

1 Introdução

O design¹ de um artefato de software normalmente envolve a compreensão do problema a ser modelado, a identificação de possíveis alternativas de solução para este problema, a análise destas alternativas e a tomada de decisões sobre quais soluções serão usadas na construção do artefato final. Os produtos finais desse processo, o artefato e sua especificação, representam apenas parte do conhecimento empregado pelos projetistas durante o design. Eles representam a solução final escolhida para um problema de design particular, mas não representam, por exemplo, as razões que levaram os projetistas a escolher esta solução entre outras alternativas de solução disponíveis, e porque estas outras alternativas foram descartadas. Ou seja, eles não representam o *design rationale* que deu origem ao artefato em questão.

Design rationale inclui todo o conhecimento empregado durante o design de um artefato, expressando não apenas as decisões tomadas, mas também as razões por trás de cada decisão, incluindo sua justificativa, outras alternativas consideradas e os argumentos que conduziram à decisão (Lee, 1997). Estas informações podem ser muito valiosas, até mesmo críticas, para as várias pessoas que lidam com o artefato.

Geralmente o *design rationale* não é bem documentado, o que leva a um alto grau de comunicação verbal entre as pessoas que precisam trabalhar com um artefato para que elas possam entender o raciocínio seguido pelo projetista. Por exemplo, isto é fundamental durante a manutenção de um artefato de software projetado por uma outra pessoa, ou quando tentamos reusá-lo no contexto de um novo design. Isto é verdade ainda no caso de um único projetista, uma vez que em muitos casos, após um longo período de tempo, o projetista pode não se lembrar de todo o *rationale* que ele mesmo usou no design de um determinado artefato de

¹ Decidimos manter o termo design em inglês, por ser este um termo comumente usado pela comunidade de computação em vários países de língua não-inglesa.

software. Portanto, registrar o *design rationale* durante o design de um artefato de forma a permitir o seu reuso é uma questão crítica.

O registro de *design rationale* permite que as experiências e o conhecimento investidos em um design não sejam perdidos. Toda a experiência adquirida pelas pessoas durante o design de um artefato pode ser transmitida e potencializada pelo uso do *design rationale* registrado em novos designs. Em um modelo de reutilização, tanto as boas decisões tomadas quanto as ruins são interessantes, pois as primeiras propõem soluções válidas para um tipo de design, e as outras propõem soluções que devem ser evitadas. Além disso, na fase de reuso podemos evoluir de modelos de soluções específicas para problemas genéricos. Ou seja, não é necessário “reinventar a roda” a cada momento. Soluções escolhidas para um artefato podem ser reusadas no design de outros artefatos (Lacaze, 2005).

De acordo com Lacaze (2005), os projetos futuros poderão lucrar (gratuitamente) de partes de *design rationale* prontos para serem reusados. Assim, quanto mais projetos existirem usando *design rationale* mais os seus uso e reuso serão importantes. Este reuso será aumentado à medida que os “padrões” de soluções tornarem-se mais claros. Com eles, os projetistas poderão se concentrar em problemas mais complexos e então melhorar a qualidade final do artefato. Desta forma, acreditamos que o reuso de *design rationale* pode melhorar a produção de software, melhorando a qualidade dos artefatos produzidos.

Neste trabalho, propomos um modelo de representação de conhecimento e a arquitetura de um ambiente de design integrado para apoiar a representação e o uso de *design rationale* durante o design de artefatos de software. O termo “design” é usado com um significado mais amplo do que aquele comumente usado em Engenharia de Software. Nesta tese, design refere-se à produção de um artefato de acordo com algum método, desde sua concepção até a sua realização, e não apenas a uma única fase do processo de desenvolvimento de software.

Neste capítulo, apresentamos inicialmente a concepção de design adotada nesta tese (seção 1.1) e o problema de pesquisa abordado (seção 1.2). Em seguida, apresentamos nossa abordagem para o problema estabelecido (seção 1.3) e as motivações para esta pesquisa (seção 1.4). Por fim, apresentamos os objetivos (seção 1.5) e a estrutura desta tese (seção 1.6).

