

2 Fundamentação Teórica

2.1. Design em IHC

O *Design* não está lá, nem cá; está no meio, ora pendendo para um lado, ora para o outro, num movimento que se assemelha a uma dança graciosa. É preciso saber o momento e o lugar certo para ser rígido ou flexível, de acordo com as intenções da performance e com a guia musical. O *Design* é assim: não cria, mas recombina; não define, mas propõe; não julga, mas avalia; não é arte, mas também gosta de dançar... Para entendê-lo, é preciso dar o primeiro passo!

Instituto Faber-Ludens de Design de Interação¹

Denomina-se design qualquer processo técnico e criativo relacionado à configuração, concepção, elaboração e especificação de um artefato.

Várias têm sido as visões e conseqüentemente as definições sobre design. Na década de 1970, o design era visto por Horst Rittel (Rittel & Weber, 1973) como um processo de negociação e deliberação, que fundamentalmente lida com incerteza e conflitos. "Atingir objetivos dentro de restrições" foi a definição de design dada por Dix e coautores (2004). Moran e Carroll (1996) definem a atividade como um processo de criação de artefatos tangíveis para atender a necessidades humanas intangíveis.

Löwgren e Stolterman (2004) entendem design como um processo único. Único, pois o contexto de design é único. Os usuários (clientes) com suas necessidades e desejos, um designer com seus conhecimentos e interpretação das necessidades e desejos dos usuários, a tecnologia e as ferramentas disponíveis para o processo de design, todos estes elementos evoluindo continuamente, compondo um contexto único.

Assim como Schön (1983), Löwgren e Stolterman (2004) veem o design como um processo inerentemente complexo. Não como 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 (único ou correto) entre a intenção do projetista e o resultado (Figura 2.1).

¹ <http://www.faberludens.com.br/pt-br/node/3>

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.

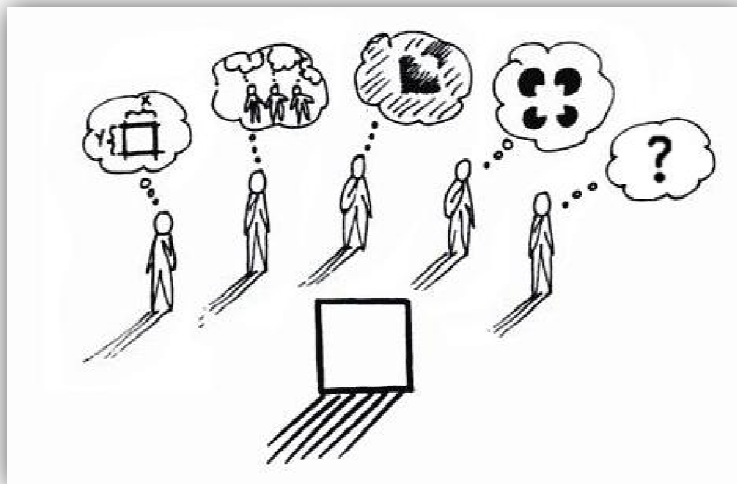


Figura 2.1: Design é um processo interpretativo onde não há verdades únicas ou absolutas. (Figura adaptada de Preece et al., 2005)

De fato existem diversas definições de design, mas, em geral, pode-se afirmar que design é a atividade intelectual (complexa) de conceber e descrever um produto a partir dos possíveis requisitos ou objetivos de seus potenciais usuários (de Souza et al., 1999).

O Design de Interação é uma área dentro do design que trata especificamente do projeto de artefatos interativos, ou seja, objetos que participam ativamente da ação humana. Como exemplo, é possível citar os sistemas computacionais. O Design de Interação pesquisa, cria e adapta esses artefatos para a interação humana. Especificamente na área de IHC, Preece e coautoras (2005) afirmam que design de interação é o design de produtos interativos que fornecem suporte às atividades cotidianas das pessoas, seja no lar ou no trabalho. Winograd (1997) entende o design de interação sob uma perspectiva linguística, “O projeto de espaços para comunicação e interação humana.”

É importante ressaltar que há diversos entendimentos quanto ao processo de design, seu ciclo de vida ou ainda modelo que contemple o design ou projeto de sistemas interativos. Alguns provenientes da engenharia de software, outros da área de IHC. De acordo com Paula (2007), IHC e engenharia de *software* “endereço” perspectivas diferentes na construção do sistema interativo para atender às necessidades do usuário final, concentrando-se em aspectos diferentes do desenvolvimento de software. Examinando a bibliografia dessas duas áreas (Dix et al., 2004; Preece et al., 2002; Sommerville, 2003), percebe-se que, enquanto a engenharia de software tem seu foco sobre a especificação, implementação e teste das funcionalidades e da arquitetura do sistema interativo, a área de IHC se concentra no projeto do sistema sob a perspectiva do usuário: as características do usuário, suas necessidades e objetivos, bem como o seu ambiente de trabalho e as tarefas que precisam ou desejam realizar através do sistema. A partir dessas definições, o designer de IHC parte para o projeto e prototipação da interface e

interação do sistema interativo, tendo como atividade constante a avaliação dos artefatos produzidos (Preece et al., 2002).

Nesse trabalho, o design de sistemas interativos é estudado sob a ótica da área de IHC, conseqüentemente, nosso foco recairá sobre as principais propostas, métodos e modelos de suporte ao projeto de sistemas interativos da área de IHC.

2.2.1. Processos de design

O processo de design contempla o conjunto de atividades (e suas relações) que ocorrem durante a concepção de um artefato. De acordo com Nühr o processo de design de interação se baseia em quatro conceitos chave: motivação, significado, modo e mapeamento (vide Figura 2.2).

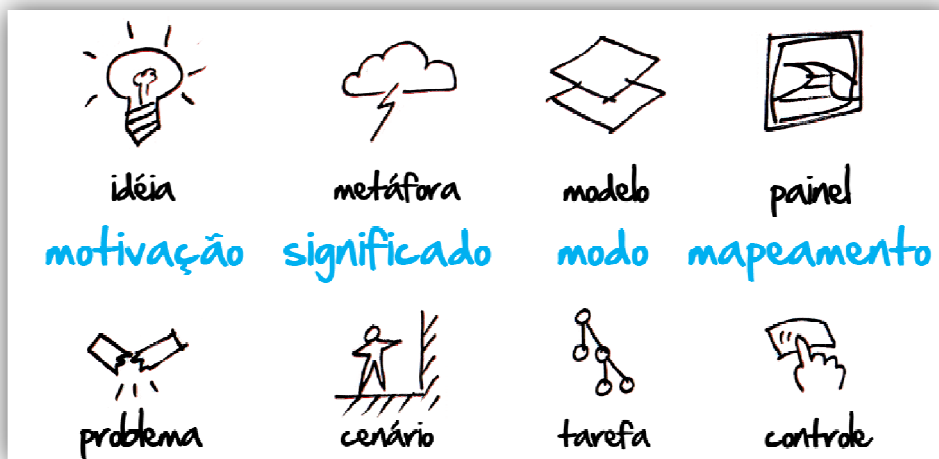


Figura 2.2: O processo de design de interação por Nühr (2009).

Para Preece e coautoras (2002), quatro atividades principais compõem o processo de design de interação (vide Figura 2.3):

- Identificação de necessidades e estabelecimento de requisitos: Quem são os usuários-alvo e que tipo de suporte um sistema interativo pode lhes oferecer para a realização de suas atividades?;
- Desenvolvimento de designs alternativos que satisfaçam esses requisitos;
- Construção de versões interativas;
- Avaliação: o sistema é útil, fácil de usar, robusto, fácil de aprender, comunica bem a ideia do designer, é “atraente”...

Em cada uma dessas etapas, pode haver a necessidade de revisão e alteração nas anteriores, já que o “problema” de design é um alvo móvel.

As autoras ressaltam três características-chave quanto ao processo de design: o envolvimento dos usuários no processo, a identificação e registro dos princípios de usabilidade e das metas decorrentes da experiência do usuário

logo no início do projeto e o fato de que a iteração é inevitável em todas as etapas do processo.



Figura 2.3: Processo de design de interação proposto por Preece e coautoras (2002).

Hix e Hartson (1993) propõem um ciclo de vida para desenvolvimento de software que tem como ponto forte a interação com o usuário durante o processo. O modelo estrela, composto por seis etapas (vide figura 2.4), difere dos outros modelos até então propostos por não pressupor ordem entre as etapas e por ter a participação dos usuários durante o projeto do sistema. O modelo pressupõe a avaliação contínua durante o processo e a interação com o usuário. A avaliação por sua vez estende os conceitos de validação, verificação e teste.

Na proposta do modelo/processo, não há necessidade de se especificar todos os requisitos antes de começar a trabalhar no projeto do sistema. Pode-se iniciar com um protótipo rápido das telas do sistema e depois ir incrementando-o com novos requisitos identificados. De acordo com esse modelo, o ciclo de vida de desenvolvimento pode ser iniciado em qualquer fase indicada pelas setas de entrada. Cada fase pode então ser seguida por qualquer outra. Esse modelo tem como ponto central a participação do usuário sob a forma de avaliações que são feitas ao término de cada etapa ou na passagem de uma fase para outra.



Figura 2.4: Modelo Estrela de Hix e Hartson (1993).

Para Dix e coautores (2004), o processo de design é baseado em quatro fases e um ciclo de interação. Este processo é apresentado na Figura 2.5 onde os retângulos representam as fases e as setas o fluxo do processo. A primeira fase é onde se estabelece o que exatamente é necessário para o sistema. Na etapa seguinte, a análise, os resultados de entrevistas e observação precisam ser ordenados de alguma forma que permita a identificação dos elementos-chave e a comunicação com outros estágios do processo de design. A fase de design, considerada uma etapa central, é onde se move do que se quer para como fazê-lo. Nessa etapa é necessário registrar todas as decisões de design tomadas, adotando alguma notação e/ou método. Sabendo-se da complexidade do design, não se espera que o alvo (design adequado) seja atingido na primeira tentativa, vem então a fase de iteração e prototipação onde o design proposto é avaliado para ver quão bem ele “funciona” e onde pode ser melhorado. Finalmente, com uma versão considerada adequada, é necessário programar a versão e instalá-la, de acordo com os autores. Aqui há a codificação, por vezes construção de hardware e escrita de documentação e manuais.

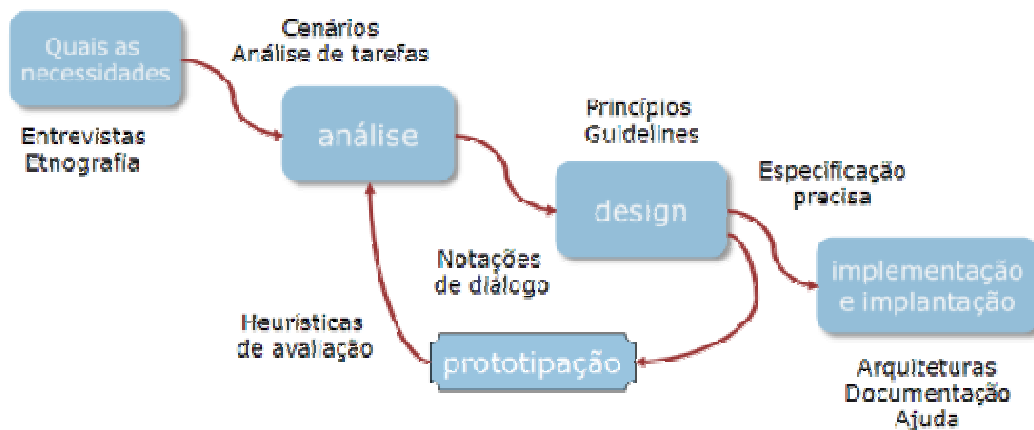


Figura 2.5: Processo de design de interação proposto por Dix e coautores. Traduzida de Dix et al (2004).

