Aquisição de Design Rationale através de Anotações Semi-Formais em Documentos Ativos de Design

Flávio Miguel Varejão
email: fvarejao@inf.puc-rio.br

Ana Cristina Bicharra Garcia
Departamento de Informática, Universidade Federal Fluminense
email: bicharra@inf.puc-rio.br

Clarisse Sieckenius de Souza
email: clarisse@inf.puc-rio.br

PUC-RioInf.MCC39/96

Abstract: In this paper, we classify the different approaches which have been proposed to document the design activity. Comparing these approaches we identify that argumentative, history and model based approaches are complementary. In order to associate the advantages of these approaches, we propose an ADD's [Gar92] extension based on the capture of structured annotations made by the user during the design process. These annotations are free text annotations organized by an index structure captured from the context where the annotation is made. This structure supports retrieval of annotations in different contexts from that the annotation is made.

Keywords: Design Rationale, Active Design Documents, ADD, Annotation

Resumo: Neste trabalho realizamos uma classificação das diversas abordagens que têm sido propostas para documentar a atividade de design. A partir da comparação das características de cada abordagem, observamos uma complementaridade entre as abordagens histórica, argumentativa e baseada em modelos de design. Para combinar as vantagens destas abordagens, propomos uma extensão de ADD [Gar92] baseada na captura de anotações estruturadas feitas pelo usuário durante o processo de design. Anotações estruturadas são anotações feitas pelo usuário em linguagem natural organizadas por uma estrutura de indexação capturada do contexto onde ela foi feita. Esta estrutura suporta a recuperação de anotações em contextos diferentes daquele no qual a anotação foi capturada.

Palavras Chave: Design Rationale, Documentos Ativos de Design, ADD, Anotação

1 - Introdução

Design é uma atividade na qual se busca criar um artefato que, quando colocado em funcionamento no ambiente de uso para o qual foi projetado, produzirá uma alteração do estado corrente de modo a satisfazer as necessidades do usuário do artefato [RG96].

Embora saibamos fazer design, esta atividade é custosa e difícil na maioria das vezes. Isto decorre principalmente do fato de que as características do espaço de design não costumam ser definidas antecipadamente, sendo estabelecidas ao longo da própria atividade de design. Adicionalmente, muitas vezes, a atividade de design é multidisciplinar, podendo envolver diferentes especialistas de cada domínio em um processo de negociação.

Tipicamente, o produto da atividade de design são descrições detalhadas do artefato projetado e indicações de como construí-lo. Considerando a existência de dificuldades para a realização do design, documentar não apenas o resultado final do design, mas *como* e *porque* um produto foi projetado pode ser extremamente útil para:

- promover maior reflexão sobre as questões de design,
- facilitar futuras alterações no produto,
- permitir que o conhecimento adquirido seja reusado no projeto automático de outros produtos,
- melhorar a qualidade de projetos futuros, dado que a documentação do design pode fornecer base para a minoração de erros futuros,
- dar suporte ao projeto colaborativo [PSL95], dado que os diferentes agentes envolvidos podem entender e verificar os motivos de conflito entre agentes e, consequentemente, resolvê-los de forma mais harmônica,
- preservar a memória institucional, desvinculando o conhecimento do projeto de um funcionário específico,
- treinamento de novos funcionários.

O tipo de conhecimento e raciocínio que fundamenta o processo de design é conhecido pelo termo genérico de *Design Rationale*. Se por um lado, documentar o processo de design é uma atividade custosa para o designer, em decorrência da costumeira exiguidade dos prazos de entrega de projetos, por outro lado, capturar automaticamente o racional de design também é uma tarefa difícil, por envolver variadas classes de conhecimento e processos de raciocínio. Apenas para mostrar algumas destas dificuldades, citamos que, muitas vezes, decisões são tomadas baseadas em critérios subjetivos, tal como conceitos estéticos, ou mesmo, baseadas no conhecimento *compilado* (intuição) do próprio designer.

Com o intuito de auxiliar os designers a documentar este tipo de conhecimento, várias abordagens tem sido propostas para a documentação de design. Cada uma destas propostas possui características próprias que as diferenciam uma das outras e que as tornam mais ou menos adequadas para a captura do rationale.

Inicialmente, apresentaremos uma classificação (não muito diferente da apresentada por Garcia [Gar92]) das propostas para captura de design rationale. Esta classificação se baseia na estratégia de captura utilizada pelas propostas, dividindo-as em quatro grupos distintos: Caderno de Anotações, Histórico, Argumentativo, Baseada em

Modelos. Utilizaremos esta classificação para ilustrar e comentar brevemente alguns dos sistemas propostos para captura e recuperação de rationale.

Para que seja possível comparar as características destas propostas, desenvolvemos uma estrutura de comparação de abordagens de captura de design rationale. Durante a comparação das propostas, pudemos notar uma certa complementaridade entre as abordagens histórica, argumentativa e baseada em modelos de design.

Em seguida, apresentamos em detalhe a abordagem utilizada por ADD (Active Design Document) [Gar92] para a captura de design rationale. Propomos, então, uma extensão de ADD que permitirá combinar algumas vantagens da abordagem argumentativa com as da baseada em modelos de design. Esta extensão é baseada na captura de anotações estruturadas feitas pelo designer durante o processo de design. Estas anotações podem ser facilmente recuperadas durante o uso da documentação em função da estruturação proposta.

2 - Classificação

A classificação a seguir se baseia na estratégia base utilizada para captura do rationale. Isto significa que, embora alguns sistemas possam utilizar mais de uma abordagem, aqui eles serão classificados de acordo com o método predominante de captura. Procura-se retratar também, através da sequência de abordagens apresentada, o processo histórico de desenvolvimento da pesquisa na área de design rationale.

2.1 - Caderno de Anotações

Geralmente, a documentação das decisões de design são registradas informalmente num caderno de anotações do projetista. Uma proposta para capturar design rationale procura incentivar os projetistas a migrarem da mídia papel para a mídia computacional fornecendo uma espécie de caderno eletrônico de anotações [LWL89]. Nele, o projetista pode fazer anotações textuais livres e representações gráficas do artefato com facilidade. Além disso, o caderno eletrônico de anotações fornece, para alguns domínios, bibliotecas de objetos gráficos para a composição das representações gráficas que podem ser posteriormente manipulados através de uma linguagem visual específica.

Embora a idéia de cadernos eletrônicos seja, em princípio, muito flexível, na prática, eles não têm sido muito usados. Em geral, os projetistas costumam ser extremamente econômicos em suas anotações. A menos que sejam requisitados (ou forçados) a documentarem o rationale, eles apenas especificam aquilo que é necessário para obter o conjunto final de desenhos que compõem o documento e as informações necessárias para construção e uso do artefato. É preciso, portanto, criar mecanismos que mostrem os benefícios da documentação do rationale aos projetistas e que os incentivem a usar estes mecanismos.

2.2 - Histórico

O enfoque de captura do rationale por histórico procura representá-lo através do registro da sequência de eventos ocorrida durante o design. Ele é baseado na idéia de que, caso os projetistas possam recuperar esta sequência, eles poderiam reusá-la para justificar o processo pelo qual o produto foi obtido ou para reproduzir parte do design em modificações do projeto original..

Neste tipo de abordagem, portanto, as intenções do projetista ficariam subentendidas pela sequência de passos tomada. O grande problema desta abordagem é exatamente este. O fato de se registrar os passos tomados pelo projetista não implica que tenhamos

compreendido qual foi a sua intenção ao fazê-lo. Por exemplo, muitas das ações tomadas pelo projetista durante o design podem ser decorrentes de um processo de aprendizado e aquisição de intuição, as quais visam lhe facilitar a compreensão do problema e dos impactos sobre o design de determinadas decisões. Neste sentido, tais ações podem ser classificadas como epistêmicas [KM94] dado que não aproximam diretamente o estado atual da solução do problema de seu estado final desejado. Assim, a ausência de compreensão das intenções do projetista nos impede de filtrar a sequência de eventos, eliminando as ações epistêmicas, o que pode comprometer em muito o reuso automático do rationale.

Outro problema é que uma leve alteração dos requisitos de design ou de uma decisão tomada pode provocar uma completa modificação do caminho seguido pelo designer. Neste caso, o rationale registrado seria praticamente inútil a partir daí. Por exemplo, em engenharia, a escolha de um tipo de material mais custoso pode tornar essencial a minimização do consumo deste material no dimensionamento de uma estrutura. A seleção de um outro tipo de material, mais barato, durante o reuso do rationale, pode não implicar numa necessidade de minimização de consumo. Ao contrário, pode ser necessário avaliar o impacto que esta escolha provocará no tamanho da estrutura. A partir deste momento, praticamente nada poderá ser reaproveitado do rationale armazenado.

Além disso, a apresentação pura e simples da sequência de passos tomados pelo projetista ao longo do design, sem levar em conta a sua relevância, pode ser altamente contra-informativa, no sentido de que o usuário desta documentação pode ficar perdido no meio de tantos detalhes e acabar desistindo.

Por outro lado, as maiores vantagens deste enfoque são a não requisição de esforço adicional por parte do designer e a baixa sobrecarga computacional requerida (é suficiente gravar um arquivo de log dos passos tomados pelo projetista).

Ganeshan et al. [GGF94] apresentam um sistema que permite capturar interativamente a história de um processo de design. Neste sistema o designer define o problema instanciando um conjunto pré-definido de classes de objetivos (restrições e critérios que devem ser atendidos). Ao longo do processo o projetista resolve o problema interagindo com o sistema através de mudanças nos objetivos e através da seleção de alternativas. Em paralelo, o sistema mantém a história do processo armazenando a sequência temporal de estados de design. Cada estado contém um registro de atividade que descreve o conteúdo dos objetos naquele instante. Desta forma, torna-se possível ao designer retornar a pontos de decisão anteriores para explorar diferentes alternativas, determinar todas as decisões que são afetadas por mudanças nos objetivos e repetir partes da história registrada.

O enfoque de Ganeshan et al. contorna parcialmente o problema da não captura das intenções do projetista, comum aos enfoques históricos, mantendo os registros de atividades de cada estado do processo. Desta maneira, o sistema está apto a mostrar, não apenas como o processo se realizou, mas também o porque de certas alternativas terem sido adotadas e outras não. Em virtude desta capacidade, o sistema de Ganeshan et al. pode ser considerado também como um exemplo da abordagem argumentativa. De qualquer forma, a ausência de um modelo de design subjacente impede que sejam capturadas as razões que levaram o projetista a adotar uma determinada sequência ou estratégia de solução.

2.3 - Argumentativo

O enfoque argumentativo requer que os designers registrem uma explanação completa do design (o rationale) de modo que ele possa ser recuperado futuramente pelos consumidores da informação. Este enfoque se difere do adotado nos cadernos de anotações pelo fato de que neste, as anotações do designer, embora livres, devem seguir uma determinada metodologia.

A grande vantagem deste enfoque é a sua flexibilidade: ele não é vinculado a nenhum domínio específico, não requer a existência de um conhecimento estabelecido de design, permite a documentação de design criativo e ainda suporta a documentação da comunicação e negociação entre os diversos projetistas. Além disso, por demandar maior reflexão do projetista durante o design, tende a minimizar a necessidade de futuros e custosos redesigns. Uma ampla discussão a respeito deste modelo pode ser encontrada em [Shu96].

IBIS (Issue-Based Information Systems) [KR70] é uma metodologia para a expressão de rationale que utiliza o enfoque argumentativo. A idéia por trás desta metodologia é que os designers devem descrever os fatores a favor e contra as diferentes alternativas às questões do processo de design. Em IBIS, as questões de design são denominadas *temas*, alternativas são chamadas *respostas* ou *posições* e a descrição de fatores favoráveis ou contrários são *argumentos*. Finalmente, as decisões sobre quais alternativas devem ser aceitas ou rejeitadas são chamadas *resoluções de conflitos*. Muitas das propostas de captura de rationale usando o enfoque argumentativo são baseadas em IBIS, tais como, gIBIS [CB89], SIBYL/DRL [Lee91] e PHI [McC91].

Já QOC (Questions, Options and Criteria) [MYBM91] requer que o designer represente o espaço de design em torno do artefato sendo produzido. De modo similar a IBIS, o design rationale é representado como uma rede de *Questões* que devem ser resolvidas durante o processo de design, *Opções* que representam as possíveis respostas as questões e um conjunto de *Critérios* que representam as razões a favor e contra as várias opções.

Embora as propostas que utilizam o enfoque argumentativo procurem reforçar a reflexão do projetista durante a realização do design, através da explicitação das razões que o levaram a tomar suas decisões e fazer suas escolhas, elas ainda não atendem as necessidades de documentação do consumidor de informação. Como é necessário capturar explicações completas antecipadamente para que possam ser posteriormente apresentadas ao usuário, seria preciso que o designer prevesse todas as possíveis questões que poderiam ser perguntadas pelo usuário e, ainda, formular previamente os rationales que deveriam ser mostrados. Isto é praticamente inviável.