1.1. Design

Atualmente, não existe na literatura um consenso quanto às definições de design e processo de design. Diferentes definições podem ser encontradas em (Goel & Pirolli, 1989), (Hubka & Eder, 1996), (Winograd, 1996) e (Bento & Feijó, 1997). Os diversos autores que tratam o tema geralmente não produzem uma definição simples, mas, contribuem para uma resposta fornecendo uma perspectiva sobre o que as pessoas fazem durante um design. Por exemplo, os autores Simon (1981) e Schön (1983) apresentam duas perspectivas bem diferentes sobre o que significa fazer design.

Segundo Simon (1981), toda pessoa que planeja cursos de ação visando transformar situações existentes em situações preferidas fazem design. A atividade intelectual que produz artefatos materiais não é fundamentalmente diferente daquela que prescreve remédios para um paciente doente, ou daquela que planeja novas vendas para uma companhia ou de uma política de bem-estar social para um estado. Design, interpretado assim, é o núcleo de todo o treinamento profissional; é a marca principal que distingue as profissões das ciências. As escolas de engenharia, assim como as escolas de arquitetura, de negócios, de educação, de direito e de medicina estão todas relacionadas com o processo de design.

Para Schön (1983), o processo de design é inerentemente complexo. Toda escolha feita pelo projetista tem efeitos intencionais e não intencionais. Design não é um processo de planejamento e execução cuidadoso, mas preferencialmente um diálogo, no qual o parceiro de diálogo – o próprio objeto sendo projetado – pode gerar interrupções e contribuições inesperadas. Ou seja, não existe um caminho direto entre a intenção do projetista e o resultado. Quando o projetista trabalha sobre um problema, ele está continuamente desenvolvendo um caminho dentro deste problema, fazendo novas apreciações e entendimentos à medida que ele faz novos movimentos.

Nesta tese, adotamos um conceito de design mais restrito, no qual o foco está na produção de um artefato. No escopo deste trabalho, design é uma atividade na qual se busca criar um artefato seguindo algum método ou processo. Tratamos especificamente de design baseado em modelo, ou seja, design visto como um processo de instanciação de um metamodelo, no qual os artefatos produzidos são

modelos abstratos comunicando diferentes características de um software. Portanto, a abordagem para a representação e o uso de *design rationale* proposta neste trabalho não é orientada especificamente para o processo de design, mas sim para a atividade de design envolvida na produção de um artefato de software.

1.2. O Problema

Embora *design rationale* tenha um valor potencial para o reuso de artefatos, ele não tem sido muito usado durante o design de artefatos de software. Uma das razões, apesar de muita pesquisa, é o consumo de tempo e o custo que geralmente são requeridos para a captura e representação de *design rationale*. Um dos problemas principais é a sobrecarga de trabalho que pode interferir no andamento natural das atividades de design e causar atraso em todo o processo de design. Este problema pode ser explicado pela falta de uma ferramenta integrada que possa tornar a captura, representação e uso de *design rationale* parte do processo de design de software. Segundo Lacaze (2005), um outro problema é que a rentabilidade do *design rationale* é difícil de mensurar, uma vez que sua lucratividade só pode ser percebida apenas ao longo de vários projetos. Na sua fase de concepção, o *design rationale* adiciona custo ao projeto, mas de um projeto para o outro este custo é reduzido. Ou seja, se comparado ao investimento inicial, este custo adicional decresce à medida que é usado em novos projetos, de tal forma que o custo total de projeto é menor para os projetos subsequentes.

Geralmente, a abordagem de representação usada determina como o *design rationale* pode ser capturado e reusado em novas situações de design. Sendo assim, esta tese aborda o seguinte problema de pesquisa:

Como o design rationale pode ser representado de forma a apoiar o reuso de designs de artefatos de software?