Garcia (1992) faz uma análise do processo de design, tendo em vista a documentação deste processo. Para o autor, o design usualmente se desenvolve em quatro fases: design conceitual, design preliminar, desenvolvimento do design e construção de desenhos, cada fase com necessidades específicas de documentação.

Na fase de design conceitual, a equipe de design trabalha sobre os conceitos básicos e os requisitos do projeto, registrando os requisitos que governarão o design. Na fase de design preliminar, é proposto um conjunto de alternativas que satisfaçam o conjunto de requisitos identificados durante o design conceitual, após negociações e aprovações entre os participantes do projeto. A documentação gerada nessa fase contempla as propostas de design aceitas a serem detalhadas. A fase de desenvolvimento de design revisa e detalha as propostas da fase anterior para a elaboração dos desenhos. De acordo com os autores, esse processo é fortemente iterativo, sendo contínuas as revisões e retomadas de fases anteriores.

User-Centered Design (UCD) é uma filosofia de design em que as necessidades, desejos e limitações dos usuários finais a que se destina o produto do design são fortemente levados em consideração, em cada etapa do processo de concepção. UCD assume que os usuários são os mais indicados para fornecer as entradas realmente necessárias ao processo de design, já que irão utilizar de fato o produto. A principal diferença entre UCD e outras abordagens é que metodologias baseadas em UCD tentam aperfeiçoar a interface usuário com base no que os usuários podem, querem, ou precisam para trabalhar, em vez de forçar os usuários a mudar a forma como eles trabalham para adequar-se às propostas dos desenvolvedores (Abrás et al., 2004).

A expressão *User-Centered Design* surgiu na década de 1980, no laboratório de pesquisa de Donald Norman, tendo suas primeiras referências publicadas no livro *User-Centered System Design: New Perspectives on Human-Computer Interaction* (Norman e Draper, 1986) e no livro *The Psychology Of Everyday Things* (Norman, 1988).

Há um padrão internacional proposto, ISO 13407: *Human-Centered Design* (HCD) *process*, que serve de base para muitas metodologias UCD. Esse padrão define um processo geral para a inclusão de “atividades centradas nos utilizadores” ao longo de todo o ciclo de vida de desenvolvimento, sem especificar exatamente nenhum método.

O processo contempla um ciclo composto por quatro atividades:

- Especificar o contexto de uso: Identificar as pessoas que irão utilizar o produto, o que elas irão usar, e em que condições.
- Especificar os requisitos: Quais são os objetivos dos usuários que devem ser alcançados para o sucesso do produto.
- Desenvolver soluções de design: Pode ser realizada em etapas, partindo de uma proposta inicial, rudimentar, de design para o design final.

- Avaliar as opções de design: A etapa mais importante do processo. Idealmente realizada com base em testes de usabilidade e de qualidade junto aos usuários.

O processo é concluído e o produto entregue, uma vez que os requisitos foram atingidos pela proposta.

Em abordagens semióticas, toda aplicação computacional é concebida como um ato de comunicação, incluindo o designer no papel do emissor de uma mensagem para os usuários dos sistemas por ele criados (Andersen et al., 1993, de Souza, 2005). Nessas abordagens, o projeto de interfaces envolve não somente a concepção e design do modelo da aplicação, mas também a modelagem da comunicação desse modelo, de maneira a revelar para o usuário o espectro de usabilidade da aplicação. Na modelagem da comunicação, o designer tem que definir o que vai dizer (conteúdo) e como vai dizê-lo (expressão da mensagem). Sendo o processo de design de uma interface um ato comunicativo, ele requer que o designer entenda o problema, avalie as oportunidades de solução e tome decisões sobre esta solução.

Na Teoria da Engenharia Semiótica (de Souza, 2005), o processo de design é visto sob a perspectiva de reflexão durante a ação (*reflection-in-action*) (Schön, 1983), na qual cada problema é visto como sendo um problema único e mutável. Único porque cada problema é caracterizado por elementos do seu contexto que definem o problema. Mutável porque não apenas não existe uma única solução “correta”, mas também porque o entendimento do problema pode evoluir à medida que o projetista pensa mais sobre ele e explora diferentes caminhos de solução. A perspectiva da comunicação na qual se ancora a teoria permite que se entenda o processo de design como o processo de construção de uma “mensagem” a ser transmitida para o usuário via interface em tempo de interação. Dessa perspectiva emergem alguns elementos importantes que devem estar presentes e serem respondidos durante o processo de design:

- Quem são emissores e destinatários dessa mensagem e qual o contexto da mensagem de design?
- Do que esses destinatários precisam e/ou querem falar? Quais as suas necessidades?

Observa-se, a partir das diferentes propostas e visões sobre o processo de design, que de modo geral ele é composto por atividades comuns: identificar e conhecer potenciais usuários e suas necessidades (reais ou potenciais), estabelecendo requisitos, com base nesses requisitos propor e analisar possibilidades de design, avaliar as possibilidades e definir uma versão a ser implementada. Como fato tem-se que esse processo é fortemente iterativo, reflexivo e complexo. Seus produtos são de natureza intelectual e única, uma vez que são produzidos a partir da união de ingredientes continuamente mutáveis (necessidades, contexto e conhecimento). Para que seja possível compreender, executar, revisar e aprender com esse processo é necessário o uso de ferramentas de suporte. Ferramentas que auxiliem o designer a:

- Registrar (documentando) e comunicar informações e tomadas de decisão sobre o processo de design;
- Direcionar-se no processo, sem por outro lado ter seu caminho e suas possibilidades de análise limitadas ou engessadas;

- Identificar possíveis pontos de melhoria ou correção, bem como a retomada do processo a partir de destes pontos;
- Entender o processo de design, suas motivações e relações, refletir e aprender sobre o problema de design e sobre o processo em si.

2.2.

Representações, modelos e abordagens de design em IHC

As atividades que compõem o processo de design são iterativas e envolvem diferentes fatores que se inter-relacionam em um processo complexo. Elas carecem de ferramentas para sua compreensão e realização. Essas ferramentas, aliadas à criatividade e à experiência do designer devem permitir elaborar um design que atenda às necessidades e preferências dos usuários. Dentre as principais ferramentas de apoio à realização das atividades do processo de design estão os modelos.

“Em pesquisa científica, os modelos são uma das mais úteis ferramentas para criticar e avançar conhecimento” (de Souza, 2005 p. 107). São ferramentas epistêmicas que permitem ao designer “conversar” enquanto concebe uma solução de interação (Schön & Bennett, 1996). O uso de modelos leva o designer a refletir enquanto os constrói e manipula sobre o que está concebendo e sobre como a solução sendo concebida poderá afetar a vida dos usuários.

Modelos são ferramentas de mediação entre pessoas, de abstração e representação, permitindo o foco em partes do problema e a abstração de outras partes. Também são ferramentas epistêmicas por permitirem a construção, manipulação e avaliação da representação da entidade/processo representado (Knuuttila, 2005; Morrison & Morgan, 1999). Morrison e Morgan (1999) atribuem uma grande parte do valor epistêmico dos modelos ao processo de sua construção e manipulação.

Para Hoover e coautores (Hoover et al., 1991), um modelo é um recorte de algo que se deseja representar com níveis apropriados de abstração. Ou, de acordo com Diaper (2004), é uma descrição ou abstração do mundo real. Sowa (2000) entende modelo como uma representação do que existe no intelecto humano como solução para um determinado problema. Como cada ser humano tem um modo diferente de interpretar o mundo, existem diferentes modelos para o mesmo tipo de problema.

Pode-se dizer que a adoção de modelos segue, em geral, a técnica da Divisão e Conquista, nada mais sendo que a divisão de um “problema” em perspectivas diferentes para poder simplificá-lo (Dijkstra, 1965). Cada modelo, portanto, trata da realidade sob uma determinada perspectiva e em um determinado nível de detalhes. Essas características fazem dos modelos importantes ferramentas para a predição dos efeitos das ações tomadas no design, para a representação e registro do *rationale* do processo (documentação), como meio de comunicação entre os participantes do processo e para auxiliar a gerenciar a complexidade do processo de design. Através da modelagem, atinge-se uma série de objetivos, dentre eles: 1) Modelos auxiliam na visualização de um sistema como ele é ou se deseja que ele seja; 2) Modelos

permitem a especificação da estrutura e/ou do comportamento de um sistema e;
3) Modelos documentam as decisões tomadas.

Com base nestas definições, entende-se que um modelo é uma representação, ou descrição, comunicável, que contém propriedades relevantes, sob certa perspectiva, de um dado artefato, em um dado nível de abstração, projetada sobre um dado meio (e.g., cérebro, papel, programa, linguagem, numa base de conhecimento), tendo dentre outros objetivos: a predição dos efeitos das ações tomadas no design; a representação e registro do DR (documentação); servir de meio de comunicação entre os participantes do processo; auxiliar na gerência da complexidade do processo de design e servir de ferramenta de reflexão e compreensão sobre o "problema" em questão.

De fato, existem modelos disponíveis para representar praticamente todas as entidades e fases na concepção de um sistema interativo, tendo como base diferentes abordagens e teorias, sendo a grande maioria baseado em teorias cognitivas (Barbosa et al., 2003).

Dentre as áreas envolvidas no projeto e construção de sistemas interativos, pode-se citar a Engenharia de Software e a IHC. A primeira se concentra em analisar e projetar, com qualidade, as funcionalidades do sistema, enquanto a área de IHC, na qual este trabalho se situa, se concentra em apoiar o projeto de interfaces, tendo em vista as características, necessidades e preferências do usuário.

Na área de IHC, têm sido propostos diversos modelos para lidar com a complexidade do projeto de sistemas interativos. Modelos do usuário, das suas tarefas, do auxílio que lhe deve ser dado, do diálogo que com este deve ser estabelecido, do domínio em que ele atua, modelos para os que concebem a interação, modelos de arquitetura, de apresentação (Trættemberg, 2002; Paternò, 1999). Alguns destes modelos e representações comumente utilizados em IHC (Barbosa et al., 2002; Trættemberg, 2002; Paternò, 1999) são:

- **Cenários**, representações em linguagem natural de forma narrativa voltadas para a descrição da interação de uma pessoa com um sistema em um dado contexto de uso (Carroll, 1995).
- **Modelos de tarefas**, como GOMS (Card et al., 1983) e CTT (Paternò, 1999), considerados os mais citados e utilizados em IHC (Paternò, 1999; Barbosa et al., 2002; Diaper, 2004). De um modo geral, descrevem a estrutura das tarefas que devem ser apoiadas pelo sistema.
- **Modelos de interação e diálogo**, como MoLIC, (Paula, 2003) e DiaMODL (Trættemberg, 2003), considerados como a especificação dos possíveis diálogos (interações) que os usuários poderão travar com o sistema.
- **Esboços e storyboards**, protótipos de baixa fidelidade extensivamente adotados em IHC. Representam graficamente de forma ainda rudimentar o objeto em desenvolvimento (Landay & Myers, 1995).

- **Statecharts**², formalismo visual de modelagem concebido por David Harel (1987) para especificar sistemas complexos. Este formalismo é na verdade uma evolução dos diagramas de transição de estados, sendo especialmente indicados para descrever sistemas complexos do tipo *event-driven*, sistemas reativos de tempo real (Köhler et al., 2000).
- **Linguagens de modelagem da interface**, como IMML (Leite, 2003), visam à especificação abstrata da interface do usuário.