Outro problema deste enfoque é a grande sobrecarga de trabalho exigida dos projetistas [FLMM91, MM94]. Além de terem de aprender uma metodologia para documentar suas idéias, eles devem dedicar seu exíguo e precioso tempo à tarefa de documentação. Ainda nesta linha, muitas vezes eles têm dificuldade de expressar seu raciocínio explicitamente. McKerlie e MacLean [MM94] reportam também que, em um experimento realizado, os projetistas deixaram de fazer design e passaram a fazer QOC, isto é, ao invés de documentarem o que era relevante ao processo de design, eles acabaram produzindo documentação detalhada de algo que não era importante.

Adicionalmente, a recuperação da informação registrada usando esta abordagem é custosa. Algumas vezes queremos reaver a informação sobre as causas de uma decisão específica. Devido a ausência de uma forma de indexação, só podemos obter esta informação procurando ao longo de toda a documentação. Pior ainda é quando

necessitamos modificar a documentação. Em muitas situações é necessário refazer todo o documento.

Segundo Shum [Shu96], modelos argumentativos são adequados para a documentação de rationale em domínios de problemas onde o conhecimento sobre o processo de design é mal estruturado ou mesmo inexistente. Nestes casos, o estilo analítico e reflexivo de design provocado por estes modelos pode ser de grande valia para aumentar a qualidade do processo de design. O alto custo de documentação do rationale é compensado pela sistematização do processo de raciocínio no qual o artefato foi idealizado. Por outro lado, em domínios onde existe um conhecimento estabelecido e bem estruturado, o processo de decisão tende a ser mais direto e implícito. Neste modo de design, o uso de modelos argumentativos para documentação de rationale costuma obstruir o andamento do projeto, tornando-se pouco atraente para os projetistas.

2.4 - Baseado em Modelos

Em certos domínios da engenharia, onde o conhecimento de design é melhor compreendido e estabelecido, é possível obter-se previamente um modelo computacional de design. Com o auxílio deste modelo, além de se fornecer suporte ao próprio design do artefato, pode-se gerar dinamicamente explicações em respostas às questões do usuário do rationale de design. Desta forma, pode-se suprir necessidades do usuário que não teriam como ser previstas antecipadamente pelo autor da documentação do rationale [GR96].

De acordo com Shum [Shu96], em princípio, a existência do modelo de design permite a geração de rationale que pode

- explicar a estrutura física do artefato: quais são os seus componentes?
- explicar a forma de operação do artefato: como o artefato interage com o ambiente externo ou como os seus componentes interagem entre si?
- explicar a função do artefato e de seus componentes: qual é o propósito do artefato e como cada componente contribui para a realização deste propósito?
- suportar verificação e avaliação do artefato: o artefato atende aos requisitos de projeto? É a melhor alternativa possível? Porque um determinado componente foi escolhido ao invés de outro?
- facilitar a modificação: quais os efeitos das seguintes alterações?
- verificar a consistência dos valores propostos: esta decisão é consistente com as demais decisões tomadas?
- auxiliar a tomada de decisão fornecendo informações passadas: quais as decisões que requisitaram análises similares à da decisão atual? Como isto foi realizado anteriormente?

Para gerar este tipo de rationale, torna-se necessário colher as informações relevantes (por exemplo, a sequência de eventos) durante o projeto e ajustá-las adequadamente ao modelo de design. Quando o usuário requer uma informação, uma explicação é construída através de um mecanismo de inferência sobre o modelo de design e sobre a informação recolhida.

Em geral, o modelo de design envolve a representação da estrutura do artefato, do seu comportamento (como os componentes interagem entre si e com o meio onde o artefato será usado), da funcionalidade pretendida e do processo de tomada de decisão.

Dentre os sistemas existentes que empregam esta abordagem, alguns modelos de design são centrados no artefato enquanto que outros são centrados no processo de design.

Nos sistemas que usam modelos centrados no artefato, o modelo de design se concentra basicamente nos aspectos estruturais e comportamentais do artefato sendo produzido e na sua funcionalidade pretendida. Os aspectos decisórios (decisões, critérios, restrições, alternativas) não são representados explicitamente no modelo e são registrados apenas indiretamente através dos passos tomados pelo designer. Segundo Alberts [Alb93], apresenta como vantagem a possibilidade de simular e verificar uma descrição do artefato contra a realidade física. Por outro lado, a ausência de representação explícita dos aspectos decisórios dificulta a elaboração do design rationale.

Baudin et al. [BSZ90] apresenta um sistema que usa um modelo da estrutura e do comportamento do artefato e um conjunto de requisitos para avaliar alternativas de design. Os modelos do artefato são fornecidos através de um editor que permite a seleção de elementos a partir de uma biblioteca de componentes. O usuário escolhe um conjunto de alternativas de design e o sistema simula o comportamento de cada alternativa e faz uma avaliação do quanto elas satisfazem aos requisitos. O design rationale é apresentado através da comparação destas avaliações.

Gruber [Gru91] usa um modelo de simulação para gerar o rationale. Para documentar a funcionalidade pretendida do artefato, o designer constrói uma demonstração, através da configuração de um cenário de simulação, de como o dispositivo executa suas funções em seu ambiente de operação. Ao configurar o cenário, o designer acaba por fornecer informações que são utilizadas para predizer o comportamento do artefato e gerar explicações que servem como demonstrações das intenções do designer.

Iwasaki et al. [IFVC93] descrevem um sistema de suporte a decisão onde a estrutura e o comportamento do artefato são representados e manipulados explicitamente. O sistema permite que o designer especifique a funcionalidade do artefato em diferentes níveis de abstração. Esta especificação permite que o sistema avalie se o comportamento obtido a partir da especificação física do artefato é capaz de atender a funcionalidade pretendida. O design rationale é representado através do registro do processo de refinamento concorrente das especificações física e funcional do artefato.

Já nos sistemas que usam modelos centrados no processo: o modelo de design se concentra basicamente no processo de tomada decisão através da representação explícita dos aspectos decisórios (decisões, critérios, restrições, alternativas). A estrutura e o comportamento do artefato são representados implicitamente no modelo apenas como elementos a serem utilizados ou definidos pelo processo de decisão. Se por um lado, tornam-se explícitas as motivações que levaram o designer a obter aquele artefato, por outro, perde-se a possibilidade de simular o comportamento do artefato e verificar se ele produz efetivamente a funcionalidade esperada. ADD [Gar92] é um exemplo de sistema de apoio ao design que utiliza um modelo centrado no processo. Na seção 4 deste artigo será apresentada uma descrição detalhada deste sistema.

Um dos problemas da abordagem baseada em modelos é que ela requer que exista conhecimento suficiente sobre a atividade de design para que grande parte do modelo possa ser definido antecipadamente. Outro problema é que, ainda que isto seja possível, esta abordagem pressupõe que a atividade de design ocorra sempre no espaço de design definido pelo modelo. Mesmo que seja possível alterar o modelo, definindo

um novo espaço de design durante o próprio processo, podem ocorrer situações ocasionais não previstas que não se enquadrem no modelo.

3 - Comparação

Para podermos comparar as diferentes classes de abordagens apresentadas na seção anterior, desenvolvemos uma estrutura de comparação baseada nos seguintes fatores: tipo do design, paradigma de descrição da atividade de design, expressividade do rationale, uso do rationale, esforço requerido do designer, modo de recuperação da informação e suporte a negociação dos conflitos. Para cada fator, apresentamos uma tabela comparativa das abordagens e justificamos a classificação apresentada.

3.1 - Por tipo

O problema de design é um problema que lida com um espaço potencialmente aberto de soluções. Enquanto em certos domínios é possível definir um espaço de problema e restringi-lo de modo a encontrar uma solução ótima ou satisfatória, em outros torna-se necessário não apenas procurar uma solução dentro do espaço de problema, mas também redefinir, em tempo de design, o próprio espaço de problema em busca de novas soluções. Neste último caso, a atividade de design adquire um caráter exploratório que a torna difícil de ser completamente modelada antecipadamente. Tal como Gero [Ger90] e de acordo com o modo de formulação do espaço de design, classificamos o tipo de design em:

- *Rotineiro (R)*: O espaço de design é definido previamente. Busca-se uma solução dentro do espaço de problema através do uso de heurísticas. A atividade de design consiste em definir e negociar valores para os atributos do artefato a ser projetado.
- *Inovador (I)*: O espaço de design é basicamente definido por antecipação, embora seja possível fazer alterações não muito radicais ao longo da atividade de design. Busca-se a solução no espaço de problema, mas permite-se a alteração do espaço através da inclusão ou exclusão de variáveis de decisão. A atividade de design consiste na definição de valores para os atributos do artefato e na eventual reformulação do problema e de suas possíveis soluções.
- *Criativo (C)*: O espaço de design é definido durante a atividade de design. A medida que o design vai transcorrendo são definidos os elementos de decisão e seus valores. A busca no espaço de problema ocorre em paralelo com a sua criação. O espaço de design pode ser completamente modificado durante o design. A atividade de design consiste na formulação e reformulação do espaço de problema, e na busca da solução dentre deste espaço. Requer grande parcela de criatividade, geralmente envolvendo a inserção de novas tecnologias ou idéias.

Na tabela 3.1, apresentamos a comparação das abordagens segundo o fator tipo de design. A abordagem histórica, embora possa ser utilizada para os três tipos de design, é limitada na qualidade do rationale que oferece. As abordagens de notebook e, em especial, a argumentativa (por incentivar a realização da documentação) se adequam melhor para a documentação de design criativo, porque a medida que o modelo é criado pode-se fazer a documentação. Por outro lado, a documentação de design rotineiro e inovador nestas abordagens tende a obstruir o processo de inferência do designer e, por conseguinte, o andamento do projeto.

A abordagem baseada em modelos se adequa ao design rotineiro e inovador, pois nestes tipos de design pode-se ter um modelo de design pré-definido que consegue capturar automaticamente boa parte do *rationale*. Por outro lado, esta abordagem

torna-se muito custosa para documentar o design criativo porque o designer teria que formular formalmente todo o modelo durante o processo de design.

Abordagem	Notebook	História	Argumentativo	Modelo
Tipo	C	R, I, C	C	R, I

Tabela 3.1 - Comparação de Abordagens por Tipo de Design

3.2 - Por paradigma de descrição da atividade de design

Segundo Dorst e Dijkhuis [DD95], os sistemas para descrição dos processos de design podem ser classificados em:

- *Rational Problem-Solving (S)*: Adota uma visão positivista de design. Considera o design como um processo de busca num espaço de problema não completamente definido. Cada passo na direção da solução é tomado de acordo com a informação disponível. Pressupõe que haja um modelo racional-científico para a solução do problema, embora este não seja necessariamente conhecido previamente em sua totalidade.
- *Reflection-In-Action (A)*: Adota uma visão construtivista de design. Considera que as questões de design, e consequentemente, o espaço de problema, vão surgindo e sendo resolvidas na medida que o design vai transcorrendo. Encara a essência da atividade de design como sendo oportunística, isto é, considera design como um problema que envolve um mundo aberto de possibilidades que não pode ser descrito ou generalizado antecipadamente. É melhor lidar com o mundo na medida que as possibilidades vão surgindo.

A tabela 3.2 mostra que as abordagens de notebook, histórica e argumentativa adotam o paradigma de *reflection-in-action*, pois a documentação é feita durante o processo de design e não pressupõe nenhum conhecimento prévio de como o design é realizado. Já a abordagem baseada em modelos se enquadra, por sua própria natureza, no paradigma de *rational problem-solving*.

Abordagem	Notebook	História	Argumentativo	Modelo
Paradigma	A	A	A	S

Tabela 3.2 - Comparação de Abordagens por Paradigma de Design

3.3 - Por expressividade do rationale capturado

A expressividade do rationale representa os tipos de questões que a abordagem pode responder. Segundo este fator, as abordagens podem ser classificadas em:

- *O Que É (Q)*: apresenta as características estruturais do artefato projetado ou de suas partes.
- *O Que Faz (QF)*: apresenta a funcionalidade do artefato projetado ou de suas partes.

- *Como Faz (CF)*: apresenta como o artefato projetado, ou parte dele, atinge uma determinada funcionalidade.
- *Como Foi Definido (C)*: apresenta o procedimento adotado (sequência temporal de eventos) que produziu uma determinada decisão de design.
- *Porque (P)*: apresenta as motivações e o raciocínio que levaram a uma determinada decisão. Identifica as relações de dependência lógica entre decisões e as razões comportamentais pelas quais aquela decisão contribui para a obtenção da funcionalidade pretendida.
- *Porque Não (PN)*: apresenta as razões pelas quais uma determinada alternativa foi preterida em relação à escolhida através de uma análise de viabilidade da alternativa ou de uma análise de desempenho comparativo entre as duas opções.
- *O Que Se (QS)*: prediz quais as alterações que haveriam no projeto caso uma ou mais modificações fossem feitas nas premissas de design.

