação: léxico, sintaxe, semântica ou uma combinação delas, conforme visto na Seção 2.2.1.

No trabalho de de Souza & Barbosa (2006), nenhuma combinação trata apenas da alteração de valores pré-definidos pelo designer, dentro de um limite também estabelecido por ele. No entanto, quando fazemos a personalização da aparência, estamos alterando valores definidos pelo designer, mas não estamos especificamente criando ou alterando o vocabulário do sistema, nem sua sintaxe ou semântica. Desta forma, achamos válido criar uma coluna que engloba esse tipo de adaptação, a qual denominamos **combinação do tipo 0**. Para cada atributo do sistema, estamos apenas mudando os valores atribuídos a ele pelo designer. A Tabela 7 encaixa os exemplos citados anteriormente nessas dimensões.

Valores (tipo 0)	- personalização da aparência da interface (Silva, 2001) - configuração de preferências (Silva, 2001)
léxico	- alteração de rótulos ou ícones - criação de atalhos para uma operação
sintático	- reordenação dos itens de interface - visibilidade dos elementos de interface - visibilidade do conteúdo/ informações - reordenação de operações ou componentes (de uma macro): Automator, Photoshop -PbD: Eager (Cypher, 1991) (macro anônima)
léxico + sintático	- integração de componentes ou funcionalidades do próprio sistema - gravação de macros: Word, Photoshop - programação por demonstração: Cocoa™, Mondrian (Lieberman)
léxico + semântico	- estilos de formatação de texto - criação/ edição de moldes
léxico + sintático + semântico	- integração de componentes ou funcionalidades em um sistema aberto (extensões do Mozilla Firefox) - programação numa linguagem de script: Hypercard™ (Hypercard, 1993), Barista (Ko & Myers, 2006)

Tabela 7: Organização dos exemplos de adaptação segundo as dimensões do sistema que sofrem a adaptação

3.1.3. Como adaptar?

A partir da análise das classificações de aplicações extensíveis da literatura que apresentamos no capítulo anterior, achamos conveniente reagrupar os diversos mecanismos numa classificação unificada que considera os seguintes tipos de estratégias de adaptação:

1. parametrização;
2. composição declarativa;
3. composição procedimental concreta;
4. composição procedimental generalizada;
5. redação utilizando linguagem de script, linguagem específica da aplicação ou de domínio; ou
6. redação utilizando linguagem de propósito geral.

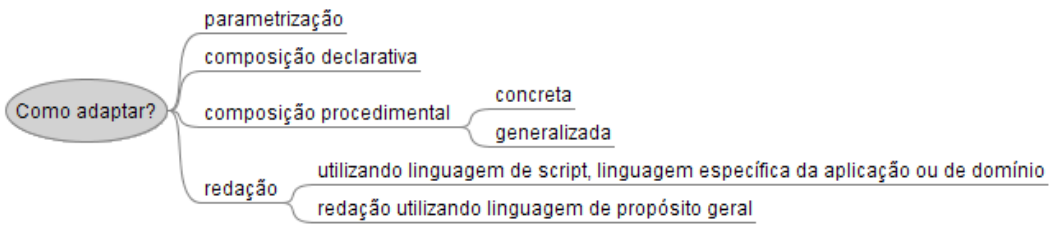


Figura 28: Como adaptar?

A **parametrização** engloba as atividades de ativação/desativação de opções e atribuição de valores diferentes aos parâmetros disponíveis pelo designer, independente se essas alterações irão afetar a interface ou a maneira como o programa apresentará suas funcionalidades. Alterações deste tipo mudam somente a semântica do sistema de significação, sob a perspectiva semiótica. O usuário não pode compor os elementos de diferentes formas, nem modificar o vocabulário usado para formar as sentenças da linguagem do sistema.

A parametrização exige que os designers pensem, já nas fases iniciais do projeto de implementação do sistema, nas opções que estarão disponíveis para ajuste do usuário final. O usuário não cria nada novo.

Esta classificação segue os mesmos princípios apontados pelos outros pesquisadores.

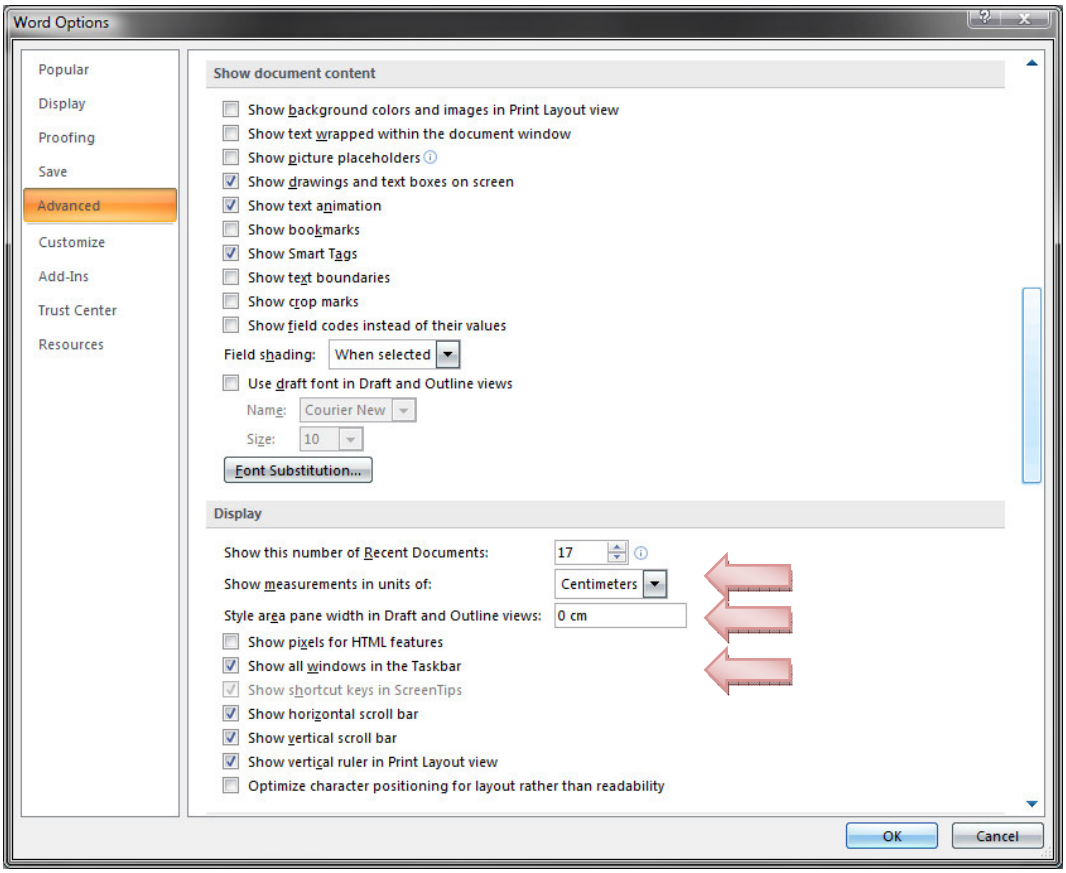


Figura 29: Microsoft® Office Word 2007 – tela de configuração de preferências

A Figura 29 mostra um exemplo de parametrização encontrado no Microsoft® Office Word 2007. Esta tela apresenta diversas opções a serem personalizadas pelo usuário. As três setas adicionadas a figura mostram diferentes formas de alteração dos parâmetros: a seleção, a partir de uma lista pré-definida; a

inserção, através da digitação; e a ativação/desativação. Note que, mesmo no caso da inserção, o designer limita o espaço dos possíveis valores informados pelo usuário (Figura 30).

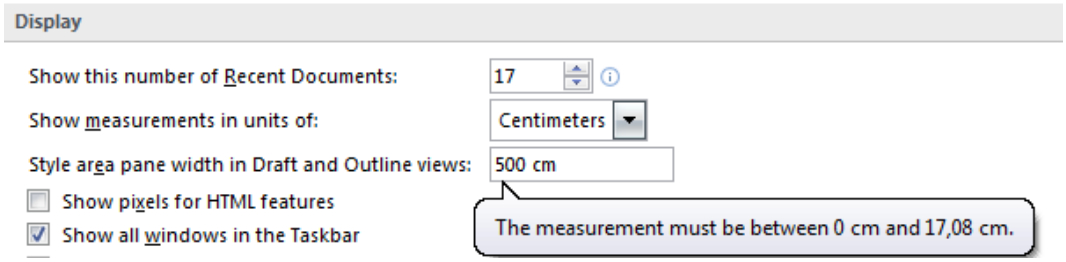


Figura 30: Microsoft® Office Word 2007 – mensagem de erro para valores inválidos

Chamamos de **composição declarativa** os mecanismos que permitem que o usuário componha um *workflow* de ações a serem executadas. As atividades envolvidas são composições, na medida em que estamos lidando com articulação de ações; e declarativa porque o usuário indica o que deve compor a extensão (quais são as ações e, eventualmente, parâmetros), mas não demonstra o procedimento para o sistema. O usuário poderá associar esta composição a um elemento da interface, de maneira que o acesso a ele seja facilitado.

Dentro deste grupo, podemos identificar mecanismos de integração, cujos componentes podem ser internos ou externos ao sistema. Um exemplo bastante comum desta técnica são os componentes do Firefox®. Os componentes são criados fora do ambiente do sistema e podem ser incorporados através de uma opção do menu do navegador. A instalação é simples e, como se trata de um sistema web, ele fornece a possibilidade do usuário atualizar os componentes, procurar por mais extensões, desativar ou desinstalar as existentes.

Figura 31: Mozilla Firefox® - componentes

O exemplo anterior mostra como novas funcionalidades que não haviam sido previstas pelo designer do sistema podem ser incorporadas ao navegador. Neste caso, a metamensagem pode ser totalmente modificada. Porém, temos o caso de integração prevista pelo designer, onde ele permite que parte da metamensagem contenha lacunas a serem preenchidas pelo usuário. Ou seja, o usuário pode integrar componentes previamente previstos pelo designer. Um exemplo típico de um mecanismo deste tipo é permitir que o usuário escolha o nível de instalação de um sistema (recomendado, avançado etc.) e, para cada nível, ele pode escolher qual componente deseja instalar. Mesmo após a instalação o usuário pode refazer a escolha de instalação/desinstalação dos componentes.

Na Figura 32 o pacote Microsoft® Office 2003 permite que o usuário escolha quais ferramentas ele deseja usar. Apesar de este exemplo exteriorizar o que estamos estudando aqui, ele retrata muito bem o exemplo de integração analisado.

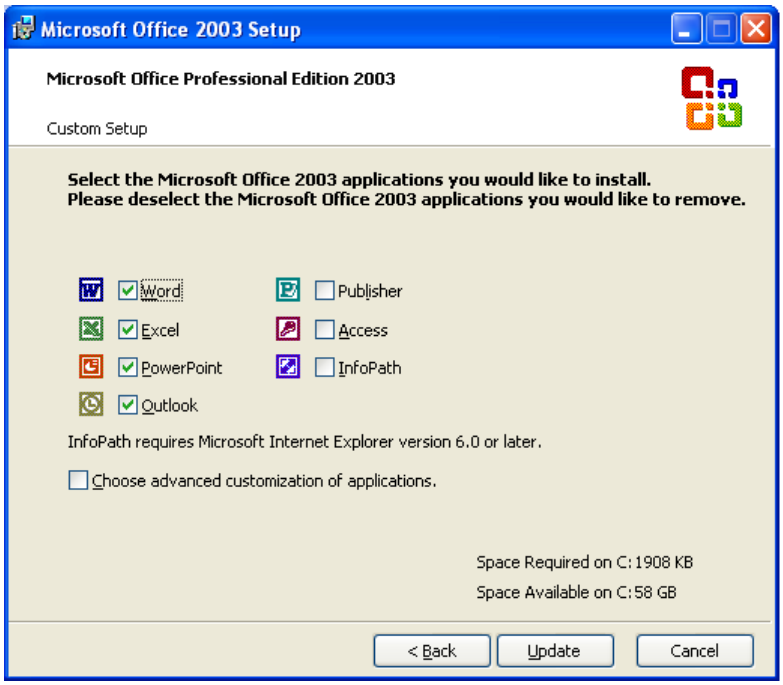


Figura 32: Microsoft® Office 2003 – Tela de escolha de componente de instalação

A criação de estilos para formatação de texto também é um tipo de composição declarativa. O usuário pode agrupar as diferentes formatações de texto e parágrafo, por exemplo, e dar um nome a este estilo, que será adicionado a uma lista. Depois, basta o usuário selecionar o texto e escolher a agrupamento de estilos desejado para que ele seja aplicado ao texto. Silva (2001) chamou este tipo de mecanismo de uso de moldes.

A criação de atalhos é outro exemplo que se encaixa neste grupo. Neste caso, ela é degenerada, pois é composta de apenas uma instrução. O usuário pode associar um comando a uma combinação de teclas ou a um botão, que pode ser inserido na barra de ferramentas do sistema.

Em contrapartida, usamos a notação de **composição procedimental** aos mecanismos em que o usuário informa, passo a passo, o que o sistema deve fazer a fim de atingir a intenção final.

Dentro desta composição, podemos ter mecanismos concretos ou generalizados. Na classificação de **composição procedimental concreta**, estão os mecanismos de gravação de macro. Esses mecanismos são concretos porque registram literalmente as operações realizadas pelos usuários.

Um exemplo interessante que envolve mecanismos deste tipo é o Automator, da Apple. Ele permite que o usuário defina ações a serem executadas em um workflow. A Figura 33 mostra a tela do programa. No lado esquerdo são lista as aplicações e as possíveis ações associadas a cada uma. O usuário deve arrastar a ação desejada para o painel do lado direito, e definir os valores necessários de cada ação. Após definir o *workflow*, o usuário pode executar a ação. É possível também nomear a lista de ações e salvar para uso posterior.