De acordo com Paula (2003), apesar dos diferentes modelos propostos para dar ao designer suporte durante o projeto de interfaces e da interação, não costuma ficar claro como esses modelos se inter-relacionam em um processo coerente de design. Também não existe um padrão na adoção dos modelos durante o processo de design. As equipes de design adotam representações e modelos distintos, conforme suas preferências e as características do projeto. Outro ponto identificado é que a comunicação sobre estas representações e modelos e suas relações requer outra representação ou modelo elaborado especificamente para esse fim.

Sendo assim, observa-se que as representações, modelos e linguagens investigadas não atendem a requisitos de um processo de design que expresse claramente a mensagem do designer sobre o projeto e conseqüentemente sobre suas intenções na interface. Tendo em vista o foco e nível de abstração de cada modelo, a simples agregação de algumas dessas representações, na tentativa de dar suporte ao processo como um todo, tem como problema a indefinição dos escopos de cada representação dentro do processo de design, causando a sobreposição de informações ou intervalos e incompatibilidade entre os produtos de um modelo e insumos de outro. Enquanto ferramentas epistêmicas, os modelos também devem ser fundamentalmente comunicativos. Eles devem “conversar” com o designer durante a concepção da solução interação (Schön e Bennett, 1996), e como também são ferramentas de mediação entre pessoas (Knuuttila, 2005; Morrison & Morgan, 1999), deve “conversar” com qualquer membro da equipe de design que precise dele extrair informações.

Além disso, de acordo com Paula (2003), o designer precisa ser apoiado também ao construir um modelo a partir de outros modelos utilizados no projeto, na integração das informações e destes modelos. O trabalho aqui proposto tem como um de seus objetivos viabilizar a integração das informações de modelos utilizados no projeto de sistemas interativos em um discurso que ofereça uma visão global do projeto, bem como as interpretações, intenções e razões que levaram o designer a uma proposta de solução de design.

2.3. Design Rationale

2.3.1. Definições

De acordo com o dicionário Cambridge, “*rationale*” é a razão ou intenção para um conjunto de pensamentos ou ações.

² Statecharts é atualmente parte integrante da UML (Rumbaugh *et al.*, 1999).

Segundo Gruber e Russel (1991), Design Rationale (DR) se refere a explicações sobre o raciocínio do design, ou seja, explicações de “como” e “por que” um artefato foi projetado de uma determinada maneira. De forma similar, Conklin e Burgess-Yakemovic (1995) definem DR como a informação que explica por que um artefato é estruturado da maneira que é e tem o comportamento que tem. Em Burge e Brown (2000), o conceito de DR é limitado pelos autores às decisões tomadas durante a fase de análise de projeto e as razões que levaram a tais decisões.

Para Lee (1997), DR 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, alternativas consideradas e os argumentos que conduziram à decisão tomada. Burge e Brown (2002a) estendem essa definição afirmando que, quando integrado ao processo de design, o DR deve cobrir também quando as decisões foram tomadas, em que ordem e o que levou de uma decisão a outra na ordem em que foram tomadas. Nesse contexto, um artefato pode ser visto como qualquer informação gerada, alterada ou usada durante o processo de desenvolvimento, podendo estar na forma de um modelo, um elemento do modelo como uma classe ou subsistema, um documento ou mesmo um arquivo contendo código executável (Francisco, 2004).

Com base nas diversas definições, entende-se que DR é o registro de todo o conhecimento produzido durante o processo de design de um artefato, reflexões, decisões, justificativas, alternativas, argumentações, critérios de avaliação, bem como a cronologia, relações entre as partes e o contexto no qual este processo está inserido.

2.3.2.

Motivações para o Design Rationale

O DR pode ser utilizado para diferentes propósitos. A seguir é definido um conjunto de possíveis utilizações para o DR (Monk et al., 1995; Burge & Brown, 2002; Daughtry et al., 2009):

- **Verificação:** o DR pode ser utilizado para verificar se as decisões de design e os produtos são o reflexo do que designer e usuários realmente querem.
- **Melhoria da qualidade:** o registro do DR contribui para a redução da arbitrariedade na tomada de decisões, visto que apresenta uma metodologia estruturada para apoiá-las, sendo que as justificativas e os argumentos são registrados e poderão ser recuperados em momentos posteriores;
- **Avaliação:** o DR pode ser utilizado para avaliar as várias alternativas de design discutidas durante o processo de design.
- **Manutenção:** usar o DR para localizar fontes de problemas no projeto, para indicar onde as mudanças são necessárias, a fim de modificar o projeto, e para assegurar que as opções rejeitadas não serão reimplementadas inadvertidamente.

- Reuso: o DR é usado para determinar que porções do projeto possam ser reutilizadas e, em alguns casos, sugerir onde e como serão modificadas.
- Ensino: o DR pode ser utilizado como recurso para o ensino de pessoas não familiarizadas com o design e/ou o sistema.
- Comunicação/Colaboração: o DR propicia uma melhor comunicação entre as pessoas (em seus diferentes papéis) envolvidas no processo de design, oferecendo desse modo suporte à colaboração.
- Apoio ao design: usar o DR para apoiar as discussões e decisões durante o processo de design.
- Documentação: o DR é usado para documentar todo o processo de design, envolvendo deliberações, alternativas discutidas, razões por trás das decisões e uma visão geral do produto.
- Suporte à criatividade: o DR pode dar suporte à criatividade fornecendo mecanismos para incentivar e capturar novas ideias durante o processo de design.

Acrescenta-se a essas funções o uso do DR como ferramenta epistêmica de reflexão para o designer sobre as questões de design.

2.3.3.

Critérios de caracterização de abordagens de DR

Shipman e McCall (1997) caracterizam as abordagens em três diferentes perspectivas de DR: argumentação, comunicação e documentação.

A perspectiva argumentativa considera que o DR está relacionado aos diversos pontos discutidos, aos argumentos apresentados e a forma como foram selecionados, ou seja, está relacionado ao raciocínio dos projetistas na resolução de problemas. O objetivo dessa perspectiva é registrar o DR para evidenciar e corrigir as deficiências dos argumentos, buscando a melhora na argumentação dos projetistas e conseqüentemente a melhora da qualidade do projeto.

A perspectiva argumentativa é, por vezes, insuficiente para justificar uma decisão dentre um conjunto de alternativas. Uma decisão de design pode ser influenciada por diversos fatores e requisitos, tanto qualitativos quanto quantitativos, alguns deles tendo maior importância ou prioridade que outros. Essas prioridades nem sempre são capturadas ou explícitas e mensuráveis a partir da perspectiva argumentativa (Tang, 2007).

De acordo com a perspectiva da comunicação, o DR está relacionado à captura e recuperação de toda comunicação que ocorre entre os projetistas, através das diversas mídias (áudio, vídeo, e-mail, anotações, diagramas etc.). (Shipman & McCall, 1997).

Na perspectiva da documentação, DR se refere ao registro da informação sobre as decisões de um projeto, ou seja, quais decisões foram tomadas, quais foram os motivos que conduziram a tais decisões e quem foram

os responsáveis por elas. É importante ressaltar que, nessa perspectiva, apenas as decisões finais e uma breve explicação sobre as mesmas são registradas.

Nas perspectivas de argumentação e documentação, a captura de informações é realizada de forma estruturada, facilitando o armazenamento e recuperação. Já na perspectiva de comunicação, as informações são capturadas da forma como são produzidas, sem uma estruturação ou ordenação, o que dificulta sua recuperação.

Considerando-se que nenhuma das três perspectivas é isenta de problemas, Shipman e McCall (1997) propõem a combinação delas para que se obtenham melhores resultados. Observa-se que a evolução das notações baseadas em argumentação se direciona para essa perspectiva combinada ou híbrida.

Já Louridas e Loucopoulos (2000) propõem uma caracterização geral das abordagens de DR com base nos seguintes critérios:

Simplicidade versus funcionalidade: A maioria das abordagens de DR visa dar suporte à deliberação e argumentação, especialmente abordagens caracterizadas por simplicidade e minimalismo de conceitos. Abordagens com um conjunto extenso de conceitos são mais “exigentes”. A justificativa argumentada por esse segundo grupo é que há um ganho na funcionalidade, facilitando a automação de operações a partir dos dados capturados.

Comunicação síncrona versus diacrônica: A comunicação interpessoal pode ser síncrona ou diacrônica. Comunicação síncrona ocorre em tempo real, por exemplo, comunicação face a face. Já a comunicação diacrônica ocorre sendo re-transmitida a partir de materiais registrados como documentação, e-mail, etc. Abordagens que visam à simplicidade tendem a dar suporte a comunicação síncrona.

Intrusiva versus não intrusiva: Uma abordagem é intrusiva se alterar o processo de design, enquanto a não intrusiva tende a ser adotada de forma transparente com pouca interferência. Abordagens complexas e rígidas tendem a ser intrusivas. Mesmo sendo simples, se a abordagem exigir uma forma rigorosa de trabalho, ela também tenderá a ser intrusiva.

Prescritiva versus não prescritiva: Uma abordagem pode ser considerada prescritiva quando propõe um conjunto detalhado de passos a serem seguidos durante seu desenvolvimento (forma rigorosa de trabalho). As não prescritivas oferecem apenas orientações, permitindo grande flexibilidade em sua aplicação. Em geral, abordagens não intrusivas tendem a ser não prescritivas e abordagens prescritivas são também intrusivas.

DR como subproduto versus “coproduto”: O DR produzido por uma abordagem pode ser um coproduto ou um subproduto do processo de design. O DR é um subproduto, se sua captura é um efeito colateral da abordagem de design. O DR é um coproduto se ele é visto como um produto separado, como um artefato em si. A abordagem *Questions, Options and Criteria* (QOC), por exemplo, entende DR como um coproduto, enquanto boa parte das outras abordagens o considera um subproduto.

Informal, semiformal ou formal: Em geral, não há abordagens completamente informais ou completamente formais; a grande maioria pode ser

considerada semiformal. Propostas formais favorecem à manipulação e operação por sistemas computacionais, sendo de difícil compreensão por pessoas, enquanto abordagens informais são compreensíveis por humanos, mas oferecem mais dificuldades para serem processadas computacionalmente. Os modelos semiformais buscam atender tanto a humanos quanto a operações computacionais.

Assim como Shipman e McCall (1997), acredita-se aqui que uma abordagem híbrida que integre características de argumentação, comunicação e documentação, e, também, voltada para a proveniência e rastreabilidade das informações identificando, explicando e relacionando as informações de design seja o mais adequado.

2.3.4. Design Rationale e os Espaços de Design

O espaço de design é um conjunto particular de problemas no design de um sistema, suas opções de solução e a relação entre eles. Para MacLean e McKerlie (1994), o espaço de design é uma representação explícita das opções alternativas de design e das razões para a escolha dentre essas opções. Entende-se que o espaço de design é composto por um espaço de problema e um espaço de solução. O espaço de problema compreende a definição dos requisitos que o sistema deve vir a atingir e o espaço de solução contempla as alternativas propostas de modo a atingir os requisitos levantados, bem como os artefatos propostos. Mais especificamente, o Espaço de problema do design contempla:

Quais são as atividades e suas respectivas tarefas, quando ocorrem, onde ocorrem, com que frequência, por quem são realizadas e em que contexto de uso e cultural;

Quem são os usuários, quais seus perfis e suas metas (flexíveis ou não, funcionais ou não, mais ou menos prioritárias, essenciais ou opcionais, vinculadas ao seu perfil ou gerais);

Se há, quem são os *stakeholders* e quais seus objetivos em relação às atividades que comporão o sistema, tendo em vista o direcionamento da empresa.

Já o espaço de solução cobre informações como:

- Quais as alternativas de design, quando ocorreram e por quem foram propostas, a qual atividade/tarefas elas se direcionam;
- Quais os fatores analisados em relação às propostas e como;
- Quais as propostas de design selecionadas e por quê;
- Que artefatos foram propostos e com base em que propostas;
- Que reflexões foram produzidas durante esse processo.

