

O conteúdo do presente relatório é de única responsabilidade do(s) autor(es).
(The contents of this report are the sole responsibility of the author(s).)

**Avaliação Heurística de um sistema altamente
dependente do domínio**

L. A. S. Romani M. C. C. Baranauskas

Relatório Técnico IC-98-26

Julho de 1998

Avaliação heurística de um sistema altamente dependente do domínio

Luciana A. S. Romani* M. Cecília C. Baranauskas[◊]

Abstract

Some knowledge domains have, among their personnel, a wide spectrum of expertise in the use of computer technology, which embraces from daily users of computational systems to beginners. We argue that specific characteristics of users in the domain should be considered in the design and software evaluation. In this paper we present the use of heuristic evaluation for usability analysis of a domain-dependent system. The system chosen for a case study was the Sistema de Acompanhamento e Avaliação de Rebanhos Leiteiros. This system is used for organizing information about productive and reproductive control of animals. It possesses a large amount of forms for entrance of data, consultations and emission of reports. Results of the heuristic evaluation point to the need for specific category heuristics, to capture specific aspects of the considered domain.

Keywords: Heuristic evaluation, design, human-computer interaction

Sumário

Alguns domínios do conhecimento possuem, entre seus profissionais, um amplo espectro de *expertise* no uso da tecnologia do computador, que abrange desde usuários que utilizam-se de sistemas computacionais em seu trabalho rotineiro, até novatos no uso dessa tecnologia. Argumenta-se que características específicas de usuários no domínio considerado devem ser observadas no *design* e na avaliação de *software*. Neste artigo, é mostrado o uso da avaliação heurística para análise de usabilidade de um sistema altamente dependente de domínio. O sistema escolhido para estudo de caso foi o Sistema de Acompanhamento e Avaliação de Rebanhos Leiteiros (ProLeite), que é usado na organização das informações de desempenho produtivo e reprodutivo dos animais de rebanhos leiteiros. Possui grande quantidade de formulários para entrada de dados, possibilita consultas e emissão de relatórios. A partir dos resultados da avaliação heurística do sistema é mostrada a necessidade de um conjunto de heurísticas de categorias específicas, para captar aspectos específicos do domínio considerado.

Palavras-chave: Avaliação heurística, *design*, interface de sistema de domínio específico, interação humano-computador

* Centro Nacional de Pesquisa Tecnológica em Informática para a Agricultura, Empresa Brasileira de Pesquisa Agropecuária - Caixa Postal 6041 - 13083-970 - Campinas, SP (Brasil) - luciana@cnptia.embrapa.br.

[◊] Instituto de Computação, Universidade Estadual de Campinas - Caixa Postal 6176 - 13083-970 Campinas, SP (Brasil) - cecilia@dcc.unicamp.br.

1. Introdução

Os grandes avanços na tecnologia do computador têm aumentado o poder das máquinas e enriquecido as possibilidades de comunicação entre humanos e computadores. Entretanto, o impacto da tecnologia por si só não é suficiente para tornar os artefatos baseados no computador - em especial o *software* - mais usáveis. Curiosamente, à medida que nossas máquinas tornam-se mais sofisticadas, seu fator de usabilidade - potencial para ser usável pelo humano - tem vindo de aplicações criativas da tecnologia que estendem o “poder” do humano (Dix et al. 1993).

A interação entre o usuário e o computador não acontece no vácuo, mas é afetada tanto pela tecnologia quanto por fatores sociais e organizacionais do contexto em que o usuário está inserido. Do lado tecnológico, paradigmas de interação têm surgido ao longo da própria história da área de *Human Computer Interaction* (HCI) e do desenvolvimento de sistemas interativos: interfaces WIMP (*windows, icons, menus e pointers*), manipulação direta, hipertexto, etc. Do lado dos fatores humanos, a literatura (Norman, 1986; Laurel, 1990; Winograd, 1996; Nardi, 1995) tem apresentado princípios para interação humano-computador que dependem menos da tecnologia e muito mais de um entendimento profundo do elemento humano na interação. Paradigmas por um lado e princípios por outro são dois lados da mesma moeda. Paradigmas de interação, considerados isoladamente não deixam claro como dão suporte ao usuário na realização de suas tarefas. Os princípios de interação, por sua vez, podem ser definições gerais e abstratas não muito úteis ao *designer*, embora a consciência dos fatores humanos possa ajudar a limitar os efeitos negativos do uso da tecnologia na interação.