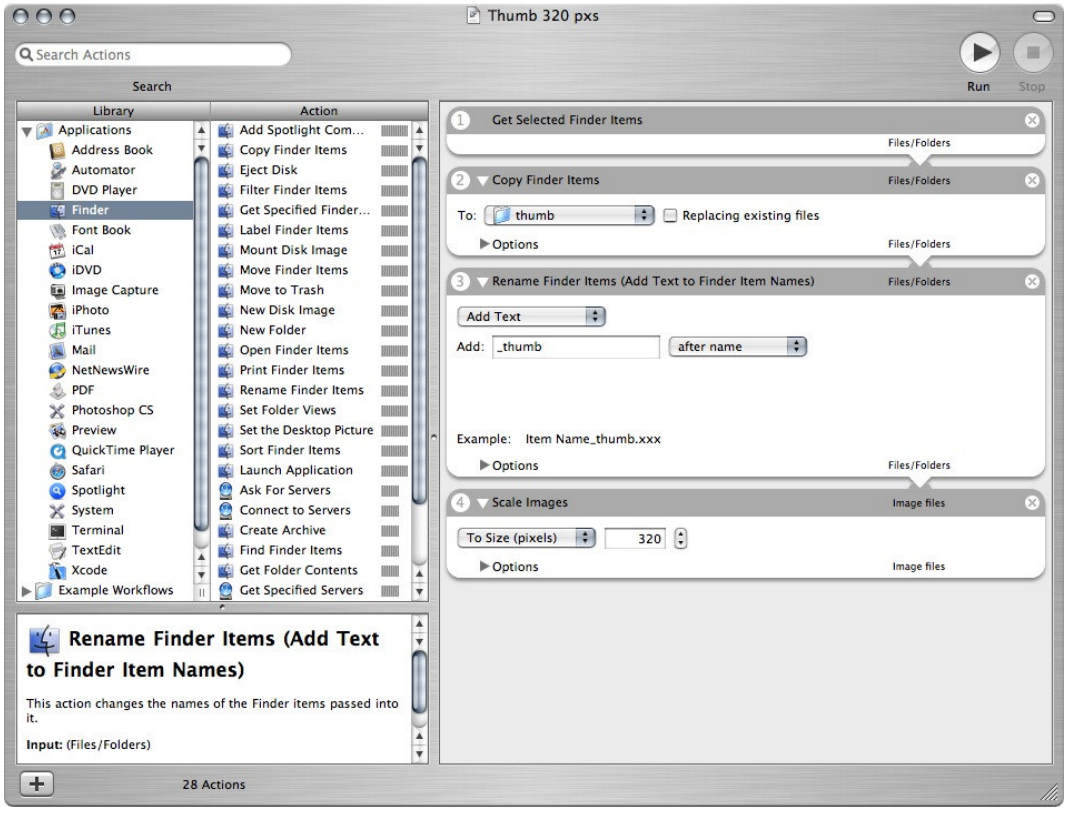


Figura 33: Automator (Apple Mac OS X) – Automação de tarefas

No exemplo, o usuário está criando *thumbs* para todas as fotos selecionadas pelo Finder (uma aplicação do sistema Mac OS X). Após a seleção, o script faz uma cópia do arquivo original, adiciona o texto "_thumb" no final de cada nome de arquivo e redimensiona o tamanho das fotos para 320px.

A criação de atalhos também é um exemplo que se enquadra neste grupo. Neste caso, estamos falando de criações que envolvam mais de uma tarefa. Por exemplo, ao final da gravação de uma macro, o usuário pode associar o conjunto de instruções gravado a uma tecla de atalho ou botão da interface.

Na classificação de **composição procedimental generalizada,** o sistema cria programas generalizados a partir das interações dos usuários. Alguns mecanismos permitem até o uso de condicionais e iteradores. Podemos ter mecanismos que atuam com a inferência do sistema, como no caso do Eager, e os que atuam sem a inferência do sistema, como no caso do Cocoa™[5].

O Eager (Cypher, 1991) apóia o usuário na execução de tarefas repetitivas. Ele constantemente observa as ações no sistema e, quando detecta uma atividade repetitiva, ele escreve um programa que executa a tarefa para o usuário. Porém, o usuário faz a escolha se deseja para passar o comando da execução das tarefas para o Eager. As tarefas repetidas nada mais são do que exemplos ou demonstrações do que o usuário deseja fazer.

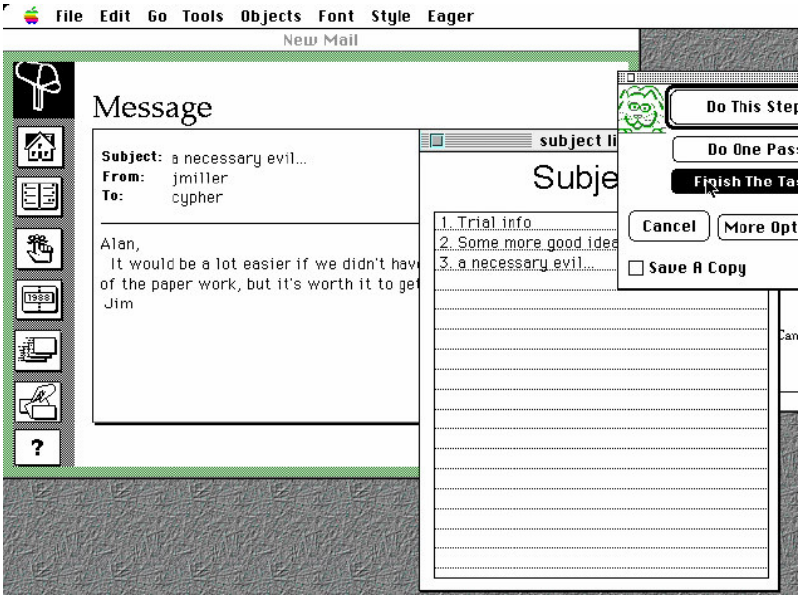


Figura 34: Eager (Cypher, 1991) – apóio a tarefas repetitivas

[5] <http://developer.apple.com/cocoa/> Acesso em 21/01/2008.

O Cocoa™ permite a criação de jogos e simulações. Ele foi desenvolvido para ser usado por crianças de oito anos de idade. Na Figura 35, o usuário está definindo como o personagem deve se comportar quando encontra uma parede. Ele pode definir movimentos para cima, para baixo, para o lado direito e para o esquerdo. Na figura, ele está criando e gravando uma regra. Após a criação, ele pode simular a execução. O usuário define vários exemplos do que o sistema deve fazer em casos específicos. Neste sistema não há inferências. As regras são sempre definidas pelos usuários.

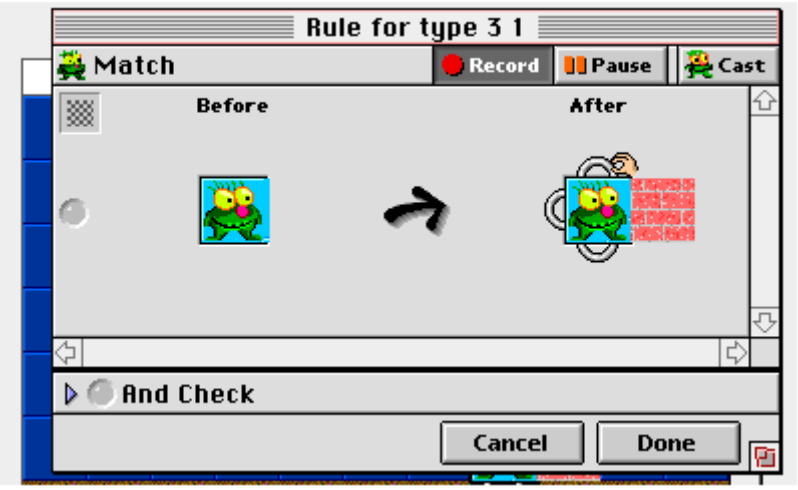


Figura 35: Cocoa™ - programação por exemplo

Até aqui não estamos lidando com nenhum tipo de programação por parte dos usuários. Apesar de alguns mecanismos exigirem o conhecimento de termos e funções específicas do sistema ou domínio, o usuário não precisa lidar com códigos e linguagem de programação.

A última classificação envolve a **redação** de códigos. A redação pode utilizar uma linguagem de script, linguagem específica da aplicação de domínio ou linguagem de propósito geral. Separamos em dois grupos os mecanismos de redação, pois os três primeiros tipos de linguagem estão bem relacionados enquanto que os de propósito geral agrupam mecanismos com características diferentes das primeiras.

O Hypercard™ é uma ferramenta que auxilia o usuário final a criar programas que utilizam linguagens de script. Ele utiliza variáveis especiais como, *it*, *me*, *next*, *previous*, e *this*, associando as variáveis a objetos. Por exemplo, o ***Me*** é usado para se referir ao objeto que contém o *handler* em execução.

Figura 36: Hypercard™ (Hypercard, 1993) – janela de diálogo para perguntas

Na Figura 36 qualquer que seja o valor digitado pelo usuário no campo será armazenado na variável **It**, que é o destino para o resultado de um subconjunto dos comandos da linguagem. O HyperCard tem uma linguagem de script com base a língua inglesa e outra chamada HyperTalk, uma linguagem de alto nível.

O Barista (Ko & Myers, 2006) é um editor de códigos Java, uma linguagem de propósito geral. O sistema possibilita a criação de classes de maneira altamente visual e interativa. Ele apóia a construção de códigos através da edição de textos, seleção de menus e arrastando elementos da linguagem para dentro do código. No exemplo mostrado na Figura 37, o usuário está selecionando a opção [for] do menu *pop-up*.

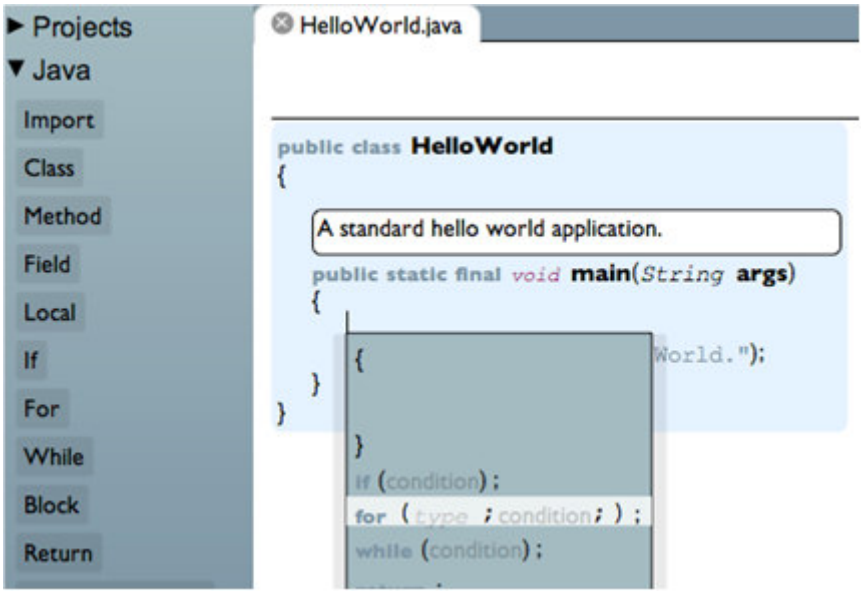


Figura 37: Barista (Ko & Myers, 2006) - auxílio à escrita de códigos

A Figura 38 mostra a edição de uma macro, realizada no Microsoft® Word® 2003.

Primeiro, o usuário criou uma macro, gravando a interação em sua forma usual. A macro transformava o texto selecionado para o formato negrito (Selection.Font.Bold = wdToogle, como no destaque). Então, o usuário mudou de idéia, e gostaria que o texto fosse transformado em itálico, ao invés do negrito.

Para isto, ele entrou no modo de edição de macro do sistema e alterou o valor correspondente (Selection.Font.Italic = wdToogle).

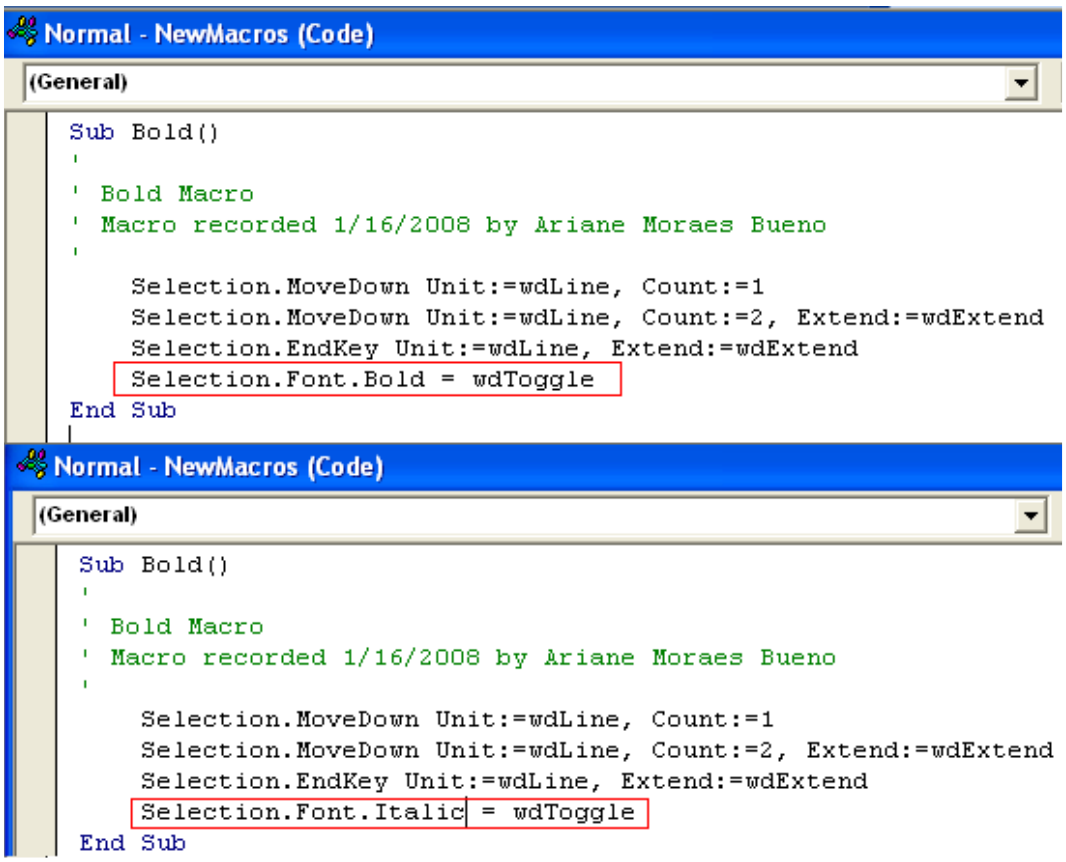


Figura 38: Edição de uma macro no MS® Word 2003

A aprendizagem por desvelamento proposta por Silva está além da classificação proposta. Identificamos este mecanismo como a relação entre os mecanismos do tipo programação por demonstração e os de escrita de código, como as linguagens de script e, por este motivo, não foi incluída na tabela.