Para Burge e Brown (2000), o DR é visto como uma ponte entre as fases do processo de design (Figura 2.6). O espaço de requisitos contempla o conjunto de requisitos que definem o sistema a ser concebido. Esses requisitos,

são mapeados em objetivos e sub-objetivos (quando necessário), que podem ser satisfeitos a partir de uma ou mais alternativas de design, esse mapeamento contempla o espaço de *rationale*. Cada alternativa é então mapeada para um artefato ou requisito para a próxima etapa do processo, contemplando o espaço de design.

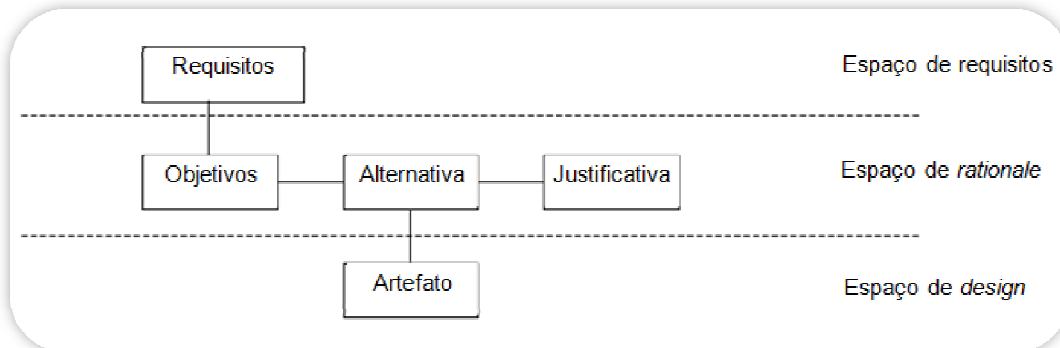


Figura 2.6: Design Rationale no Processo de design, ligando requisitos e design. Traduzida de (Burge e Brown, 2000)

Deste modo, como apresentado por Burge e Brown (2000), o DR funciona, ou pode funcionar, como uma ponte ligando as etapas do design. Sob esta perspectiva, acredita-se nesta pesquisa, que é parte do papel do DR integrar e relacionar o espaço de problema do design ao espaço de solução, apresentando desta forma não apenas propostas, justificativas e decisões, mas as motivações e o contexto no qual tudo está inserido, indo inclusive além da proposta apresentada por Burge e Brown (2000).

2.3.5. Requisitos para o DR

Com base nas características e nos principais problemas identificados nas abordagens mais conhecidas de DR, Medeiros (2006) gerou a seguinte lista de requisitos para a representação de DR:

Representação explícita de decisões: Uma mesma alternativa de projeto pode ser considerada, ou não, uma solução para questões de projeto diferentes. Portanto, as decisões tomadas sobre a aceitação ou não dessa alternativa como uma solução para cada questão de projeto devem ser registradas separadamente.

Distinção entre argumentos e justificativa final: Algumas vezes, o projetista decide aceitar uma ideia de solução que possui apenas argumentos contrários. Podem existir, também, ideias que, pelo número de argumentos a favor, parecem representar uma melhor solução de projeto do que a ideia escolhida pelo projetista para uma dada questão. Isto pode indicar que existe um *rationale* que não está sendo registrado ou que a escolha dessa ideia de projeto deve ser reconsiderada pelo projetista. Geralmente, o *rationale* que está faltando pode ser registrado como um novo argumento. Porém, em alguns casos, não se trata de um novo argumento para a ideia de solução proposta, mas, sim, de uma justificativa final para a decisão tomada pelo projetista, após analisar os argumentos apresentados. Por exemplo, essa justificativa final poderia registrar os motivos pelos quais o projetista decidiu usar uma ideia de solução, apesar de todos os argumentos apresentados para ela serem contrários à sua aceitação.

Integração da argumentação com as descrições dos artefatos gerados: A complexidade e o volume de informações presentes nas representações de DR tornam difícil para o projetista compreender as diversas ideias de solução e o resultado das decisões tomadas. A relação entre a argumentação registrada e o artefato produzido pode facilitar esta compreensão e ajudar o projetista a decidir se vale a pena reusar um artefato ou não.

Registro de informações sobre o contexto de projeto: Registrar apenas argumentação usada pelos projetistas não é suficiente para compreender o conhecimento usado por eles durante o projeto. É necessário, também, ter informações sobre o contexto de projeto, como, por exemplo, o método de projeto utilizado, as atividades realizadas e os responsáveis pelas decisões tomadas.

Integração do DR com o método de projeto: Representações genéricas e informais de DR dificultam o uso do *rationale* em novos projetos. A integração do *rationale* com a semântica formal definida pelos métodos de projeto torna o conteúdo registrado mais rico e expressivo, possibilitando a realização de operações computáveis, capazes de apoiar novos usos de DR, como, por exemplo, a combinação de *rationale* de artefatos similares.

2.3.6. Design Rationale como Ferramenta Epistêmica

Adler e Winograd (1992) argumentam que novas tecnologias serão mais eficazes quando projetadas para aumentar, em vez de substituir, as habilidades dos usuários.

Winograd e Flores (1986) entendem que artefatos computacionais são fundamentalmente linguísticos. Por essa razão, o desenvolvimento desses artefatos não pode seguir o mesmo paradigma de racionalismo técnico utilizado pelas engenharias. O racionalismo técnico resume a atividade de design ao enquadramento do problema de design em uma tipologia previamente definida, cuja solução seja conhecida pela adoção de leis, princípios, normas e valores estabelecidos (como uma receita de bolo). Os autores acreditam que cada artefato computacional é uma solução única a um dado problema. A construção e uso desses artefatos passam, necessariamente, por processos de comunicação e de interpretação onde o designer comunica sua visão de design pela interface do sistema e os usuários a interpretam fazendo uso do sistema.

O filósofo e pedagogo norte-americano Donald Schön, em sua obra *The Reflective Practitioner* (1983), oferece como alternativa ao racionalismo técnico o uso de ferramentas epistêmicas que auxiliem o designer a refletir e aprender sobre problema e solução de design. Essas ferramentas devem apoiar o designer, levando-o a interagir com o modelo do problema em um ciclo onde o modelo “responde” ao designer apresentando resultados surpreendentes, e o designer por sua vez, a partir da busca pela compreensão desses resultados, formula novas soluções de design. Trata-se de um processo reflexivo, batizado pelo autor de reflexão em ação (*reflection-in-action*).

Um grande desafio, apresentado por Daughtry e coautores (2009) para a área de pesquisa de DR, é a criação de um framework em que *rationale* e criatividade possam ser conectados. Para isso, a estrutura proposta para esse registro deve ser pensada e proposta de modo a favorecer o ato reflexivo.

2.3.7. Representações de DR

A seguir são apresentadas algumas das principais representações de DR baseadas em argumentação e algumas das ferramentas que as implementam.

O modelo Toulmin

Uma das bases teóricas para o DR foi o trabalho do filósofo Inglês Stephen Toulmin, que devotou seu trabalho ao estudo da natureza da argumentação. No livro *The Uses of Argument* (1958), Toulmin propõe um modelo descritivo de análise que especifica e relaciona os elementos potenciais de qualquer argumentação. Esse modelo, conhecido como Modelo de Toulmin, utiliza uma representação gráfica semiformal (Figura 2.9) para a visualização estrutural da argumentação, sendo composto por seis elementos inter-relacionados.

Três elementos são considerados fundamentais: o dado ou fato, denominado no modelo de *Datum* (**D**); a assertiva ou conclusão desejada, chamada de *Claim* (**C**); e a justificativa, chamada de *Warrant* (**W**). É possível apresentar um argumento contando apenas com estes elementos, cuja estrutura básica é: “a partir de um dado D, já que W, então C”. Há a possibilidade de se especificar em que condições a justificativa apresentada é válida ou não, indicando um peso para tal justificativa. Para isso, acrescenta-se ao argumento qualificadores modais chamados *Qualifiers* (**Q**). Da mesma forma, é possível especificar em que condições a justificativa não é válida ou suficiente para dar suporte à conclusão. Nesse caso, é apresentada uma refutação chamada de *Rebuttal* (**R**). A justificativa, que apresenta um caráter hipotético, pode ser apoiada por uma alegação que lhe dê suporte, denominada *Backing* (**B**) ou conhecimento básico. O *backing* é uma garantia baseada em alguma autoridade, uma lei jurídica ou científica, por exemplo, que fundamenta a justificativa.

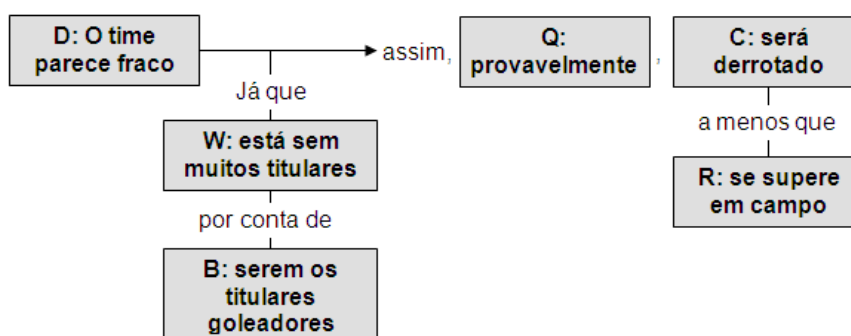


Figura 2.7: Exemplo de argumentação baseado na estrutura proposta por Toulmin. Adaptada de Toulmin (1958).

Apesar de sua importância enquanto precursor, o modelo apresenta deficiências expressivas no contexto de DR. De acordo com a análise apresentada em Lee e Lai (1991), o modelo não contempla o conceito de estado das alternativas, como, por exemplo, “rejeitada” e “aceita” ou permite representar decisões tomadas. Ou seja, seu escopo é limitado à representação de argumentos, mas não à representação do que pode ter sido definido como aceito ou decidido no processo argumentativo. O que se herda de Toulmin é a noção

clara da argumentação no que diz respeito a apresentação de propostas, argumentos e justificativas.

A Notação IBIS

O modelo de Toulmin pode ser visto como um precursor dos modelos de representação baseados em argumentação para DR (Shum, 1991). Entretanto, historicamente, o movimento de DR iniciou-se com o *Issue-Based Information System* (IBIS) proposto por Kunz e Rittel (1970). A notação tinha como principal objetivo suportar a coordenação e o planejamento de processos de decisões políticas. A notação é baseada no princípio de que o processo de projetar soluções de problemas complexos (*wicked problems*) é fundamentalmente uma conversação entre os atores envolvidos neste projeto, onde cada um colabora com sua especialidade e pontos de vista (Conklin & Begeman, 1988). IBIS permite uma representação explícita do raciocínio de decisões, ou seja, das deliberações que antecedem uma decisão, oferecendo suporte especialmente no contexto de decisões tomadas em grupo. A proposta não estava relacionada a software, mas sim a um modo de modelar a argumentação em geral.

O foco de IBIS não está em como o problema é resolvido, de onde vem as alternativas, como são avaliadas ou como se chega a um determinado consenso. A ênfase recai sobre representação do processo pelo qual as decisões são tomadas. Deste modo, o intuito é fornecer suporte à estruturação da discussão de modo a auxiliar na resolução das questões apresentadas.

Há três entidades-chave compondo a proposta: tema, posição e argumento. Essas entidades se inter-relacionam por meio de relacionamentos (apoia, é contra, responde a, sugere etc.), dando forma à argumentação (vide Figura 2.10). As questões são articuladas como temas ou questões (*issues*). Cada tema é seguido por uma ou mais propostas ou posições que respondem a este tema. Cada posição pode vir a ser solução para o tema apresentado ou pode ser rejeitada, dependendo dos argumentos favoráveis ou contrários a essas posições.