A tabela 3.3 demonstra que todas as abordagens são capazes de responder *O Que É* o artefato, porque ao longo da documentação são registrados todas as características necessárias a construção do artefato. Por outro lado, o único tipo adicional de questão que pode ser respondida pela abordagem histórica é *Como Foi Definido.* Já a abordagem de notebook é ainda mais restritiva, porque só responde a perguntas do tipo *Porque* quando estas tiverem sido registradas. A abordagem argumentativa é capaz de responder as questões *Como Foi Definido, Porque* e a um conjunto parcial de perguntas *Porque Não* (as que tenham respostas diretas a partir da documentação). Por fim, a abordagem baseada em modelos pode, em princípio, responder a qualquer tipo de questão possível desde que guarde um registro histórico dos eventos. Embora esta generalidade da abordagem baseada em modelos seja real, ela não significa, na maioria das vezes, que os sistemas que utilizem esta abordagem sejam capazes de fazê-lo por completo.

Abordagem	Notebook	História	Argumentativo	Modelo
Expressividade	Q, P*	Q, C	Q, C, P, PN*	Q, QF, CF, C, P, PN, QS

Tabela 3.3 - Comparação de Abordagens por Expressividade

3.4 - Uso do rationale capturado

Este fator classifica as abordagens segundo o tipo de uso a que elas se adequam.

- *Descrição (Dc)*: descreve as características do artefato projetado. Usada para a construção física do artefato. Tipicamente associada as perguntas *O Que É* e *O Que Faz.*
- *Explicação (E)*: demonstra por que o artefato foi projetado da maneira que foi. Usada para facilitar posteriores modificações de projeto e para aprendizado. Tipicamente associada as perguntas *Como Faz, Porque, Porque Não e Como Foi Definido.*

* atende apenas parcialmente

- *Simulação (S)*: permite avaliar o impacto das alterações das premissas de design sobre o projeto do artefato. Usada para aprendizado e para verificação durante o próprio design. Tipicamente associada a pergunta *Porque Não* e *O Que Se.*
- *Repetição (R)*: permite a reedição dos passos de design. Usada para reaproveitamento automático do processo para o design de artefatos similares.
- *Design (Di)*: apóia o design de projetos futuros através de fornecimento de casos.

Na tabela 3.4 vemos que as abordagens de notebook e argumentativa só podem ser empregadas na descrição e em parte da explicação do design. Já a abordagem histórica, além da descrição e parte da explicação, pode ser usada para repetição dos passos de design, quando combinada com um modelo formal da atividade de design. A abordagem baseada em modelos, pode ser usada para descrição, explicação, design, simulação (já que possui um modelo do espaço de design) e mesmo para a repetição do design, quando combinada com um registro histórico do processo.

Abordagem	Notebook	História	Argumentativo	Modelo
Uso	Dc, E*	Dc, E*, R	Dc, E*	Dc, E, S, R, Di

Tabela 3.4 - Comparação de Abordagens por Uso

3.5 - Por esforço requerido do designer durante a captura

Este fator identifica a quantidade adicional de trabalho que o projetista será submetido ao fazer a documentação. Ele é importante porque o quanto mais o projetista tiver que se desviar da sua atividade principal, o design, menos ele sentirá incentivado a realizar a documentação.

- *Pequeno (P)*: a participação do designer no processo de captura consiste em definir as características do artefato. Enquanto o projetista procede normalmente o design, o sistema computacional obtém automaticamente o rationale.
- *Médio (M)*: A maior parte do rationale é adquirida pelo sistema computacional. No entanto, quando o sistema não é capaz de gerá-lo por si só é requerida a participação do designer para que ele forneça o rationale.
- *Grande (G)*: O rationale é totalmente fornecido pelo designer. O sistema computacional é apenas uma ferramenta que facilita a documentação feita pelo designer.

A tabela 3.5 mostra que a abordagem de notebook pode requerer um esforço médio ou grande do designer, de acordo com quanto ele se disporá a documentar. Já a abordagem histórica pode ser sempre capturada automaticamente e, por consequência, requer nenhum ou pouco esforço do designer. No outro lado está a abordagem argumentativa que sempre demanda um grande esforço de documentação por parte do designer. Já abordagem por modelos pode requerer um esforço pequeno ou médio, de acordo com a adequação do modelo de design para refletir a atividade de design.

Abordagem	Notebook	História	Argumentativo	Modelo
Esforço	M, G	P	G	P, M

Tabela 3.5 - Comparação de Abordagens por Esforço Requerido do Designer

3.6 - Modo de recuperação da informação

Este fator classifica as abordagens segundo os mecanismos que podem ser oferecidos para a recuperação do rationale.

- *Realizada pelo Usuário (U)*: Usuário recupera informação percorrendo e analisando sequencialmente o rationale documentado.
- *Realizada Através de Mecanismo de Navegação (N)*: Sistema fornece mecanismo para navegação no documento, através da ligação em grafo das informações interrelacionadas do rationale. O acesso a informação se dá através da navegação no documento pelo caminhamento no grafo.
- *Realizada Através de Mecanismo de Indexação (I)*: Sistema fornece mecanismo para indexação da informação contida no documento. O acesso a informação se dá através de questões feitas pelo usuário.
- *Realizada Através de Mecanismo de Explicação (E)*: Sistema fornece mecanismo para explicação, onde o usuário pode consultar a informação desejada através da formulação de perguntas, que serão respondidas sob demanda por inferência no documento.
- *Realizada Através de Mecanismo de Simulação (S)*: Sistema fornece mecanismo para simulação, onde o usuário pode consultar a informação desejada através da formulação de cenários de design.

A tabela 3.6 mostra a comparação segundo este fator. As abordagens de notebook, histórica e, frequentemente, a argumentativa oferecem apenas para o usuário a opção de consultar a informação desejada no documento passivo através de uma busca não orientada pelo documento. Contudo, na abordagem argumentativa podem ser fornecidos mecanismos para navegação no documento e para indexação da informação. Já a abordagem de modelos permite a recuperação da informação através de explicação e simulação, em virtude da facilidade de inferência proporcionada pelo modelo que lhe é subjacente.

Abordagem	Notebook	História	Argumentativo	Modelo
Recuperação	U	U	U, N, I	E, S

Tabela 3.6 - Comparação de Abordagens por Modo de Recuperação da Informação

3.7 - Suporte a negociação dos conflitos

Este fator analisa o tipo de apoio fornecido pela abordagem para a resolução de conflitos num ambiente onde diferentes agentes interagem.

- *Nenhum Suporte (NS)*: Computador não dá apoio a resolução de conflitos. Os usuários negociam diretamente entre si a solução dos conflitos.
- *Dirigida pelo Usuário (DU)*: Computador suporta a negociação de forma passiva, ou seja, funciona como meio de comunicação dos conflitos e dos diferentes rationales aos agentes envolvidos. Quem define a solução para os conflitos são os usuários.
- *Dirigida pelo Sistema (DS)*: Computador resolve autonamente alguns dos conflitos, mas não prescinde da participação do usuário na solução de outros.

Neste último caso, ele suporta a decisão fornecendo as informações que dispõe sobre o conflito.

- *Realizada pelo Sistema (RS)*: Computador resolve autonomamente todos os conflitos que aparecem.

A tabela 3.7 mostra a comparação segundo este fator. Enquanto as abordagens de notebook e histórica não fornecem nenhum suporte computacional a resolução dos conflitos que não seja o próprio documento produzido, a abordagem argumentativa pode suportar a abordagem dirigida pelo usuário ou até a dirigida pelo sistema [PSL95]. Por fim, a abordagem por modelos pode, potencialmente, fornecer qualquer tipo de suporte computacional.

Abordagem	Notebook	História	Argumentativo	Modelo
Negociação	NS	NS	DU, DS	DU, DS, RS

Tabela 3.7 - Comparação de Abordagens por Suporte a Negociação dos Conflitos

3.8 - Análise da Comparação das Abordagens

Através da análise das tabelas 3.1 a 3.7, constatamos que a abordagem baseada em modelos, em geral, é capaz de produzir rationale mais rico do que as demais. Mesmo assim, ela não é capaz de atender sozinha todos os requisitos necessários a uma boa documentação de design. Nos fatores expressividade e uso do rationale, a abordagem histórica fornece um bom complemento a abordagem de modelos. Por sua vez, nos fatores tipo de design e paradigma de descrição, a abordagem argumentativa oferece melhor suporte ao design criativo e ao método *reflection-in-action*. Além disso, os mecanismos de navegação e indexação que podem ser fornecidos para recuperação da informação na abordagem argumentativa podem completar os mecanismos de recuperação da abordagem por modelos. Assim posto, concluímos que a combinação destas três abordagens, com ênfase na abordagem por modelos, é a forma mais adequada de abordar o problema de documentação do design rationale.

4 - Descrição do ADD

ADD (*Active Design Document ou Documento Ativo de Design)* [Gar92] se baseia na idéia de se capturar *rationale* através da transformação dos documentos de design de repositórios estáticos de dados em modelos computacionais do próprio design. Assim, ao invés do designer registrar diretamente os dados relativos ao *rationale* na medida que o design transcorre, ele apenas interage com um modelo computacional do processo de design. Este modelo é utilizado por ADD para registrar diretamente os dados relevantes do *rationale*. Posteriormente, quando torna-se necessária a recuperação do *rationale*, o documento ativo pode gerar uma explicação a partir do modelo computacional e das informações registradas.

Desta forma, além de dar suporte ao próprio design, ADD facilita o processo de captura de *rationale*, liberando o designer da árdua tarefa de registrar passo a passo todo o seu processo de raciocínio e decisão, e recuperação do *rationale*, ao gerar explicações por demanda.

Portanto, para que o ADD possa ser utilizado, é necessário que ele incorpore um modelo computacional do processo de design do artefato a ser produzido. Em geral, ADD iniciará sua execução com um modelo inicial elicitado dos especialistas do

domínio por engenheiros de conhecimento. À medida que o design vai transcorrendo este modelo vai sendo ajustado pelo próprio designer para retratar de maneira adequada os processos de raciocínio e decisão que foram utilizados.

É importante ressaltar que, em princípio, nada impede que ADD seja usado sem um modelo inicial. Quando isto ocorre, o designer deve fornecer todo o modelo computacional na medida que procede o design.

ADD começa sua interação com o designer permitindo que ele defina os valores que especificam o artefato a ser projetado. Baseado no seu modelo computacional, ADD pode gerar expectativas para as decisões de design na medida que os designers vão propondo valores. Quando as expectativas estão corretas, isto é, o valor calculado pelo modelo coincide ou se aproxima do valor fornecido pelo designer, isto significa que o modelo computacional é representativo do processo de decisão do designer e pode ser utilizado para gerar *rationale*. Por outro lado, quando existe uma diferença relevante entre o valor proposto e o valor calculado, isto significa que é necessária uma correção do modelo ou dos valores propostos. Neste momento, ADD interage com os designers apresentando a forma de raciocínio que o levou a obter um valor diferente do fornecido e solicitando que eles reconsiderem o valor proposto ou que ajustem o seu modelo, para que ele possa refletir o novo processo de design. Caso o designer não queira nem alterar o valor proposto nem ajustar o modelo, ele pode, ainda, forçar a execução com dados inconsistentes.

O modelo computacional de design é representado em ADD através de uma rede de dependência paramétrica. Cada parâmetro desta rede representa uma entidade de design que pode ser utilizada no desenvolvimento do processo e na elaboração e definição do artefato. Por exemplo, estas entidades podem representar propriedades da estrutura física do artefato (como o material que será composto, a altura, a largura e o comprimento), condições operacionais e ambientais sob as quais é suposto que o artefato irá funcionar (como pressão e temperatura), propriedades relacionadas com aspectos pragmáticos do contexto de design (como custo financeiro de parte do artefato e facilidade de construção e manutenção) e, mesmo, entidades auxiliares que modelem aspectos cognitivos do processo de decisão do designer (como casos similares ao design em questão).

Cada parâmetro da rede é associado aos parâmetros dos quais depende para ser definido. Além disso, ele detém conhecimento de como estes parâmetros o influenciam. Desta forma, pode-se sempre determinar quais parâmetros influenciam a outro e quais parâmetros são influenciados por um determinado parâmetro. Este tipo de informação habilita ADD a saber quando um determinado parâmetro pode ser calculado, a calcular o valor deste parâmetro e a identificar os parâmetros que necessitam ser recalculados quando ocorre uma alteração em outro.