Apesar da problemática discutida, é senso comum o fato de construirmos sistemas mais usáveis que no passado, embora exista a consciência de que muito se tem a conhecer e fazer, ainda, em termos de *design* de sistemas interativos. Dix et al. (1993) colocam que, o objetivo primeiro de um sistema interativo é permitir ao usuário alcançar metas particulares em algum domínio de aplicação; isto é, o sistema interativo deve ser “usável”. Para tal, o *designer* deve lidar com duas questões em aberto:

*como um sistema interativo pode ser desenvolvido para assegurar essa usabilidade
como a usabilidade do sistema pode ser demonstrada ou medida (Dix et al., 1993, p. 117)*

Sempre que objetos são criados para pessoas usarem - *software* não é diferente - decisões são tomadas e tais objetos são construídos carregando uma intenção do *designer* para com o usuário final, de como esse objeto deveria ser percebido ou usado. A Engenharia de Usabilidade (Nielsen, 1993) encoraja o uso de metas de usabilidade no processo de *design*, como forma de explicitar a intencionalidade do *designer*.

Os princípios básicos de usabilidade envolvem três categorias principais: facilidade com que novos usuários podem efetivamente começar a interagir e alcançar máxima performance; multiplicidade de maneiras com que o usuário e o sistema trocam informação e o nível de suporte que o usuário tem para determinar seu sucesso e a avaliação de suas metas. Embora práticas iterativas de *design* embasadas nesses princípios não garantam a usabilidade do produto, elas favorecem o julgamento de usabilidade dos sistemas interativos. Molich e Nielsen (1990) propuseram um método conhecido como “avaliação heurística”, em resposta à necessidade de efetivamente se “depurar” o design de sistemas interativos. A avaliação heurística não é inteiramente subjetiva, uma vez que, um critério específico é usado para guiar a avaliação. A abordagem é simples e relativamente rápida, embora um certo conhecimento e um certo número de avaliadores sejam necessários para a aplicação das heurísticas de usabilidade.

O usuário de software no domínio agropecuário pertence a um largo espectro de *expertise* no uso da tecnologia de computadores. Inclui desde usuários que utilizam-se de sistemas

computacionais em seu trabalho rotineiro (pesquisadores de empresas agropecuárias, por exemplo) a usuários novatos no uso dessa tecnologia (funcionários de cooperativas, agricultores, pecuaristas, etc.). Características importantes da categoria de usuários no domínio considerado devem ser observadas no *design* e avaliação de software nesse domínio. Como exemplo, pode-se citar a falta de precisão para movimentos leves e precisos, do usuário acostumado a equipamentos pesados e rudes como a enxada, tornando imperativo facilitar a manipulação de alguns elementos de interface.

Neste artigo, tem-se por objetivo aplicar e discutir princípios de usabilidade à análise de um sistema interativo de domínio específico: o Sistema de Acompanhamento e Avaliação de Rebanhos Leiteiros¹ (ProLeite). Este sistema visa organizar as informações de desempenho produtivo e reprodutivo dos animais de rebanhos leiteiros. Mais especificamente, discute-se a aplicação da avaliação heurística ao *design* da interface do sistema e propõe-se um conjunto de heurísticas específicas para a classe de aplicações ilustrada pelo ProLeite, como um suplemento às heurísticas gerais. O artigo é organizado da seguinte forma: na seção 2, é apresentado o referencial teórico do trabalho; na seção 3, é apresentado o estudo de caso em questão: o ProLeite como uma instância de sistemas desenvolvidos para o domínio da agropecuária, metodologia de avaliação e resultados preliminares; na seção 4 discute-se uma possível extensão para o conjunto de heurísticas utilizado e finalmente na seção 5, são apresentadas algumas conclusões.

2. Referencial teórico

Avaliação é uma parte integral do processo de *design* de software e deve ter lugar ao longo do ciclo de vida do *design*. Portanto, ela não deve ser pensada como uma fase única no processo de desenvolvimento do *software*, nem como uma atividade a ser realizada ao final do processo, se o cronograma permitir. Fazendo um paralelo com os novos conceitos de qualidade em manufatura, a avaliação deve ocorrer durante o processo de *design*, com resultados gerando melhorias contínuas e alimentando modificações no produto final - o *software*. Embora o usuário final em seu contexto de trabalho seja a razão primeira de se pensar em qualidade da interface, em geral, não é possível realizar testes experimentais extensivos continuamente durante o *design*.

Conforme Hix (1993), as avaliações da qualidade de interfaces com usuários são de natureza subjetiva e giram em torno de sua usabilidade. Isto é, as avaliações estão associadas às condições oferecidas aos usuários na utilização da funcionalidade da aplicação. Segundo Nielsen (1990), experiências têm demonstrado que muitos desenvolvedores acham caros, difíceis de usar e demorados muitos dos métodos de avaliação de usabilidade. Entretanto, técnicas analíticas e informais podem e devem ser usadas, como por exemplo a avaliação heurística, método proposto por Nielsen e Molich (1990) para superar essa problemática. Também chamado “*discount usability engineering*”, é um método barato, rápido e fácil de usar (Nielsen, 1993). Além disso, a literatura tem mostrado em estudos experimentais comparativos com outras técnicas de avaliação como por exemplo uso de *guidelines*, *walkthroughs* cognitivos e testes de usabilidade, que a avaliação heurística tem apresentado melhores resultados encontrando problemas mais sérios com menos esforço despendido (Jeffries et al. 1991).