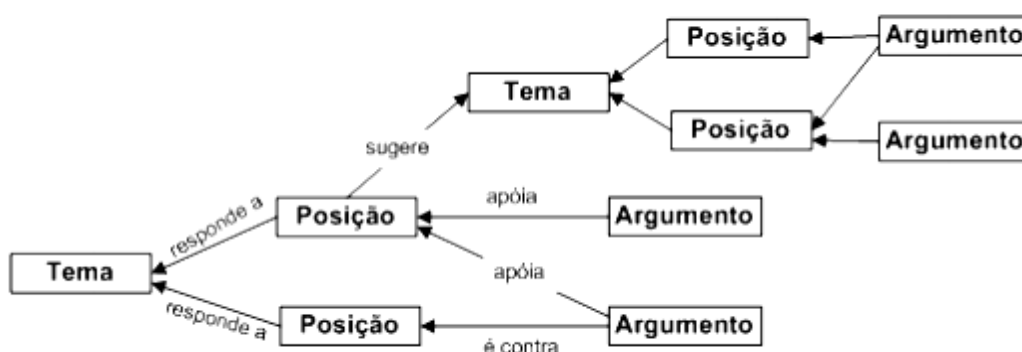


Figura 2.8: Elementos e relações da notação IBIS. Adaptada de Kunz e Rittel (1970).

Uma vantagem de IBIS é que o DR pode ser capturado de maneira informal e em um estágio inicial do processo, assegurando que as questões do projeto estão entendidas. O modelo pode ser tanto utilizado por usuários que trabalham sozinhos como em equipe. A estrutura utilizada na discussão é muito útil e serve para mostrar opiniões pessoais, sinalizações de concordância e

tornam as suposições e definições mais claras (Conklin & Begeman, 1988). Entretanto, devido à sua simplicidade, contendo apenas três tipos de nós, muitas informações acabam sendo representadas em um único nó (Monk et al., 1995). O nó “argumento”, por exemplo, pode vir a armazenar também conceitos como requisitos, restrições e objetivos.

A ausência de uma representação hierárquica gera uma rede complexa quando são armazenados muitos nós (Francisco, 2004). Outro ponto negativo é que, devido à sua estrutura, o modelo não permite a exploração estruturada dos assuntos, dificultando a recuperação de informações específicas.

De acordo com Shum (1991), IBIS foi a primeira notação direcionada para o DR, por essa razão, há muitos sistemas, modelos e notações propostos para apoiar a captura de DR baseadas em IBIS. São exemplos a ferramenta de hipertexto graphical IBIS (gIBIS) (Conklin & Begeman, 1988), que estende o vocabulário de IBIS e adiciona uma representação gráfica, a notação *indented text* IBIS (itIBIS) (Conklin & Burgess-Yakemovic, 1995), a técnica DialogMapping implementada pela ferramenta QuestMap™ e a ferramenta Compendium (Conklin et al., 2003), que é uma melhoria de QuestMap™. Também são baseadas em IBIS as notações DRL e PHI, que serão apresentadas a seguir.

A Notação PHI

A notação *Procedural Hierarchy of Issues* (PHI), proposta no trabalho de Ray McCall (McCall, 1991), foi a primeira notação proposta estendendo IBIS. PHI amplia o conceito de tema (*issue*) proposto em IBIS adicionando subtemas, ou seja, a possibilidade de um tema servir a outro. Estes subtemas devem ser resolvidos para a resolução do tema. Outra alteração é a substituição da estrutura de relacionamento de temas pela relação "serve" (vide Figura 2.11). O subtema A serve a um tema B se A ajuda a resolver B.

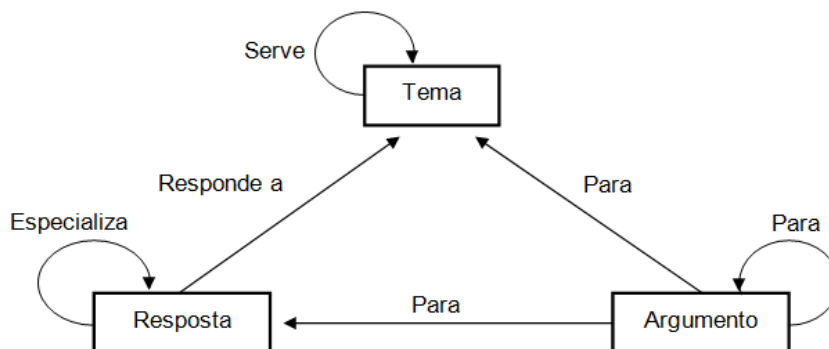


Figura 2.9: Elementos e relações do modelo PHI. Adaptada de McCall (1991).

De acordo com Medeiros (2006), o ponto forte de PHI é o foco na deliberação de temas que servem às afirmações estabelecidas no design (um tema principal), evitando tratar questões triviais e irrelevantes.

O modelo PHI tem sido implementado em diversas ferramentas para captura de DR. JANUS (Fischer et al., 1989) que é um sistema desenvolvido para o design arquitetural (projeto de cozinhas planejadas) que utiliza a notação PHI. JANUS conta com um editor de CAD, ferramentas para o registro do DR e um sistema de críticas que dá suporte ao designer. A notação PHI também é utilizada por PHIDIAS (McCall, 1990), um sistema hipermídia usado para

projetos gráficos 2D e 3D. Uma representação simplificada da notação PHI é utilizada por DocRationale (Francisco, 2004).

O Modelo Potts & Bruns

Parcialmente inspirados no sistema de hipertexto gIBIS, Potts & Bruns (1988) propuseram um modelo genérico para a representação das deliberações do processo de design de software e suas relações com os artefatos de design. A inovação da proposta se baseia na inclusão de "artefatos intermediários", estendendo a notação proposta em IBIS. Outra alteração do modelo sobre IBIS foi a adoção de um único elemento "justificativa" contemplando toda a argumentação, em vez de um conjunto argumentos separados. O principal propósito da proposta foi adaptar a notação de IBIS especificamente para o design de software. O modelo é composto pelos nós de deliberação: tema, alternativa e justificativa, sendo os designs intermediários representados por nós artefato. A Figura 2.12 apresenta um esquema relacionando os elementos do modelo.

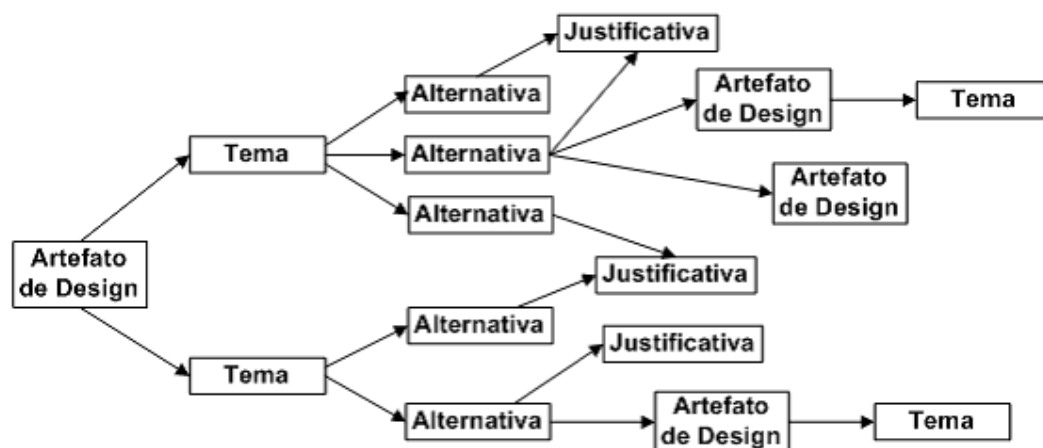


Figura 2.10: Elementos do modelo Potts & Bruns. Adaptada de Potts & Bruns (1988).

Potts

A abordagem Potts (Potts, 1989) é uma evolução (extensão) da proposta Potts e Bruns. Potts segue mais a fundo o modelo IBIS, introduzindo um conceito específico para a captura das ações de design. Os seguintes elementos compõem a abordagem:

Questões ou temas levantados a partir de **passos de design**. **Posições ou alternativas**, que visam resolver as **questões**. **Argumentos** apoiam ou objetam as **posições**. **Artefatos** por sua vez representam os objetos do processo de design. E **passos** que representam ações no processo de design.

A Linguagem DRL

A abordagem Potts & Bruns inspirou a criação da linguagem *Decision Representation Language* (DRL) (Lee, 1990). Em DRL, Potts & Bruns é revisto e estendido de modo a oferecer mais poder expressivo e computacional. De acordo com o autor Jintae Lee, o principal objetivo de DRL é a representação

dos aspectos qualitativos do processo de tomada de decisão. Esse objetivo, por sua vez, é motivado pelo compartilhamento do conhecimento, suporte a decisão e resolução de problemas baseado em argumentação.

Os objetos fundamentais de DRL são: Objetivos, Alternativas e Alegações (*Claims*). Alternativas representam as opções a escolher, objetivos especificam as propriedades da opção ideal e alegações constituem argumentos relevantes para a escolha. Outros objetos são casos especiais dos três objetos apresentados (e.g., problema de decisão é uma subclasse de objetivo). Esses objetos se relacionam por um conjunto de atributos, como, por exemplo: *é uma sub-decisão de*, *é um objetivo para*, *é uma alternativa para*, *é uma sub-alternativa de*, *facilita*, *suporta*, *nega*, *qualifica*, *influencia*, *responde* etc. (vide Figura 2.13)

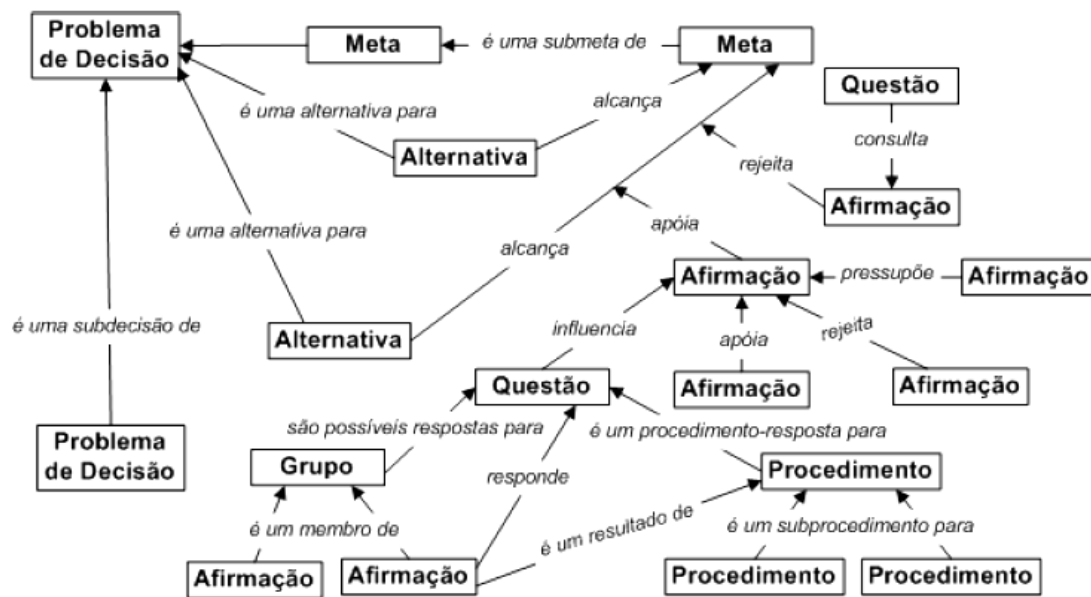


Figura 2.11: Vocabulário de DRL, fonte: Medeiros (2006).

