

5 Discussões

Neste capítulo discutimos a utilização da ferramenta ComunIHC-ES no apoio à comunicação entre os profissionais de IHC e os engenheiros de software. Alguns resultados obtidos no estudo de caso apresentado no capítulo anterior são utilizados durante a exposição das questões elucidadas aqui.

5.1 ComunIHC-ES à luz da Engenharia Semiótica

A introdução desta tese ressaltou a motivação teórica para se criar uma ferramenta que tenha como objetivo fazer a ponte entre as áreas de IHC e a de engenharia de software, mais especificamente, uma ferramenta que apóie a construção da camada da aplicação (funcionalidades internas do software) compatível com o projeto da camada de interação (conversa usuário-sistema). A busca desta compatibilidade tem como objetivo evitar que decisões sobre a semântica da interação sejam ignoradas, modificadas ou mal entendidas pelos engenheiros de software durante a construção da aplicação, podendo gerar assim, problemas na interação usuário-sistema. É importante ressaltar novamente que o objetivo final é a construção de um sistema interativo que atenda aos usuários e que tenha uma alta qualidade de uso.

Para tentar diminuir problemas na comunicação entre estes profissionais, foi proposta nesta tese a utilização de um conjunto de representações, que formam a ferramenta ComunIHC-ES. Entretanto, existe a seguinte questão: A ComunIHC-ES é um signo (Peirce, 1931-1958) criado por um designer de IHC, e como um signo, ela não tem um significado único (de Souza, 2005, p.39). O interlocutor, ao recebê-la, pode gerar diversas interpretações, através de um processo ilimitado – a semiose ilimitada. Fatores subjetivos, situacionais e culturais influenciam o fruto deste processo, ou seja, a atribuição de

significados a um signo, no caso, a ferramenta ComunIHC-ES. Desta forma, não podemos prever qual será o resultado da leitura de um engenheiro de software sobre a representação do projeto de IHC.

A ferramenta ComunIHC-ES foi pensada de forma a tentar convergir o máximo possível o significado atribuído a ela pelos profissionais de IHC e pelos engenheiros de software. Para isto, houve o cuidado de se deixar claro através dela os resultados das tomadas de decisão dos profissionais de IHC quanto ao projeto de IHC. Esta explicitação da lógica de design colabora para a formação de um significado coerente, pelos engenheiros de software, com o que foi proposto pelos profissionais de IHC. A criação da parte 3 da ferramenta – comunicação do projeto de IHC – tem este papel.

A ComunIHC-ES, como já foi explicado, é fundamentada na teoria da engenharia semiótica. Por esta ser uma teoria de IHC, durante o desenvolvimento deste trabalho, houve a preocupação sobre o entendimento de suas partes e, para isto, da engenharia semiótica, pelos engenheiros de software – profissionais que não necessariamente conhecem os fundamentos desta teoria e que estão habituados com uma outra forma de entender e decompor um problema do mundo computacional. Esta preocupação foi desfeita durante a execução do estudo de caso. Houve uma total aceitação do aprendizado e entendimento dos fundamentos²¹ da teoria de engenharia semiótica como base para o uso da ferramenta ComunIHC-ES. Os engenheiros de software não tiveram problemas em entender o projeto da interação usuário-sistema como uma conversa entre o usuário e um representante do designer que está cristalizado na interface. Eles entenderam esta metáfora e a utilizaram ao fazer as comunicações e negociações com o avaliador durante o estudo de caso.

A ComunIHC-ES, uma nova ferramenta de IHC baseada na teoria da engenharia semiótica, agora é acrescida às ferramentas de apoio ao design e avaliação de IHC que fazem parte desta teoria. Podemos citar como ferramentas de design desta teoria a Manas (apoio ao design de sistemas multiusuários) (Barbosa, 2006) e a de construção de sistemas de ajuda online, já citada (Silveira, 2002) e, como ferramenta de avaliação, o método de

²¹ Estes fundamentos foram apresentados durante o minicurso do estudo de caso, cujo material se encontra no Apêndice C desta tese.

avaliação de comunicabilidade, também já citado (Prates et al., 2000). Outras ferramentas podem ser vistas em (de Souza, 2005).