A representação de *design rationale* tem sido estudada exaustivamente. As abordagens existentes para a representação de *design rationale* normalmente variam entre o formal (aqui entendido como computacionalmente tratável) e o informal (não computacionalmente tratável). Uma abordagem formal permite que o computador processe os dados, mas nem sempre a informação resultante é apresentada em um formato compreensível para um ser humano. Já uma abordagem informal fornece dados em formatos facilmente gerados e

compreendidos pelos seres humanos, mas que não podem ser processados pelo computador. As abordagens híbridas tentam explorar as vantagens de ambas, combinando estrutura semântica (formal e computável) e conteúdo (informal) para representar o conhecimento gerado.

O nível de formalidade usado em uma representação de *design rationale* híbrida geralmente depende do propósito de quem usa esta representação, uma vez que em algum nível todas as abordagens são formais e informais. No entanto, acreditamos que é possível acrescentar um nível de formalização a mais em uma representação de *design rationale* híbrida, a fim de possibilitar o reuso de seu conteúdo no design de novos artefatos. Este nível de formalização é obtido pelo uso da semântica formal fornecida pelos métodos de design na representação de *design rationale*, conforme será abordado ao longo desta tese.

1.3. A Abordagem

Embora existam várias abordagens diferentes para representar *design rationale* relatadas na literatura, a maioria delas gera representações incompletas ou informais, não permitindo o reuso efetivo de *design rationale* no design de novos artefatos. Além disso, quando aplicadas a artefatos definidos formalmente (tais como artefatos de software), sua informalidade nos impede de considerar, automaticamente, as alternativas prescritas pelos métodos de design utilizados, assim como incorporar suas restrições. Ou seja, não é possível avaliar a semântica do artefato fornecida pelo modelo formal que o descreve.

Para muitos domínios de conhecimento, e em particular, no domínio de design de software, existem modelos formais que descrevem os artefatos deste domínio. Em muitos casos, estes modelos apresentam descrições semânticas que permitem inclusive raciocínio sobre os artefatos produzidos. Nesta tese, este tipo especial de domínio de design é chamado “design baseado em modelo”.

Design baseado em modelo é uma categoria de problemas de design que podem ser vistos como um processo de instanciação de um metamodelo. Este metamodelo representa os modelos formais usados para descrever os artefatos produzidos. Um exemplo de tal modelo formal em design de software é a linguagem de especificação UML (OMG, 2003) usada para descrever um diagrama de classes.

Em nossa abordagem para representação de *design rationale* propomos o uso dos modelos formais dos artefatos na instanciação do modelo de representação apresentado nesta tese. Esta abordagem permite a criação de um processo semi-automático para representar *design rationale*. Neste processo, o ambiente de design integrado usa o modelo formal do artefato para sugerir opções de design a cada passo do design, e para registrar as escolhas feitas pelos projetistas, usando uma linguagem de propósito especial que será descrita mais adiante. Dependendo da riqueza do modelo formal do artefato que está sendo projetado, o ambiente pode sugerir novas alternativas de design e também validar a consistência das decisões tomadas pelo projetista.

Teoricamente, quando existe uma semântica formal para os artefatos que estão sendo projetados, é possível construir sistemas capazes de gerar artefatos de forma automática. No entanto, não tratamos de automação de design nesta tese. O design é apoiado considerando a intervenção do usuário na definição das opções de design ou das operações necessárias para construir o artefato.

1.4. Motivação

Uma representação de *design rationale* definida formalmente, enriquecida com a semântica formal dos artefatos sendo projetados, nos permite dar significado explícito ao conteúdo registrado, tornando mais fácil para as máquinas automaticamente processarem e integrarem as informações registradas. Isto possibilita uma nova forma de reuso de design, realizada através da integração de *design rationales*. Esta integração é possível quando as representações de *design rationale* são geradas usando o mesmo esquema de representação (ou esquemas compatíveis), e os artefatos são construídos usando o mesmo modelo formal e representam o mesmo domínio de aplicação. Por exemplo, é possível integrar os *rationales* de dois diagramas de classes UML (modelo formal) diferentes, criados para representar o modelo conceitual de uma loja de CDs (mesmo domínio), sendo estes *rationales* representados usando o vocabulário de uma mesma ontologia.