Apesar de ter também uma notação semiformal, DRL tem um maior nível de formalidade na descrição da informação se comparada às outras notações. Esta representação explícita do conhecimento é apropriada para possibilitar que o sistema forneça mais serviços ao projetista.

DRL foi desenvolvida para ser utilizada no sistema SIBYL (Lee, 1990), um sistema de gerenciamento de decisão qualitativo, com o objetivo de fornecer suporte a serviços computacionais.

A linguagem DRL é estendida em RATSpeak por Burge e Brown (2006) para atender melhor a engenharia de software. RATSpeak é utilizada como linguagem de representação no sistema *Software Engineering Using RAtionale* (SEURAT) (Burge & Brown, 2007), desenvolvido para apoiar o uso de DR em manutenção de software. SEURAT é um plug-in Java integrado com o ambiente de desenvolvimento Eclipse.

Outro sistema que toma como base a notação DRL é *Inferencing over Rationale* (InfoRat) (Burge & Brown, 2000). O sistema faz inferências sobre o DR capturado, buscando detectar inconsistências e para avaliar o impacto das

mudanças. Em InfoRat, o DR é visto como uma ponte entre as fases do processo de design.

Para Lee e Lai (1991), uma representação deve apoiar uma variedade de tarefas de projeto, como responder questões sobre o progresso do projeto, alternativas geradas, avaliações que levaram à escolha de determinadas alternativas e se necessário a transferência de conhecimento a outros projetos e pessoas. A notação DRL foi desenvolvida visando suportar todas essas questões. Sua ênfase é gerenciar os elementos qualitativos das tomadas de decisão e do gerenciamento de suas dependências.

A Notação QOC

Questions, Options, and Criteria (QOC) (MacLean et al., 1991) é uma notação semiformal, similar a IBIS e PHI, baseada em argumentação, desenvolvida no centro de pesquisa Xerox Cambridge EuroPARC. O foco da notação é a análise do espaço de design. QOC requer que o designer represente o espaço de design em torno do artefato sendo produzido, sendo essa sua principal diferença em relação a notações baseadas em IBIS, cujo propósito principal é registrar a deliberação do design.

Em QOC, o DR é representado como uma rede de “Questões” (problemas chave) que devem ser resolvidas durante o processo de design, tendo “Opções” como possíveis respostas às questões e um conjunto de “Critérios” que representam as razões a favor e contra as várias opções. Esses elementos e relações são apresentados na Figura 2.14. Além desses elementos, existem também os “julgamentos” (*assessments*), que são os relacionamentos entre opções e critérios. Em sua forma mais simples, os julgamentos podem possuir dois tipos de relacionamentos: “apoia” ou “é contra”. Os argumentos são usados para conduzir a discussão sobre a situação dessas entidades e seus relacionamentos.

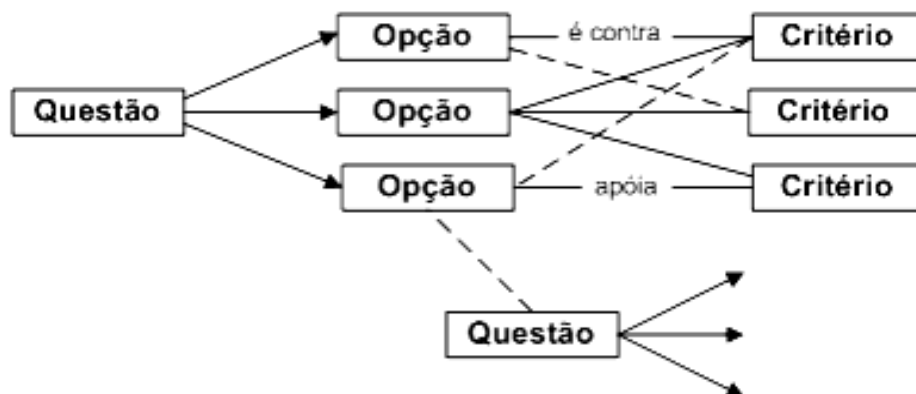


Figura 2.12: O vocabulário de QOC. Adaptada de MacLean e coautores (1991).

Outras notações estendem QOC, como a notação *Traceability, Exploration and Analysis Mode* (TEAM) que estende QOC para tratar sistemas interativos e também abordar algumas das questões de usabilidade de QOC. DQN é uma notação baseada em QOC, projetada para apoiar o desenvolvimento usando métodos formais. A notação DQN mostra que DR também pode ser formal.

DQN

Design Question Notation (DQN) (Bramwell, 1995) é uma notação baseada em QOC. Seu principal objetivo é possibilitar a estruturação, manipulação e raciocínio sobre as opções de design no que diz respeito ao desenvolvimento formal. O principal objetivo por trás da notação é proporcionar um elo entre a pesquisa sobre DR e os métodos formais. A unidade básica em DQN é chamada “questão de design”, que pode ser utilizada para capturar o refinamento das relações entre uma especificação inicial e as “opções de design”.

De acordo com o autor, DQN se destina a ser uma notação de uso geral e, portanto, aplicável a opções de design expressas em diferentes métodos de desenvolvimento.

REMAP

Representation and Maintenance of Process Knowledge (REMAP) é uma abordagem de representação do DR baseada em IBIS, com entidades e relacionamentos adicionais de modo a torná-la semanticamente mais rica (Ramesh e Dhar, 1992). Mais especificamente as seguintes entidades são introduzidas aos elementos tema/questão, posição e argumento de IBIS gerando a proposta REMAP:

- **Requisitos:** representam os objetivos/metapas que guiam o processo de design, gerando questões específicas;
- **Decisões:** são as ações por meio das quais uma resolução foi atingida e uma posição selecionada;
- **Restrições:** representam compromissos, criados a partir de decisões, que devem ser satisfeitos;
- **Objetos de design:** representam os artefatos em questão;
- **Pressupostos:** proveem o contexto no qual um argumento é considerado para ser aplicável.

Essa extensão fornece uma ligação entre requisitos e objetos de design ao DR. REMAP possibilita a captura da história sobre as decisões de design durante o ciclo de desenvolvimento. Entretanto, com o aumento do número de entidades, cresce proporcionalmente o número de relações possíveis. Este crescimento de complexidade torna difícil, se não impraticável, o uso da notação sem suporte computacional, diferentemente da proposta IBIS. REMAP, em virtude de sua complexidade, foi concebido para ser implementado em ferramentas computacionais.

ADD e ADD+

Active Design Documentation (ADD) (Garcia, 1992) é um modelo computacional integrado para auxiliar designers a documentar o projeto em tempo de design. A abordagem provê um sistema computacional que apoia a documentação (dados e *rationale*) das atividades do designer durante o processo de design.

O modelo ADD representa o DR como uma combinação de *rationale* baseado em argumentação e baseado em modelo. ADD se baseia na ideia de capturar o *rationale* através da transformação dos documentos de design de repositórios estáticos de dados em modelos computacionais do próprio design. Assim, o projetista interage com o modelo computacional do processo de design utilizado pelo sistema ADD para registrar diretamente os dados relevantes referentes ao *rationale*.

Apesar dos sistemas ADD serem ferramentas importantes no apoio ao processo e a documentação de design, existem algumas dificuldades no seu uso. A aquisição de conhecimento para a construção e manutenção dos modelos de domínio demanda um enorme esforço. Em ADD+, Garcia e de Souza (1997) propõem uma extensão ao modelo ADD, visando aperfeiçoar a capacidade retórica e explanatória do modelo, permitindo responder perguntas do tipo “E se..”, “Por que não...” etc. Comparado a ADD, ADD+ tem um modelo de comunicação explícito com o usuário para transmitir mensagens que reforçam a usabilidade do modelo.

Kuaba

Kuaba, que significa conhecimento no idioma tupi guarani, é uma abordagem para representação de Design Rationale direcionada para o reúso de designs baseados em modelo. A abordagem faz uso de uma ontologia de mesmo nome desenvolvida por Medeiros (2006). Esta ontologia descreve um modelo de representação de conhecimento para o DR. De acordo com Santos (2007), a ontologia Kuaba permite a adição de semântica ao conteúdo do processo de raciocínio capturado, possibilitando a realização de inferências e outras computações sobre o *DR*.

Kuaba é apresentada como uma ferramenta capaz de suprir, no design baseado em modelos, a falta de uma ferramenta integrada que possa tornar a captura, representação e uso de DR parte do processo de design de software, apoiando principalmente o reúso de designs de artefatos de software.

A ontologia Kuaba estende a estrutura de argumentação da notação IBIS, enriquecendo-a com as decisões e as justificativas para essas decisões, além dos relacionamentos entre os argumentos e os artefatos gerados (Medeiros et al., 2005). A possibilidade da definição das decisões dentro do processo de design, e não apenas o raciocínio, é um diferencial da proposta em relação a outras propostas como IBIS, QOC, PHI. De acordo com a autora, a abordagem de representação possibilita um tipo de reúso de design de software em um nível mais alto de abstração, onde *rationales* podem ser integrados e reempregados no design de um novo artefato.

O vocabulário da ontologia Kuaba é constituído fundamentalmente pelos seguintes elementos:

- **Elementos de raciocínio:** são os itens principais, divididos em 3 categorias:
- **Questões:** representam os questionamentos que surgem na elaboração do artefato, e que podem possuir várias alternativas de solução.

- **Objetivos:** designam um objetivo ou meta a ser alcançada, a demanda deve ser satisfeita, um problema a ser resolvido, uma questão a ser discutida, de um modo geral qualquer coisa que possa surgir no processo de concepção e requeira esforço para ser realizada. Elementos conceitualmente semelhantes a esse podem ser encontrados em outras abordagens, por exemplo, a entidade “Problema de Decisão” (*Decision Problem*) de DRL ou ainda o objeto “Requisito” (*requirement*) de REMAP, o conceito “Problema” (*Issue*) das propostas IBIS, PHI, Potts e Bruns e Potts, e a “Questão” (*Question*) em QOC e Kuaba.
- **Hipóteses:** que designam qualquer sugestão, proposta ou ideia sobre a resolução de um problema no processo de design. Hipóteses são formas de satisfazer os objetivos. O conceito Hipótese corresponde às “Opções” (*Options*) da notação QOC, “Posições” (*Positions*) de IBIS e Potts e “Respostas” (*Answers*) de PHI, “Ideias” de Kuaba, abrangendo também a semântica do objeto “Alternativa” (*Alternative*) de Potts e Bruns e DRL.
- **Ações de Design:** que representam as medidas tomadas no âmbito de um projeto de design. A entidade compreende o objeto de decisão (*Decision*) de REMAP e Kuaba e o conceito “Etapa” (*Step*) em Potts.
- **Justificativas:** representam quaisquer afirmações feitas durante o processo de design que podem ser concebidas como uma alegação sobre algo sendo questionado. Esta entidade compreende o objeto “Alegação” (*Claim*) de DRL, o objeto “Argumento” (*Argument*) de IBIS, Potts, PHI, Kuaba e QOC, e o objeto “Justificação” (*Justification*) de Potts e Bruns. Também as “Suposições” (*Assumptions*) em REMAP.
- **Artefatos:** designam qualquer objeto que é usado, produzido ou trabalhado durante o processo de design. Essa entidade compreende tanto o objeto “Artefato” (*Artifact*) de Potts e Bruns, Potts, Kuaba e DRL, bem como os “Objetos de Design” (*Design Objects*) de REMAP.

O segundo conjunto de elementos, os elementos “dinâmicos” identificados, consiste de quatro atividades:

- **Definição do Problema:** momento em que são identificados problemas a serem resolvidos;
- **Análise do problema:** análise do domínio do problema, a fim de encontrar alternativas para resolver as questões estabelecidas;
- **Avaliação de Soluções:** avaliação das alternativas propostas;
- **Resolução do Problema:** resolução das questões, escolhendo entre as alternativas restantes.