5.2 ComunIHC-ES como *Boundary Object*

Boundary Objects são objetos que apóiam a interseção de mundos sociais diferentes e fornecem informações para cada mundo. As características de um *boundary object* de acordo com Star e Griesemer (Star e Griesemer, 1989) são:

1. Faz a interseção entre áreas diferentes;
2. Todas as partes envolvidas contribuem para a sua construção;
3. Possui as informações requeridas por cada área;
4. É flexível o suficiente para adaptar a necessidade de cada área e robusto o suficiente para manter uma identidade comum entre as áreas;
5. É estruturado fracamente para o uso comum, mas torna-se fortemente estruturado para o uso individual de cada área;
6. Aumenta a autonomia e a comunicação entre as áreas;
7. Contém o essencial para manter a coerência de informação;
8. Atua como ponte entre as áreas e não impõe a visão de uma.

A proposta de Walenstein (Walenstein, 2003), cujo trabalho foi citado na Introdução desta tese, e de outros autores como Green (Green, 1989), é de se usar *boundary objects* entre as áreas entre IHC e engenharia de software. Segundo Walenstein, os profissionais de IHC e engenheiros de software podem utilizar mais de um *boundary object* durante as fases do processo de desenvolvimento de software, cada um com um certo objetivo. Em seu trabalho, Walenstein utiliza uma lista inicial de tipos *boundary objects* proposta por Star e Griesemer, e traz para o contexto de IHC. Segue a lista:

- Repositórios (*repositories*): conjunto indexado de objetos, como, por exemplo, uma biblioteca. Walenstein caracteriza GUI *toolkits* como um repositório, pois eles contêm uma biblioteca reusável de componentes de interface. Eles contemplam características da engenharia de software (reuso, *framework*, etc) e características de IHC (princípios e estilos de interação, etc);
- Tipo Ideal (*ideal type*): é um objeto como, por exemplo, um diagrama, que não descreve em detalhes um item. São abstrações de domínios. Walenstein cita como exemplo os *design patterns*;
- Limites coincidentes (*coincident boundaries*): objetos que possuem o mesmo limite, mas cujo conteúdo interno é diferente para cada domínio. Star cita como exemplo os mapas geográficos, que possuem um limite fixo, mas podem conter informações diferentes para comunidades distintas. Walenstein fornece como exemplo os cenários, que podem ter uso distinto por IHC e pela engenharia de software;
- Formulários padronizados (*standardized forms*): objetos padronizados para apoiar um tipo de comunicação que ocorre em grupos distintos, por exemplo, um formulário com estrutura fixa.

Pelo que foi analisado no estudo de caso apresentado no capítulo anterior, acreditamos que a ComunIHC-ES pode ser considerada um *boundary object* entre as áreas de IHC e a de engenharia de software. Este *boundary object* tem com o objetivo comunicar a solução do projeto IHC (e o conhecimento que está por trás dela) para os engenheiros de software, apoiando a negociação e a interseção entre estas duas áreas (primeira característica de *boundary objects*, como foi descrito acima).

A segunda característica de um *boundary object* é: “Todas as partes envolvidas contribuem para a sua construção”. No estudo de caso apresentado, houve muita passagem de conhecimento no sentido IHC→Engenharia de Software, mas não no sentido oposto, isto é, questões de engenheiros de software afetando a solução via ComunIHC-ES. Acreditamos que a simplicidade do domínio não promoveu uma maior negociação nesta direção, ou seja,

Engenharia de Software→ IHC. Estudos de caso reais e mais complexos são necessários para verificar como a ComunIHC-ES se comporta quanto a esta característica.

