

# Aplicabilidade de Padrões de Engenharia de Software e de IHC no Desenvolvimento de Sistemas Interativos

André Constantino da Silva<sup>1</sup>, Júnia Coutinho Anacleto Silva, Rosângela Aparecida Delosso Penteado, Sérgio Roberto Pereira da Silva.

**Abstract** -- A Pattern is a way to express knowledge using words and sketches in a structured form. The knowledge expressed by the pattern is a success solution for a recurrent problem in a determined context. It has many characteristics and two main advantages: 1) it supplies a vocabulary between their users and 2) it captures experience. Patterns can be used in Software Engineering and Human-Computer Interaction to transfer knowledge and to improve the communication between the development team, resulting in a better participation of users of interactive systems. Therefore, to better use the pattern's potential in these areas, it is necessary that its use don't be *ad hoc*. Thus, studies are necessary to point out better ways to use patterns in an interactive system development process. For the first step, potential pattern's uses in the many process models proposed by these areas were identified, include for them integration.

**Index Terms** -- *Computing Patterns, Human-Computer Interaction, Interactive systems, Software engineering, User interfaces.*

## I. INTRODUÇÃO

Dois áreas da Ciência da Computação se destacam no desenvolvimento de sistemas interativos: a Engenharia de Software (ES) e a Interação Humano-Computador (IHC). Entretanto, essas áreas divergem quanto ao foco dado durante o desenvolvimento de tais sistemas. Enquanto o foco de ES é mais voltado para a tecnologia utilizada, o de IHC são os aspectos da interação entre o ser humano e a máquina [1].

Apesar dessas duas áreas divergirem quanto ao foco, ambas propõem o desenvolvimento de sistemas interativos de forma sistemática, definindo modelos de processo, métodos e técnicas. Pode-se citar como modelos de processo propostos pela ES o Cascata [2], o Incremental [3], a Prototipação [3], o Espiral [3], o *Rapid Application Development* (RAD) [3], o *Rational Unified Process* (RUP) [4] e os métodos ágeis como o *Extreme Programming* (XP) [5]. Como modelos de processo propostos pela IHC cita-se o Projeto Centrado no Usuário (PCU) [6], o Estrela [7], a Engenharia de Usabilidade [8] e o

Design Participativo (DP) [9]. Além da apresentação de tais práticas, ambas as áreas estão identificando e aplicando padrões (*patterns*) para o processo de desenvolvimento [10]-[14].

Embora os modelos de processo, métodos e técnicas expressem o foco da área que a propõe, percebe-se que é possível desenvolver sistemas interativos considerando práticas de ambas as áreas através da consideração dos aspectos relevantes à ES em conjunto aos considerados pela IHC, objetivando o desenvolvimento de sistemas de forma mais abrangente. Logo, considerando os aspectos dessas duas áreas durante o desenvolvimento, é possível construir sistemas que não somente sejam de fácil manutenção, e que satisfaçam o usuário quanto ao prazo de entrega e ao custo, mas que também o tornam mais confiável e de fácil utilização pelo usuário. Esses resultados são desejáveis, pois o mercado está se tornando cada vez mais competitivo, sendo a interface com o usuário um diferencial de produto [15].

A integração de ES e IHC não é uma tarefa fácil embora traga benefícios ao processo de desenvolvimento, principalmente devido às divergências de focos e de formação dos profissionais. Para transpor essa dificuldade, é necessário que ocorra uma identificação mais detalhada das tarefas a serem realizadas durante o processo de desenvolvimento, não se limitando somente a responsabilizar o engenheiro de software pela parte tecnológica e o especialista em IHC pela interação e interface com o usuário. Essa divisão não é suficiente, pois aspectos de interação e tecnologia se influenciam mutuamente.

Outra questão a ser considerada na integração entre a ES e a IHC é a necessidade de uma boa comunicação entre os especialistas das duas áreas durante o desenvolvimento do sistema. A falta de comunicação influencia no produto construído, como por exemplo, a adição de funcionalidades não necessárias, código de difícil compreensão e manutenção.

Nesse contexto, padrões são considerados como uma solução para amenizar os problemas citados. Padrões são usados aqui como uma forma de expressar conhecimento através de textos e esboços em um formato estruturado, cujo conhecimento é expresso como uma solução de sucesso para um problema que ocorre frequentemente em um determinado contexto [16].