Os parâmetros da rede de dependência são subdivididos em três classes: Primitivos, Derivados e Decididos. Parâmetros Primitivos são aqueles que só podem ser definidos pelo usuário ou pelo contexto de uso do sistema. Parâmetros Derivados podem ser definidos pelo usuário ou podem ser calculados diretamente (através da aplicação de uma fórmula ou de um algoritmo exato) a partir dos parâmetros que o influenciam. Por fim, Parâmetros Decididos são aqueles que modelam o processo de decisão do designer. Eles também podem ser definidos pelo usuário ou podem ser calculados a partir dos parâmetros que o influenciam. O que os diferencia dos parâmetros derivados é o processo de cálculo envolvido. Nos parâmetros decididos, cada parâmetro têm associado a ele um conjunto de alternativas, critérios, restrições e funções de avaliação

destas alternativas a partir dos critérios e restrições. O valor calculado do parâmetro decidido é a alternativa que melhor satisfez aos critérios e restrições existentes.

A Figura 4.1 ilustra a representação utilizada em ADD apresentando parte de uma rede paramétrica de um sistema para design de processo em plataformas de petróleo em águas profundas. A parte da rede representada objetiva identificar qual o projeto de plataforma existente no banco de dados que mais se assemelha ao atual. Os parâmetros primitivos Vazão Máxima de Óleo, Vazão Máxima de Gás e Vazao Máxima de Água (indicados na figura por elipses) são definidos pelo usuário e são utilizados para calcular, através da sua soma, o parâmetro derivado Vazão Total do Fluido (indicado por um retângulo). Este, por sua vez, é utilizado como critério, juntamente com o parâmetro primitivo Tipo da Plataforma (usado como restrição), para seleção dentre as alternativas existentes, da Plataforma Similar (parâmetro decidido, indicado por um retângulo de bordas arredondadas) cuja Vazao Total de Fluido mais se aproxima da atual e que seja do Tipo de Plataforma especificado.

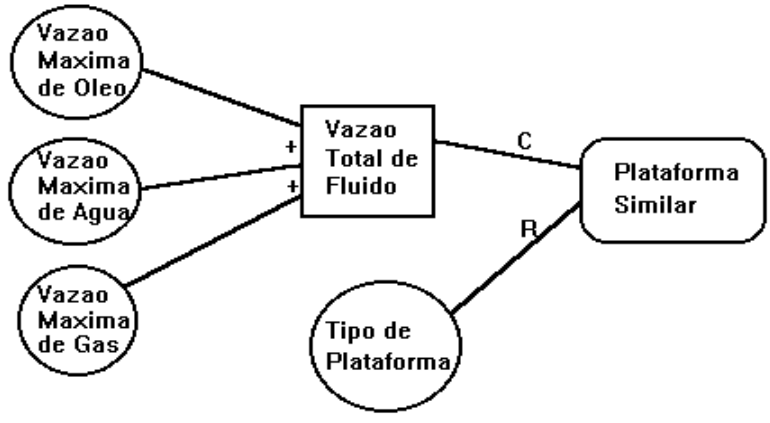

Figura ?.1 - Rede Paramétrica

Num determinado momento, a rede paramétrica pode não representar adequadamente o modelo do processo de design empregado por um designer específico. Quando isto acontece, ADD permite que a rede seja alterada para se ajustar ao novo modelo. A aquisição de conhecimento no ADD pode ser realizada através de ajustes nas funções de avaliação de critérios e alternativas, ajustes nos procedimentos de cálculo dos parâmetros e através da própria alteração da topologia da rede. Neste último caso, ADD possibilita que parâmetros sejam eliminados e criados, que as associações sejam refeitas e que novos procedimentos de cálculo sejam incorporados.

Em geral, ao final do processo de design, o modelo está ajustado para refletir o procedimento de raciocínio e decisão adotado pelo designer. De posse deste modelo é possível gerar explicações sobre:

- qual o valor de determinado parâmetro?

- porque foi obtido este valor?
- porque não se escolheu uma outra alternativa?
- qual o impacto de se alterar um determinado parâmetro?

Adicionalmente, ADD também registra, na forma de histórico, todos os seus cálculos de parâmetros e todas as suas interações com o usuário. Desta forma é possível apresentar toda a sequência de eventos realizados e suportar, juntamente com o modelo final de design, a inferência de parte das motivações (por exemplo, quais os motivos para alterar o modelo original) que levaram o designer a proceder daquela forma.

Portanto, a base informacional que ADD utiliza para gerar *rationale* é composta pelo modelo final do processo de design e pela sequência cronológica de interações e cálculos. A apresentação do *rationale* em ADD consiste na apresentação do histórico do processo, da parte da rede paramétrica envolvida e de uma análise comparativa das alternativas consideradas ao se tomar uma decisão, integrados por um texto explicativo em linguagem natural. Isto é superior ao *rationale* que os modelos históricos e argumentativos apresentam porque permite a geração, por demanda, de um tipo mais variado e manipulável de rationale.

A interação de ADD com o designer inicia com o designer propondo valores para um conjunto de parâmetros de design. Quando o designer questiona o valor de outros parâmetros, ADD entra no modo de antecipação. Neste modo, ADD pode calcular o valor dos parâmetros a partir do seu modelo computacional de design e dos valores fornecidos pelo designer. O designer pode não ficar satisfeito com os valores obtidos e resolver alterar alguns dos valores que havia proposto. Sempre que recebe valores de parâmetros, ADD entra no modo de propagação e verifica se os parâmetros alterados influenciam outros parâmetros cujos valores já foram calculados ou definidos. Caso isto ocorra, ADD recalcula estes parâmetros. É importante notar que ADD atua sempre como um assistente de design e, portanto, nunca avança na rede além do ponto onde o designer chegou, seja por atribuição ou por questionamento de valor.

Caso os valores recalculados por ADD conflitem com valores que foram propostos pelo designer, ADD entra no modo de reconciliação. Neste modo, ADD apresenta a linha de raciocínio que o levou a calcular um valor diferente do proposto pelo usuário. ADD solicita, então, que o usuário altere o valor proposto ou que ajuste o seu modelo para refletir a linha de raciocínio utilizada pelo designer. Caso, por algum motivo, o usuário não queira alterar o valor proposto ou ajustar o modelo, ele pode ainda forçar a execução com o valor proposto. O problema desta atitude é que o modelo de ADD não mais representará a linha de raciocínio empregada pelo usuário. A alternativa que ADD fornece nessas situações é permitir que o usuário faça uma anotação em texto livre explicando as motivações para a sua decisão.

Ao final do processo de design, o artefato foi definido e ADD possui um modelo que representa a linha de raciocínio utilizada pelo designer e um histórico dos eventos realizados durante o processo. Esta informação é armazenada para que, no futuro, quando for necessário recuperar o *rationale*, ela seja reincorporada ao ADD e o usuário possa requerer as informações desejadas sobre o processo.

5 - Proposta para a Extensão do ADD

Apresentamos a seguir uma proposta para extensão do ADD que visa torná-lo ainda mais expressivo como ferramenta para captura e recuperação de design rationale. Serão abordados sucessivamente as motivações que nos levaram a propor uma

extensão do ADD baseada em anotações estruturadas, uma breve introdução às anotações e as formas de representação, aquisição e recuperação das anotações.

5.1 - Motivação para a Extensão do ADD

Como foi visto na seção 4, o modelo adotado por ADD permite a representação de uma ampla gama de conhecimentos e metodologias empregadas frequentemente em design. Como a atividade de design, além de envolver a busca de soluções num espaço de estados, também consiste num processo de exploração criativa onde são produzidos novos espaços de estados a serem avaliados [QG96], ADD possibilita que o seu modelo seja alterado para refletir novas soluções e novos métodos de design.

Embora ADD forneça esta possibilidade de ajuste do modelo formal de conhecimento, existem situações onde o designer pode preferir fornecer informações complementares às já representadas no modelo ou no histórico, sem ter que alterar o modelo formal diretamente. Tais situações podem ocorrer porque:

a. O usuário quer reforçar o que está representado no modelo. Neste caso, o modelo é adequado para a realização do design mas o usuário quer complementar o conteúdo da explicação apresentando uma argumentação informal que descreve aspectos não expressos no modelo.

b. Embora o modelo seja representativo do processo de decisão (isto é, o valor calculado por ADD equivale ao valor proposto pelo designer), ele não revela alguma intenção do usuário ao propor este valor para o parâmetro. Por exemplo, o usuário pode ter proposto um valor para ajustar outro parâmetro afetado por ele. Esta informação não seria registrada explicitamente pelo modelo.

c. O usuário não consegue se expressar através da rede paramétrica. Neste caso, o conhecimento que ele quer expressar é representável através da rede mas existe uma dificuldade do usuário em fazê-lo.

d. Não convém expressar o conhecimento a ser fornecido através da rede paramétrica. Nestes casos, embora o conhecimento seja relevante para o caso particular e possa ser representado na rede, ele é totalmente ocasional (isto é, provavelmente não será repetido com frequência em outros designs). Exemplos deste tipo de conhecimento são o conhecimento episódico [Vis95], no qual o designer usa uma analogia com uma situação externa (um episódio) ao domínio de design para embasar sua decisão e o conhecimento proveniente de resultados de prototipação.

e. O conhecimento a ser inserido é completamente novo (não previsto no modelo) e sua introdução na rede paramétrica envolveria alterações extremamente trabalhosas e complexas. Por exemplo, o lançamento de um novo tipo de componente eletrônico no mercado poderia alterar completamente parte do modelo de configuração de computadores. Provisoriamente, até quando o modelo fosse ajustado, as informações do novo componente e como ele afeta o design poderiam ser fornecidas complementarmente.

f. O usuário quer justificar uma alteração no modelo. Neste caso, ele pode querer explicar porque o novo procedimento de cálculo é mais adequado que o anterior.

O ADD, através de sua opção de representação de conhecimento e raciocínio, fornece aos usuários uma linguagem de expressão, claramente demarcada, onde um conjunto prévio de abstrações sobre o domínio e intenções de ação e comunicação é fixo. Ocorre que, como foi visto na seção 3.2, a atividade de design é por natureza flexível. Neste sentido as situações enumeradas acima caracterizam uma necessidade de expansão do meio expressivo de ADD, seja no sentido de alcançar uma meta linguagem (como em (a) ou (f)), seja no sentido de uma extensão expressiva ao nível da própria linguagem ADD (como em (b), (c) ou (d)). Já casos como (e) esbarram em limitações computacionais básicas, no sentido de que nenhuma linguagem artificial é capaz de conter em si mesma uma componente de criatividade e inovação capaz de gerar por si só novos tipos de objetos no domínio e seus respectivos modos de expressão e restrições comunicativas.

Uma forma de capturar estes tipos de informações é permitir que o designer faça anotações livres sobre o processo de design. Nas situações em que o designer opta por não ajustar o modelo e forçar uma decisão, ADD já possibilita que ele incorpore uma anotação livre que servirá posteriormente como *rationale* daquela decisão. Este tipo de solução está em consonância com a conclusão obtida na seção 3.8, que indica a necessidade de combinarmos as abordagens histórica, argumentativa e baseada em modelos para a obtenção de uma documentação mais rica.

Contudo, o mecanismo de anotações livres faz um uso muito restrito destas informações. Embora este tipo de anotação atenda, em parte, as necessidades de explicação do porque daquela atitude, ela acaba por desprezar informações importantes do contexto da decisão, que o modelo e o histórico de ADD podem proporcionar.

Uma outra possibilidade seria utilizar um modelo argumentativo [KR70, MYBM91, Shu96] para que o designer possa expressar as informações complementares usando uma metodologia de representação de *rationale*. Embora este tipo de anotação incentive uma maior reflexão sobre o design e permita que o designer forneça as informações de uma forma mais organizada, ela também desconsidera informações contextuais importantes que podem ser proporcionadas pelo modelo e pelo histórico de ADD.

Tais informações contextuais podem servir, juntamente com informações contextuais fornecidas pelo próprio anotador, como indexadores de um mecanismo de anotações estruturadas que permitirão uma maior utilização e uma recuperação mais adequada destas anotações. Anotações estruturadas são, por assim dizer, anotações informais em linguagem natural organizadas segundo uma estrutura de indexação que permita captar o contexto no qual as anotações foram feitas e que possibilite a recuperação desta anotação quando ela for necessária.

De fato, as grandes vantagens da integração do mecanismo de anotações estruturadas ao ADD são proporcionar uma estrutura de informações ortogonal ao modelo formal de ADD e permitir capturar conhecimento de modo informal.