Estas primitivas formam o que os autores denominaram de modelo de ciclo de raciocínio apresentado na Figura 2.16.

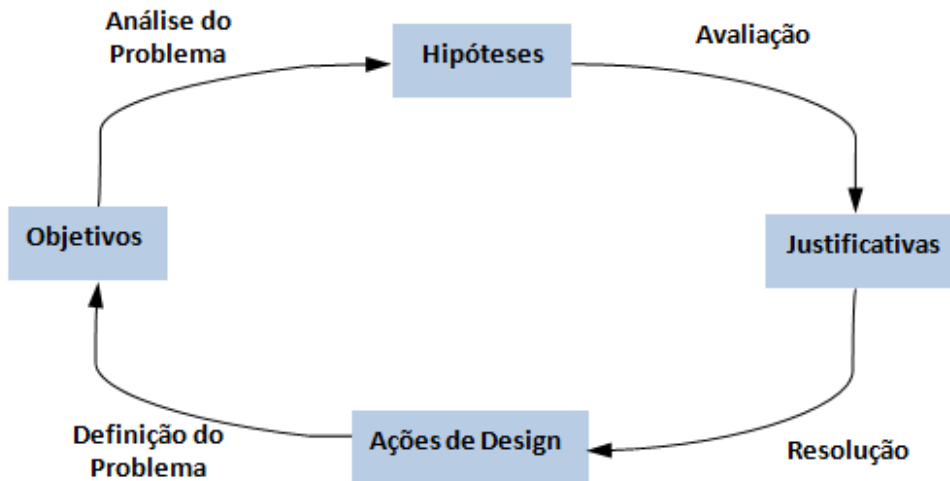


Figura 2.14: Ciclo de design proposto por Louridas e Loucopolos (2000).

De acordo com os autores, esse modelo é adequado a um processo reflexivo. Na etapa de Definição do Problema, são gerados os objetivos. Nesse momento, os designers definem os objetivos a serem atingidos, as demandas a serem preenchidas, os problemas a serem resolvidos, assuntos a serem discutidos. Na Análise do Problema, são definidas as hipóteses. A equipe de designers analisa o domínio problema, buscando sugestões, propostas ou ideias sobre a resolução dos problemas identificados. Na etapa de Avaliação das Soluções, a equipe de designers produz justificativas para as alternativas propostas. Na Resolução de Problemas são geradas as ações de design. Essas ações geram/alteram os artefatos na busca da resolução dos problemas. Designers tomam medidas que afetam os objetos produzidos ou utilizados durante o processo de design. Problemas existentes podem ser resolvidos e novos problemas podem ser identificados.

2.3.8. Considerações Gerais

É importante ressaltar que há diversas outras propostas e abordagens não citadas neste trabalho, mas que, de um modo geral, descendem de alguma das abordagens aqui apresentadas, estendendo-as ou alterando elementos de modo a atender a novas situações ou requisitos específicos.

Com base em uma perspectiva comunicativa do processo de design de sistemas interativos, pode-se dizer que o DR representa o discurso do designer (e sua equipe) sobre parte relevante do projeto do sistema. Esse discurso deve ser coerente e conexo, citando e relacionando atividades, ferramentas, produtos (e.g., requisitos, modelos, artefatos) e contexto do processo de design a que representa. O registro desse discurso deve possibilitar ou dar subsídios, dentre outras coisas:

- À proveniência das informações de design e sua rastreabilidade;
- A uma melhor compreensão dos artefatos propostos;
- À comunicação deste processo para os diferentes participantes do projeto (e.g., usuários, clientes, designers e desenvolvedores);

- À reflexão com base em experiências já vivenciadas;
- A posterior explicação do sistema projetado a seus usuários por meio da ajuda do sistema (Silveira et al., 2000).

Analisando as abordagens propostas de registro do DR, sob estes cinco “requisitos”, observa-se que um dos problemas recorrentes é que, na maioria das propostas estudadas (Moran & Carroll, 1996), o DR não está integrado aos insumos e produtos (e.g., requisitos, tarefas, modelos, artefatos) do processo de design que visa representar. Deste modo, essas propostas são consideradas como coproduto do processo de design (Fischer et al., 1995; Louridas & Loucopoulos, 2000) e não integradas ao mesmo. Conseqüentemente, integram de forma pouco coesa ou simplesmente não integram as informações provenientes das diferentes etapas do processo.

Outra característica comum a boa parte das abordagens é que elas têm o foco de seu registro em etapas específicas do processo de design, fazendo com que o “discurso” registrado não tenha o relacionamento e abrangência necessários. Propostas como IBIS, QOC e PHI atuam sobre as decisões de design, mas em geral não vinculam essas decisões à modelagem do problema de design. Já propostas como a de Dutoit e Paech (2002) são focadas no registro do *rationale* do espaço de problema.

Para Medeiros (2006), de um modo geral, as abordagens existentes para DR geram representações incompletas ou informais, não cobrindo detalhes importantes como, por exemplo, as decisões tomadas pelos projetistas durante o design.

Dentre as abordagens estudadas, três propostas se sobressaem, tomando como base os elementos necessários para um discurso coerente e conexo, interligando as atividades do processo de design, são eles: Kuaba, ADD+ e Modelo de Design Reflexivo (*Reflective Design Model*).

Na proposta do Modelo de Design Reflexivo, Louridas e Loucopoulos (2000) fazem uma análise bastante completa de diferentes representações de DR, identificando e caracterizando um conjunto de elementos “estáticos” e “dinâmicos” considerados relevantes para a composição de um modelo para o design reflexivo. Um ponto negativo do modelo que deve ser observado é a ausência em sua notação da representação do contexto. Concorda-se com Medeiros (2006) que o contexto de design é um requisito importante para a expressividade da modelagem do DR.

Kuaba, por sua vez, apresenta uma ontologia bastante completa para o registro do DR. Assim como no Modelo de Design Reflexivo, Kuaba traz em sua ontologia de DR não apenas elementos que permitem o raciocínio de design, mas também a tomada de decisão. A nosso ver, seu ponto mais importante é a busca pela integração do DR com o registro do processo de design. Apesar de suas contribuições, que inspiram nossa pesquisa, em uma análise mais cuidadosa da ontologia de Kuaba, observam-se algumas limitações na proposta. Um primeiro ponto frágil desponta na integração de Kuaba com os diferentes tipos modelos (formais ou informais) que podem vir a ser adotados para dar suporte às diferentes atividades do processo de design. A ontologia impõe o uso de modelos formais para o suporte às atividades. Essa decisão exclui ferramentas importantes adotadas no design baseado em modelos, por exemplo, cenários de uso, modelagem e esboços.

Outro ponto negativo é a limitação da ontologia no relacionamento das atividades de design e de seus produtos (artefatos). Esta limitação impacta, por exemplo, na rastreabilidade das informações do design. Ter o reuso como principal motivação é outro ponto delicado da proposta de Kuaba, considerando a natureza única e complexa do design (Schön, 1983; Löwgren & Stolterman, 2004; de Souza et al., 1999).

A proposta ADD+ tem como principal característica o caráter mais comunicativo, expressivo e conseqüentemente epistêmico imposto ao DR. Outro ponto relevante de ADD+ é a integração do ato de documentar ao processo de design.

Entretanto, assim como em outras propostas, faltam meios para o registro do contexto de design (e.g., dados do projeto, participantes, motivação). Não foi possível identificar como o modelo trata as atividades de design individualmente, como ele se relaciona com o uso de outros modelos e quais os elementos e relações do modelo de tomada de decisão. O passo inicial de engenharia do conhecimento para a geração do modelo (e posterior manutenção) também pode ser visto como uma questão problemática, já que exige tempo e conhecimentos específicos para tal.

Estes modelos, e conseqüentemente as notações nas quais eles se ancoram, contribuíram na formalização do modelo proposto neste trabalho, seja pelas benesses que eles aportam ao DR, seja pelas limitações que nos indicaram onde seria possível melhorar.

2.4. Rastreabilidade de Requisitos

A conformidade de um sistema interativo com os requisitos a partir dos quais foi projetado é uma exigência básica da indústria de desenvolvimento de software. Entretanto, esta é uma exigência que nem sempre é alcançada. Por esta razão, surgiu a área de Gerência de Requisitos (GR).

A Gerência dos Requisitos (GR) é uma área da engenharia de software ligada ao processo de acompanhamento de mudanças de requisito. O conjunto de atividades que compõem a Gerência dos Requisitos estão definidas no CMMI, na área de Desenvolvimento de Requisitos. É por esta razão que esta área e suas atividades são fortemente associadas à qualidade do desenvolvimento de software. A atividade chave da GR é a rastreabilidade de requisitos, já que requisitos não podem ser efetivamente gerenciados sem rastreabilidade. Quanto a definição, há diversas propostas para a rastreabilidade de requisitos (Sayão e Leite, 2005). Dentre elas a definição dada por Gotel e Finkelstein:

“A rastreabilidade se refere à capacidade de descrever e seguir a vida de um requisito, em ambas as direções, para frente e para trás (ou seja, desde as suas origens, através do seu desenvolvimento e especificação, para sua implantação e utilização posteriores, e através de todos os períodos do curso de aperfeiçoamento e interação em qualquer destas fases).” (Gotel & Finkelstein, 1994).

Para Ramesh e Jarke (2001), a rastreabilidade de requisitos é uma característica de um sistema em que os requisitos são claramente vinculados às

suas fontes e aos artefatos criados durante o de ciclo de vida de desenvolvimento do sistema com base nesses requisitos.

Diversos trabalhos apresentam a rastreabilidade como um dos mais importantes pré-requisitos para o desenvolvimento de software de qualidade (Ramesh & Jarke, 2001; Sayão & Leite, 2005; Kotonya, 1998), mas suas utilidades vão além, podendo apoiar gerentes e desenvolvedores em tarefas de grande valia como (Sayão & Leite, 2005):

- verificar a alocação de requisitos a componentes do software;
- resolver requisitos em conflito;
- verificar a cobertura dos requisitos;
- corrigir problemas;
- validar com o usuário final os requisitos;
- analisar o impacto na evolução do sistema;
- prever custos e prazos;
- gerenciar riscos;
- fazer melhorias de *hardware* e/ou *software*;
- reutilizar componentes.

Uma atribuição que inclui-se nesta lista é o redesign. Saber, na história de um requisito, os pontos de início do processo de design e que propostas foram produzidas, possibilita a retomada de uma proposta abandonada ou a identificação do início do design para o requisito.

Para que um requisito seja rastreável são necessárias algumas informações. De acordo com Kirkman (1998), um requisito é rastreável se é possível detectar:

- A origem do requisito;
- A razão para a existência do requisito;
- A que o requisito se relaciona;
- Como o requisito se refere a outras informações do processo de design;
- O processo de tomada de decisão que levou ao requisito;
- O status do requisito.

Segundo Sommerville (2003) um requisito é rastreável se é possível descobrir quem sugeriu o requisito (a fonte), por que o requisito existe (rationale), que outros requisitos estão relacionados a ele (dependência entre

requisitos) e como o requisito se relaciona com outras informações tais como o design do sistema, implementação e documentação do usuário.