Padrões podem ser utilizados para melhorar a comunicação entre os profissionais das diferentes áreas, permitindo também que eles expressem conhecimentos obtidos durante o

A. C. da Silva (e-mail: andrecons@dc.ufscar.br), J. C. A. Silva (e-mail: junia@dc.ufscar.br) and R. A. D. Penteado (e-mail: rosangel@dc.ufscar.br) are with Federal University of São Carlos, Departamento de Computação, Rod. Washington Luis, km 235, Caixa Postal 676, São Carlos-SP CEP 13565-905.

S. R. da Silva (e-mail: srsilva@din.uem.br) is with State University of Maringá, Departamento de Informática, Av. Colombo, 5790, zona 07, Maringá – PR, CEP 87020-900.

<sup>1</sup>A. C. da Silva é bolsista financiado pela CAPES.

desenvolvimento de sistemas, de tal forma que possam ser reaproveitados em outros projetos, permitindo diminuir custos e aumentar a qualidade dos sistemas desenvolvidos.

Conceitos envolvendo padrões são apresentados na Seção II, e nas Seções III e IV é discutida a aplicabilidade de padrões nos modelos de processos propostos pela ES e pela IHC, respectivamente. Na Seção V é discutida a aplicabilidade dos padrões de ES e de IHC no desenvolvimento de sistemas interativos. Finalmente, na Seção VI são apresentados as considerações finais e trabalhos futuros com a aplicação de padrões.

## II. PADRÕES

Padrões podem ser vistos como uma forma de expressão, contendo informações textuais e esboços, que surgiram durante o Renascimento, no século XV, quando Francesco di Giorgio expressou alguns de seus projetos em um formato que permitisse que outras pessoas aproveitassem suas soluções em outros projetos [10].

O mais antigo estudo em larga escala envolvendo padrões ocorreu durante as décadas de 60 e 70, na Arquitetura, com o trabalho de Christopher Alexander [16][17]. Alexander definiu um padrão como uma solução para um problema em um contexto e apresentou diversas características dos padrões, além de observar que os padrões não estão isolados.

A seguir são revisadas algumas das características gerais de padrões reunidas por Fincher [18]. As definições apresentadas tentam integrar definições da ES e da IHC:

**1) Formato de Apresentação:** padrões são descritos dividindo sua apresentação em um determinado número de elementos. Os elementos são escolhidos de acordo com a ênfase e os detalhes que o autor deseja destacar no padrão. Logo, percebe-se que não existe um formato único [19][20];

**2) Captura da Prática:** um padrão apresenta uma solução para um problema existente na prática cuja solução pode ser aplicada, de modo eficiente, a diversos casos;

**3) Abstração:** os padrões não podem ser abstratos demais e nem muito específicos. Quando um padrão é muito abstrato, o usuário do padrão necessita redescobrir como aplicar a solução, impedindo um melhor aproveitamento da solução do padrão. Padrões específicos demais também não são desejáveis, pois impediria uma maior aplicação do padrão;

**4) Princípio de Organização:** Alexander observou que os padrões não estão isolados, existe um relacionamento entre eles [16]. Padrões atualmente são organizados seguindo um dos dois princípios:

– *Catálogo de Padrões:* apresenta um conjunto de padrões;

– *Linguagem de Padrões:* apresenta um conjunto estruturado de padrões, o relacionamento entre eles e informações relevantes sobre o domínio e a aplicabilidade da linguagem. Na linguagem, os padrões que a compõem devem cobrir todos os aspectos importantes de um dado domínio [21];

**5) Facilidade de Compreensão:** um padrão é escrito de forma compreensível, simples e coesa para facilitar a sua

leitura e aplicação.

Através dessas características, padrões possuem, basicamente, duas vantagens:

- **Fornecimento de um Vocabulário:** padrões podem ser utilizados para melhorar a comunicação entre a equipe de desenvolvimento ou, também, permitir que os usuários (clientes da equipe de desenvolvimento) opinem e se expressem melhor. A segunda alternativa foi explorada pelos trabalhos de Alexander;

- **Captura da Experiência:** como padrões apresentam soluções de sucesso que foram identificadas e evoluídas ao longo do tempo, eles capturam a experiência. Logo, padrões podem ser utilizados para a transferência de conhecimento entre pessoas de níveis de experiência diferentes. A utilização de padrões para transferência de conhecimento é amplamente explorada e utilizada pela ES.