Optamos por não inserir nesta relação a flexibilização, proposta por Trigg (1987). Muitas vezes os mecanismos de flexibilização são confundidos com funcionalidades do sistema.

A Tabela 8 apresenta a classificação unificada relacionando-a com as classificações apresentadas no capítulo 2.

Classificação proposta	Trigg (1987)	Fischer e Gingersohn (1990)	Cypher (1993)	Mørch (1997)	Silva (2001)		Blackwell (2006)
Parametrização	Parametrização	Definição de parâmetros (como em um diálogo de opções)	Preferências	Customização	Programação paramétrica	Personalização da aparência da interface Configuração de preferências	
Composição declarativa (integração - building blocks de parte da metamensagem ou de novos sistemas, criação "degenerada" de atalhos)	Integração	n/a	n/a	Integração	n/a		
	Tailoring	n/a	n/a	Extensão	Programação paramétrica	Uso de moldes	
							Spreadsheets (por não precisar saber o que uma função faz)
Composição procedimental concreta			Gravadores de macros		Programação imitativa	Gravação de macros	
Composição procedimental generalizada			Programação por demonstração			Programação por demonstração ou exemplo	Programação por demonstração ou exemplo
							Reescrita gráfica de sistemas
Redação (escrita de código) utilizando linguagem de script, linguagem específica da aplicação ou linguagem de domínio. A linguagem pode ser textual ou visual. Pode-se apoiar a redação com uma interface via menus ou formulários.							Linguagens visuais
			Linguagem de script		Programação descritiva	Roteiros	Scripting
Redação (escrita de código) utilizando linguagem de propósito geral			n/a			n/a	

Tabela 8: Classificação unificada das técnicas e mecanismos de extensão

efeitos sobre / procedimentos	valores (tipo 0)	léxico	sintático	léxico + sintático	léxico + semântico	léxico + sintático + semântico
		- "renaming" -"aliasing" (criação de atalhos)				
parametrização	- personalização da aparência da interface (Silva, 2001) - configuração de preferências (Silva, 2001)		- reordenação dos itens de interface - visibilidade dos elementos de interface - visibilidade do conteúdo/ informações	- integração/ eliminação interna (instalação ou desinstalação de funcionalidades já disponíveis no sistema)		
composição declarativa			- Automator - reordenação de operações ou componentes (de uma macro) - Photoshop	- PbD: Cocoa™	- estilos de formatação de texto - criação/ edição de moldes (+ parametrização)	- integração/ eliminação externa (componentes)
composição procedimental concreta				- gravação de macros (+ parametrização) - Photoshop (record macro / insert menu item)		
composição procedimental generalizada			-PbD: Eager (Cypher, 1991) "Anônimo"	- PbD Mondrian (Lieberman)		
redação - linguagem especifica						Hypercard™ (Hypercard, 1993)
redação - linguagem geral						Barista (Ko & Myers, 2006) (+ composição procedimental)

Tabela 9: Tabela comparativa dos diversos efeitos relacionados com os procedimentos

A Tabela 9 faz um cruzamento entre os efeitos (o que é adaptado) e os procedimentos (como a adaptação é feita).

Nada adianta incluir no sistema adaptações se os usuários não sabem como utilizá-las e, no pior caso, perderem mais tempo adaptando do que realizando a tarefa do jeito que estão acostumados a fazer ou do jeito que aprenderam como fazer. É necessário deixar claro para os usuários como eles devem realizar as adaptações, quais são exatamente os passos a serem seguidos, quais os benefícios essas adaptações irão trazer e, principalmente, como eles podem desfazer o que fizeram. O ideal seria que eles pudessem testar as adaptações antes da efetiva finalização, como um preview.

3.1.4. Em função de quê o usuário realiza a adaptação?

Há diversos fatores que podem ser influenciadores da adaptação. Levando em conta os diversos tipos de usuários, precisamos definir suas **preferências**, as **características físicas** e **culturais**, seus **interesses** e suas **necessidades**. Cada um desses pontos podem influenciar uma proposta de design que leva a uma extensão do sistema. Além disto, as **ferramentas ou os sistemas similares** que os usuários utilizam (ou já utilizaram), também pode ser um fator relevante para a escolha da adaptação. Este ponto nos informa sobre a familiaridade do usuário com algum mecanismo de extensão existente.

As diferentes **estratégias** utilizadas pelos usuários, como o uso de mouse vs. teclado ou interação top-down vs. bottom-up, por exemplo, é outro fator no qual a adaptação pode estar relacionada. Além disto, a curva de aprendizado, ou seja, o **perfil de conhecimento do usuário** que varia de novatos a experientes pode ser exemplificado como outro fator de relacionamento.

Por outro lado, uma adaptação pode ser relacionada à necessidade de obter (ou visualizar) informações diferentes de acordo com a **intenção ou conhecimento prévio do usuário** sobre um determinado objeto (ou tarefa). Por exemplo, um determinado arquivo possui um conjunto de informações associadas. Um usuário sabe *a priori* uma determinada informação sobre um arquivo, mas desconhece as outras. O sistema fornece diferentes visualizações para cada tipo de informação, ou seja, sendo um arquivo de imagem, o usuário pode visualizar um *preview* da imagem. Desta forma, se ele desconhece o nome do arquivo, mas sabe

qual é a imagem que ele deseja procurar, ele pode manipular o conjunto de arquivos de acordo com a informação desejada. Mais precisamente, o sistema pode adaptar a visualização dos diferentes tipos de arquivos de acordo com a intenção do usuário.

Fatores relacionados às tarefas também podem ser associados a adaptação. A **freqüência de realização de uma tarefa** pode levar o usuário a criar atalhos, por exemplo. Ainda sobre a freqüência, podemos ter diferentes variações. Podemos associar poucas variações a criação de macros, variações médias a programação por demonstração e muitas variações a redação de código.

Quando uma **tarefa possui diversas decomposições possíveis**, onde não existe uma forma natural, ou seja, única ou clara de realização, pode ser necessário disponibilizar meios de adaptação ou ainda flexibilização.

O usuário pode ter diferentes tipos de **foco** sobre uma tarefa, ou sobre as ferramentas e conteúdos de um sistema, ou até mesmo sobre o domínio do sistema.

Com relação ao domínio, podemos ter **tipos de conteúdo diferentes** e, nesse caso, devemos apoiar a percepção do conteúdo. Além disto, podemos ter **variações na cardinalidade do conteúdo**, onde a adaptação da ordenação, da formatação ou da visualização pode ser necessária.

3.1.5. Ao quê a adaptação está vinculada? Até quando a adaptação fica em vigor?

Uma adaptação pode possuir um vínculo **temporal**. Uma adaptação pode não persistir, ou seja, estar vinculada a uma única ocorrência de uma operação; pode persistir apenas durante uma sessão, ou ainda entre sessões (disponível da próxima vez em que o sistema é executado). O vínculo também pode ser a um **usuário** ou grupo de usuários, a um **dispositivo**, a um **arquivo** ou ainda a outros **fatores contextuais**, como por exemplo, o local em que o usuário se encontra ou a rede à qual está conectado.

O Microsoft® Office Word 2007, quando o usuário usa um estilo de *bullets*, o sistema modifica uma lista que contém os estilos de *bullets* usados no documento, os da biblioteca e o estilo que está em uso atualmente. A modificação desta lista é feita pela observação do sistema sobre o que o usuário está fazendo.

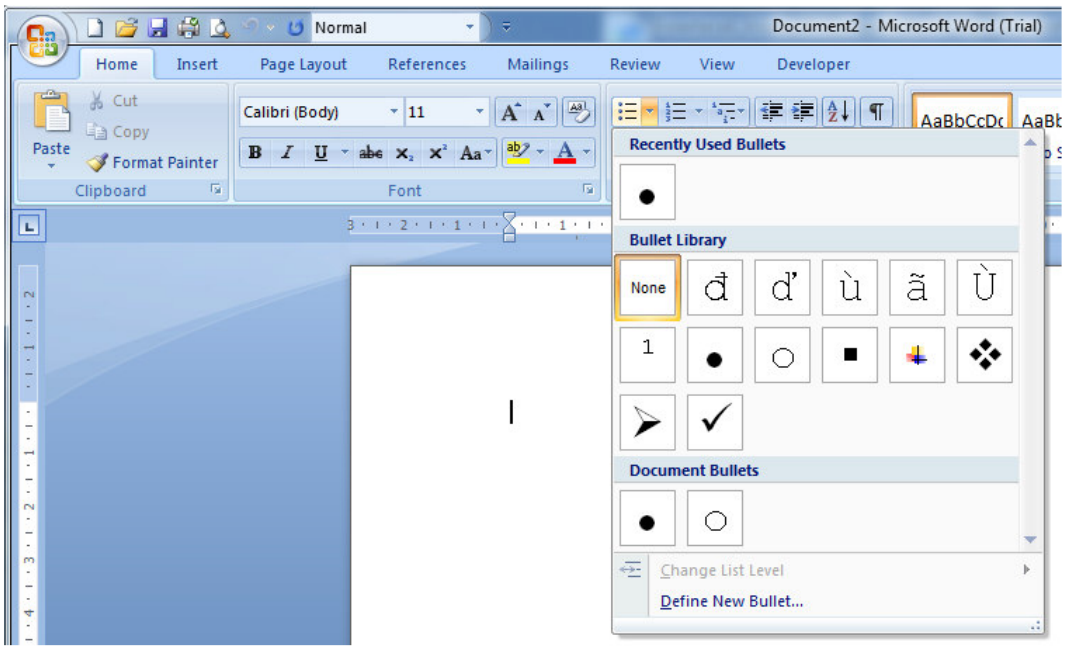


Figura 39: Lista de *bullets* do Microsoft® Office Word 2007

A lista de *bullets* apresentada na Figura 39 possui um número limitado de espaços destinados a novos *bullets* que serão usados. Na medida em que todo o espaço for tomado, o primeiro *bullet* usado recentemente será retirado deste conjunto, dando lugar ao novo *bullet* usado recentemente. A lista estará sempre presente, mas o conteúdo dela será modificado com o tempo e uso do sistema.

Uma adaptação também pode estar vinculada a outros elementos. Podemos ter adaptações relacionadas aos usuários, quando uma adaptação fica disponível ao usuário toda vez que ele se identifica no sistema, independente de dispositivo de acesso.

Podemos ter o caso em que a adaptação está relacionada ao tipo de documento ou artefato em que o usuário está trabalhando. Por exemplo, os estilos definidos pelo usuário no Microsoft Word® ficam atrelados ao documento onde foram definidos.

O mesmo ocorre para os dispositivos. Um sistema desenvolvido para rodar em diferentes dispositivos deve estar preparado para sugerir adaptações que sejam válidas quando ocorre a mudança dos mesmos. Por exemplo, um *browser* pode ser desenvolvido para rodar em um computador de mesa ou em um telefone celular. Seria interessante que algumas adaptações pudessem ser realizadas em ambos os dispositivos. A escolha da página principal é um exemplo deste tipo de adaptação, e esta escolha pode ser a mesma para os dois dispositivos. Porém, pode ser que

uma adaptação para um celular não sirva para um computador. Um exemplo disto é a definição de não mostrar imagens das páginas navegadas pelo celular para poupar tempo e quantidade de *bytes* carregados. Por outro lado, em um computador de mesa, usando internet banda larga, esta intenção perde totalmente o sentido e esta adaptação já não é mais válida.

Há ainda diversos outros fatores a que adaptações podem estar vinculadas, como informações de contexto do uso de um sistema. Como último exemplo, vamos citar a diferença no local de uso de um sistema. Um usuário que trabalhe em casa ou no trabalho poderia definir em seu notebook uma impressora default diferente para cada local.

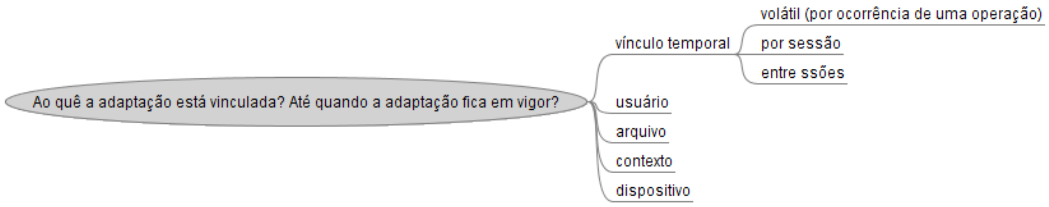


Figura 40: Ao quê a adaptação está vinculada? Até quando a adaptação fica em vigor?