¹ Produto em desenvolvimento pelo Centro Nacional de Pesquisa Tecnológica em Informática para a Agricultura (CNPTIA) em parceria com o Centro Nacional de Pesquisa de Gado de Leite (CNPGL) – unidades da Embrapa e a Associação Brasileira de Criadores (ABC), com apoio da Associação de Criadores de Bovinos do Brasil.

A literatura em HCI (Dix et al. 1993; Preece et al. 1994) tem feito uma distinção entre avaliação do *design* de um sistema interativo e avaliação de uma implementação (completa ou protótipo). Enquanto a primeira tem o foco na avaliação pelo *designer*, sem o envolvimento do usuário, a última estuda usos reais do sistema. Discutiremos avaliação sob a primeira ótica, embora algumas técnicas possam ser aplicadas em ambos os casos.

A intenção básica da avaliação é identificar elementos que possam causar dificuldades ao usuário, porque violam princípios cognitivos conhecidos ou ignoram os resultados empíricos já bem aceitos. São 3 as metas principais da avaliação: examinar a funcionalidade do sistema, o efeito da interface no usuário e identificar problemas específicos de *design*. O foco de nosso trabalho é a terceira meta, que relaciona tanto funcionalidade quanto usabilidade da interface. Trata-se de identificar os aspectos negativos do *design*: elementos que, quando usados em seu contexto intencional, causam resultados não esperados ou confusão entre os usuários.

A avaliação heurística é o método analítico de laboratório escolhido neste trabalho, por sua adequabilidade à avaliação contínua do processo, com baixo custo, conforme discutido anteriormente. Envolve especialistas avaliando o *design* com base em um conjunto de critérios de usabilidade ou heurísticas. O *design* é examinado em busca de instâncias nas quais esses critérios são violados. Os critérios de usabilidade são relacionados a princípios e *guidelines* e podem ser selecionados ou derivados deles. Um conjunto de critérios, conforme proposto originalmente por Nielsen e Molich (1990) inclui:

- diálogo simples e natural;
- fala a língua do usuário;
- minimiza a carga cognitiva do usuário;
- consistência;
- feedback;
- saídas marcadas claramente;
- atalhos;
- mensagens de erro precisas e construtivas;
- previne erros;
- ajuda e documentação.

Embora simples e relativamente rápido, o método requer conhecimento do avaliador, para aplicação das heurísticas. Além disso, resultados experimentais mostram que são necessários de 3 a 5 avaliadores para se conseguir um bom resultado. Neste trabalho, investigamos a necessidade de um conjunto específico de heurísticas, dependentes de características do domínio, a ser acrescentado às heurísticas gerais.

3. Estudo de caso: avaliando um sistema altamente dependente do domínio

O método de avaliação heurística tem sido amplamente utilizado em interfaces de sistemas de uso genérico do tipo processadores de texto e sistemas de informação *online*. Entretanto, verifica-se que este método também pode ser aplicado à avaliação de interfaces que são direcionadas para um grupo de usuários especializados. Nesta seção, são apresentadas informações detalhadas sobre o uso da avaliação heurística para detecção de problemas de usabilidade em um sistema de domínio especializado.

3.1 O Sistema de acompanhamento e avaliação de rebanhos leiteiros

Entender os detalhes do ProLeite requer conhecimentos específicos sobre manejo e avaliação de rebanhos leiteiros. No entanto, detalhes muito específicos sobre o sistema não são necessários para o entendimento de sua aplicabilidade geral. Brevemente, o ProLeite provê uma interface gráfica que permite organizar as informações de desempenho produtivo e reprodutivo de animais de rebanhos leiteiros. O sistema emite relatórios com indicadores técnicos de desempenho individual dos animais.

O sistema é útil para instituições que realizam o acompanhamento de rebanhos leiteiros. Além disso, é um instrumento gerencial do produtor/criador para a sua orientação na tomada de decisões no processo de planejamento das atividades de rotina dos sistemas de produção de leite. O sistema possui vários módulos para cadastro de informações sobre os animais, os rebanhos, as associações, os núcleos de criadores, as regiões, as fazendas e os criadores. A partir destas informações cadastrais, um conjunto de eventos pode ser controlado como a classificação linear de tipo, controle leiteiro, análise de leite, lactações em andamento/encerradas, cobertura, diagnóstico de prenhez, eliminação de animais dos rebanhos, encerramento da lactação e parto. O sistema permite a emissão de vários relatórios que servem de subsídio para o manejo e a avaliação dos rebanhos.

A interface do sistema em questão está organizada com base em menus que, em uma tela inicial, organizam e agrupam as tarefas viabilizadas ao usuário. As tarefas propriamente ditas que, em geral referem-se ao cadastramento, gerenciamento de dados e consulta, são viabilizadas por janelas do tipo *form-fill* que contêm um conjunto específico de botões para operação. As figuras 1 e 2 ilustram duas telas do sistema. A tela inicial do sistema mostra um nível do menu de opções com a organização da seqüência de tarefas disponíveis no sistema, como pode ser visto na Figura 1. A Figura 2 apresenta uma das telas de entrada dos dados cujo estilo é usado largamente no ProLeite.