Esta representação mais formal de *design rationale* possibilita um tipo de reuso de software em um nível mais alto de abstração, onde *rationales* podem ser integrados e re-empregados no design de um novo artefato. Começando com

artefatos existentes, o projetista pode analisar seus *rationales* e decidir integrá-los para obter uma solução de design mais completa. Assim, ele pode revisar ou estender a solução obtida, adicionando novas alternativas ou fazendo novas escolhas com respeito às alternativas já definidas, gerando um novo *design rationale*. Deste ponto de vista, ambas as tarefas de manutenção e evolução de artefatos de software podem ser consideradas simplesmente como uma continuação de um design anterior, capturado em um dado *design rationale*.

Quando tais representações de *design rationale* estão disponíveis em um ambiente distribuído, é possível considerar a colaboração entre projetistas com apoio semi-automatizado, no qual representações de *design rationale* podem ser pesquisadas, recuperadas e integradas durante o design de um novo artefato. Tal disponibilidade pode ser a base para design colaborativo (e ainda participativo), entre projetistas trabalhando no design de um dado artefato.

1.5. Objetivo da pesquisa

Design rationale tem um potencial significativo para apoiar o reuso de artefatos de software, mesmo não sendo muito usado com esta finalidade. Nesta pesquisa decidimos investigar como *design rationale* pode ser representado de forma que operações computáveis possam ser realizadas sobre o seu conteúdo para apoiar o design de novos artefatos. Em resumo, o principal objetivo é mostrar que com uma abordagem de representação apropriada, as representações de *design rationale* podem ser processadas computacionalmente para apoiar o reuso de designs baseados em modelo. Para isto, propomos a ontologia Kuaba² (Medeiros et al., 2005a, b) como uma linguagem de representação formal para *design rationale*, e o uso dos modelos formais de artefatos, como definidos pelos métodos de design, como parte do processo de instanciação desta ontologia.

Os resultados esperados desta pesquisa são:

- uma linguagem expressiva para descrever *design rationale*;
- um conjunto de operações computáveis para apoiar o uso de *design rationale* no design de novos artefatos;

² Kuaba significa “conhecimento” em Tupy-guarani, a língua de um dos povos nativos do Brasil.

- um conjunto de regras para a realização de inferências e validações sobre o *design rationale* registrado;
- a definição de uma representação de *design rationale* que considere a semântica formal dos artefatos fornecida pelos métodos de design de software.

1.6. Estrutura da Tese

O restante desta tese apresenta, de forma mais detalhada, nossa abordagem para a representação de *design rationale* e a arquitetura conceitual de um ambiente de design para apoiar a captura, representação e o uso de *design rationale*. O capítulo 2 apresenta as diversas abordagens existentes na literatura para a representação de *design rationale*. Este capítulo apresenta, também, os principais problemas observados nestas abordagens e os requisitos que julgamos necessários à representação de *design rationale*.

O capítulo 3 apresenta a ontologia Kuaba, criada como um modelo de representação de conhecimento para o domínio de *design rationale*. Inicialmente, apresentamos um resumo do formalismo lógico usado na descrição da ontologia. Em seguida, apresentamos o vocabulário da ontologia Kuaba e as regras definidas para apoiar a representação de *design rationale*.

O capítulo 4 contém alguns exemplos mostrando como os modelos formais dos artefatos, prescritos pelos métodos de design de software, podem ser usados no processo de instanciação da ontologia Kuaba para representar *design rationale*.

O capítulo 5 aborda o uso de *design rationale* e apresenta alguns exemplos de uso das representações de *design rationale* geradas com Kuaba.

O capítulo 6 apresenta a arquitetura conceitual do ambiente de design integrado proposto nesta tese e as operações definidas sobre as representações de *design rationale* para apoiar o reuso de design de software.

O capítulo 7 apresenta as conclusões sobre este trabalho, indicando suas principais contribuições, alguns trabalhos relacionados e sugestões de trabalhos futuros.

Nos apêndices são apresentados: um resumo da sintaxe da linguagem usada para implementar as operações definidas neste trabalho; e os códigos contendo a definição da ontologia Kuaba e a implementação dessas operações.