3.2. Ponte entre Análise e Design de Mecanismos de Extensão

A escolha e definição das adaptações podem ocorrer nas diversas etapas do desenvolvimento do software. De maneira geral, elas podem ocorrer na fase de levantamento e análise de requisitos, na fase de modelagem, no design, na implementação e no próprio uso. Sem contar, é claro, quando é feita uma análise sobre um sistema existente para propor modificações de funcionalidades extensíveis.

Na fase de análise de requisitos, é feito um estudo sobre quem são os usuários. É inevitável que os usuários possuam expectativas e necessidades diferentes, e a partir das definições dessas necessidades, juntamente com suas características, será possível determinar quais funcionalidades poderiam ser incluídas no grupo de objetivos do sistema de maneira a auxiliar esses usuários.

A fase de modelagem deve capturar *insights* não só de um sistema específico, mas sobre o domínio do problema. A visualização da interação pode fornecer pistas sobre a localização das extensões, assim como irregularidades com possíveis sugestões.

A fase de design é uma das mais importantes, onde as decisões sobre as extensões podem ocorrer de forma significativa. Isto ocorre porque nesta fase há a transformação dos problemas em soluções de design. Algumas formas de adaptação se enquadram na parametrização ou configuração de preferências. A escolha dos itens de interface, componentes de menu e funcionalidades são profundamente discutidos nesta etapa, e podem gerar sugestões para possíveis adaptações. Além disto, uma alternativa de design pode levar a um futuro componente de customização.

Na fase da implementação, a escolha da linguagem de programação é primordial para a definição da possibilidade de construir as idéias de extensões. Ou seja, nesta fase os designers poderão avaliar se as soluções de extensão propostas até aqui poderão de fato serem desenvolvidas.

A fase de uso, quando o usuário usa o sistema e efetivamente executa as extensões pode fornecer informações sobre a eficácia das soluções propostas. Nesta fase é possível avaliar se os usuários entenderam o sistema e aprenderam como ampliá-lo para o uso.

A partir das perguntas de Hackos & Redish (1998) apresentadas na Seção 2.3.2, compilamos um conjunto de perguntas que não se limitam a decisões de IHC, mas que podem auxiliar o designer na tomada de decisão sobre quais mecanismos de extensão seria interessante incluir no sistema sendo projetado.

Primeiro definimos um conjunto de fatores que englobam perguntas de natureza semelhante. Para cada fator, criamos um fluxograma para guiar o designer de acordo com algumas respostas das perguntas da análise selecionadas. Ao final de cada caminho do fluxograma, identificamos uma tupla <*quem, o que, como , em função de que, ao quê/até quando*>, que corresponde a um mecanismo de extensão mapeado àquela pergunta, tal como encontrado em nossos estudos. Eventualmente não definimos os itens "ao quê/até quando", nos casos em que essa decisão deve ser tomada depois da escolha do mecanismo. Nesses casos, representamos a ausência de definição como "___". Além disto, algumas vezes o mecanismos não possui um item "em função de quê" atrelado, representado por um sinal "-" em nosso fluxograma.

A partir desta tupla, exemplificamos um possível mecanismo de extensão retirado dos sistemas analisados. Cada exemplo é identificado com um número e a lista de exemplos encontra-se no final desta seção.

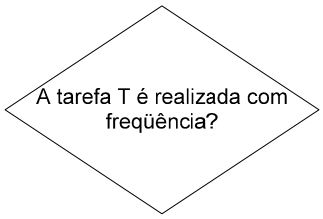
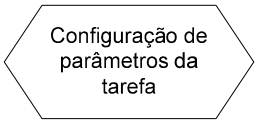
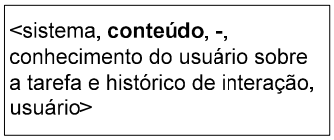
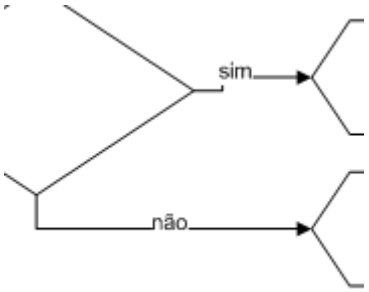
Representação	Explicação
	Este símbolo indica o início de um fluxograma, referenciado por um fator chave. No exemplo, este fator é "sobre a freqüência das tarefas".
	Este símbolo representa uma pergunta derivada do conjunto de perguntas da análise de usuários e tarefas. Esta pergunta indica uma reflexão a ser feita com relação ao sistema.
	Este símbolo indica uma consideração a ser feita com relação a pergunta relacionada.
	Este símbolo indica o final de um caminho do fluxograma, caracterizando um mecanismo de adaptação, através da tupla <*quem, o que, como , em função de que, ao quê/até quando*>.
	Possíveis caminhos a serem seguidos no fluxograma: "sim" e "não".
	Número indicativo do exemplo que se encontra na lista de exemplos do final desta seção.

Tabela 10: Listagem de representações e suas funções no fluxograma

Com relação às características físicas e culturais dos usuários, bem como suas preferências, selecionamos as seguintes perguntas:

- O projeto considera questões de acessibilidade (em particular, deficiências visuais)?
- Os usuários expressaram preferências no que tange a cores, fontes etc.?
- Existem diferenças ou preferências lingüísticas ou jargão entre os usuários?

O fluxograma a seguir mapeia essas perguntas em algumas considerações e nos mecanismos de extensão correspondentes (Figura 41).

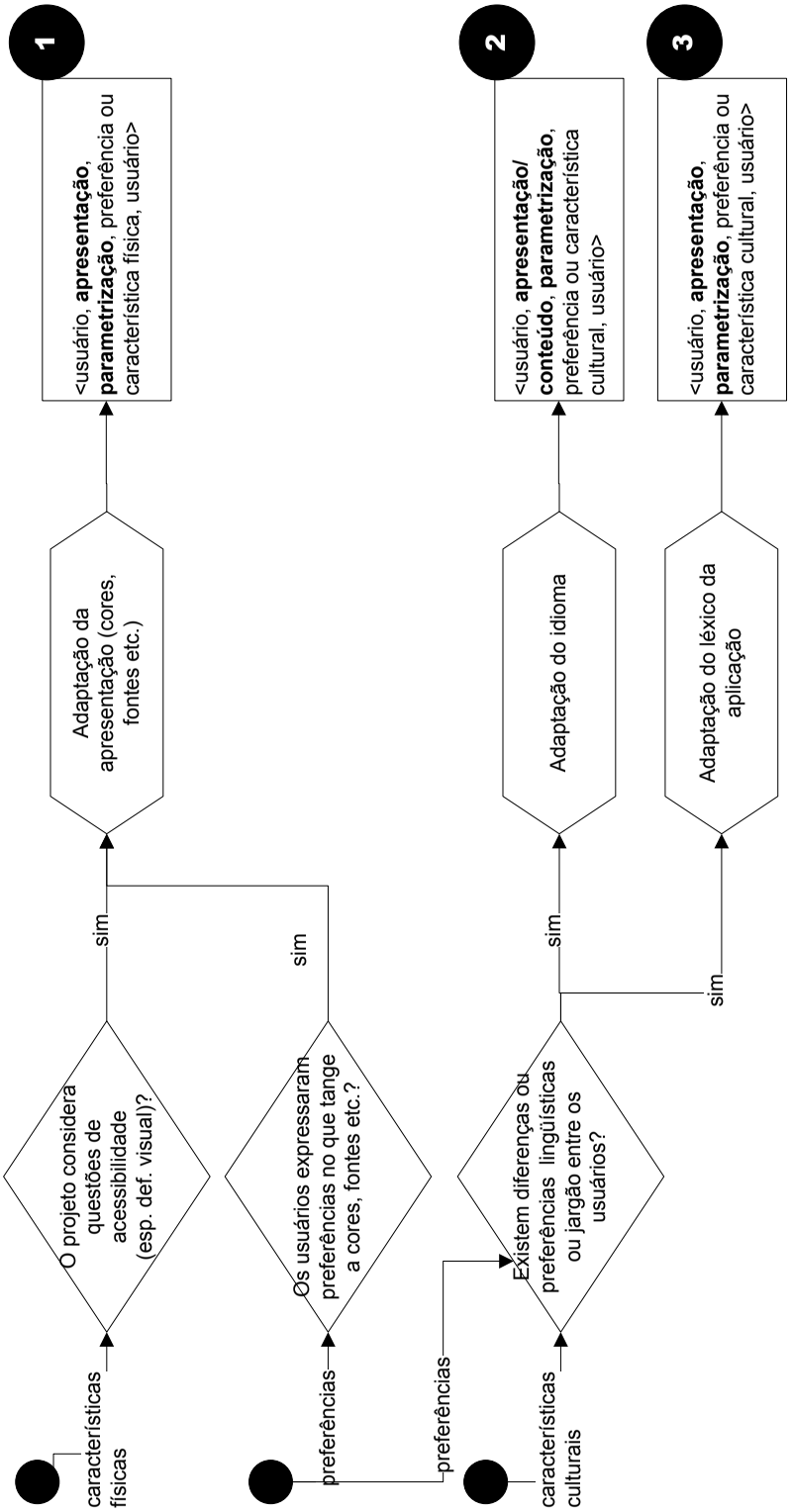


Figura 41: Mapeamento - características físicas, preferências e características culturais.

Com relação ao uso das ferramentas, as perguntas selecionadas foram:

- O usuário tem familiaridade com ferramentas semelhantes?

- As ferramentas possuem mecanismos de extensão?
- Ele deve se tornar um especialista nas tarefas com a ferramenta?
- O usuário utiliza esses mecanismos?

O fluxograma a seguir mapeia essas perguntas em algumas considerações e nos mecanismos de extensão correspondentes (Figura 42):

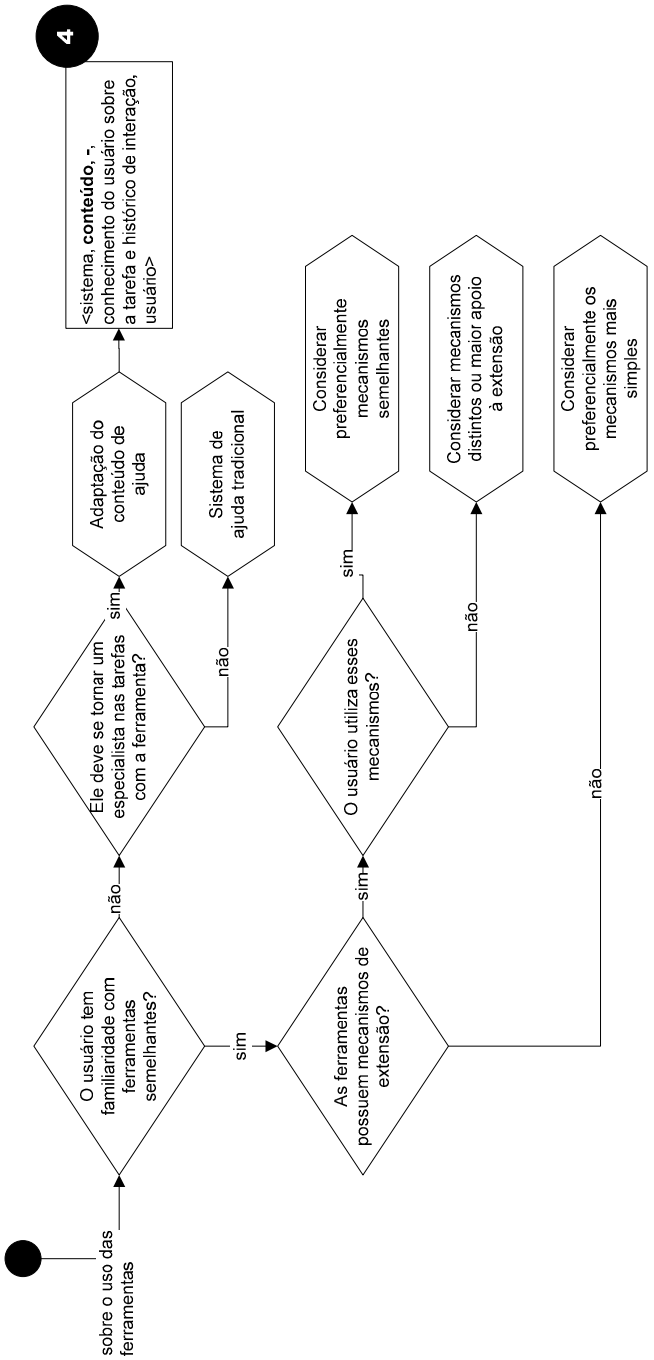


Figura 42: Mapeamento – uso das ferramentas

Algumas questões estão fortemente relacionadas aos conjuntos de tarefas que cada papel de usuário deve poder realizar no sistema:

- Uma tarefa T possui uma forma única de ser realizada?
- É possível mapear cada forma de realizar T a um grupo de usuários?