Fig. 1. Tela inicial do ProLeite

Fig. 2. Formulário para entrada de dados sobre a produção leiteira

3.2 Metodologia

O procedimento utilizado para a avaliação heurística da interface do sistema consistiu de quatro fases: uma pré-avaliação, uma avaliação, uma reunião para discussão e uma sessão de classificação criteriosa, durante a qual o observador estimou a gravidade dos problemas de usabilidade encontrados durante as avaliações.

As três primeiras etapas levaram aproximadamente 3 dias. O resultado do trabalho desenvolvido nestas quatro fases foi a elaboração de um relatório com uma lista de problemas de usabilidade encontrados na interface do sistema, o qual foi encaminhado ao grupo de desenvolvimento.

A avaliação do ProLeite foi realizada com a participação de três avaliadores com formação na área de informática, sendo que um deles possui conhecimento sobre o desenvolvimento do sistema. Para o propósito deste estudo, não foi considerado nenhum especialista do domínio (engenheiro agrônomo, veterinário, por exemplo) para julgar se a terminologia utilizada na interface era compreensível e adequada ao domínio específico. Este aspecto não foi contemplado porque o sistema foi desenvolvido com a supervisão de um especialista do domínio, que ficou encarregado de verificar todos os termos usados na interface do sistema durante seu desenvolvimento.

Na avaliação heurística, o avaliador deve possuir um conhecimento mínimo sobre técnicas de *design* de interfaces, o que possibilita que o mesmo possa avaliar e identificar características da interface que não estejam de acordo com as heurísticas. No entanto, no caso de uma aplicação de domínio específico, pode ser necessária a presença de um observador durante a sessão de avaliação, para esclarecimento de eventuais dúvidas dos avaliadores não especialistas no domínio.

Na fase de pré-avaliação, uma explicação verbal foi fornecida aos avaliadores para salientar os aspectos mais significativos do sistema. Em seguida, uma lista com as heurísticas que deveriam ser usadas na avaliação, foi entregue aos avaliadores. Juntamente com as heurísticas, foi entregue um texto explicativo sobre as mesmas e eventuais dúvidas foram esclarecidas pelo observador.

A Tabela 1, a seguir, apresenta o conjunto de heurísticas derivadas de uma análise de problemas de usabilidade desenvolvidas por Nielsen (1993), que foi entregue aos avaliadores:

- | |
|--|
| 1. Estética e design mínimo: Os diálogos não devem conter informações irrelevantes ou que |
|--|

sejam raramente necessárias. Toda unidade de informação extra em um diálogo compete com as unidades relevantes de informação e diminui sua visibilidade.
2. Coerência entre o sistema e o mundo real: O sistema deve falar a língua do usuário, com palavras, frases e conceitos familiares ao usuário, ao invés de termos orientados ao sistema. O sistema deve seguir convenções do ambiente real e fazer as informações aparecerem em uma ordem lógica e natural.
3. Reconhecer é melhor que lembrar: O sistema deve fazer objetos, ações e opções visíveis. O usuário não precisa lembrar informações de uma parte de um diálogo em outro. Instruções para usar o sistema devem ser visíveis ou fáceis de serem recuperadas sempre que forem necessárias.
4. Consistência e padronização: Usuários não devem ter surpresas se diferentes palavras, situações ou ações significam a mesma coisa. O sistema deve seguir uma plataforma de convenções.
5. Visibilidade do status do sistema: O sistema deve sempre manter os usuários informados sobre o que está sendo feito, através de um <i>feedback</i> apropriado dentro de um tempo razoável.
6. Permite ao usuário um controle e liberdade de opções: Usuários frequentemente escolhem funções erradas e necessitam de clareza nas opções de "saídas de emergência" sem ter que atravessar um extenso diálogo. O sistema deve prover <i>undo</i> e <i>redo</i> .
7. Flexibilidade e eficiência de uso: Aceleradores não conhecidos por usuários novatos podem agilizar a interação com usuários experientes. O sistema pode atender as necessidades de ambos usuários inexperientes e experientes. O sistema deve permitir ao usuário ajustar suas ações frequentes.
8. Ajuda os usuários a reconhecer, diagnosticar e recuperar erros: Mensagens de erros devem ser expressas em linguagem plana, sem códigos, indicando o problema precisamente e sugerindo uma solução construtivamente.
9. Previne erros: A presença de mensagens explicativas deve ser um cuidado de <i>design</i> que previne a ocorrência de problemas.
10. Documentação e ajuda: É sempre melhor que o sistema possa ser usado sem documentação, mas ela pode ser necessária. Qualquer informação deve ser facilmente encontrada e enfocar a tarefa do usuário. Uma lista de passos concretos deve ser disponibilizada e não deve ser extensa.