A ComunIHC-ES contém todo o projeto da interação modelado por um profissional de IHC, ou seja, toda a informação que ele acredita ser necessária para se especificar esse projeto. O estudo de caso realizado nos mostrou que a ComunIHC-ES também fornece todas as informações necessárias sobre o projeto da interação para o trabalho dos engenheiros de software. Desta forma, esta ferramenta possui as informações requeridas por cada área (terceira característica de um *boundary object*).

Também podemos ver as características “é flexível o suficiente para adaptar a necessidade de cada área e robusto o suficiente para manter uma identidade comum entre as áreas” e “é estruturado fracamente para o uso comum, mas torna-se fortemente estruturado para o uso individual de cada área” se manifestarem na ComunIHC-ES. Como já foi dito, o objetivo desta ferramenta é comunicar a camada de interação para os engenheiros de software e permitir que eles utilizem este projeto durante a especificação das funcionalidades do software. Ou seja, o objetivo é que eles utilizem o projeto de IHC no contexto da engenharia de software. Durante o estudo de caso, pudemos perceber que isto ocorreu, isto é, eles utilizaram a ComunIHC-ES no momento da modelagem UML. Então, ela atendeu às necessidades desta área, e claramente, atende às necessidades da área de IHC. Além disto, esta ferramenta foi concebida para permitir uma negociação sobre o projeto de IHC entre as áreas. Desta forma, o projeto que ela representa está “aberto” para possíveis modificações, de acordo com as restrições impostas pelas áreas envolvidas.

Durante o estudo de caso, também pudemos perceber que a ComunIHC-ES serviu como “base” para a discussão do projeto de IHC que estava representado ali. Através dela foi possível compartilhar entre os profissionais o projeto de IHC, permitindo que cada um deles seguisse com o seu trabalho, baseando-se em uma visão comum da camada de interação. Então, uma vez que o projeto de IHC tenha sido comunicado, entendido e todos estejam de acordo com o que está representado, podemos dizer que a ComunIHC-ES aumenta a autonomia dos profissionais, permitindo que cada um siga o seu trabalho, com o mesmo projeto em mente (sexta característica de um *boundary object*).

No estudo de caso não foi levantada, pelos engenheiros de software, nenhuma questão sobre a inutilidade da informação contida na ferramenta (foram apontadas algumas redundâncias apenas). Desta forma, podemos dizer que a ComunIHC-ES contém o essencial para manter a coerência de informação sobre a camada de interação (penúltima característica listada acima).

A última característica de um *boundary object* apresentada é: “Atua como ponte entre as áreas e não impõe a visão de uma”. Esta questão de imposição de visões foi uma outra preocupação que surgiu durante a elaboração deste trabalho, pois segundo a ordem apresentada, o trabalho de projeto da solução de IHC vem antes do trabalho de especificação das funcionalidades internas do software. Não era intenção da ComunIHC-ES impor decisões de IHC que engessassem o trabalho dos engenheiros de software já ditando a forma de eles trabalharem na especificação das funcionalidades do software. Esta questão foi levantada durante as entrevistas no estudo de caso e todos os participantes afirmaram que o que está representado na ComunIHC-ES não impõe decisões de projeto de engenharia de software. Ou seja, a ferramenta só limita o trabalho dos engenheiros de software relacionado à camada de interação. Ela não interfere em decisões de engenharia relacionadas a, por exemplo, modularização, re-uso, componentização, etc.

Quanto aos tipos de *boundary objects* citados acima (repositórios, tipo ideal, limites coincidentes e formulários padronizados), acreditamos que a ComunIHC-ES chega mais perto do “Tipo Ideal”, pois ela contém representações (inclusive um diagrama) que descrevem em um nível abstrato a solução de IHC.