O fluxograma a seguir mapeia essas perguntas em algumas considerações e nos mecanismos de extensão correspondentes (Figura 43):

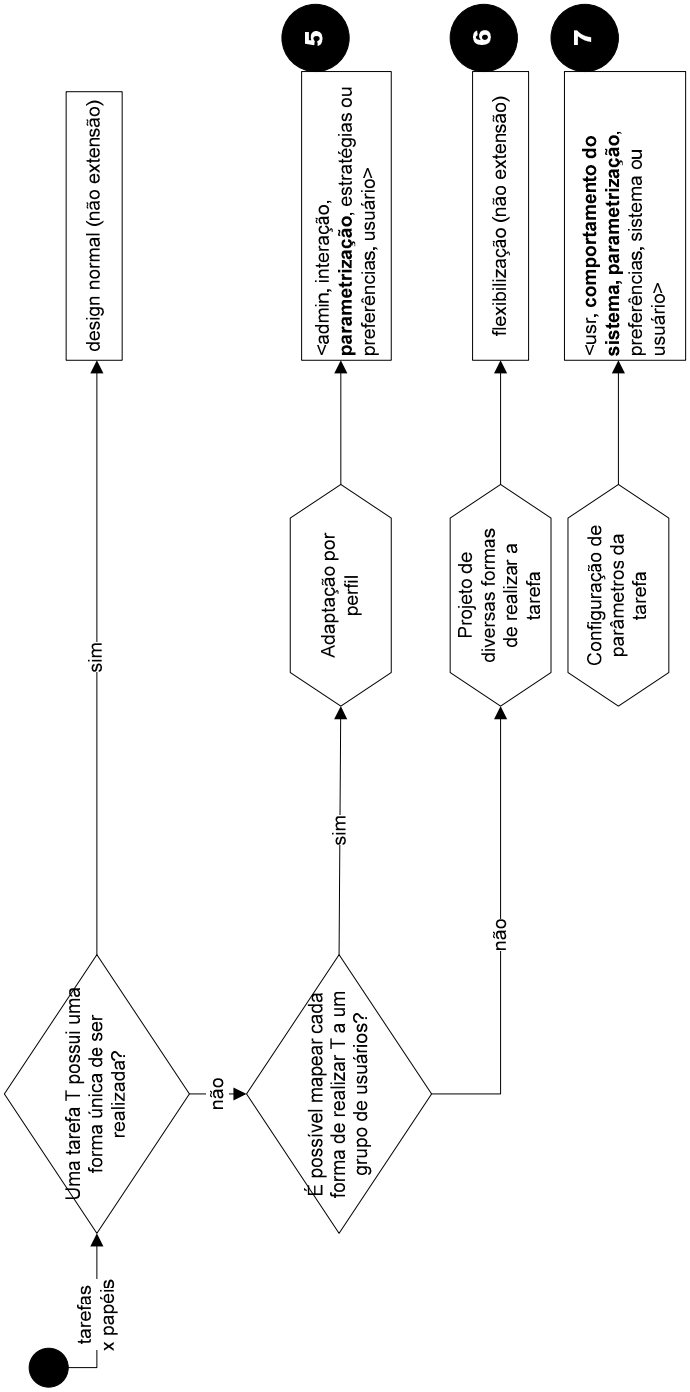


Figura 43: Mapeamento – tarefas vs. papéis

Além do conjunto de tarefas por papel, alguns usuários têm objetivos diferentes para atingir no sistema, dentre os quais podem selecionar aqueles que desejarem. Duas perguntas que investigam esse ponto são:

- Todos os usuários precisam de toda funcionalidade?
- O conjunto de funcionalidades desejadas é bem conhecido (e fechado)?

Como resposta a estas perguntas, pode-se definir mecanismos de extensão por composição declarativa, permitindo integrar componentes interna (tipicamente através da configuração de instalação da aplicação) ou externamente (tipicamente através da incorporação de *plug-ins*), como ilustrado na Figura 44.

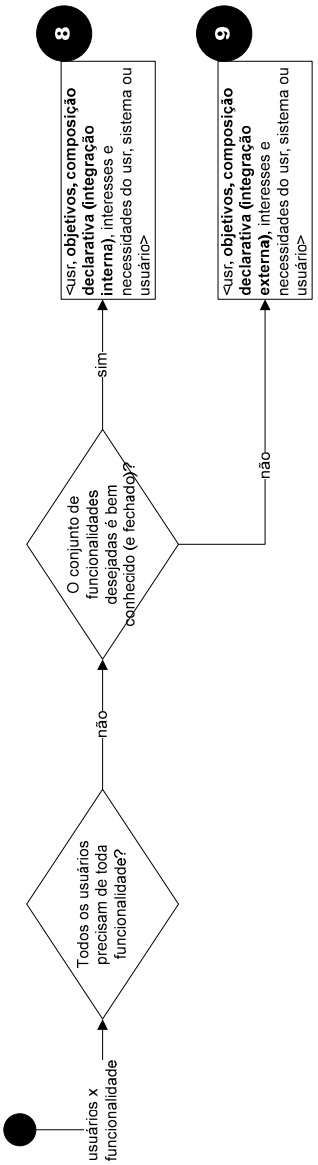


Figura 44: Mapeamento - usuários vs. funcionalidade

Com relação à freqüência em que uma tarefa é realizada, surgem diversas perguntas:

- A tarefa T é realizada com freqüência?
- Já existe um elemento de interface correspondente à tarefa?
- As operações que compõem T são claramente identificadas por signos (e.g. nome)?
- Quanta variação existe na realização da tarefa?

A partir dessas perguntas, foram encontrados os mapeamentos ilustrados na Figura 45:

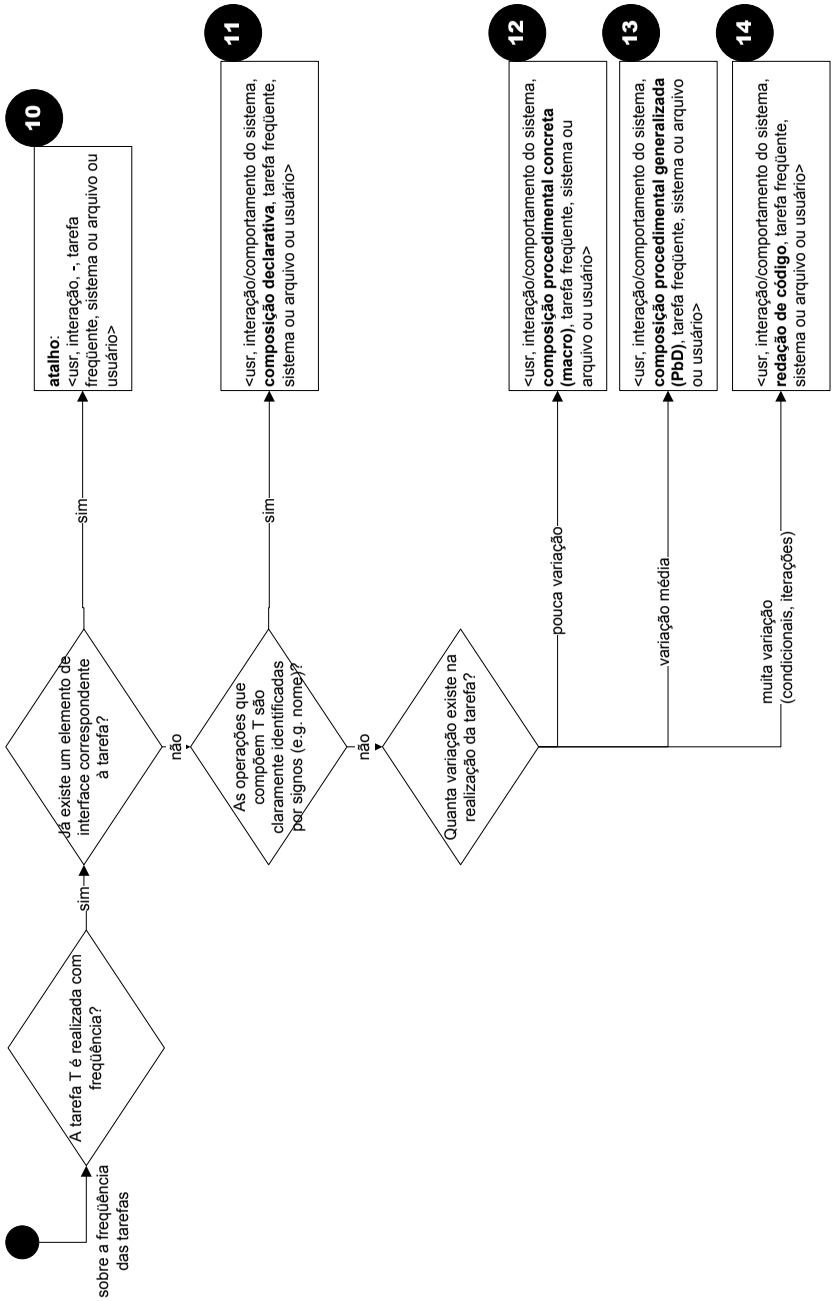


Figura 45: Mapeamento - freqüência das tarefas

Algumas extensões estão ligadas não apenas às tarefas em si, mas também aos conceitos e informações do domínio, que podem ser analisados utilizando as seguintes perguntas:

- Várias tarefas utilizam um mesmo conjunto de informações?
- Todas as informações do conjunto são homogêneas?
- Uma tarefa se beneficia de um subconjunto específico dessas informações?
- Uma tarefa se beneficia de uma perspectiva específica dessas informações?
- O sistema é capaz de diferenciar os conteúdos?
- Várias tarefas utilizam um mesmo conjunto de operações?
- Uma tarefa se beneficia de um subconjunto específico dessas operações?

A partir dessas perguntas, foram encontrados os mapeamentos ilustrados na Figura 46:

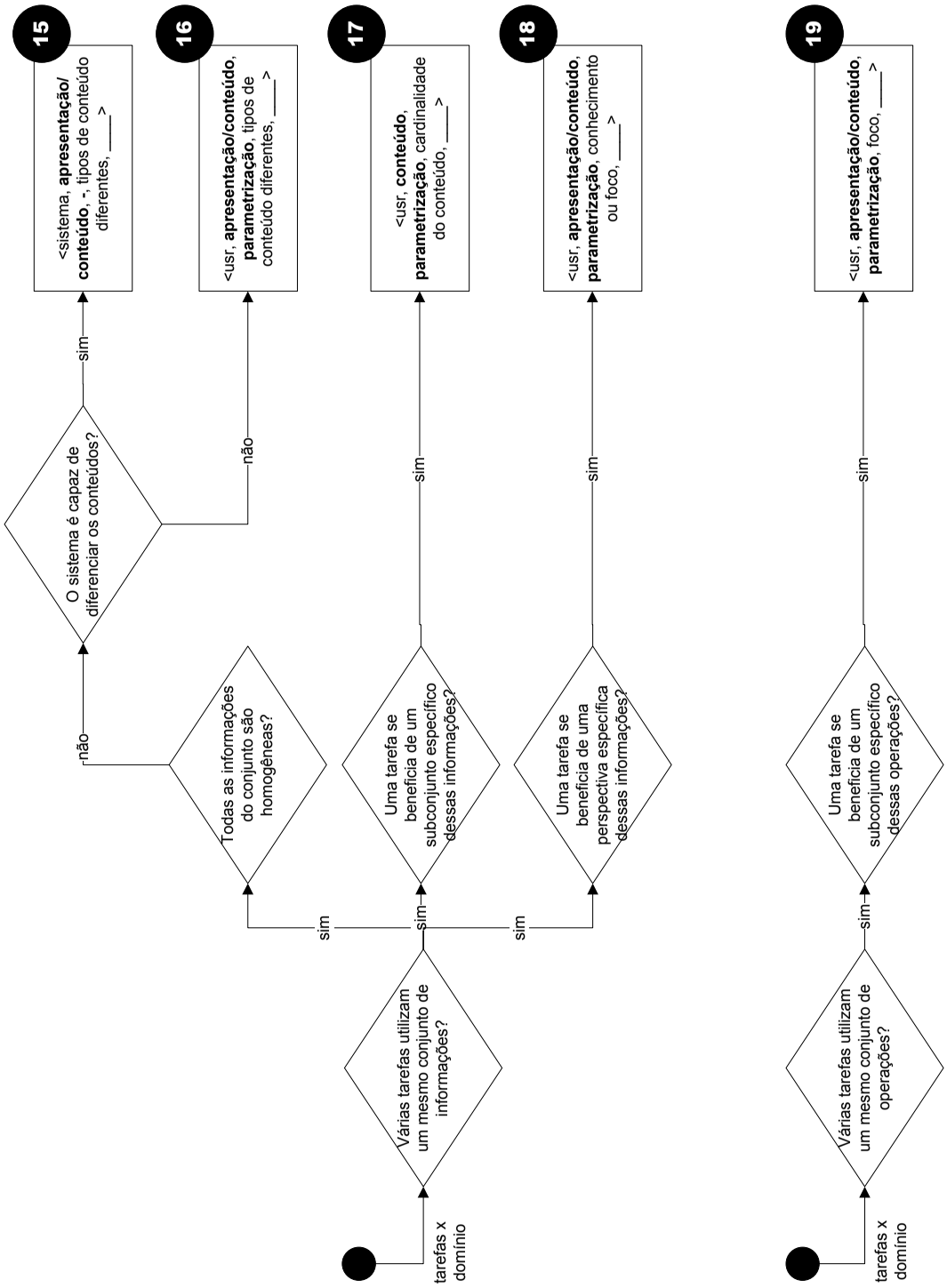


Figura 46: Mapeamento – tarefas vs. domínio

Lista de exemplos

Esta seção apresenta exemplos de extensão para cada item mapeado na seção anterior.

1. Adaptação da apresentação (cores, fontes etc)

O SquirrelMail permite que o usuário modifique a combinação de cor das telas e fontes do sistema. O designer fornece um conjunto de temas para a escolha do usuário.