Tabela 1. Lista de heurísticas utilizadas

Após a explicação sobre o sistema e a distribuição da lista de heurísticas para os avaliadores, foi solicitado a eles que usassem o sistema procurando a orientação de uso do mesmo, através dos menus e botões de comando. Como o sistema restringe-se basicamente a entrada de dados, consulta e emissão de relatórios, nenhum cenário específico foi projetado. Apenas houve uma recomendação de que os avaliadores usassem o sistema de forma exploratória tentando averiguar instâncias com problemas de usabilidade. Após a explanação sobre o sistema e esclarecimento de dúvidas, os avaliadores estavam aptos a usá-lo sem a necessidade de auxílio do observador.

Nas sessões de avaliação, cada avaliador conduziu sua avaliação individualmente e independentemente dos outros avaliadores e do observador. Foi solicitado aos avaliadores que não discutissem com os outros até que a última avaliação tivesse sido feita. A cada avaliador foram atribuídas de 2 a 3 horas para realizar sua avaliação.

O observador reforçou a recomendação de que os avaliadores fizessem a avaliação em duas etapas: primeiro, verificando a interface de forma geral, percorrendo a maioria das telas e, em uma

segunda etapa investigando com maiores detalhes os itens que fazem parte da lista de usabilidade entregue a eles no início da sessão.

Com esse conjunto de informações e observações, todos os avaliadores completaram uma sessão de uso do sistema sem grandes dificuldades, apesar da ausência de conhecimento especializado do domínio. Aos avaliadores, foi solicitado que procurassem encontrar tantos problemas de usabilidade quanto fosse possível, incluindo problemas maiores e menores. Além disso, eles deveriam anotar os problemas encontrados para posterior elaboração de um relatório que deveria ser entregue ao observador.

Uma outra atividade dos avaliadores foi a de classificar os problemas de usabilidade detectados dentro dos dez itens da lista de heurísticas entregue a eles no início da sessão. Durante as sessões de avaliação, todas as observações e problemas detectados pelos avaliadores foram anotadas por eles mesmos.

3.3 Resultados preliminares

Após o término da última sessão de avaliação, os avaliadores se reuniram com o observador para discutir os problemas de usabilidade detectados na interface do sistema durante as avaliações. Foram discutidos e enumerados os problemas mais graves que sugerem a necessidade de um *redesign*. Embora o método de avaliação heurística não proponha buscar soluções para os problemas de usabilidade levantados, algumas sugestões foram dadas pelos avaliadores para possíveis soluções de alguns problemas detectados na interface do sistema.

Como resultado desta reunião, ficou a cargo do observador classificar todos os problemas de usabilidade encontrados na interface do ProLeite, de acordo com o seu grau de comprometimento do uso e elaborar um relatório para o grupo de desenvolvimento. Além disso, o relatório também deveria conter sugestões para a solução dos problemas detectados na interface do sistema.

A seguir, são comentados sucintamente os resultados obtidos na avaliação. Os itens com o marcador (+) indicam aspectos positivos do sistema em relação à heurística analisada e os itens com o marcador (-) indicam aspectos negativos.

1. Estética e *design* mínimo

(+) As telas e formulários da aplicação são sóbrios e livres de objetos desnecessários.

(+) A apresentação visual é agradável e facilita a leitura e identificação das funções, dos campos e formatos de entrada e da apresentação de informação.

(-) Na tela de abertura da aplicação, onde é solicitada a senha, a ilustração com figuras e rótulos - “ProLeite” e raças de animais – assemelha-se a uma página de um *site* na *Internet*, podendo levar o usuário a confundir os rótulos com *links* ou opções da aplicação.

2. Coerência entre o Sistema e o Mundo Real

(+) O sistema apresenta-se de forma clara e atual. As palavras e objetivos do programa embora técnicos, podem ser utilizados por usuários conhecedores ou não do domínio específico.

(-) Uso de termos em linguagem estranha ao domínio como por exemplo “eventos” e “registro”.

3. Reconhecer é Melhor que Lembrar.

(+) O sistema apresenta ícones e rótulos de fácil identificação e compreensão. Os menus são separados por temas bem objetivos.

(-) Algumas listas (*listboxes*) utilizadas para selecionar o valor de chaves estrangeiras em determinados formulários (como os códigos de rebanho e proprietário no formulário de animais) quando não acionadas apresentam somente o código e não o nome das entidades relacionadas.

(-) A impossibilidade de visualização de várias janelas ao mesmo tempo dificulta o uso do sistema, pois algum usuário pode querer verificar informações em um formulário enquanto entra dados em outro, por exemplo.

(-) Na opção “Utilitários” do menu principal, as subopções “Cópia de Segurança” e “Restauração da Cópia de Segurança” deveriam estar em posições adjacentes e todas as subopções poderiam ser agrupadas, utilizando-se separadores.