Diante dessas características e vantagens, os padrões possuem uma ampla aplicabilidade no desenvolvimento de sistemas interativos. Essa aplicabilidade de padrões, tanto na ES quanto na IHC, será discutida nas próximas seções.

## III. APLICABILIDADE DOS PADRÕES NOS MODELOS DE PROCESSO DE ENGENHARIA DE SOFTWARE

Com a experiência em desenvolvimento de sistemas, mais precisamente durante a elaboração do projeto, Gamma, Helm, Johnson e Vlissides publicam, em 1995, um livro contendo uma coleção de padrões de projeto de sistemas orientados a objetos [11]. Segundo esses autores, o objetivo da sua coleção de padrões é fornecer meios de tornar o sistema mais reusável, e também anular alternativas que comprometam a reutilização. Entretanto, esse trabalho se destaca devido à utilização de padrões para a transferência de experiência entre profissionais de níveis de conhecimento diferentes, ignorando a participação do usuário do sistema durante o desenvolvimento na etapa de projeto. Segundo Fincher [22], Gamma *et al.* desconsideraram a participação do usuário do sistema motivados pela não necessidade dos usuários conhecerem internamente o sistema.

Além dos padrões de projeto de Gamma *et al.*, diversos outros padrões da ES foram identificados. Em 1996, Buschmann *et al.* [12] publicaram um livro contendo diversos padrões. Nesse livro, os padrões da ES são divididos em três categorias:

- **Padrões Arquiteturais:** expressam o esquema ou organização estrutural para sistemas de software ou hardware, dividindo o sistema em componentes ou em subsistemas, e apresentando o relacionamento entre eles;

- **Padrões de Projeto:** fornecem um esquema para refinar os subsistemas ou componentes de um sistema de projeto de software;

- **Padrões de Programação ou Idiomas:** são padrões de baixo nível, específicos para uma linguagem de programação, que descrevem como implementar aspectos particulares dos componentes ou a relação entre eles usando características de uma determinada linguagem.

Além destas três categorias definidas por Buschmann *et al.*,

existem outras categorias, entre elas pode-se citar:

**Padrões de Análise:** descrevem soluções para problemas de análise de sistemas;

- **Padrões de Interface:** são casos particulares de padrões de projeto e definem soluções para problemas comuns no projeto de interface de sistemas;

- **Padrões de Processo:** define soluções para os problemas encontrados nos processos envolvidos na ES, como por exemplo, desenvolvimento, gerenciamento de configurações e testes;

- **Padrões Organizacionais:** descrevem abordagens provadas e de sucesso para organizar e gerenciar pessoas envolvidas com os processos na ES.

Analisando as categorias acima definidas pode-se notar a relação existente entre as fases de desenvolvimento de um sistema e o melhor uso de padrões em cada uma delas. Na fase de especificação de requisitos pode-se aplicar padrões de análise. Essa fase ocorre:

- na etapa de análise de requisitos no modelo de processo Cascata ou Incremental;

- na etapa de engenharia do produto no modelo de processo Prototipação;

- na atividade de engenharia no modelo de processo Espiral;

- no *workflow* de análise e projeto no modelo de processo RUP; ou

- na etapa de iterações e lançamentos no modelo de processo XP.

Exemplos de padrões de análise são os contidos na linguagem de padrões para Gestão de Recursos de Negócio (GRN) [23], cujos padrões auxiliam na especificação de requisitos do domínio de gestão de recursos, como: localizar, vender, manter recursos.

Na elaboração do projeto, os padrões arquiteturais podem ser utilizados para apoiar a escolha da arquitetura do sistema. Isso acontece:

- durante a etapa de projeto de sistemas e de software no modelo Cascata;

- na atividade de Engenharia do modelo Espiral;

- na etapa de Projeto do Incremental;

- no *workflow* análise e projeto do RUP; ou

- ao obter o projeto simplificado ou durante a construção da metáfora do sistema no XP.