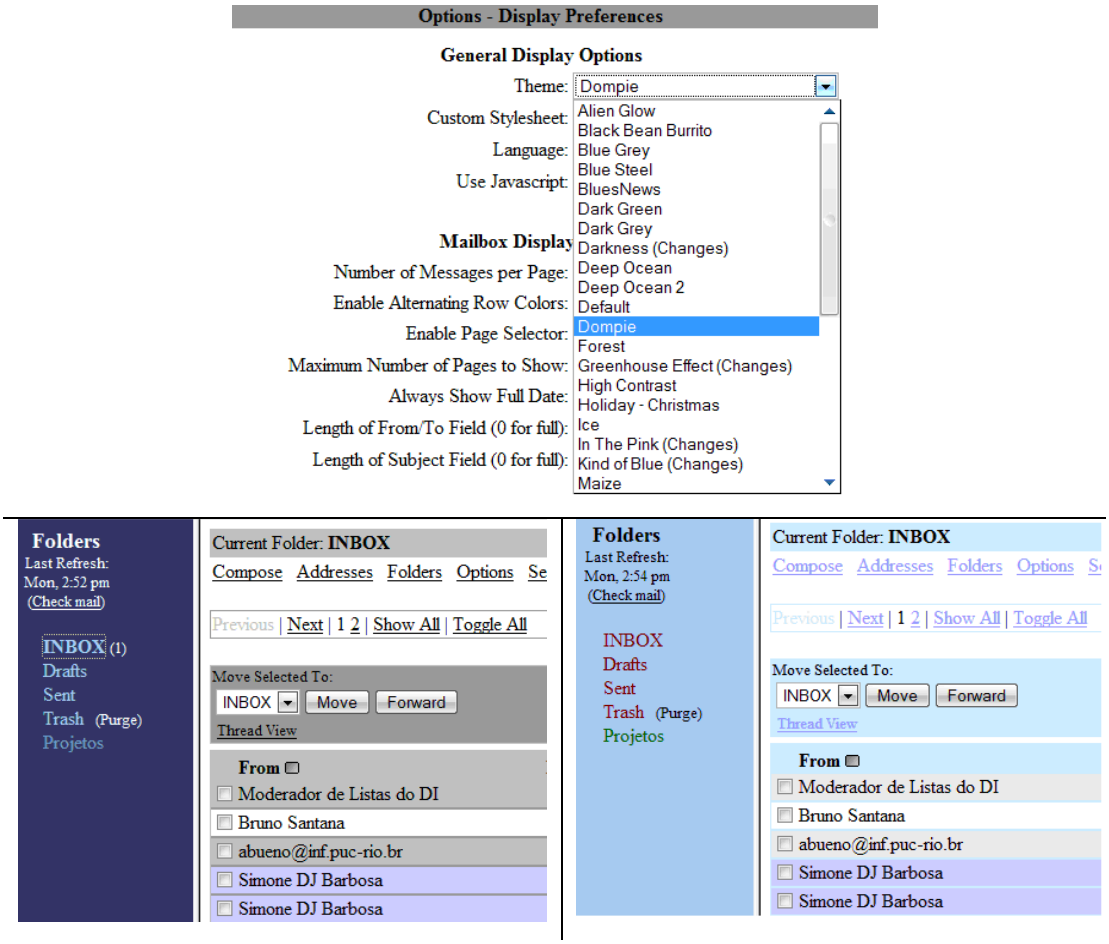


Figura 47: SquirrelMail - adaptação da apresentação

2. Adaptação do idioma

No Microsoft® Office® 2007 o usuário pode escolher o idioma dos menus e diálogos. As versões anteriores não traziam esta funcionalidade e, caso o usuário precisasse que a aplicação suportasse outro idioma, ele deveria instalar outra versão específica do idioma desejado.

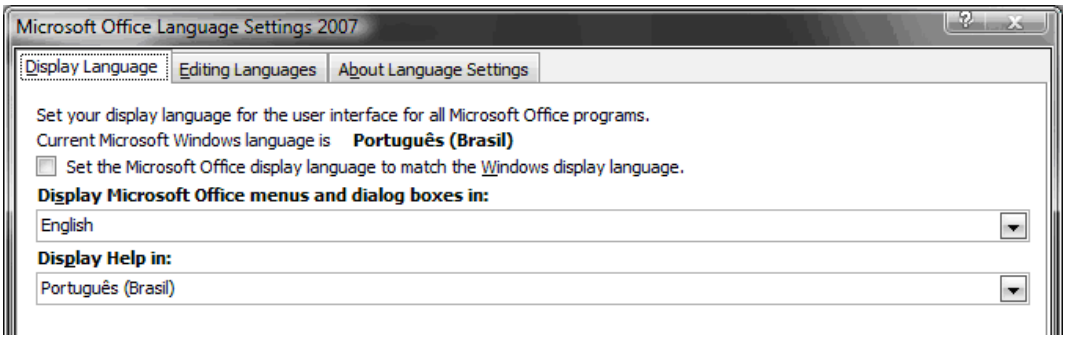


Figura 48: Adaptação do idioma - Microsoft® Office® 2007

3. Adaptação do léxico da aplicação

No Microsoft® Office® 2007, o usuário pode definir quais ferramentas ele deseja adicionar ou remover da barra de ferramentas de acesso rápido.

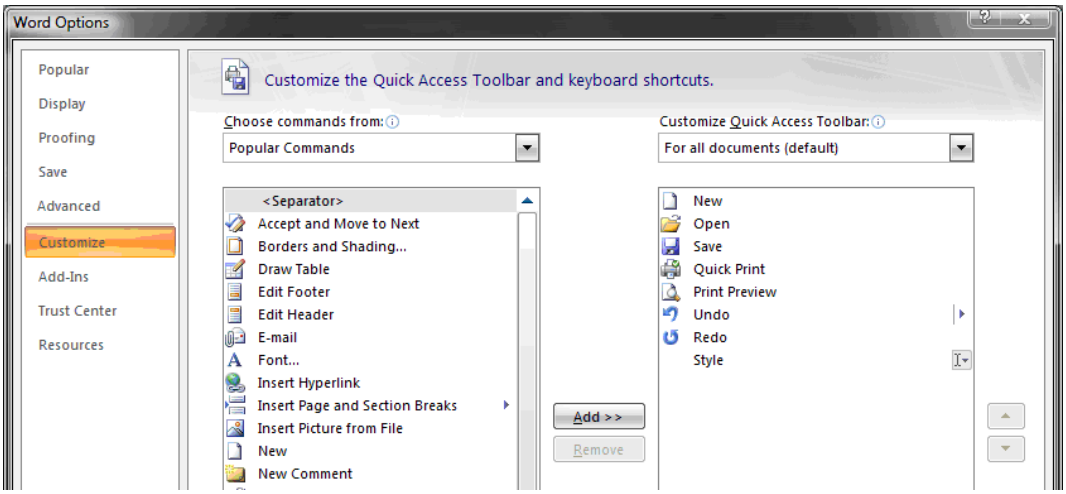


Figura 49: Adaptação do léxico da aplicação - Microsoft® Office® 2007

4. Adaptaçao de conteúdo de ajuda

O conteúdo de ajuda apresentado ao usuário pode ser diferente com base no perfil do usuário (e.g. novato ou experiente). Esse conteúdo pode ser selecionado a partir de conteúdos predefinidos ou gerados pela própria aplicação.

5. Adaptação por perfil

Para a tarefa de criação de contas de e-mail, podemos ter o caso em que o administrador pode criar várias contas através de script (criação do tipo *batch*) e um usuário não administrador que cria uma conta de cada vez utilizando a interface de formulários (GUI).

6. Projeto de diversas formas de realizar uma tarefa (flexibilização, não extensão)

Na Seção 2.1.3, Figura 15, mostramos o exemplo de criação de *playlists* do iTunes, que pode ser feita de três formas diferentes. 1) através do menu File> New Playlist (ou sua variação – atalho ctrl + N); 2) selecionando as músicas e depois através do menu File> New Playlist from Selection; e 3) através do menu File> New Smart Playlist, para selecionar uma opção de busca e o conteúdo referente a ela. Mais uma vez destacamos que este é um exemplo de flexibilização e não se relaciona com uma extensão do sistema.

7. Configuração de parâmetros da tarefa

No Microsoft® Office® 2007, o usuário pode definir várias opções com relação ao armazenamento automático. No exemplo a seguir, ele pode definir o caminho para os arquivos de trabalho e a freqüência em que são armazenados.

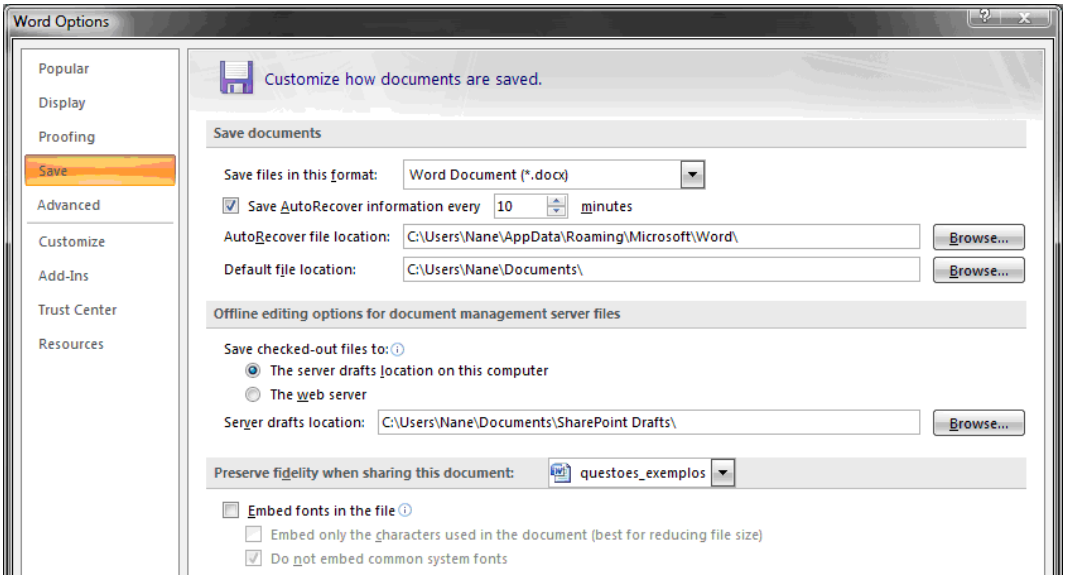


Figura 50: Configuração de parâmetros da tarefa.

8. Integração interna

Instalação do pacote Microsoft® Office® 2003 (Figura 32). Ele possui diversas ferramentas como o Word®, Excel®, Power Point®, entre outros, onde o usuário pode escolher quais ferramentas ele deseja instalar. Este exemplo foi explicado anteriormente na Seção 3.1.3.

9. Integração externa

Instalação de componentens do Mozilla Firefox (Figura 31). O usuário pode incluir funcionalidades novas no sistema, estendendo seu objetivo. Este exemplo foi explicado anteriormente na Seção 3.1.3.

10. Criação de atalhos

O Microsoft® Office® 2007 permite que o usuário modifique as teclas de atalho das funções do sistema ou crie novas para funções que não possuem atalhos relacionados.

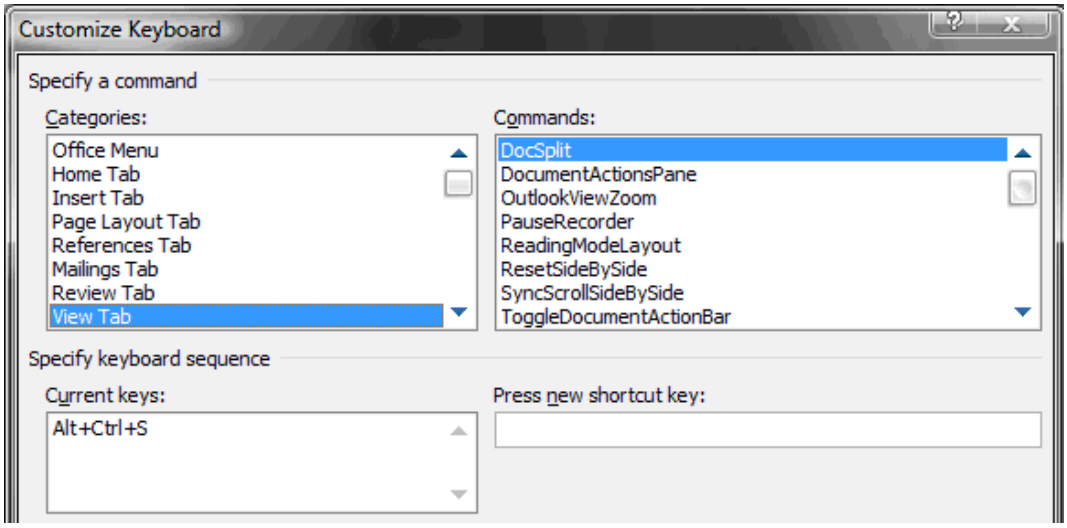


Figura 51: Criação de atalhos no Microsoft® Office® 2007

11. Operações claramente identificadas por signos

O exemplo do Automator (Figura 33) apresentado na 3.1.3 pode ser mapeado como um sistema que fornece operações identificadas por signos. Uma lista de operações é fornecida para que o usuário componha uma ou mais, formando um *workflow* que resultará na execução de uma tarefa.

12. Tarefa freqüente com pouca variação

Um exemplo deste tipo de tarefa é a gravação de macros, já descrita anteriormente no decorrer deste trabalho.

13. Tarefa freqüente com variação média (programação por demonstração)

Um exemplo deste tipo de tarefa é a programação por demonstração, já descrita anteriormente no decorrer deste trabalho.

14. Tarefa freqüente com muita variação (redação de código)

O Barista é um editor de códigos Java, uma linguagem de propósito geral (Figura 37). O sistema possibilita a criação de classes de maneira altamente visual e interativa. Ele apóia a construção de códigos através da edição de textos, seleção de menus e arrastando elementos da linguagem para dentro do código. Este exemplo foi explicado anteriormente na Seção 3.1.3.

15. O sistema é capaz de diferenciar os conteúdos

A busca por um item no Google pode retornar diversos tipos de dados. Cada tipo de dado pode ser apresentado de uma forma diferente. Por exemplo, a busca pelas palavras "santa geração" retornou um link para um site de cifras e um link para um vídeo do *YouTube*. No segundo caso, apareceu um *preview* do vídeo (Figura 52).