4. Consistência e Padronização.

(+) Percebe-se muita consistência e padronização na nomenclatura utilizada, nas convenções adotadas e mesmo na disposição dos elementos da interface.

(-) A divisão entre “Consulta” e “Relatórios” no menu principal é estranha, uma vez que as consultas sobre lactações e os relatórios presentes na aplicação têm funcionamento análogo.

(-) O sistema apresenta inconsistência em relação às mensagens de erro; estas são apresentadas em idiomas diferentes.

5. Visibilidade do *status* do Sistema

(+) O sistema mostra em cada janela um título na barra superior da janela, informando em qual tarefa o usuário está trabalhando.

(+/-) O sistema orienta parcialmente o usuário sobre a seqüência de operações a serem feitas.

(-) Para qualquer opção apontada no menu principal ou mesmo subopção dos menus abertos a partir do menu principal, permanece a mesma frase na linha de *status*, relativa ao botão da barra de ferramentas mais próximo da opção clicada.

(-) Ao entrar em um formulário, não é permitido alterar os valores dos campos dos registros até que se acione o botão “Alterar”.

6. Permite ao Usuário um controle e liberdade de Opções

(+) O sistema permite que o usuário tenha a possibilidade de cancelar uma inserção ou alteração ainda não confirmada através do botão cancelar.

(-) A opção fechar janela dos formulários de edição de dados (padrão do *Windows*) permite sair de um formulário sem pedir confirmação, mesmo no modo de edição de dados, podendo ocasionar a perda acidental de informações digitadas.

(-) Não há opções de “undo” e “redo” para as transações executadas na aplicação.

7. Flexibilidade e Eficiência do Uso.

(-) O sistema não faz distinção entre usuários experientes e não experientes.

(-) Apesar de apresentar algumas teclas de atalho na tela principal, as mesmas estão confusas pois muitas letras da barra de tarefas são as mesmas do menu, por exemplo, o código ALT-C, que deveria corresponder a “Consulta”, segundo a letra grifada no menu principal, abre o formulário de “Análise de Laboratório” e o código ALT-R ocasiona a saída do sistema, sem confirmação.

(-) A tecla F1 (acesso ao *Help* no padrão *Windows*) não está habilitada em formulários de edição de informações.

8. Ajuda os usuários a reconhecer, diagnosticar e recuperar erros.
- (+/-) O sistema só apresenta esse tipo de ajuda a respeito da manutenção e recuperação das tabelas, em outros casos as mensagens de erros não estão ajudando o usuário a reconhecer os erros. Elas devem ser um pouco mais trabalhadas.
 - (-) Falta verificar a consistência das informações digitadas nos campos de entrada de dados. Por exemplo, o registro de um animal cadastrado pode ser igual ao do seu pai e ao de sua mãe.
 - (-) A aplicação não faz a consistência do nome do arquivo/diretório para recuperação de cópia de segurança.
9. Previne Erros
- (-) Na tela de abertura da aplicação, falta uma mensagem para orientar o usuário a inserir uma senha válida e então clicar sobre o botão “Confirmar”, ou então “Cancelar” a entrada na aplicação.
 - (-) Como mencionado no item anterior, as mensagens de erro devem ser trabalhadas para melhor compreensão do usuário, pois na maioria das vezes, quando elas não estão em inglês, as mesmas não estão em forma objetiva e clara.
10. Documentação e *Help*
- (+) O sistema possui recurso de *help on-line* aderente ao padrão atualmente vigente para aplicações *Windows*.
 - (-) A ajuda só está disponível na janela principal e muitas vezes ela é necessária durante o cadastramento, mas não é possível acessá-la durante a inclusão, navegação ou alteração.

4. Discussão

A detecção e a análise detalhada dos problemas de usabilidade de uma interface não devem ser realizados em uma única sessão de avaliação. Neste trabalho, a avaliação criteriosa dos problemas de usabilidade do ProLeite foi feita posteriormente por observadores a partir dos relatórios dos avaliadores e do resultado da reunião para discussão dos problemas de usabilidade da interface.

O observador é especialista em HCI e conhece o sistema avaliado. Um número maior de observadores pode levar a resultados melhores no processo de avaliação criteriosa dos problemas de usabilidade detectados. No entanto, em sistemas simples um único observador que seja especialista no método e detenha bons conhecimentos sobre o sistema a ser avaliado pode ser suficiente.

Durante as avaliações foram detectados vários problemas de usabilidade na interface do ProLeite que foram detalhados na seção anterior. Em sua maioria, os problemas detectados são de fácil solução, como por exemplo a tela de abertura que possui uma imagem grande com excesso de informações desnecessárias. Uma solução possível seria usar uma imagem menor e destacar o campo para entrada de senha.

Outros problemas detectados são mais severos como as mensagens de erro que são pouco explicativas e em alguns casos escritas em inglês, em função do ambiente de implementação. Solucionar este problema, não é complicado, bastando apenas traduzir as mensagens e redigí-las melhor.