Redes de dependência paramétrica são suficientes para representar qualquer relação causal necessária para expressar como um determinado parâmetro é calculado a partir de outros. Desta forma, é sempre possível utilizar uma rede deste tipo como modelo de como determinado artefato foi projetado. Contudo, essas redes apenas registram as relações de dependência existentes no domínio. A integração do mecanismo de anotações estruturadas permitirá ao ADD capturar, expressar e inferir outros tipos de relações (causais, temporais, estruturais, organizacionais, funcionais, dentre outras)

através de um conjunto de anotações que se correlacionam formando uma rede de informações ortogonal ao modelo de ADD. Por exemplo, através das anotações poderão ser expressas explicitamente as relações entre estrutura e funcionalidade do artefato, que são utilizadas como princípio básico para muitos sistemas de design [RG96, IFVC93]. Outro exemplo deste tipo de uso das anotações seria permitir refletir claramente a estrutura organizacional num ambiente multi-agente, isto é, o processo de negociação e a estrutura hierárquica na qual os diversos agentes estão envolvidos, o que é de suma importância em design concorrente [FKS95].

Junto com o mecanismo de anotações pode-se incorporar ainda uma classificação dos parâmetros da rede de dependência segundo o tipo de entidade do domínio de design que eles representam. Esta classificação pode ser utilizada para ampliar as formas de recuperação das anotações.

Desta forma, com a incorporação do mecanismo de anotações estruturadas ao ADD aumenta-se o poder de expressão de ADD em termos de espaço de design *rationale* que pode ser representado, capturado e recuperado.

Além disso, ADD será capaz de capturar informalmente o rationale que não necessita ser expresso no modelo formal, em particular, o conhecimento de design oportunístico que não pode ser antecipadamente formalizado no modelo. Segundo Shum [Shu96], a idéia de se permitir a incorporação de argumentação informal encontra suporte em recentes pesquisas na área de design colaborativo:

> *Dentro da área de pesquisa em hipertexto e design colaborativo é, portanto, encorajador ver a recente ênfase dada na importância de permitir que os designers se expressem ambiguamente, ao invés de precisa e explicitamente.* [Shu96]

Por outro lado, as vantagens que esta extensão de ADD tem sobre modelos puramente argumentativos são a liberação do designer da árdua tarefa de registrar todo o rationale, pois grande parte deste conhecimento já está embutido no modelo, e o fato de que o conhecimento informal se torna muito mais fácil de recuperar, pois esta informação pode ser organizada de acordo com o contexto na qual é fornecida e pode ser acessada através de mecanismos de indexação e inferência.

5.2 - Anotações em ADD

Podemos traçar um paralelo da abordagem que utilizaremos com a de GHS (General Help Systems) [Sou92, Sou93]. GHS é um sistema que se baseia em anotações estruturadas para dar suporte a busca de informação, navegação e aprendizado em redes de informação, como a Internet. Tal como no nosso problema, nestas redes não se pode prever, antecipadamente e por completo, quais serão as necessidades do usuário, devido ao grande volume e variedade de informação e a seu constante crescimento. A seguinte citação enfatiza a necessidade de se lidar com situações contingenciais.

> *A maneira como os usuários adaptarão recursos às necessidades é por definição algo que não pode ser completamente antecipado pelos projetistas, desde que esta adaptação emerge de características completamente contingenciais.* [Sou93]

Em decorrência deste fato, GHS adota o mecanismo de anotações estruturadas para **facilitar** aos usuários fornecerem e adquirirem informações sobre os recursos disponíveis e poderem, por conseguinte, resolver seus problemas. Este é também o objetivo de nossa proposta. A compreensão de GHS, portanto, nos forneceu algumas indicações de como a captura e recuperação de rationale baseada em anotações

estruturadas devem ser feitas. Reconhecendo este fato, denominamos de **ADD-GHS** o ambiente proposto para estender o ADD.

A figura 5.1 apresenta a estrutura de anotações de GHS [Sou93]. A partir dela podemos identificar a questão central tratada por de Souza [Sou92, Sou93] em relação ao GHS. A parte estruturada das anotações cumpre determinadas funções linguísticas como as de indexação, definição do discurso e metalinguagem (conforme indicado na figura pelo texto em itálico). Já a parte de texto livre atende ao caráter criativo e inovador de toda atividade inteligente humana. Estes fatores vêm justamente ao encontro das considerações feitas na seção 5.1, sobre a contribuição de um mecanismo de anotações no ambiente ADD.

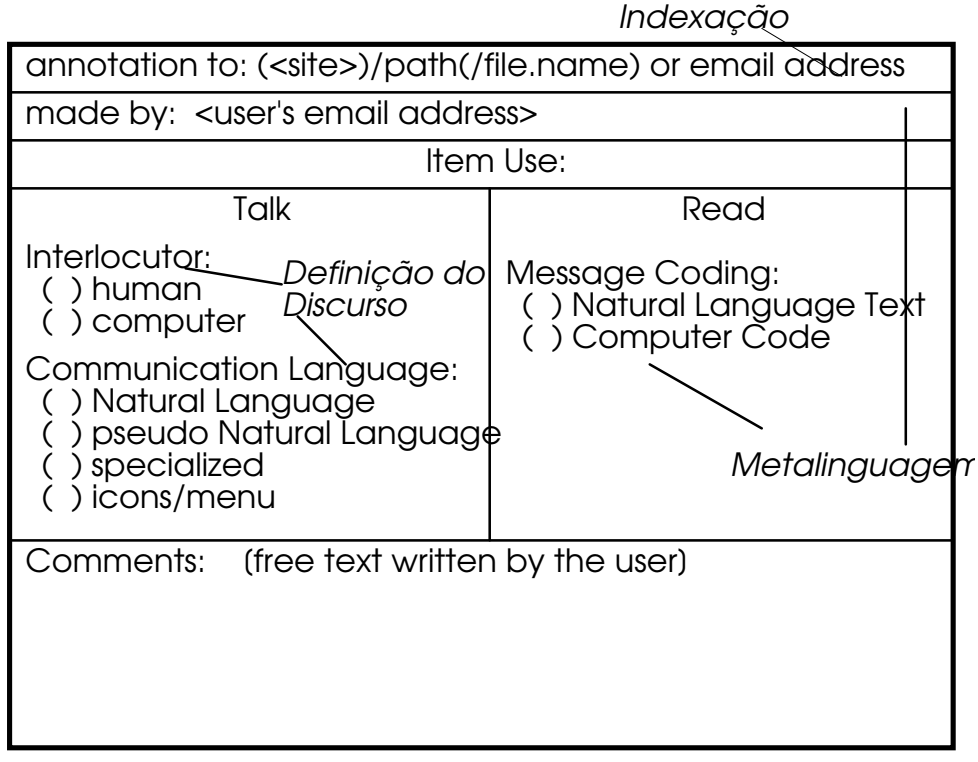

Figura 5.1 - Estrutura da Anotação em GHS

Podemos destacar ainda, dentre as indicações fornecidas por GHS, os requisitos que o mecanismo de anotações de ADD-GHS deve cumprir para ser efetivamente utilizado pelo designer:

1. deixar claro para o designer quais os benefícios que ele terá ao usar as anotações. No caso de ADD-GHS, os benefícios são permitir a inserção de qualquer conhecimento adicional necessário a documentação de design e a possibilidade de recuperação contextualizada desta informação durante uma explicação ou no próprio design;
2. não deve exigir que o designer tenha de preencher uma grande quantidade de informações para fornecer e recuperar as anotações pois isto aumentaria o custo do uso deste mecanismo e, consequentemente, o desestimularia a utilizá-lo;
3. procurar fornecer apenas as informações que são relevantes para o usuário quando da recuperação da informação (nada adianta fornecer as informações relevantes misturadas com um número incontável de

informações que não dizem respeito a questão formulada pelo designer pois ele teria o custo adicional de separar o que lhe interessa, o que pode inviabilizar o uso do mecanismo);

4. sempre fornecer a informação relevante existente (se a informação existe, deve ser possível recuperá-la).

Para definirmos o mecanismo de anotações de ADD-GHS, além de termos em mente os fatores acima, é importante termos um bom entendimento sobre o que são e para que servem estas anotações. Para isto, discutimos a seguir os objetos aos quais a anotação pode se referir, os motivos que levam o designer a fazer uma anotação e os usos que podem ser feitos das anotações.

5.2.1 - Objeto da Anotação

As anotações podem ser diferenciadas de acordo com o objeto ou objetos a que se referem quando capturadas. Se a anotação se refere a um único parâmetro ela é **específica** e, portanto, relacionada a uma única decisão. Este tipo de anotação expressa uma extensão do modelo de ADD no sentido de aumentar a perspicuidade sobre um elemento do domínio.

Caso a anotação se refira a um grupo de parâmetros ela é **coletiva**. Neste caso, ela revela relacionamentos não capturados pela rede paramétrica. Este tipo de anotação expressa uma extensão do modelo de ADD no sentido de elevar os níveis de abstração representados no domínio.

Por sua vez, se ela se refere ao modelo como um todo ou a uma parte do domínio que não tenha sido representada formalmente através de parâmetros, ela é **extra**. Este tipo de anotação visa fornecer informações que impactam o uso do modelo como um todo ou de partes do modelo. Desta maneira, elas expressam um aumento de perspicuidade sobre todo o modelo ou uma extensão do próprio conjunto de objetos do domínio.

5.2.2 - Motivo da Anotação

As anotações também podem ser classificadas segundo o contexto de captura que as motivou. Este tipo de classificação é possível de ser realizada porque a existência do modelo de ADD restringe o conjunto de situações plausíveis em que o designer pode desejar fazer uma anotação.

Basicamente, uma anotação pode ser feita para justificar uma decisão, para justificar uma alteração do modelo, para complementar o modelo ou para justificar um procedimento de design adotado pelo projetista.

Quando a anotação é feita para **justificar uma decisão**, o anotador pode estar objetivando:

- explicar um valor *proposto* para o parâmetro,
- explicar porque um valor foi *forçado*,
- explicar porque ele concorda com o valor *calculado* por ADD,
- para apresentar argumentos a favor ou contra valores do parâmetro num processo de negociação ou para justificar o valor *negociado*.

Como anotações feitas para justificar uma decisão objetivam explicar o valor de um parâmetro, elas são sempre específicas.

Quando a anotação é feita para **justificar a alteração do modelo**, os motivos podem ser relacionados com:

- a *alteração* da forma de cálculo de um parâmetro,
- a *inclusão* de novos parâmetros na rede,
- a *exclusão* de parâmetros na rede.

Neste caso, como as alterações são sempre realizadas sobre o modelo paramétrico, anotações que justificam a alteração do modelo podem ser específicas, quando a alteração é realizada sobre um parâmetro, ou coletivas, quando a alteração envolve um grupo de parâmetros. Se é específica, ela referencia a fórmula de cálculo de um parâmetro derivado ou referencia a alteração (inclusão, exclusão, modificação) de alternativas, restrições e critérios de parâmetros decididos. Se é coletiva, ela enfoca porque o novo método de design daquela parte do modelo é mais efetivo que a anterior.

Quando a anotação é feita para **complementar o modelo**, o designer pode estar querendo:

- *incluir* novas informações ao modelo,
- *excluir* informações do modelo,
- *discordar* das informações existentes no modelo.

Anotações para complementar o modelo são as de uso mais amplo pois podem ser específicas, coletivas ou extras. Se ela é específica serve para que o designer possa discordar ou reforçar uma fórmula de parâmetro derivado ou alternativas, restrições, e critérios de parâmetros decididos. Se ela é coletiva serve para que ele enfatize relacionamentos que não sejam de dependência entre os parâmetros ou para fornecer informações da conjuntura de design sobre a validade de um tipo de procedimento. Se ela é extra, ela permite que sejam fornecidas informações da conjuntura de design que impactem a aplicação do modelo como um todo ou para a inserção de novas informações que não estão expressas no modelo formal. Ainda, anotações complementares extras servem para expressar informações a respeito das relações entre estrutura, funcionalidade e conjuntura de uso do artefato que não podem ser expressas via rede de dependências e para expressar informações relativas ao aspecto organizacional da equipe de design, quando em ambientes multi-agentes.

Quando a anotação é feita para justificar um **procedimento de design**, o anotador pode, entre outras coisas, estar querendo indicar:

- um processo de *ajuste por tentativa e erro*,
- um processo de *geração e comparação de alternativas*.

Anotações que justificam um procedimento de design podem ser específicas ou coletivas, dependendo da amplitude de alteração do modelo necessária para a geração de um ciclo de design. Se é necessário alterar apenas um parâmetro para gerar uma alternativa ou para ajustar o parâmetro desejado, ela é específica. Por outro lado, se é necessário alterar vários parâmetros, ela é coletiva.