A escolha será realizada através da comparação entre os requisitos do sistema com as forças apresentadas pelo padrão. O padrão arquitetural escolhido irá apresentar quais são os componentes a serem utilizados e como se realiza a comunicação entre eles. Padrões de projeto da ES são aplicados para refinar esses componentes e a comunicação entre eles. Por exemplo, no caso de sistemas interativos, poderia ser utilizado o padrão arquitetural *Model-View-Controller* (MVC) ou o padrão arquitetural *Presentation-Abstraction-Control* (PAC), podendo utilizar os padrões de projeto de Gamma *et al.* para refinar os componentes e a comunicação entre eles considerando a reusabilidade do sistema, pois esses padrões, segundo seus autores, tornam o

sistema mais reutilizável. Nessa etapa de desenvolvimento também é possível aplicar os padrões de interface para auxiliar na definição de interfaces do sistema em desenvolvimento com outros sistemas.

No caso de sistemas orientados a objetos (OO), quando se deseja utilizar um banco de dados não OO para a persistência dos dados, existem padrões que auxiliam a isolar do engenheiro de software os detalhes relacionados com a implementação da persistência, como é o caso do padrão *Persistent Layer* [24]. Também se pode utilizar um outro padrão que auxilie na definição e distribuição de responsabilidades relacionadas a operações de leitura e escrita de dados no banco de dados, como é o caso do padrão CRUD [24]. Este padrão mapeia funcionalidades da classe a operações possíveis de serem realizadas em banco de dados relacionais. Padrões também podem ser utilizados para projetar o banco de dados relacional e otimizá-los, como é o caso dos padrões de [25] e [26].

Padrões de processo e organizacionais podem ser aplicados em todas as etapas do processo de desenvolvimento de sistemas interativos. Exemplos de padrões de processo são apresentados por Ambler [27], que apresenta também uma discussão de como e quando esses padrões podem ser aplicados durante o desenvolvimento.

Outros exemplos de padrões de processo são os de testes, que podem ser utilizados pelos engenheiros de software na criação de casos de testes, apresentando quais das técnicas de testes poderiam ser aplicadas para um determinado problema. Logo, padrões de testes podem ser aplicados:

- na etapa de implementação e teste de unidade e na etapa de integração e teste de sistemas do modelo Cascata;

- na atividade de construção e entrega do Espiral;

- na fase de teste e entrega do modelo RAD;

- na etapa de teste do Incremental;

- no *workflow* de testes do RUP; ou

- na fase de iterações e lançamentos do XP, mais precisamente na atividade de testes.

Além dessas aplicações de padrões da ES, padrões podem ser aplicados na condução de processos de reengenharia. Recchia [28] elaborou uma família de padrões de processo para a realização de engenharia reversa de sistemas legados Clipper, e padrões para conduzir a engenharia avante desses sistemas para sistemas em Delphi.

A aplicação de padrões durante o processo de desenvolvimento pode trazer diversos benefícios. Por exemplo, pode tornar o código do sistema menos pessoal, o que auxilia em modelos de processo como o XP, que padroniza tal tipo de código, pois o mesmo é compartilhado por toda a equipe de desenvolvimento.

Outro benefício da aplicação de padrões no desenvolvimento de sistema é a melhora na comunicação entre a equipe de desenvolvimento, através do fornecimento de um vocabulário comum. Esse benefício se torna útil em modelos de processo que exigem uma ampla comunicação entre a equipe de desenvolvimento, como o modelo de processo RAD e os modelos de processo ágeis como o XP.

A aplicação de padrões também pode aumentar a clareza no projeto e no código do sistema, desde que sua aplicação seja facilmente identificada no artefato. O engenheiro de software, habituado com os padrões, compreenderá a documentação de modo mais eficiente, pois já reconhece o problema que o padrão trata, além das forças que estavam em conflito e levaram à aplicação do padrão.

Percebendo os benefícios da aplicação de padrões, diversos padrões estão sendo identificados e utilizados no desenvolvimento de sistemas pelos engenheiros de software [11][12][23][28]. Esses padrões podem ser aplicados durante as diversas etapas do desenvolvimento de sistemas. Entretanto, percebe-se que tal aplicação é aleatória, podendo ser útil uma forma sistemática para melhor aproveitar os benefícios da aplicação de padrões.

#### IV. APLICABILIDADE DOS PADRÕES NOS MODELOS DE PROCESSO DE INTERAÇÃO HUMANO-COMPUTADOR

Em 1998, Tidwell apresenta sua linguagem de padrões voltada para o desenvolvimento do projeto da interação entre o homem e a máquina, considerando aspectos de IHC [13]. Segundo Fincher [18], esse trabalho se destaca devido ao seu pioneirismo e também por ser a maior em escala e influência em IHC. Entretanto, esse trabalho não está finalizado, pois alguns padrões não estão totalmente escritos e outros não foram escritos, mesmo tendo sido identificados.