Alguns problemas são graves como por exemplo a falta de *feedback* para o usuário que tenta digitar em um formulário e não recebe nenhum aviso que não pode alterar o campo antes de pressionar o botão Alterar ou Inserir. Uma solução para este problema, seria emitir um *beep* e

apresentar uma mensagem caso o usuário insistisse em clicar sobre os campos de edição. Melhor ainda seria não obrigar o usuário a pressionar o botão de Alterar para poder iniciar o preenchimento dos campos.

Além dos problemas de usabilidade, alguns erros de programação (*bugs*) foram descobertos durante a avaliação heurística. Estes erros não podem ser classificados como problemas de usabilidade, mas *bugs* sempre provocam um impacto negativo na usabilidade do sistema. A principal conclusão sobre a descoberta de *bugs* durante a avaliação heurística é que os usuários sempre encontram novos modos de exercitar um programa. Entretanto, este fato evidencia que uma fase de beta teste na qual o sistema é utilizado por um grupo de usuários reais, é extremamente importante antes da exposição do sistema para público externo.

A presença de vários problemas na interface não significa que os desenvolvedores são incompetentes e insensíveis a questões de usabilidade. No entanto, o foco desta discussão são os problemas de usabilidade da interface e não seus aspectos positivos.

Para realizar uma avaliação mais criteriosa dos problemas de usabilidade encontrados no sistema e gerar um relatório síntese com os problemas hierarquizados dos mais graves para os menos graves, o observador também utilizou o sistema para verificar alguns detalhes dos problemas apontados pelos avaliadores. Durante esta avaliação/verificação da interface do sistema, o observador pôde verificar que alguns problemas de usabilidade não foram detectados pelos avaliadores em seus relatórios. Um dos motivos para essa omissão de problemas pode ser a falta de experiência dos avaliadores com o método de avaliação heurística. No entanto, analisando melhor os tipos de problemas, nota-se que são específicos do tipo de aplicação, ou seja, não estão cobertos pelas heurísticas gerais utilizadas.

Com base nestes resultados, constatou-se a necessidade e a possibilidade de se desenvolver heurísticas de categorias específicas como uma forma de se dar um “zoom” em questões de usabilidade específicas de uma classe de produtos, como um suplemento às heurísticas gerais propostas por Nielsen (1990). Algumas são heurísticas novas e outras são complementações às heurísticas originais, como pode ser visto na Tabela 2, a seguir:

1. Opções de menu significativas e agrupadas logicamente: O agrupamento e os nomes das opções do menu são pistas para o usuário encontrar a opção requerida.
2. Uso de botões com ícones e legenda: Se usar ícones, eles devem ser simplificados, com legenda para atender aos usuários que têm dificuldade em entender o significado pictográfico. Adicionalmente, o sistema deve usar um balão explicativo do uso do botão.
3. Permite opções de configuração: Para facilitar o uso do <i>mouse</i> para usuários com dificuldade de manipulação deste periférico, o sistema deve permitir alterar o tamanho dos botões para tamanhos maiores. Além disso, o sistema deve permitir a alteração das cores dos rótulos e cor dos campos de entrada, para facilitar a leitura e prever a possibilidade de execução do sistema em qualquer tipo de configuração de tela (por exemplo, 640X480, 800X600 (fontes grandes e pequenas), etc) sem perda de qualidade dos diálogos.
4. Facilidade no modo de operação: O sistema deve prover um modo de operação facilitado, principalmente para sistemas de entrada de dados. Tais sistemas são construídos a base de formulários de entrada de dados sendo importante minimizar a quantidade de toques do usuário.
5. Títulos de diálogos, rótulos de campos e instruções compreensíveis e significativos: No sistema não devem ser usados termos da área específica de computação como por exemplo “Registro”, “Indexação”. Devem ser usadas palavras significativas para o usuário.
6. Agrupamento lógico e seqüencial dos campos: O sistema deve dispor os campos de forma lógica e seqüencial nos formulários. O agrupamento lógico facilita a entrada de dados tanto para usuários