5.2.3 - Uso da Anotação

Uma anotação pode ser utilizada para atender a diferentes propósitos. O primeiro tipo de uso que identificamos é o de **explicação**. Neste tipo de uso se enquadram as apresentações contextualizadas das anotações quando da resposta às perguntas de porque, porque não e como determinada decisão foi tomada. O segundo tipo de uso é o de **suporte a tomada de decisão** do projetista. As anotações que vão sendo

realizadas ao longo do desenvolvimento de diferentes projetos são armazenadas numa base de anotações que pode ser consultada a qualquer momento pelo projetista. Deste modo, antes de tomar uma decisão, ele pode obter informações, feitas no passado, a respeito de decisões similares àquela que ele se depara atualmente, através de um mecanismo de recuperação de anotações que infere quais as anotações que podem ser relevantes. Um outro tipo de utilidade para as anotações é o seu uso como fonte de **aquisição de conhecimento**. À medida que a base de anotações for crescendo pode ser interessante fazer uma análise destas anotações para identificar o que necessita ser mudado no modelo formal de ADD para refletir de modo mais adequado o processo de design, diminuindo assim a necessidade do designer de utilizar as anotações. É possível vislumbrar ainda um outro tipo de uso das anotações, embora este seja muito mais difícil de ser implementado. Elas poderiam ser usadas para o **aprendizado automático**, o qual envolveria uma análise da sua estrutura de indexação, do texto da anotação e do contexto de captura e uso.

Nosso interesse no desenvolvimento desta extensão é utilizar as anotações para a explicação, suporte ao design e para aquisição do conhecimento. Contudo, vamos enfatizar neste trabalho a sua utilização na explicação e como elemento de suporte ao design.

5.3 - Representação das Anotações

Apresentamos nesta seção o modelo de representação que será utilizado para armazenamento das anotações. Tomando como base as informações da seção anterior, propomos diferentes estruturas de indexação para as anotações específicas, coletivas e complementares, cujos esquemas serão apresentados ao longo desta seção. Antes, porém, introduzimos um modelo de classificação de parâmetros que deve ser incorporada a base de conhecimento de ADD para permitir um uso ainda mais amplo das anotações específicas e coletivas.

5.3.1 - Classificação dos Parâmetros segundo o Domínio de Design

Para que se possa usar o mecanismo de anotações em toda a sua potencialidade, é necessário identificar as classes de entidades envolvidas no processo de design. Em ADD, os parâmetros da rede de dependência representam as entidades do domínio de design. Estas entidades podem ser descritoras:

1. da **estrutura** física do artefato - descrevem as características do artefato ou de suas partes componentes. Por exemplo: altura, largura e peso de um equipamento.
2. do **estado** físico com o qual o artefato vai interagir - descrevem as características do ambiente sobre o qual o artefato vai atuar. Por exemplo: pressão, temperatura, cargas.
3. das **funcionalidades** que o artefato deve cumprir quando em operação - descrevem as alterações no ambiente que o artefato deve ocasionar quando em operação. Estes parâmetros descrevem a funcionalidade primária do artefato, pois o objetivo básico da criação de um artefato é sempre alterar o ambiente de modo a atender uma determinada expectativa. Variação de temperatura e variação de pressão são exemplos de parâmetros deste tipo.
4. de **propriedades** que o artefato deve possuir - descrevem características que o artefato deve possuir para melhor satisfazer as necessidades do usuário ou do construtor do artefato. Estes parâmetros são representativos das funcionalidades secundárias (adicionais) que o artefato deve atender.

Por exemplo: baixo custo de construção, facilidade de manutenção, durabilidade.

5. do **método de cálculo** utilizado: descrevem elementos auxiliares que são utilizados para calcular os valores dos parâmetros descritores da estrutura, do estado, das funcionalidades e das propriedades de design. Por exemplo: fatores de ajuste, probabilidades, percentuais, somatórios.
6. do **processo de design** utilizado: descrevem elementos utilizados na modelagem do processo cognitivo de inferência do designer. Por exemplo: casos similares (abdução).

Embora em ADD estas entidades sejam representadas pelos parâmetros primitivos, derivados e decididos da rede, esta classificação é apenas implícita, pois não existe nenhuma estrutura que diferencie os parâmetros segundo o tipo de entidade do domínio que ele representa. De fato, os parâmetros que representam as funcionalidades primárias e secundárias são frequentemente associados à restrições e critérios, mas mesmo este tipo de diferenciação não é suficientemente explícita para os nossos propósitos.

A incorporação de uma estrutura de classificação ao ADD permite uma maior utilização do mecanismo de anotações pois as anotações realizadas sobre um parâmetro de uma determinada classe poderão, por exemplo, ser generalizadas para outros parâmetros da mesma classe ou de suas subclasses. Além disso, a associação desta estrutura com o modelo formal de ADD fornecerá base para uma análise comparativa dos contextos de captura e uso da anotação, quando for necessário realizar recuperações mais elaboradas.

O modelo de classificação que propomos toma como base as seis categorias indicadas acima para criar uma hierarquia descritiva dos parâmetros. Esta hierarquia é altamente dependente do contexto pois deve refletir o domínio de aplicação do sistema. Apresentamos na figura 5.2 uma parcela da hierarquia estrutural utilizada num sistema para design de processo em plataformas de petróleo. Não obstante ser apenas uma pequena parte da hierarquia, ela serve para ilustrar o tipo de classificação que propomos e o uso pretendido.

Classe: *Estrutura*
  Subclasses: *Sistemas, Equipamentos, Atributos*
Classe: *Sistemas*
  Instâncias: *Recebimento, Aquecimento, Separação, Compressão de Gás, Resfriamento*
Classe: *Equipamentos*
  Subclasses: *Recebedor, Trocador de Calor, Separador, Compressor, Filtro, Vaso*
Classe: *Trocador de Calor*
  Subclasses: *Casco e Tubo, Elétrico, Forno*
Classe: *Casco e Tubo*
  Instâncias: *Trocador1, Trocador7, Trocador9*
...
Classe: *Atributos*
  Subclasses: *Dimensões, Propriedades*
Classe: *Dimensões*
  Subclasses: *Lineares, Quadráticas, Cúbicas*
Classe: *Lineares*
  Subclasses: *Altura, Largura, Comprimento*
Classe: *Altura*
  Instâncias: *AlturaTroc1, AlturaTroc7, AlturaTroc9, AlturaVaso3*
...

Figura 5.2 - Exemplo de Hierarquia de Classes

Na hierarquia apresentada, as instâncias (Recebimento, Trocador1, ..., AlturaTroc1, ...) são os parâmetros representados no modelo formal de ADD. Anotações feitas sobre a classe *Sistemas* são herdadas para todas as suas instâncias. Da mesma forma anotações feitas sobre a classe *Trocador de Calor* são herdadas para todas as sua subclasses (*Casco e Tubo, Elétrico, Forno*), para as suas instâncias e para as instâncias das suas subclasses. Anotações feitas sobre a instância *AlturaTroc1* podem, de acordo com o contexto, serem exportadas para as outras instâncias irmãs (*AlturaTroc7,AlturaTroc9, AlturaVaso3)* ou para apenas algumas delas (*AlturaTroc7, AlturaTroc9*).

5.3.2 - Representação de Anotações Específicas

A Figura 5.3 apresenta um esquema dos campos que compõem um registro de uma anotação específica. O esquema é dividido em três campos: Contexto, Significado e Comentários. Embora todos estes três campos contenham informações a respeito do contexto no qual a anotação foi feita, usamos o campo Contexto para representar apenas informações contextuais diretamente relacionadas com o modelo formal e com o histórico de ADD. Já o campo Significado procura fornecer informações a respeito do conteúdo da anotação e é subdividido nos subcampos Foco e Objetivo. Por fim, o campo Comentários é onde é colocada a anotação textual livre propriamente dita e é subdividido nos subcampos Texto e Palavras Chave.

Cabe ainda, antes de descrevermos a semântica de cada um dos campos do esquema, ressalvarmos que o processo de captura destes campos envolve a composição de mecanismos de aquisição automática no modelo de ADD e de requisição direta ao usuário. Estes mecanismos serão apresentados na seção 5.4.

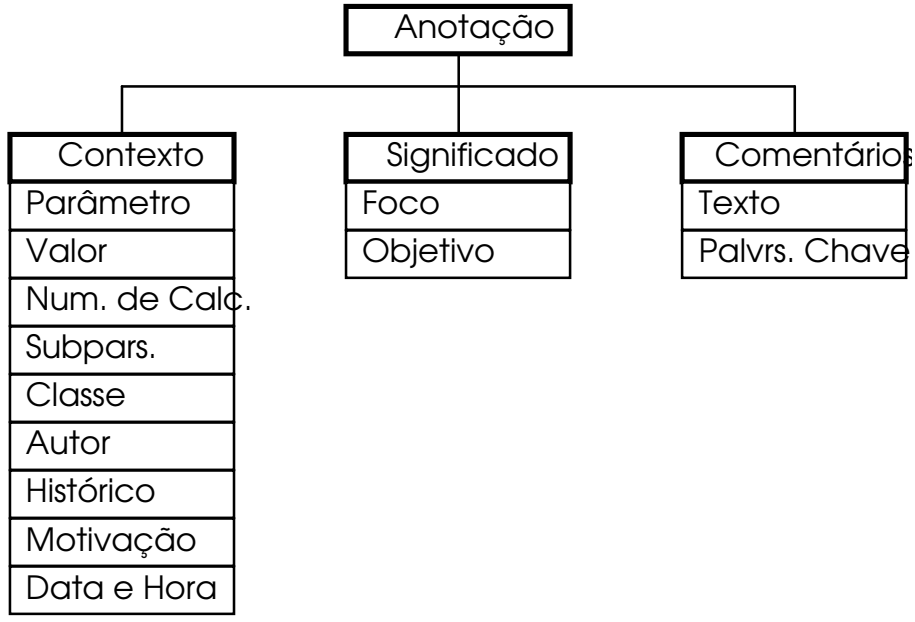

Figura 5.3 - Esquema de Anotação Específica

O campo Contexto é subdividido em vários subcampos. Os três primeiros campos identificam o parâmetro sobre o qual é feita a anotação específica, o seu valor no momento da anotação e o número de vezes que este valor foi modificado até aquele momento. O campo seguinte representa os subparâmetros que estão ativos no momento da anotação e seus valores. O campo Classe identifica a categoria, dentro do domínio, a que o parâmetro pertence. Embora esta informação possa ser inferida a partir do parâmetro, ela nos dá uma boa indicação a respeito do tema da anotação. Todos estes campos são importantes porque mostram a configuração do modelo formal no instante da anotação. Em especial, o número de vezes que o valor foi modificado indica o grau de compromisso que o projetista tem com aquele valor. Se ele foi modificado diversas vezes isto significa que este parâmetro foi bastante avaliado antes de ser assumido e, portanto, presume-se que o grau de compromisso com o valor atual é grande.

Já o campo Autor identifica o responsável pela anotação. Esta informação é essencial quando o sistema é multi-agente, quando existe mais de um responsável pela execução do mesmo sistema ou quando estamos consultando projetos passados. Deste modo, podemos saber se quem fez a anotação foi um especialista renomado ou um estagiário e, portanto, identificar qual o grau de confiança que devemos colocar naquela anotação.

O campo Histórico representa qual foi o procedimento (registro das ações que foram realizadas sobre aquele parâmetro) do projetista antes de decidir-se por fazer uma anotação informal. Este registro proporciona informações a respeito do contexto exato no qual o projetista decidiu fazer a anotação.

O campo seguinte, Motivação, apresenta uma indicação do motivo pelo qual o usuário resolveu fazer a anotação. Este campo demanda um processo heurístico de interpretação e classificação durante a captura da anotação. No caso da Motivação, é necessário interpretar o contexto (modelo e histórico) para poder classificá-lo dentre um conjunto de possíveis motivações da anotação. Como este processo é heurístico, a

informação colhida neste campo é muito mais evidencial do que exata. Por outro lado, a vantagem da existência deste campo é permitir uma recuperação mais flexível da anotação do que a que seria obtida pela comparação de campos históricos.

O último campo do Contexto armazena a data e a hora em que anotação foi realizada. Este campo proporciona uma organização temporal das anotações. Esta organização permite que, no caso de anotações com informações conflitantes, seja verificado qual delas é a mais recente e, presumivelmente, a mais atualizada. Além disso, ela também permite que seja mostrado sequencialmente o conjunto de anotações que foram feitas sobre o parâmetro. Este tipo de informação pode ser importante para acompanharmos a evolução da interpretação daquele parâmetro pelos projetistas.

O propósito do campo Significado é permitir que o anotador forneça maiores informações sobre o conteúdo de sua anotação livre. Estas informações devem ser estruturadas para que o ADD-GHS possa utilizá-las formalmente para analisar a relevância da anotação. O subcampo Foco, portanto, possibilita que ele registre quais serão os focos, dentre um conjunto de propriedades de um parâmetro, que serão enfatizados na anotação. Já o campo Objetivo visa possibilitar que o anotador indique mais especificamente quais são as suas intenções, dentre um conjunto de combinações possíveis, ao fazer a anotação.