Diversos outros padrões de IHC foram propostos, entre eles se destacam os identificados por Welie e Trætteberg [29], cujos padrões são voltados para o projeto da interação e interfaces com o usuário. No formato proposto por esses autores existe um elemento chamado “Princípio de Usabilidade”, que descreve quais princípios de usabilidade o padrão está conectado. Esses princípios estão relacionados de acordo com as categorias de problemas do usuário do sistema propostas por Norman [30]. Esses autores também apresentam no elemento “Razão” (*Rationale*), quais aspectos mensuráveis de usabilidade são incrementados ao aplicar o padrão.

Outro trabalho que se destaca na identificação de padrões de IHC é o de Borchers [10]. Borchers apresenta uma linguagem de padrões de IHC, mais especificamente para projeto, focando a interação com quiosques e fornecendo certo grau de transparência da interação. Além de padrões de IHC, Borchers também apresenta padrões arquiteturais da ES e padrões específicos do domínio da aplicação, que em seu campo de estudo é a música.

Em 2003, Tidwell divulga uma coleção de padrões voltada especificamente para a construção de interfaces com o usuário [31]. Baseando-se em sua experiência na identificação e na aplicação de padrões no desenvolvimento de sistemas, Tidwell [32] discute que “Exemplos”, “Aplicabilidade” e “Evidência” são os elementos mais importantes para os usuários dos padrões. Logo, enfatizando essas características, em detrimento de alguns aspectos formais, como a completude, o formato, e a diferença entre uma técnica e um padrão, a apresentação dos padrões na sua coleção não é uniforme.

Segundo Tidwell [32], os padrões de Welie foram escritos

apresentando explicações do porquê o padrão funciona e a razão de se utilizar a solução do padrão do que uma outra. Ela também considera a especificação e a aplicabilidade características-chaves desse trabalho. Diversos padrões de IHC foram identificados por Welie [14], e são apresentados tanto padrões para a interação na Web quanto padrões para interfaces gráficas com o usuário (GUI) e também para a interação com sistemas móveis. Além da identificação de padrões de IHC, Welie e Veer propõe a divisão de seus padrões nas seguintes categorias [33]:

- **Metas de Negócio:** descrevem propósitos ou razões para a existência do sistema;
- **Padrões de Postura:** descrevem a estrutura que é comumente usada por sistemas relacionados;
- **Padrões de Experiência:** descrevem as principais metas e tarefas do usuário;
- **Padrões de Tarefas:** apresentam soluções para pequenos problemas do usuário;
- **Padrões de Ação:** especifica o uso de *widgets* bem conhecidos ou descrevem *widgets* customizáveis.

Assim como na ES, não existe uma classificação bem definida para os padrões de IHC, pois o assunto ainda está evoluindo nessas áreas. Segundo Alpert [34], há indícios que existam duas categorias de padrões na IHC:

- **Padrões de IHC:** relacionados com assuntos de alto nível, e possivelmente com *guidelines*, envolvendo a psicologia do usuário do sistema, podendo incluir também padrões arquiteturais;
- **Padrões de Interface com o Usuário:** relacionados com problemas de interação específicos, sendo sua solução baseada em componentes de interface com o usuário.

Além das categorias citadas, é possível encontrar na literatura outros tipos de padrões, como os padrões para Hipermídia, padrões de usabilidade, padrões para Web e padrões para avaliação de usabilidade.

Padrões podem ser utilizados para detalhar tarefas realizadas pelos usuários do sistema e como o sistema pode suportá-las. Esta atividade ocorre:

- na etapa de projeto do modelo de processo Projeto Centrado no Usuário (PCU);
- na etapa de análise de sistemas/ tarefas/ usuários/ funcionalidades do modelo Estrela; ou
- no estágio de conhecer o usuário no modelo Engenharia de Usabilidade.

Deste modo, uma linguagem de padrões para refinar as soluções propostas para as tarefas é útil, tendo como produto final a especificação da interação e da interface com o usuário.