iniciantes quanto experientes.
7. Diferenciação entre campos não editáveis, obrigatórios e opcionais: O sistema deve prover cor de fundo diferenciada para campos não editáveis sinalizando para o usuário que determinados campos em um formulário não precisam ser digitados. O sistema deve fornecer alguma forma de identificação para campos obrigatórios para auxiliar na entrada de dados pois evidencia quais campos devem ser preenchidos e quais podem ficar em branco. Facilita a entrada dos dados e acaba evitando erros.
8. Permite identificação do tipo de dado e quantidade de caracteres: No sistema deve estar em evidência para o usuário o tipo de dado para cada campo, através de mensagem explicativa (<i>hint</i>), formatação do campo (por ex. --/--/---- para datas) ou no próprio rótulo. O sistema deve alertar sobre a quantidade de caracteres possível por campo.
9. Agilidade na movimentação do cursor: O sistema deve facilitar a mudança de um campo para outro nos formulários através de teclas de atalho além do <i>mouse</i> , como o TAB ou ENTER.
10. Facilidade na correção de erros durante a entrada de dado: O sistema deve permitir a correção rápida de erros durante a entrada de dados através de teclas como DEL, BACKSPACE ou outra. Além disso, ele deve possibilitar <i>overtyping</i> nos campos e a remoção de um campo inteiro.
11. Aproveitamento de dados entrados anteriormente: Para sistemas de entrada de dados, é extremamente importante o aproveitamento de dados na digitação de novos registros. Esta característica diminui consideravelmente a quantidade de toques por registro.
12. Localização de informação rapidamente: O sistema deve permitir a localização de um registro específico durante a entrada do dado. Esta opção de localização deve possibilitar buscas elaboradas não apenas por um código ou chave que identifique o registro na base de dados.

Tabela 2. Conjunto de heurísticas de categorias específicas

De posse deste conjunto específico de heurísticas, os avaliadores poderiam detectar que o sistema ProLeite não possui opções de configuração para alteração de tamanho de botões, cor de rótulos, etc. Estas características facilitam o uso do sistema por usuários com dificuldade de manipulação do *mouse* ou leitura de fontes pequenas. Outro ponto relevante não observado durante as sessões de avaliação, é a ausência de diferenciação entre campos obrigatórios, não editáveis e opcionais no sistema em questão. Este aspecto teria sido verificado se o avaliador tivesse em mãos a heurística específica número 7. Pode-se citar ainda, a não observação de problemas de usabilidade tais como, a falta de aproveitamento de dados entrados anteriormente, questões relativas ao modo de operação, à movimentação do cursor e possibilidade de correção de erros durante a entrada de dados que não foram anotadas durante a avaliação. Apenas uma menção ao modo de operação foi feita em relação a necessidade de pressionar o botão Alterar antes de modificar um determinado formulário.

Desta forma, adicionando-se estas heurísticas específicas à lista de heurísticas originais, poder-se-ia obter um resultado de avaliação heurística ainda melhor para esta categoria de sistemas. Possivelmente o número de problemas de usabilidade encontrados poderia ser ainda maior. A identificação do maior número possível de problemas na fase de protótipo possibilitaria a correção dos mesmos antes da versão final do sistema.

5. Conclusão

O método de avaliação heurística utilizado para a inspeção do sistema ProLeite mostrou-se eficiente na detecção de problemas de usabilidade, além de ser barato, rápido e fácil, comprovando

as expectativas iniciais. No entanto, vários dos problemas de usabilidade específicos da categoria à qual pertence o sistema em questão, não foram verificados pelos avaliadores que utilizaram o conjunto de heurísticas originais em cada sessão de avaliação. Este resultado levou à proposição de um novo conjunto de heurísticas direcionadas para avaliação de sistemas pertencentes ao domínio específico tratado neste trabalho, que permite a detecção de outros problemas de usabilidade não contemplados nas heurísticas originais.

Desta forma, propõe-se que em novas avaliações de sistemas na classe deste domínio específico sejam utilizados os dois conjuntos de heurísticas para detecção do maior número possível de problemas de usabilidade. Isto irá contribuir enormemente para o aumento da qualidade do produto final, uma vez que o método de avaliação heurística deve ser utilizado como parte do ciclo de desenvolvimento de *software*.

6. Referências

- DIX, A.; FINLAY, J.; ABOWD, G.; BEALE, R. *Human-computer interaction*. New York: Prentice-Hall, 1993. 570p.
- HIX, D.; HARTSON, H.R. *Developing user interfaces: ensuring usability through product and process*. New York: John Wiley, 1993. 381p.
- JEFFRIES, R.; MILLER, J.R.; WHARTON, C.; UYEDA, K.M. *User interface evaluation in the real world: a comparison of four techniques*. ACM, 1991.
- LAUREL, B. *The art of human-computer interaction*. Reading: Addison-Wesley, 1990.
- NARDI, B. *A small matter of programming: perspectives on end user computing*. Cambridge: MIT Press, 1993.
- NIELSEN, J. *Usability engineering*. Boston: Academic Press, 1993.
- NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: EMPOWERING PEOPLE - CHI'90 CONFERENCE *Proceedings*. New York: ACM Press, 1990.
- NORMAN, D.; DRAPER, S. *User centered system design*. Hilldale: Lawrence Erlbaum Associates, 1986.
- PREECE, J.; ROGERS, Y.; BENYON, D.; SHARP, H.; HOLAND, S. *Human-computer interaction*. Reading: Addison-Wesley, 1994.
- WINOGRAD, T.; BENNET, J.; DEYOUNG, J.; HARTFIELD, B.,ed. *Bringing design to software*. New York: ACM Press; Reading: Addison-Wesley, 1996. 321p.