Por fim, o campo Comentários é onde o anotador escreve a anotação em linguagem natural. No subcampo Texto é armazenado o texto da anotação e no subcampo Palavras Chave são armazenados padrões morfológicos que ocorreram no texto acima de determinada frequência. Estes padrões serão guardados como palavras chaves que servirão como índices para a recuperação da anotação.

5.3.3 - Representação de Anotações Coletivas

Como pode ser observado na figura 5.4, o esquema de anotações coletivas possui uma estrutura geral semelhante com o das anotações específicas. Tal como nestas últimas, o esquema de anotações coletivas é dividido nos campos Contexto, Significado e Comentários. As principais diferenças do esquema de anotações coletivas para o de anotações específicas são nos subcampos do campo Contexto e no conteúdo dos subcampos do campo Significado.

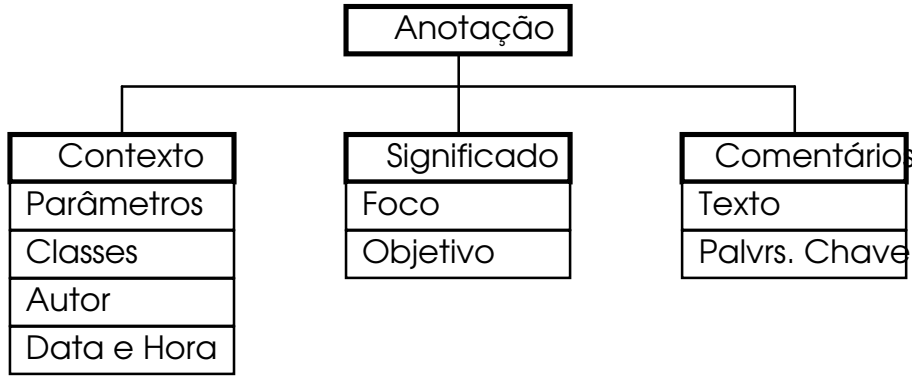

Figura 5.4 - Esquema de Anotações Coletivas

Nas anotações coletivas o campo Contexto é subdividido nos campos Parâmetros e Classes (que indicam, respectivamente, quais os parâmetros e classes a que a anotação

se refere), Autor e Data e Hora. O campo Parâmetros permite definir precisamente quais os parâmetros que são associados à anotação. Já o campo Classes permite que sejam feitas anotações sobre grupos de parâmetros relacionados pela hierarquia de classificação dos parâmetros. Por exemplo, ao fazer uma anotação sobre *Trocadores* (exemplo da seção 5.3.1), ela seria associada a todos os parâmetros que fossem instâncias das suas subclasses.

O campo Significado é dividido em Foco e Objetivo. O Foco expressa os possíveis relacionamentos entre os parâmetros associados à anotação. Estes relacionamentos podem ser causais (no caso de justificativa de alteração da rede paramétrica), abstracionais (no caso de agrupamento de parâmetros para criar uma nova abstração), validativos (quando atestam ou negam a validade do relacionamento causal dos parâmetros), comportamentais (quando justificam um procedimento de design), funcionais (quando relacionam os parâmetros segundo a funcionalidade do artefato ou de suas partes), estruturais (quando relacionam os parâmetros segundo a estrutura do artefato ou de suas partes), organizacionais (quando relacionam os parâmetros segundo aspectos organizacionais da instituição ou equipe de design) e conjunturais (quando os parâmetros são correlacionados por uma conjuntura externa que afeta o design, construção ou uso do artefato).

5.3.4 - Representação de Anotações Extras

As únicas diferenças do esquema de anotações extras (figura 5.5) para o de anotações coletivas é a ausência dos subcampos Parâmetros e Classes do campo Contexto.

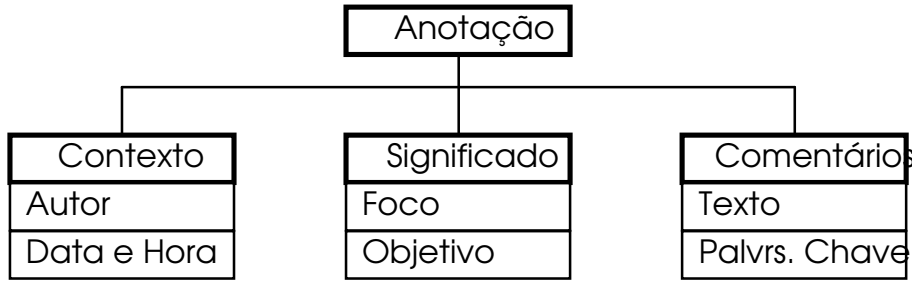

Figura 5.5 - Esquema de Anotações Extras

O subcampo Foco do campo Significado expressa os possíveis tópicos que uma anotação extra pode enfocar. Estes tópicos podem ser causais (no caso de inserção de novo conhecimento sobre o método de design), abstracionais (no caso de se querer apresentar argumentos em níveis mais abstratos do que o do modelo), validativos (quando atestam ou negam a validade do modelo como um todo), funcionais (quando as anotações apresentam uma nova funcionalidade do artefato ou de suas partes), estruturais (quando as anotações fornecem uma nova parte da estrutura do artefato), organizacionais (quando enfocam os aspectos organizacionais da instituição ou equipe de design) e conjunturais (quando apresentam uma situação da conjuntura externa que afeta o design, construção ou uso do artefato).

5.4 - Captura das Anotações

A captura de anotações em ADD-GHS é invocada pelo usuário no momento em que ele julgar adequado. Ela consiste em obter o texto da anotação do próprio usuário e as informações que servirão como índices desta anotação. ADD-GHS promove a indexação das anotações segundo três métodos. O primeiro método consiste em

capturar, a partir do modelo de ADD e do histórico, o contexto no qual foi realizada a anotação. O segundo método é baseado na requisição de um conjunto de informações ao próprio projetista (numa espécie de *template* a ser preenchido). Por sua vez, o terceiro método obtém automaticamente os índices do texto da anotação através da identificação de padrões morfológicos que ocorrem acima de uma determinada frequência (como, por exemplo, no programa WAIS).

É importante ressaltar que as informações requisitadas ao projetista são opcionais (caso o usuário não deseje preenchê-los, ele poderá não fazê-lo) e que propositalmente exigem uma pequena carga de trabalho para serem fornecidas, de modo a não desestimular o anotador a fazer a anotação. No caso particular onde o projetista não fornece nenhuma informação adicional, a anotação será indexada apenas pelos dados de contexto e pelos padrões morfológicos encontrados no texto. Do mesmo modo, esta anotação só poderá ser recuperada através da busca a estes dados ou a estes padrões.

5.4.1 - Captura de Anotações Específicas

Descreveremos, agora, como os campos das anotações específicas podem ser capturados. Os campos Parâmetro, Valor, Número de Cálculos, Subparâmetros, Classe, Autor, Histórico e Data e Hora podem ser obtidos diretamente a partir do modelo formal de ADD e do histórico de design. Já o campo Texto é fornecido livremente pelo anotador e o campo Palavras Chave é obtido automaticamente a partir da análise de frequência de padrões morfológicos neste texto.

Para a captura do campo Motivação é necessário identificar quais foram as circunstâncias em que a anotação foi decidida através de uma interpretação e classificação do histórico. Com o intuito de ilustrar o processo de captura deste campo apresentamos algumas das possíveis relações entre o histórico e a sua motivação:

1. O usuário pode ter tentado vários valores, depois tentou alterar a rede e, finalmente, forçou e fez a anotação. Isto provavelmente significa que ele não discordava em princípio do modelo, mas ao tentar verificou que havia algo diferente a ser representado, tentou fazer a alteração mas não conseguiu e resolveu forçar explicando informalmente as suas razões. Logo, classificamos este tipo de motivação como *falha simples de conhecimento identificada no processo*.
2. O usuário tentou várias vezes e resolveu forçar. Significa provavelmente que ele concordava com o modelo mas verificou que ele não lhe atendia e resolveu forçar logo porque a alteração era por demais complexa. Neste caso, classificamos este tipo de motivação como *falha complexa de conhecimento identificada no processo*.
3. O usuário forçou direto. Havia uma discordância prévia do modelo. Esta situação é classificada como *falha complexa de conhecimento identificada previamente.*
4. O usuário propõe um valor que conflita com o calculado pelo ADD, mas durante reconciliação ele adota o valor de ADD e resolve fazer uma anotação. Isto significa que ele quer reforçar os argumentos a favor do valor de ADD em comparação com o que havia proposto. Nossa classificação é *justificativa para valor reconciliado*.

5. O usuário propôs um valor que não conflita com o calculado por ADD mas, ainda assim, quis fazer a anotação. Ele concorda com o procedimento de cálculo do modelo, mas quis apresentar informações adicionais sobre o seu intento. Nossa classificação é *justificativa para valor proposto.*
6. O usuário altera o modelo e decide fazer uma anotação sobre um dos parâmetros. Neste caso, o usuário quer explicar o porque da alteração do modelo. Logo, nossa classificação é *justificativa para alteração de cálculo.*
7. Em caso de sistemas multi-agente [VG96], alguns parâmetros serão motivo de conflito entre agentes. No caso da ocorrência de um conflito, o usuário pode querer fazer uma anotação para explicar quais fatores o levaram a resolver o conflito. Quando o designer aceita o valor externo (proposto por outro agente), classificamos como *justificativa para valor externo em conflito.* Quando o designer mantém o seu próprio valor, classificamos como *justificativa para valor próprio em conflito.*

Já para a captura do campo Foco, o anotador pode selecionar como tópicos de sua anotação zero, um ou mais ítens de uma lista que depende do tipo do parâmetro sobre o qual se faz a anotação. Se o parâmetro é Primitivo, a lista é composta dos tópicos *parâmetro, valor, classe do domínio e superparâmetros*. Se o parâmetro é Derivado, a lista é composta dos tópicos *parâmetro, valor, classe do domínio, fórmula, subparâmetros e superparâmetros.* Se o parâmetro é Decidido, a lista é composta dos tópicos *parâmetro, valor, classe do domínio, subparâmetros, superparâmetros, alternativas, critérios e restrições.* No caso do anotador selecionar algum dos ítens *subparâmetros, superparâmetros, alternativas, critérios e restrições*, ele poderá ser ainda mais específico selecionando também quais elementos do ítem selecionado a anotação se refere. Por exemplo, se ele houver selecionado *alternativas*, ele poderá indicar quais as alternativas, dentre as existentes, que ele enfocará na anotação. Se o anotador achar que o foco de sua anotação não é nenhum da lista de opções ele pode, ainda, estabelecer ele próprio o foco da anotação através de definição de palavras chaves.

Por fim, a captura do campo Objetivo é composta por duas partes. Na primeira parte o anotador pode especificar se a anotação visa *adicionar informação sobre o parâmetro*, *restringir informação sobre o parâmetro* ou *explicar a informação do modelo sobre o parâmetro* (ou seja, reexplicar o modelo informalmente). A segunda parte só existe quando a motivação da anotação foi um conflito entre agentes ou entre ADD e o agente. Nestes casos, o anotador pode selecionar seus objetivos dentre as seguintes opções: *reforçar valor proposto, refutar outros valores* (se ele manteve ou forçou o seu próprio valor) ou *reforçar valor assumido, refutar outros valores* (se ele assumiu o valor de ADD ou o valor de outro agente). Da mesma forma que no campo foco, o anotador pode definir ele mesmo o objetivo, em forma de palavras chaves, caso nenhum objetivo da lista lhe atenda.

5.4.2 - Captura das Anotações Coletivas

Quando o anotador decide fazer uma anotação coletiva, as informações que podem ser extraídas diretamente do modelo se tornam bastante reduzidas, dado que a anotação pode ser feita a qualquer momento sobre qualquer parte do modelo. Desta forma, o campo Contexto contém apenas quatro subcampos. Os campos Autor e Data e Hora são capturados automaticamente. Já os campos Parâmetros e Classes são especificados

pelo anotador através da seleção em listas contendo, respectivamente, os parâmetros e as classes do modelo. Presume-se que quando o designer seleciona parâmetros ele não selecionará simultaneamente classes e vice-versa, embora não haja impedimento para fazê-lo.