Apesar da rastreabilidade de requisitos já ser utilizada há alguns anos, códigos fonte freqüentemente evoluem sem a atualização da documentação (Ramesh et al. 1997). Isso ocorre porque manter a consistência e a rastreabilidade entre abstrações de alto-nível, funcionalidades e componentes de *software* é custoso e consome tempo (Antoniol et al., 2005). Tentando minimizar esse problema, pesquisadores e a indústria de software desenvolveram diferentes técnicas (Hull et al., 2005) e ferramentas (Sayão e Leite, 2005) visando difundir e aperfeiçoar a gerência de requisitos, que, até então, infelizmente falharam em alcançar uma adoção generalizada (Neumüller et al., 2006) sendo esta atividade, muitas vezes, apenas um requisito desejável dos processos de desenvolvimento de software. Problemas relacionados ao registro e uso da rastreabilidade não são recentes: Gotel e Filkelstein (1994) observavam, já em 1994, que a rastreabilidade de requisitos era citada como problema-chave entre indústria e comunidade de engenharia de requisitos. Dentre os vários problemas identificados, um problema central é o fato de que muitos desenvolvedores veem a rastreabilidade como uma atividade opcional, para os quais existem muito poucos recursos disponíveis e de que eles veem muito poucos benefícios diretos tendo em vista o custo de implementação da rastreabilidade (Stout, 2001).

Uma forma possível de implementar a base da rastreabilidade de requisitos, minimizando o esforço inerente a essa atividade é associá-la ao processo ao de design de IHC, que já demanda boa parte dos requisitos da rastreabilidade. Deste modo, integrando esforços necessários em ambos os processos e diminuindo a “distância” entre as áreas de IHC e Engenharia de Software. Sob esta perspectiva a literatura de rastreabilidade oferece insumos importantes para a geração de um modelo que favoreça paralelamente a rastreabilidade de software e o design de sistemas interativos.

2.5. Proveniência de Dados (*Provenance*)

A pesquisa sobre proveniência ou procedência (em inglês *provenance*), também designada na literatura por linhagem, pedigree, filiação ou genealogia, trata da história da informação, incluindo sua origem, os principais eventos que ocorrem ao longo do seu ciclo de vida, e outras informações de rastreabilidade associados à sua criação, processamento e arquivamento (Ram & Liu, 2007).

Rastreabilidade é um conceito fortemente vinculado à proveniência. Pode-se dizer que rastreabilidade é uma característica de sistemas nos quais há a ligação clara dos requisitos às suas fontes e aos artefatos criados durante o ciclo de vida de desenvolvimento do sistema baseado nesses requisitos (Ramesh & Jarke, 2001). Apesar do conceito de rastreabilidade ser bem mais difundido na engenharia de software, mais especificamente na gerência de requisitos, sua noção e importância também se aplica ao design de sistemas interativos em IHC, sendo requisito importante para prover explicações sobre as informações de design.

É importante ressaltar que, em geral, as definições de proveniência variam de acordo com a área de estudo ou aplicação do conceito. O conceito de

proveniência pode ser associado não apenas aos dados “transformados” ou produtos de dados, mas também com o conjunto de processos que permitiram a criação dos dados. Simmhan e coautores (2005) definem proveniência como a informação que ajuda a determinar a história de uma derivação de dados de um produto, a partir de suas fontes originais.

Proveniência pode ter um grande número de aplicações (Pearson, 2002). Ram e Liu (2007) afirmam que rastrear a procedência das informações possibilita aos usuários compartilhar, descobrir e reusar os dados, facilitando o aprendizado e evitar erros já cometidos. Analisando as diferentes aplicações da proveniência, Goble (2002) as classifica em cinco grupos:

Qualidade: a linhagem pode ser usada para estimar a qualidade e confiabilidade dos dados com base em sua fonte e suas transformações. Ele também pode fornecer dados sobre as demonstrações de prova da derivação dos dados.

Auditoria: Proveniência pode ser usada para fazer a auditoria da trilha dos dados, para determinar o uso de recursos, e detectar erros na geração de dados.

Replicação de derivação: informações detalhadas de proveniência podem permitir a repetição da derivação de dados, e servir de mapa para a replicação do processo.

Atribuição: o pedigree dos dados pode estabelecer os direitos autorais e propriedade dos dados, permitir sua citação e determinar a responsabilidade no caso de dados errados.

Informação: A utilização genérica da linhagem é a consulta com base na linhagem de metadados para a descoberta de dados. Ele também pode ser consultado para fornecer um contexto para interpretar os dados.

Pearson (2002) afirma que as informações de proveniência devem ser capturadas com a esperança de serem compreensíveis o suficiente para serem úteis no futuro. Para isso, é necessário que as informações estejam contextualizadas no domínio de que se originam. É desejável adquirir uma profunda compreensão da semântica de origem e identificar os conceitos-chave associados a elas. Alguns modelos e ontologias, avançando na pesquisa sobre proveniência, propõem o registro da semântica da proveniência (Ram & Liu, 2007; McGuinness et al., 2007; Marins, 2008). Dentre eles o modelo W7 (apresentado a seguir) se mostra bastante completo.

Ancorados fundamentalmente na pesquisa Bunge (1977) sobre “história ou proveniência”, que tem como elementos centrais as noções de evento, espaço, tempo, ação e agente, Ram e Liu (2007) propõem o modelo ontológico W7. O modelo visa à captura da semântica da proveniência dos dados e conceitua proveniência como consistindo de sete dimensões interconectadas (Figura 2.18): o que (*what*), quando (*when*), onde (*where*), quem (*who*), como (*how*), qual (*which*), e por quê (*why*).

- **O que (*What*):** é o elemento base do modelo, sua semântica é definida como uma sequência de eventos que afetam um objeto de dados durante o seu tempo de vida;

- **Quando (When):** a semântica do “quando” representa o conjunto de *timestamps* associados aos vários eventos de proveniência;
- **Onde (Where):** o conjunto de todas as localidades (físicas, lógicas ou geográficas), onde os eventos de proveniência acontecem;
- **Como (How):** documenta as ações que levam à ocorrência de um evento;
- **Quem (Who):** se refere aos agentes que tratam dos eventos;
- **Qual (Which):** descreve os dispositivos que são usados na criação, análise e transformação dos dados;
- **Por quê (Why):** representa o conjunto de razões para os eventos de proveniência.

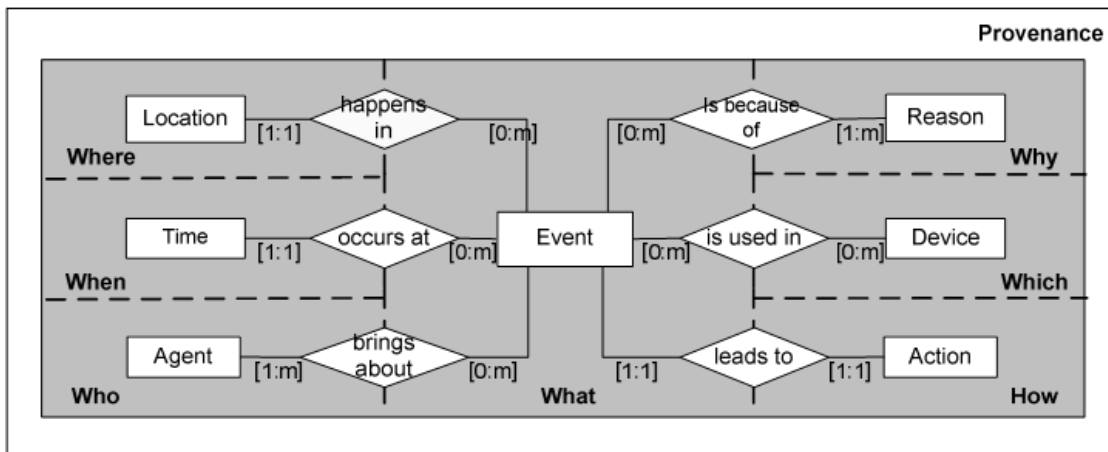


Figura 2.15: Ontologia de proveniência W7 (Ram & Liu, 2007).

Analisando as propostas de proveniência, observa-se que o registro do processo de design aqui pretendido tenciona, mas vai além do que propõem as definições e pesquisas de proveniência (Pearson, 2002). Enquanto proveniência propõe o registro de informações com seus objetivos direcionados unicamente ao resultado do registro (e.g., estimar qualidade, auditoria de dados, replicação de processos e rastreabilidade das informações), o registro das atividades do processo de design por si só, deve ter, antes de qualquer outra coisa, um caráter e objetivo epistêmico. O ato de registrar as informações do processo já é por si só um objetivo, pois deve levar o designer a refletir sobre o problema, além de guiá-lo durante o processo. Alguns outros objetivos do registro do processo de design são possibilitar a reconstrução da metamensagem de design, permitir a comunicação entre membros da equipe através do modelo, registrar reflexões acerca do processo, além do registro do DR de modo a relacionar os espaços de Design. Essas razões exigem que esse registro tenha uma semântica bem mais específica do que os exemplos de modelos e ontologias de proveniência.

É necessário, a partir do registro do processo, por exemplo, poder responder a questões mais profundas e específicas do que as propostas de proveniência podem prover, por exemplo:

- A quem ou a que essa informação atende?

- Qual a alternativa a essa informação?
- Que comentários foram feitos durante o processamento dessa informação?
- Qual a relação (e o tipo de relação) entre esta informação e outras informações do contexto?

Desse modo, acredita-se que as pesquisas em proveniência podem sim auxiliar, indicando requisitos na construção de um modelo de suporte ao registro das atividades do processo de design, como as sete dimensões endereçadas no trabalho de Ram e Liu (2007). Entretanto, nem todos os requisitos necessários são previstos a partir dessa área de pesquisa.

2.6. Conclusões do capítulo

Neste capítulo foram analisados conceitos, propostas e ferramentas de suporte ao design de sistemas interativos, tendo sido observados as nuances e requisitos desta atividade, o design, e tendo em vista diferentes perspectivas sobre o mesmo. Pode-se afirmar que o design é um processo complexo, único, dado o seu contexto e de cunho intelectual e comunicativo. Este processo pode ser dividido em atividades que se interrelacionam a partir de seus produtos e insumos. Diferentes modelos propostos dão foco a atividades específicas deste processo ou o vêem em um nível diferente de abstração. Observa-se, entretanto, que independente das visões e abstrações aplicadas sobre as atividades, visando minimizar a complexidade deste processo, há a necessidade de ver este processo e trabalhá-lo de forma global, permitindo sua compreensão enquanto processo e as relações e interseções entre suas partes. Para atingir a estes objetivos, é necessário relacionar as atividades do processo por meio de seus insumos e produtos. Um efeito desta visão global é a possibilidade de identificar e compreender as relações entre os requisitos que motivaram o processo e os resultados obtidos a partir do mesmo.

As notações e propostas de DR visam, prioritariamente, o registro do processo decisório das atividades de design. Burge e Brown (2000), entretanto, veem, além das funções comumente atribuídas ao DR (e.g., melhoria da qualidade do design, avaliação, manutenção, reuso, ensino, documentação), a função de servir como ponte entre as etapas ou fases do processo de design, dando o suporte necessário a integração das atividades do processo. Entretanto, apesar de ser apresentado como panacéia para o design, Medeiros (2006) afirma que as notações de DR existentes ainda geram representações incompletas ou informais, não cobrindo detalhes importantes como, por exemplo, as decisões tomadas pelos projetistas durante o design. Mesmo abordagens que oferecem um maior relacionamento entre as atividades e elementos do processo como Kuaba, ADD+ e Modelo de Design Reflexivo (*Reflective Design Model*) ainda deixam lacunas quanto à proveniência necessária para a compreensão clara das atividades e suas realções, favorecendo também a rastreabilidade das informações de design.

A compreensão e análise dos conceitos, requisitos e propostas de proveniência e rastreabilidade, aliados aos modelos de design e conceitos e notações de DR, oferecem o substrato necessário para a proposição de um

modelo que atenda ao suporte do processo de design. Suporte este, ancorado no registro integrado e reflexivo do processo de design, possibilitando a compreensão deste processo e de seus resultados, bem como das relações entre ambos, favorecendo, dentre outras coisas, a rastreabilidade das informações de design.