Quando existe a necessidade de se apresentar diversas soluções, como no caso do estágio de projeto detalhado da Engenharia de Usabilidade, padrões podem ser utilizados para oferecer uma base para as diversas soluções. Como os padrões não limitam a criatividade do projetista [35], diversas soluções irão ser propostas, entretanto com características em comum definidas pelos padrões escolhidos antes do início das atividades desse estágio.

Outro benefício da não limitação da criatividade do

projetista é a possibilidade de explorar alternativas de projeto, pois a aplicação do padrão não implica em uma única solução. Isso ocorre devido à liberdade de criatividade, oferecida através do grau de abstração. Essa aplicação de padrões é útil durante a realização do estágio de projeto paralelo da Engenharia de Usabilidade.

Percebe-se que os padrões podem ser utilizados seguindo a idéia de Alexander: fornecer um vocabulário que permita uma maior participação do usuário no desenvolvimento. Essa idéia se difere da ES, que utiliza padrões para transferência de conhecimento. Logo, em modelos de processo de IHC que exigem uma ampla comunicação entre a equipe de desenvolvimento e o usuário, como é o caso do Design Participativo, padrões podem ser utilizados como um vocabulário entre a equipe de desenvolvimento e os usuários finais, tornando a comunicação entre eles mais efetiva.

Existe também a possibilidade de expressar o *feedback* do usuário do sistema através de padrões, desde que se permita ao usuário o acesso a eles. Os usuários poderiam indicar quais padrões eles acreditam que seriam uma melhor alternativa para o problema identificado no sistema. Esse emprego é útil durante o retorno à etapa de projeto no modelo de processo PCU, após a etapa de avaliação, ou na etapa de avaliação de usabilidade do modelo Estrela, ou na Engenharia de Usabilidade, no estágio de coleta de *feedback* de estudo de campo.

Outro modo de aplicação de padrões é o seu uso no planejamento e na condução de avaliações, durante o processo de desenvolvimento. Avaliações ocorrem no modelo PCU durante a etapa de avaliação, ou na etapa de avaliação de usabilidade do modelo Estrela, ou nos estágios de teste empírico e projeto interativo na Engenharia de Usabilidade. Um exemplo de padrões para avaliações é a linguagem proposta por Gellner e Forbrig [36] que é voltada ao planejamento e condução de avaliações de usabilidade, e que, segundo seus autores, pode ser utilizada independentemente do modelo de processo seguido.

Do mesmo modo como padrões podem auxiliar no planejamento e condução de avaliações, eles também podem ser utilizados para construir cenários mais realísticos para essas avaliações.

Além dessas aplicabilidades no processo de desenvolvimento, padrões estão sendo utilizados para expressar melhor os *guidelines* e as heurísticas propostas por especialistas em IHC. Como os *guidelines* podem não possuir exemplos, não serem construtivos e serem conflitantes entre si, e como as heurísticas também não possuem exemplos, não são construtivas e são muito abstratas [10][37], padrões, através de seu formato e de suas características, podem expressar melhor essas ferramentas.

Percebe-se, então, que padrões podem ser aplicados nas diversas atividades para o desenvolvimento de um sistema interativo seguindo as práticas da IHC. Essa aplicação traz como benefício o uso de conhecimento de projetos anteriores, e auxilia na garantia dos princípios de alguns modelos de processo dessa área, como é o caso do princípio de

envolvimento ativo do usuário do sistema no modelo de processo PCU e DP.

#### V. APLICABILIDADE DOS PADRÕES DE ENGENHARIA DE SOFTWARE E DE INTERAÇÃO HUMANO-COMPUTADOR PARA DESENVOLVIMENTO DE SISTEMAS INTERATIVOS

Em processos de desenvolvimento de sistemas interativos que consideram os aspectos tanto da ES como da IHC, a comunicação entre os integrantes da equipe de desenvolvimento e também entre essa e os usuários do sistema é de fundamental importância para o sucesso do produto. Devido ao fornecimento de um vocabulário comum favorecer o aumento da comunicação entre os envolvidos no processo, e também devido à facilidade de compreensão, padrões podem ser utilizados para melhorar a comunicação entre a equipe e os usuários do sistema. Neste contexto, padrões permitem um maior envolvimento dos usuários no processo de desenvolvimento fornecendo-lhes um vocabulário para expressar suas opiniões. Esses usuários também podem utilizar os padrões para expressar seus requisitos como, por exemplo, através de padrões de análise, e refinar os requisitos através de padrões de tarefas do usuário.