Figura 52: Adaptação do conteúdo conforme tipo de informação.

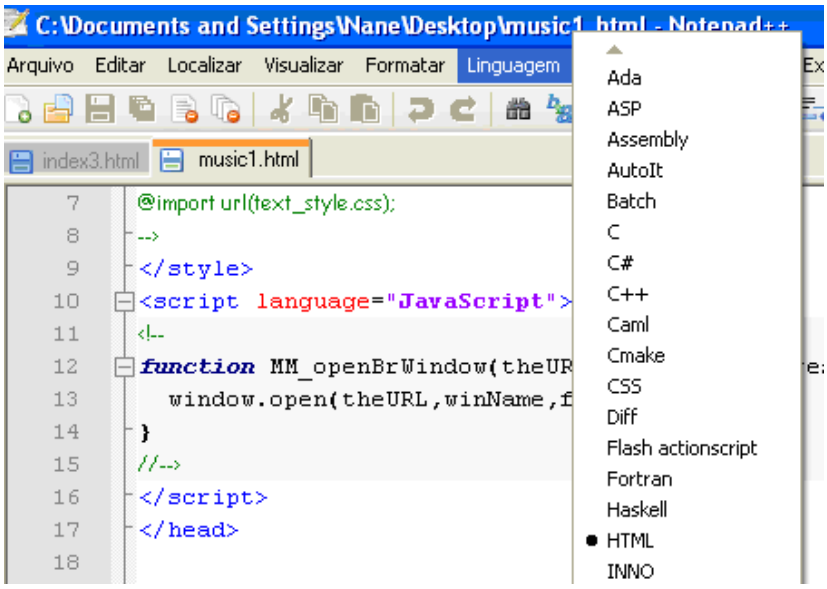


Figura 53: Adaptação da formatação do conteúdo conforme tipo de informação.

Um outro exemplo de adaptação conforme o tipo da informação encontra-se no Notepad++. Nesse sistema, o usuário pode escolher qual a linguagem está relacionada com o arquivo que está sendo usado e, de acordo com a linguagem selecionada, a coloração do código fonte muda (Figura 53).

16. O sistema NÃO é capaz de diferenciar os conteúdos

No Microsoft® Outlook® 2007, o usuário pode escolher como o texto da mensagem deve aparecer de acordo com as seguintes tarefas: 1) escrever novo e-mail; 2) responder e-mail e/ou repassar; ou 3) criar novo e-mail utilizando texto simples (sem html).

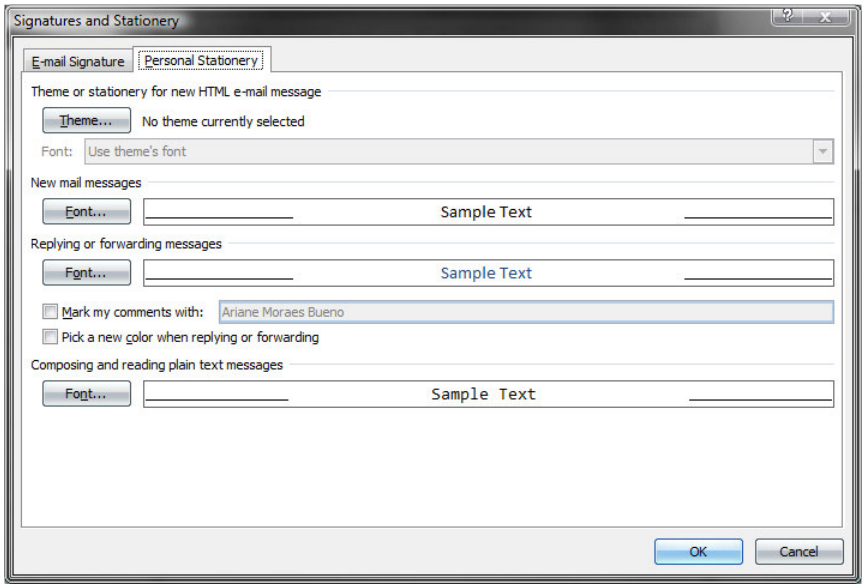


Figura 54: Microsoft® Outlook® 2007 – Criação de estilos diferentes para novos e-mails, e-mails respondidos ou encaminhados ou e-mails de texto simples.

Outro exemplo pode ser encontrado no Microsoft® Word® 2003, quando utiliza marcas de revisão. Ele pode selecionar como deseja ver as diferentes marcas. Na figura a seguir, o usuário definiu que as inserções são apresentadas com o estilo sublinhado e receberá a cor da fonte de acordo com o autor. Cada inserção de cada autor receberá uma cor diferente. As linhas modificadas receberão destaque através de uma borda no canto esquerdo da linha (Figura 55).

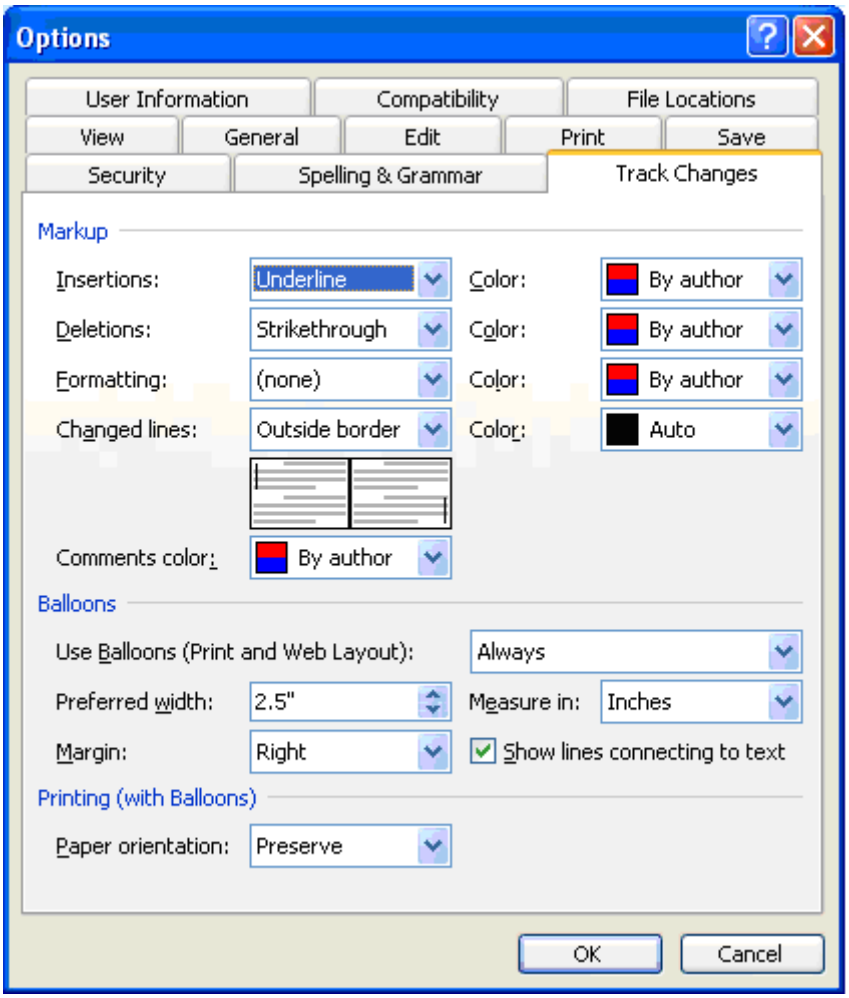


Figura 55: Adaptação conforme preferências do usuário.

17. Subconjunto específico de informações

No Thunderbird o usuário pode fazer uma busca e salvar o resultado da busca dentro de uma pasta que pode ser criada no momento da busca (Figura 56).

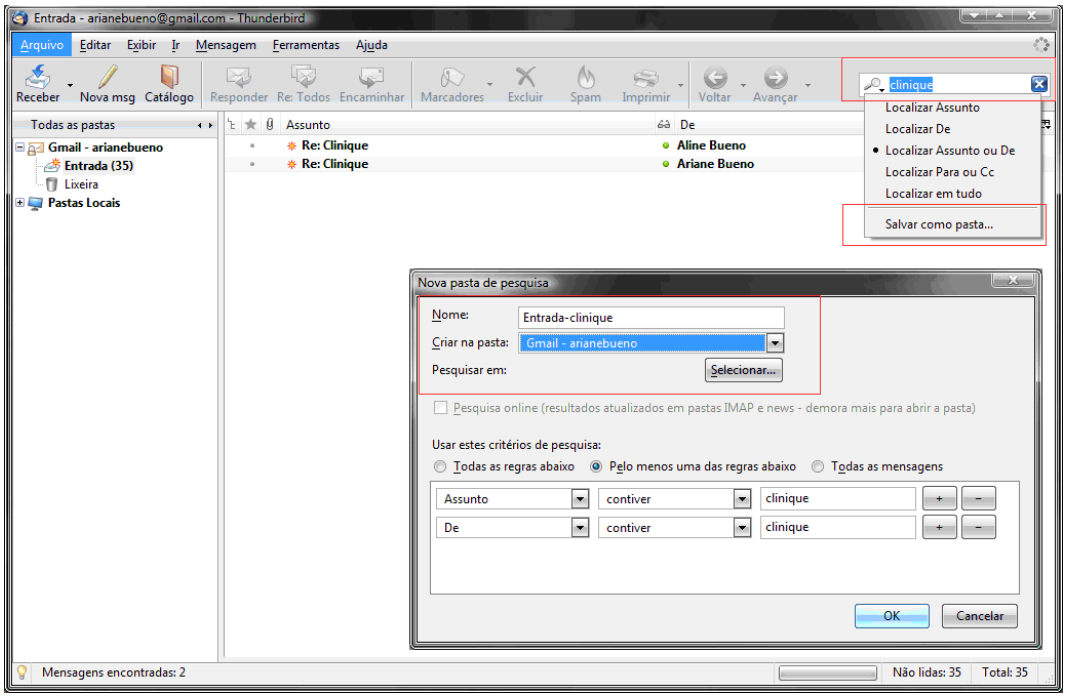
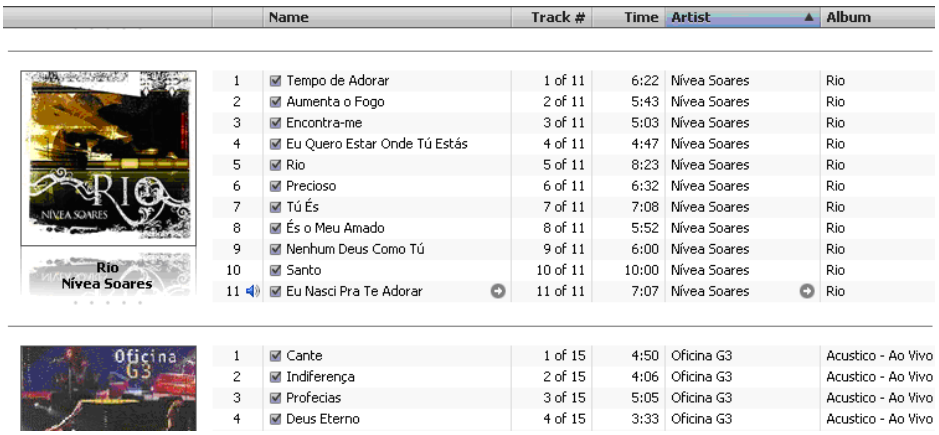


Figura 56: Adaptação do conteúdo.

No iTunes, há três formas de visualização do conteúdo: ver todas as listas de músicas (Figura 57), visualizar a lista de álbuns (Figura 58) ou visualizar um álbum de cada vez (Figura 59). O default inicial é ver as listas de músicas. No entanto, se o usuário modifica a visualização e fecha o sistema, na próxima vez que o abrir, a última visualização utilizada é recuperada.

Name	Track #	Time	Artist	Album	Grouping	Genre	Ratir
Eu Nasci Pra Te Adorar	11 of 11	7:07	Nívea Soares	Rio		Gospel	
Cante	1 of 15	4:50	Oficina G3	Acustico - Ao Vivo		Gospel	
Indiferença	2 of 15	4:06	Oficina G3	Acustico - Ao Vivo		Gospel	
Profecias	3 of 15	5:05	Oficina G3	Acustico - Ao Vivo		Gospel	
Deus Eterno	4 of 15	3:33	Oficina G3	Acustico - Ao Vivo		Gospel	
Quem	5 of 15	4:04	Oficina G3	Acustico - Ao Vivo		Gospel	
Magia Alguma	6 of 15	5:21	Oficina G3	Acustico - Ao Vivo		Gospel	
Mi Pastor	7 of 15	5:13	Oficina G3	Acustico - Ao Vivo		Gospel	
Pirou	8 of 15	3:48	Oficina G3	Acustico - Ao Vivo		Gospel	

Figura 57: Visualização por listas de músicas.



	Name	Track #	Time	Artist	Album
1	Tempo de Adorar	1 of 11	6:22	Nívea Soares	Rio
2	Aumenta o Fogo	2 of 11	5:43	Nívea Soares	Rio
3	Encontra-me	3 of 11	5:03	Nívea Soares	Rio
4	Eu Quero Estar Onde Tú Estás	4 of 11	4:47	Nívea Soares	Rio
5	Rio	5 of 11	8:23	Nívea Soares	Rio
6	Precioso	6 of 11	6:32	Nívea Soares	Rio
7	Tú És	7 of 11	7:08	Nívea Soares	Rio
8	És o Meu Amado	8 of 11	5:52	Nívea Soares	Rio
9	Nenhum Deus Como Tú	9 of 11	6:00	Nívea Soares	Rio
10	Santo	10 of 11	10:00	Nívea Soares	Rio
11	Eu Nasci Pra Te Adorar	11 of 11	7:07	Nívea Soares	Rio
1	Cante	1 of 15	4:50	Oficina G3	Acustico - Ao Vivo
2	Indiferença	2 of 15	4:06	Oficina G3	Acustico - Ao Vivo
3	Profecias	3 of 15	5:05	Oficina G3	Acustico - Ao Vivo
4	Deus Eterno	4 of 15	3:33	Oficina G3	Acustico - Ao Vivo

Figura 58: Visualização por lista de álbuns.