5.3 ComunIHC-ES: uma Ferramenta Epistêmica

Como foi visto na seção anterior, a ComunIHC-ES faz o papel de um *boundary object* entre as áreas de IHC e engenharia de software. Ela é uma representação a ser compartilhada entre as duas áreas, com o papel de servir como base para a discussão sobre o projeto da interação. Como foi visto na descrição da ferramenta (capítulo 3), ao utilizar a ComunIHC-ES, os profissionais de IHC têm que analisar o projeto da interação para levantar as decisões que estão ali representadas e que precisam ser ressaltadas para os

engenheiros de software via as perguntas que estão disponíveis na parte 3 da ferramenta (O que é isto? Para que serve isto? etc). Ou seja, antes de apresentar o projeto para os engenheiros, o profissional de IHC tem que passar pelo processo de analisar o seu próprio trabalho e verificar o que não está explícito na modelagem, mas que precisa ser destacado para o trabalho dos engenheiros de software. Todo este processo leva este profissional a refletir novamente sobre a solução por ele definida, dando-lhe oportunidade a adquirir novos conhecimentos sobre a solução ou até mesmo sobre o problema sendo resolvido. Desta forma, podemos dizer que a ComunIHC-ES tem a capacidade de ser uma ferramenta epistêmica. Essa característica da ComunIHC-ES manifesta-se ainda em dois outros momentos.

Quando a solução de IHC é apresentada aos engenheiros de software, na medida em que estes buscam entender o que está representado na ComunIHC-ES, eles aumentam o seu conhecimento sobre o sistema interativo a ser desenvolvido, ou seja, sobre como o usuário poderá alcançar suas metas ao interagirem com a aplicação. Esse conhecimento serve como insumo também para a atividade de especificação das funcionalidades do software, ajudando os engenheiros de software a tomarem as suas decisões.

O próprio profissional de IHC também pode ser levado novamente a refletir quando é questionado pelos engenheiros de software sobre as decisões que estão representadas na ComunIHC-ES. Por exemplo, o engenheiro de software pode, durante o uso da ferramenta, apontar restrições de arquitetura que impedem que determinada escolha de IHC possa ser implementada. Neste momento, o profissional de IHC deve analisar outras possíveis soluções e negociar a viabilidade com os engenheiros de software. Desta forma, ele estará refletindo sobre o problema e suas possíveis soluções.

Assim, podemos dizer que a ComunIHC-ES tem a capacidade de ser uma ferramenta epistêmica em três momentos. De fato, seria interessante avaliar, num trabalho futuro, o que confere a *boundary objects* o caráter de ferramenta epistêmica: uma característica intrínseca do artefato ou o seu uso, como por exemplo como apoio a comunicação e a negociação.

5.4 Uso da ComunIHC-ES na Prática de Desenvolvimento de Software

A ComunIHC-ES, como já foi dito, não pretende substituir outras formas de comunicação utilizadas entre profissionais de IHC e engenheiros de software durante o processo de desenvolvimento. Continua sendo válida a comunicação face-a-face, o uso de, por exemplo, cenários de uso, *storyboards* e protótipos. O objetivo desta ferramenta é contribuir ainda mais para a troca de informações entre estas áreas.

Através do estudo de caso realizado, pudemos perceber que a criação de uma ferramenta computacional irá facilitar ainda mais o uso da ComunIHC-ES na prática de desenvolvimento, ainda mais se ela contar com alguns tipos de automatizações, como por exemplo, a geração de esqueletos de diagramas UML a partir das correspondências citadas na seção 3.4 (tendo sempre em vista a não interferência em decisões de projeto relativos à engenharia de software). Além disto, a ferramenta computacional também poderá contar com o recurso de geração de cenários de uso, baseados no projeto de IHC, para apoiar ainda mais o entendimento da solução pelos engenheiros de software.

Durante o estudo de caso, o processo de leitura da ComunIHC-ES proposto na seção 3.5 foi utilizado pelos engenheiros de software, que ressaltaram que este é um importante recurso associado à ComunIHC-ES. Segundo eles, o processo é útil para guiar a leitura de toda a documentação produzida. Entretanto, apesar destas opiniões, acreditamos que um processo mais detalhado talvez pudesse auxiliar ainda mais os engenheiros de software, principalmente quando o domínio modelado for mais complexo. Então, o fornecimento de um processo contendo uma orientação de leitura da ComunIHC-ES mais detalhada precisa ser analisado.