A característica de facilidade de compreensão também pode ser explorada no sentido de permitir que especialistas de uma área compreendam as decisões tomadas pelos especialistas da outra área envolvida no processo de desenvolvimento. Esse conhecimento é desejável, pois facilita as tomadas de decisões que envolvam opiniões dos especialistas das diferentes áreas.

Devido à natureza dos padrões em fornecer uma solução para um problema que é formado por forças em conflito, eles podem ser utilizados para expressar soluções para os conflitos entre engenheiros de software e especialistas de IHC que ocorrem de forma recorrente no desenvolvimento de sistemas.

#### VI. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Através da aplicação de padrões é possível desenvolver o sistema de modo mais abrangente, considerando e tratando os aspectos relevantes de ES e de IHC. Entretanto, é necessário que os padrões não sejam aplicados de modo aleatório, para que se possa aproveitar melhor o seu potencial no processo. Isso ocorre devido a enorme quantidade de padrões já identificados, e a crescente identificação de novos padrões. Um estudo é necessário para que os padrões já identificados e usados pelas áreas relacionadas com o desenvolvimento de sistemas sejam mais bem explorados. Uma alternativa identificada é a composição dos padrões relacionados com o desenvolvimento de sistemas interativos em uma coleção mais abrangente do que as propostas pelos autores dos padrões. Essa coleção deverá reunir os diversos padrões existentes e fornecer diretrizes mais específicas relacionadas a sua aplicação durante o processo de desenvolvimento.