Name	Track #	Time	Artist	Album	Grouping	Genre	Ratir
Abertura	1 of 11	1:26	Diante do Trono	Águas Purificadoras		Gospel	
Tempo de Festa	2 of 11	5:45	Diante do Trono	Águas Purificadoras		Gospel	
Eu Quero o Avivamento	3 of 11	6:05	Diante do Trono	Águas Purificadoras		Gospel	
A Vitória da Cruz	4 of 11	8:02	Diante do Trono	Águas Purificadoras		Gospel	
Quando Deus Escolhe Alguém	5 of 11	4:31	Diante do Trono	Águas Purificadoras		Gospel	
À Sombra do Altíssimo	6 of 11	7:53	Diante do Trono	Águas Purificadoras		Gospel	
O Cheiro das Águas	7 of 11	6:16	Diante do Trono	Águas Purificadoras		Gospel	
Águas Purificadoras	8 of 11	16:06	Diante do Trono	Águas Purificadoras		Gospel	
O Bálsamo de Gileade	9 of 11	6:24	Diante do Trono	Águas Purificadoras		Gospel	

Figura 59: Visualização um álbum de cada vez.

18. Perspectiva específica de informações

Esse tipo de adaptação também pode ser feito automaticamente pelo sistema, como é o caso da formatação das pastas de arquivos de imagem do Windows®, que são apresentadas com *preview* das imagens nelas contidas.

No Microsoft® Excel® 2003, o usuário pode definir o formato de uma célula. No exemplo, ele está definido o formato para datas. O sistema oferece diversos tipos para que o usuário possa escolher.

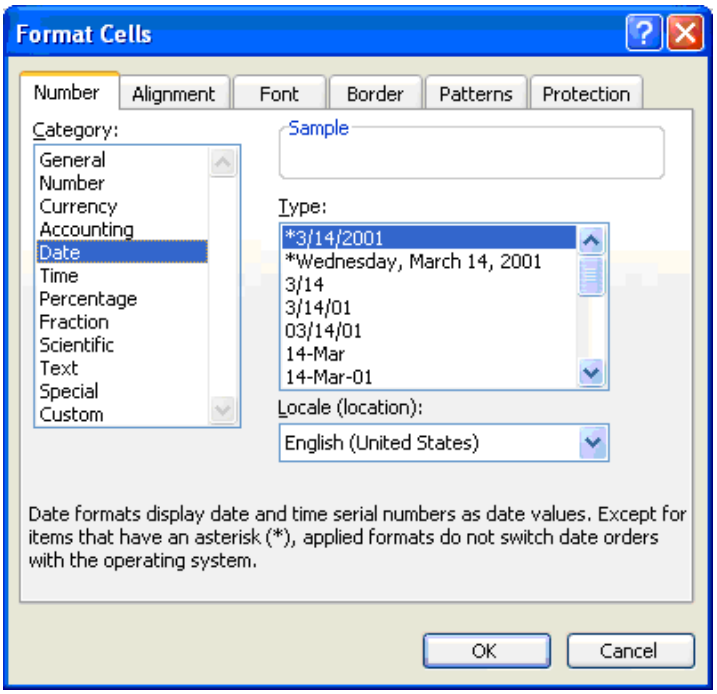


Figura 60: Configuração do formato do dado.

19. Subconjunto específico de operações

No Microsoft Windows®, o usuário pode configurar a ordenação default das pastas de arquivo. Essa configuração pode ser aplicada a uma pasta específica ou a todas as pastas do sistema.

No Adobe Photoshop®, o usuário pode definir quais janelas de configurações/opções devem aparecer. Mesmo abrindo outro documento, as janelas abertas permanecem abertas e posicionadas no mesmo local (Figura 61).

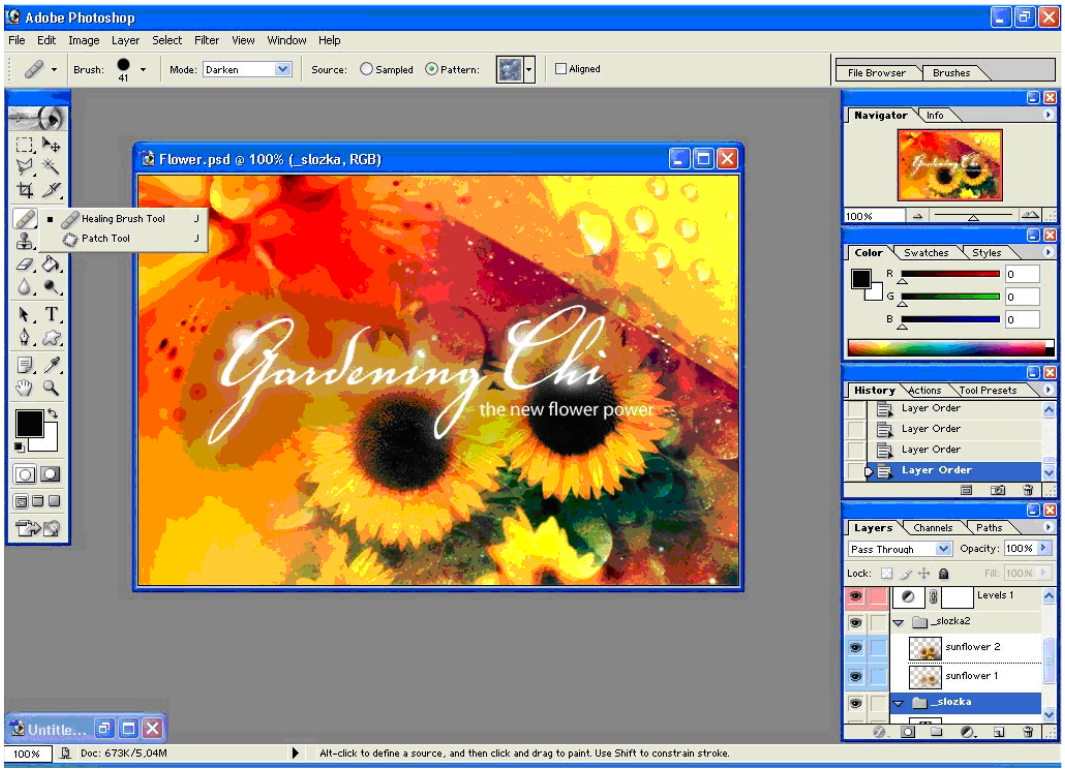


Figura 61: Persistência na exibição de janelas e barras de ferramentas.

No Adobe InDesign CS3®, o usuário pode salvar diferentes *workspaces* com configurações específicas de painéis (habilitação e posicionamento) e menus. O sistema permite que o usuário salve e eventualmente remova um workspace. Sempre que precisar trocar de um workspace para outro ele pode fazer isto através da devida seleção.

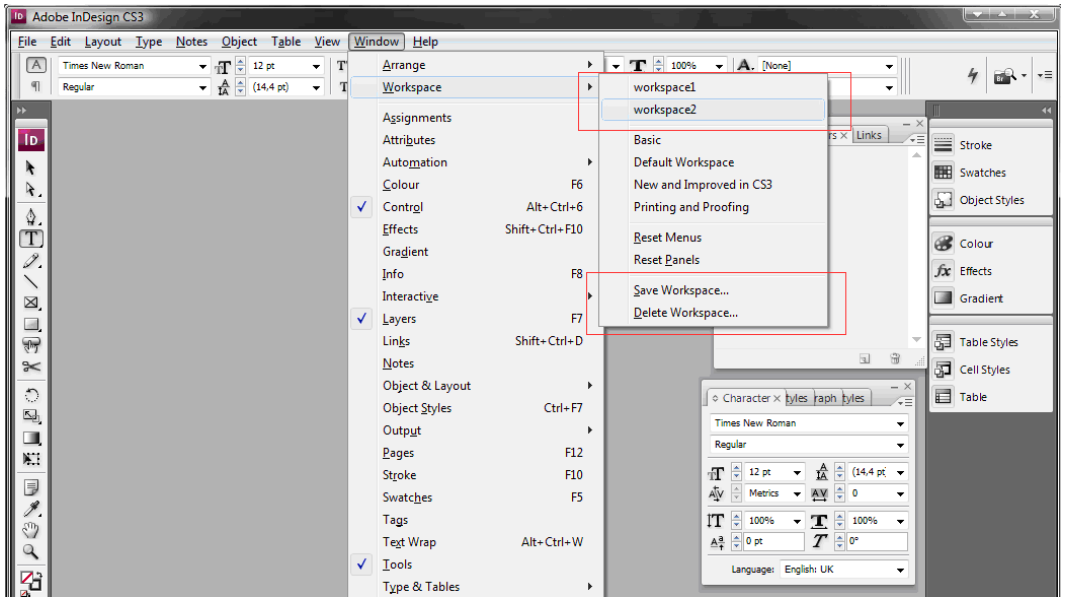


Figura 62: Persistência através de arquivo de configuração.

Um outro exemplo ocorre no Microsoft® Outlook® 2003, que permite que o usuário configure a disposição dos painéis na tela. Ele pode optar por: 1) visualizar o painel de caixa de entrada em uma coluna do lado esquerdo e o painel com o conteúdo da mensagem selecionada no primeiro painel em uma coluna no lado direito [right]; 2) visualizar o painel de caixa de entrada como uma caixa na parte superior e o painel de mensagem como outra caixa na parte inferior [bottom]; ou 3) visualizar apenas o painel da caixa de entrada [off] (Figura 63). Apesar de não haver consenso sobre isto ser ou não uma extensão, esta configuração é mantida entre sessões, independente do *default* inicial.

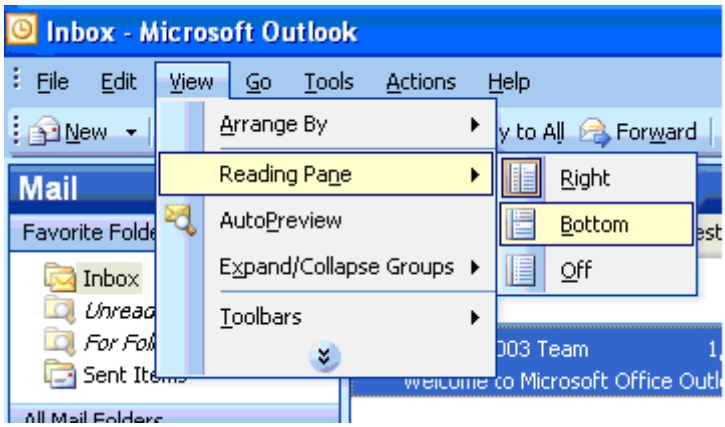


Figura 63: Configuração do leiaute dos painéis do Microsoft Outlook®.

No iGoogle, o usuário pode posicionar os agrupamentos de conteúdo da forma que desejar (Figura 64). Ele pode ainda habilitar ou desabilitar cada porção de conteúdo.

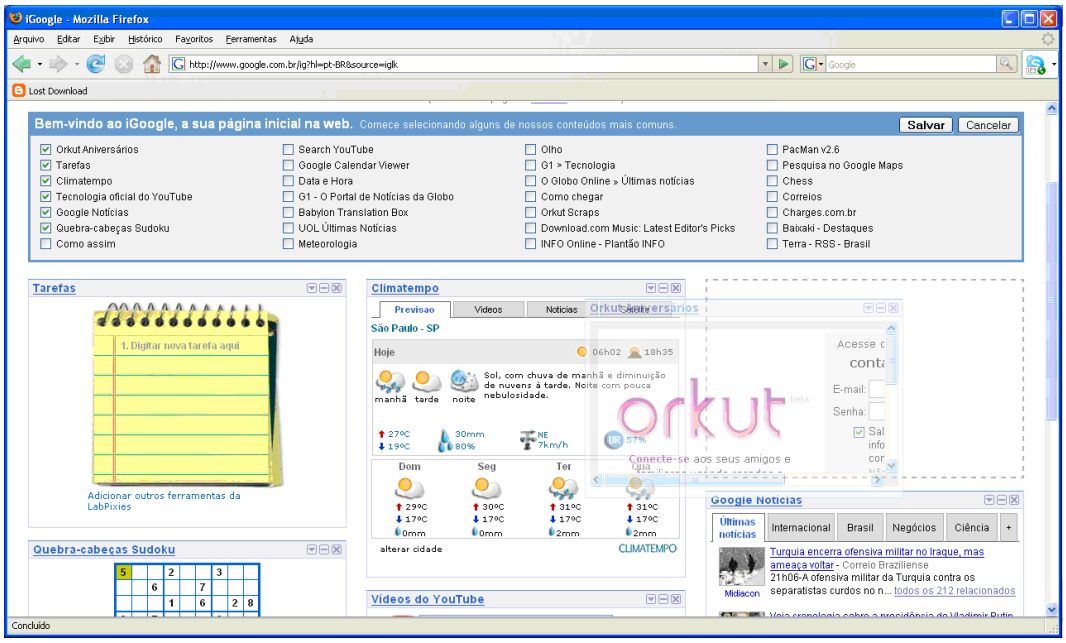


Figura 64: Configuração do leiaute de conteúdo no iGoogle.

É importante observar que o conjunto de perguntas apresentadas nos mapeamentos não pretende ser exaustivo, e que as tuplas que representam extensões não são recomendações diretas conforme as respostas às perguntas, mas sim mapeamentos resultantes dos trabalhos encontrados na literatura. O uso desse mapeamento não substitui um trabalho cuidadoso de design, apenas ajuda a organizar o espaço de design.