## VII. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J. Brown, "Methodologies for the Creation of Interactive Software," Victoria University of Wellington., Wellington, Nova Zelândia, Tech. Rep. CS-TR-96/1, May, 1996.
- [2] I. Sommerville, *Engenharia de Software*, 6.ed. São Paulo: Addison-Wesley Pub. Co., 2003, p. 592.
- [3] R. S. Pressman, *Engenharia de Software*, 5.ed. Rio de Janeiro: McGraw-Hill, 2002, p. 843.
- [4] P. Kruchten, *The Rational Unified Process – An introduction*. 2.ed. Nova Jersey: Addison-Wesley Pub. Co., 2000, p. 320.
- [5] Extreme Programming. [Online]. Available: <http://www.extremeprogramming.com>.
- [6] J. Preece, *A Guide to Usability: Human factors in computing*. USA: Addison-Wesley Pub. Co., 1993, p. 143.
- [7] D. Hix, H. R. Hartson, *Developing User Interface: Ensuring Usability Through Product & Process*. USA: Wiley & Sons Inc., 1993, p. 381.
- [8] J. Nielsen, *Usability Engineering*. California: Academic Press, 1993, p. 362.
- [9] A. J. Dix, J. E. Finlay, G. D. Abowd, R. Beale, *Human-Computer Interaction*, 2.ed. England: Prentice Hall Europe, 1998, p. 454.
- [10] J. Borchers, *A Pattern Approach to Interaction Design*. England: John Wiley & Sons Ltd., 2001, p. 246.
- [11] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston: Addison-Wesley, 1995, p. 395.
- [12] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, E. Stal, *Pattern-Oriented Software Architecture. Volume 1: A System of Patterns*, Vol. I. England: John Wiley & Sons Ltd., 1996, p. 476.
- [13] J. Tidwell, "Common Ground: a Pattern Language for Human-Computer Interface Design," presented at the 5th PLoP, Monticello, USA, 1998.
- [14] M. Van Welie, Pattern in Interaction Design. [Online]. Available: <http://www.welie.com>.
- [15] X. Ferré, "Approaches to HCI Integration into Software Engineering Processes: How Much We Still Need to Do," in *Proc. Workshop WIHCES in CLICH 2003*, pp. 11-17.
- [16] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahling, S. Angel, *A Pattern Language: Towns, Buildings, Construction*. Oxford: Oxford University Press, 1977, p. 1216.
- [17] C. Alexander, *The Timeless Way of Building*. Oxford: Oxford University Press, 1979, p. 568.
- [18] S. Fincher, "What is a Pattern Language?," presented at the 16th CHI, Pittsburgh, USA, 1999.
- [19] S. Fincher. HCI Pattern-Form Gallery. [Online]. Available: [://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html](http://www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html).
- [20] W. Cunningham, Wiki Wiki Web. [Online]. Available: <http://c2.com/cgi/wiki?WelcomeVisitors>.
- [21] J. O. Coplien, "Software Design Patterns: Common Questions and Answers," in *The Pattern Handbook: Techniques, Strategies, and Applications*, L. Rising, Ed. New York: Cambridge University Press, 1998, pp. 311-320.
- [22] S. Fincher, What is a Pattern Language?, Presented at INTERACT'99. [Online]. Available: [http://media.informatik.rwth-aachen.de/patterns/tiki/tiki-download\\_file.php?fileId=11](http://media.informatik.rwth-aachen.de/patterns/tiki/tiki-download_file.php?fileId=11).
- [23] R. T. V. Braga, F. S. R. Germano, P. C. Masiero, "A Pattern Language for Business Resource Management," presented at the 6th PLoP, Monticello, USA, 1999.
- [24] J. W. Yoder, R. E. Johnson, Q. D. Wilson, "Connecting Business Objects to Relational Database," presented at the 5th PLoP, Monticello, USA, 1998.
- [25] K. Brown, B. Whitenack, "Crossing Chasms: A Pattern Language for Object-RDBMS Integration," in *Pattern Languages of Programming Design*, vol. 2, J. M. Vlissides, J. O. Coplien, N. L. Kerth, Eds. Reading, MA: Addison-Wesley Pub. Co., 1996, pp. 227-238.
- [26] W. Keller, J. Coldewey, "Accessing Relational Databases: A Pattern Language," in *Pattern Languages of Programming Design*, vol. 3, R. Martin, D. Riehle, F. Buschmann, Eds. Reading, MA: Addison-Wesley Pub. Co., 1998, pp. 313-343.
- [27] S. W. Ambler, *Process Patterns: Building Large-Scale Systems Using Object Technology*, United Kingdom: Cambridge University Press, 1998, 582 p.
- [28] E. L. Recchia, R. A. D. Penteado, "Uma Família de Padrões para Reengenharia Orientada a Objeto," in *Proc. of The Second Latin American Conference on Pattern Languages of Programming - SugarLoafPlop*, pp. 237-252.
- [29] M. Van Welie, H. Trætteberg, "Interaction Patterns in User Interfaces," presented at the 7th PLoP, Monticello, USA, 2000.
- [30] D. Norman, *The Design of Every Things*. USA: Basic Books, 1988, p. 272.
- [31] J. Tidwell, User Interface Patterns And Techniques. [Online]. Available: <http://time-tripper.com/uipatterns>.
- [32] J. Tidwell, "Perspectives on HCI Patterns: Concepts an Tools," Position paper presented at the Workshop on Perspectives on HCI Patterns in 20nd CHI, Flórida, USA, 2003.
- [33] M. Van Welie, G. C. Van der Veer, "Pattern Languages in Interaction Design: Structure and Organization," presented at the 9th. INTERACT, Zürich, Suíça, 2003.
- [34] S. R. Alpert, "Getting Organized: Some Outstanding Questions and Issues Regarding Interaction Design Patterns," Position paper presented at the Workshop on Perspectives on HCI Patterns in 20nd CHI, Flórida, USA, 2003.
- [35] J. Vlissides, "Patterns: The Top Ten Misconceptions," Object Magazine, 1997.
- [36] M. Gellner, P. Forbrig, "A Usability Evaluation Pattern Language," presented at the 9nd INTERACT, Zürich, Suíça, 2003.
- [37] M. Van Welie, G. C. Van der Veer, A. Eliëns, "Patterns as Tools for User Interface Design," presented at the Int. Workshop on Tools for Working with Guidelines, Biarritz, French, 2000.

## VIII. BIOGRAFIAS

**André Constantino da Silva:** graduado em Ciência da Computação pela Universidade Federal de Goiás, mestrando em Ciência da Computação pelo PPG-CC/UFSCar.

**Júnia Coutinho Anacleto da Silva:** professora adjunta do Departamento de Computação da UFSCar, doutora em Física Computacional pelo IFSC/USP.

**Rosângela Aparecida Dellosso Penteado:** professora adjunta do Departamento de Computação da UFSCar, doutora em Física Computacional pelo IFSC/USP.

**Sérgio Roberto Pereira da Silva:** professor do Departamento de Informática da UEM, doutor em Ciência da Computação pela PUC/RJ.