Na captura do campo Foco, o anotador pode selecionar como tópicos de sua anotação zero, um ou mais ítens de uma lista composta pelas seguintes opções:

- *Causalidade*: no caso de justificativa de alteração da rede paramétrica - quando a rede é alterada para inserir um novo método de design ou alterar o método existente e é realizada uma anotação, o relacionamento existente entre os parâmetros inseridos e/ou alterados é de causalidade.
- *Abstração*: no caso de agrupamento de parâmetros para criar uma nova abstração - como a rede paramétrica de ADD é plana (parâmetros representam entidades de diferentes níveis de abstração numa única rede não nivelada), pode-se fazer anotações sobre um grupo de parâmetros para indicar que eles representam um conceito mais abstrato. Por exemplo, poderia-se agrupar todos os atributos relativos a um equipamento numa anotação e indicar que eles podem ser representados pelo equipamento num nível mais abstrato de interpretação.
- *Validação*: quando atestam ou negam a validade do relacionamento causal entre os parâmetros - uma anotação pode ser feita para prover evidências a favor ou contra um determinado relacionamento causal entre parâmetros. Por exemplo, pode-se querer discordar de um método de design sem ter de alterar o modelo paramétrico.
- *Comportamento*: quando justificam um procedimento de design - uma anotação pode ser feita para indicar que foi feito uma série de atribuições de valores a um grupo de parâmetros para se poder definir outros.
- *Funcionalidade*: quando relacionam os parâmetros segundo a funcionalidade do artefato ou de suas partes - pode-se indicar que um grupo de parâmetros determina uma nova função do artefato ou de seus componentes.
- *Estrutura*: quando relacionam os parâmetros segundo a estrutura do artefato ou de suas partes - pode-se indicar numa anotação que um conjunto de parâmetros fazem parte de um componente, subcomponente ou do próprio artefato.
- *Organização*: quando relacionam os parâmetros segundo aspectos organizacionais da instituição ou equipe de design - uma anotação pode indicar que um grupo de parâmetros foi objeto de negociação entre os diferentes agentes e que foram decididos através da obtenção de consenso entre eles.
- *Conjuntura*: quando os parâmetros são correlacionados por uma conjuntura externa que afeta o design, construção ou uso do artefato - pode-se indicar que uma região do modelo composta por um grupo de parâmetros não pode ser utilizada em virtude de um acontecimento contingencial. Por exemplo, caso durante o design, um determinado componente esteja em falta do mercado, uma anotação pode justificar a não utilização deste componente.

Há também a possibilidade do usuário especificar um outro foco para a anotação através do mecanismo de palavras chave, caso ele julgue que nenhuma destas opções o atende.

Já na captura do campo Objetivo o anotador pode especificar se a anotação visa *adicionar informação ao modelo*, *restringir informação do modelo*, *explicar a informação do modelo* ou *explicar negociação* ou, ainda, utilizar o mecanismo de palavras chaves.

5.4.3 - Captura das Anotações Extras

No caso de anotações extras o procedimento de captura é idêntico ao das anotações coletivas. Por sua própria natureza extraordinária, as únicas informações de contexto que podem ser capturadas são o autor da anotação e a data e hora da anotação. De resto, as diferenças para as anotações coletivas são no significado das opções do Foco e na ausência da opção *Comportamento*. As opções do Foco são:

- *Causalidade*: no caso de inserção de novo conhecimento sobre o método de design - uma anotação pode inserir uma nova forma de cálculo do artefato em substituição ou para complementar o modelo existente.
- *Abstração*: no caso de se querer apresentar argumentos em níveis mais abstratos do que o do modelo. Por exemplo, num sistema multi-nivelado pode-se querer indicar relações em cada nível deste sistema.
- *Validação*: quando atestam ou negam a validade do modelo como um todo - uma anotação pode ser feita para prover evidências a favor ou contra o modelo. Por exemplo, pode-se querer dizer que o modelo é adequado porque foi bem sucedido em outras aplicacões semelhantes a atual.
- *Funcionalidade*: quando as anotações apresentam uma nova funcionalidade do artefato ou de suas partes - pode-se indicar que o artefato ou seus componentes possuem outras funcionalidades além daquelas representadas no modelo.
- *Estrutura*: quando as anotações fornecem uma nova parte da estrutura do artefato - pode-se indicar uma mudança tecnológica através da inserção de uma anotação com um novo tipo de componente não previsto no modelo.
- *Organização*: quando enfocam os aspectos organizacionais da instituição ou equipe de design - uma anotação pode indicar qualquer aspecto decisório relacionado com o processo de negociação entre os agentes. Por exemplo, a anotação pode fornecer a relação de autoridade entre os diversos agentes do processo.
- *Conjuntura*: quando apresentam uma situação da conjuntura externa que afeta o design, construção ou uso do artefato - por exemplo, pode-se indicar que o modelo foi invalidado porque o local de construção do artefato apresenta características não previstas no modelo.

Aqui também há a possibilidade do usuário especificar um outro foco para a anotação através do mecanismo de palavras chave, caso ele julgue que nenhuma destas opções o atende.

5.5 - Recuperação de Anotações

Como já mencionamos, anotações podem ser recuperadas para satisfazer necessidades de explicação do usuário e para suportá-lo em suas decisões. A recuperação de anotações em ADD-GHS consiste em permitir ao usuário resgatar as informações desejadas em qualquer instante. Deste modo, além dele poder invocar o mecanismo de recuperação das anotações durante o design e em explicações, ele pode fazê-lo antes

de criar uma anotação, para verificar se já não há anotações contendo o que pretende declarar.

Sabendo como as anotações são representadas e capturadas, podemos apresentar o mecanismo de recuperação contextual destas anotações. Este mecanismo é composto por:

- uma base de dados contendo as anotações,
- um mecanismo de consulta a base de dados,
- uma base de conhecimento composta pela rede paramétrica e pela hierarquia de classificação dos parâmetros,
- um mecanismo de inferência que navega pela base de conhecimento e recupera as anotações que são relevantes para o uso pretendido.

A base de dados é organizada através do conjunto de índices obtidos durante a captura da anotação. O mecanismo de consulta permite que as informações da base de dados sejam acessadas através dos procedimentos comums de busca em banco de dados. A base de conhecimento contém referências às anotações - parâmetros e classes têm conhecimento sobre quais anotações os referem. Deste modo, o mecanismo de inferência pode percorrer a base de conhecimento e recuperar as anotações.

Portanto, o modo de recuperação de anotações consiste de uma composição entre uma consulta direta a base de dados e um processo de inferência na base de conhecimento. Quando é necessário consultar a base de anotações, captura-se informações fornecidas pelo contexto e pelo próprio usuário. Estas informações são utilizadas como índices para a consulta à base de anotações e como dados para o mecanismo de inferência. Descreveremos a seguir quais são os procedimentos utilizados para inferência. Fundamentalmente, o mecanismo de inferência consiste de três procedimentos:

- *Identificação de Relevância*: procedimento que compara, através de uma função avaliadora, o contexto de uso e o contexto de captura das anotações, selecionando aquelas que são relevantes para apresentação.
- *Generalização*: procedimento que consiste em estender o uso de uma anotação feita sobre um parâmetro da rede para outro parâmetro.
- *Especialização*: procedimento que consiste em estender o uso de uma anotação feita sobre uma classe ou conjunto de parâmetros para as suas subclasses ou suas instâncias.

Para identificar a relevância da anotação é necessário obter informações do contexto no qual o mecanismo de anotações foi invocado. Por exemplo, no caso de explicação, deve-se identificar qual a pergunta que vai ser respondida, qual o parâmetro que está sendo questionado, qual a configuração da rede naquele momento, quantas tentativas de atribuição de valores já foram feitas sobre aquele parâmetro, dentre outras. A seleção das anotações relevantes é realizada a partir da comparação destas informações com as informações armazenadas nas anotações.

Tal como em design baseado em analogia [QG96], os processos de generalização e especialização das anotações podem ocorrer a nível de conhecimento superficial e a nível de conhecimento profundo.

Quando ocorrem a nível de conhecimento superficial, a base do processo é o relacionamento categórico entre os parâmetros, ou seja, leva-se em conta apenas as classes dos parâmetros e suas relações na estrutura hierárquica do domínio. Em virtude

de sua própria natureza, este tipo de conhecimento é altamente dependente do domínio de design.

Deste modo, uma anotação específica pode ser generalizada para parâmetros que pertencem a mesma classe da dele. Um exemplo deste tipo de uso da anotação seria a requisição de anotações sobre a altura de um tipo de equipamento em situações onde ela foi superior a um dado valor. Neste caso, as anotações podem ser recuperadas de qualquer design ou etapa de design onde aquele tipo de equipamento foi utilizado.

Da mesma maneira, uma anotação coletiva sobre uma classe pode ser especializada para todos os parâmetros que são instâncias daquela classe. Por exemplo, uma anotação feita sobre um tipo de equipamento, pode ser utilizada para a explicação e suporte ao design de qualquer instância daquele equipamento. Da mesma forma, uma anotação feita sobre um tipo de equipamento pode ser utilizada por qualquer instância de subtipos daquele equipamento.

Por sua vez, quando os processos de generalização e especialização se dão a nível de conhecimento profundo, eles se tornam mais elaborados e envolvem uma análise do processo de tomada de decisão dos parâmetros, das suas relações de causalidade e do seu impacto sobre o comportamento e funcionalidade do artefato. Este tipo de conhecimento é mais relacionado com a tarefa de design e, portanto, menos dependente do domínio.

Anotações podem ser recuperadas de parâmetros submetidos a processos similares de tomada de decisão. Para identificar processos similares de tomada de decisão temos de investigar quais os parâmetros que influenciam a decisão, quais são seus valores, quais atuam como critérios e quais atuam como restrições.

Uma forma deste tipo de generalização ocorre quando descobre-se dois parâmetros que são decididos a partir da atuação do mesmo critério ou restrição sobre seus subparâmetros. Neste caso, uma anotação que enfoque este critério ou restrição, feita sobre a decisão de um dos dois, pode ser extrapolada para o outro, se o contexto de recuperação assim o determinar. Por exemplo, se durante a escolha de dois equipamentos de tipos distintos para um sistema, a mesma restrição sobre a altura deles é aplicada, uma anotação que enfoque esta restrição feita sobre a decisão de um pode ser aplicada para o outro.

Outra possibilidade é o uso de uma anotação feita em um parâmetro para explicar ou suportar o design de um dos parâmetros da cadeia de seus superparâmetros. Neste caso, uma anotação que tenha como foco o superparâmetro em questão pode contribuir para o entendimento ou tomada de decisão deste superparâmetro. Por exemplo, uma anotação pode informar que a largura do equipamento foi definida de modo a minimizar a área de circulação da sala onde o equipamento será colocado.

5.5.1 - Recuperação de Anotações Específicas

Durante o processo de explicação das decisões de design, quando um parâmetro que estiver sendo explicado tiver associado uma anotação, é conveniente que ela possa ser apresentada ao usuário. No caso do valor atual do parâmetro ser igual ao valor com que foi feita a anotação, ela deve ser apresentada juntamente com a explicação. Em caso contrário, deve-se sinalizar para o usuário que existem anotações relacionadas com o parâmetro em questão que podem contribuir para o seu entendimento e permitir que ele as consulte. Da mesma forma, deve-se sinalizar também quando existirem anotações associadas a sua cadeia de subparâmetros que possam ser relevantes.

Durante o próprio processo de decisão, o designer pode querer obter mais informações que suportem a sua decisão. Neste caso, ele pode recuperar anotações de acordo com o seu interesse. Para isto ADD-GHS permite que ele forneça, opcionalmente, informações sobre o conteúdo das anotações que ele deseja recuperar. Com estes dados, ADD-GHS pode usar sua estrutura de anotações e o contexto de design para recuperar aquelas que julgar relevantes.

As informações adicionais para recuperação são fornecidas pelo usuário através do preenchimento dos campos Valor, Autor, Classe, Data e Hora, Motivação, Foco e Objetivo de um formulário. Nos campos Valor e Data e Hora, o usuário pode utilizar expressões regulares formadas a partir dos operadores relacionais $<$, $>$, $=$, $\neq$, $\leq$, $\geq$ e dos operadores lógicos E e OU para definir intervalos de valores ou períodos de tempo válidos para a anotação que se deseja recuperar. No campo Autor, o usuário usa os operadores relacionais $=$ e $\neq$ e os operadores lógicos E e OU para selecionar as anotações feitas por determinado autor ou para eliminar aquelas feita por um outro, por exemplo. No campo Classe, o usuário pode selecionar quais as classes do domínio que ele quer ver anotações. Nos campos Motivação, Foco e Objetivo ele seleciona, dentre o conjunto dos valores possíveis de cada campo, os valores que lhe interessam. Caso queira, o usuário pode especificar ainda, nos campos Foco e Objetivo, uma cadeia de caracteres a ser buscada dentre as palavras chaves estabelecidas pelos anotadores. Este mesmo procedimento pode ser utilizado no campo Comentários para buscar anotações segundo os índices extraídos do texto.

No caso do designer não fornecer nenhuma informação adicional e solicitar a recuperação de anotações, ADD-GHS utiliza apenas o contexto de design para recuperar as anotações. O procedimento adotado consiste em recuperar, em ordem de prioridade:

1. as anotações específicas do parâmetro submetidas à um contexto análogo,
2. as anotações específicas do parâmetro em qualquer contexto,
3. as anotações feitas sobre parâmetros da mesma classe do da decisão submetidos à um contexto análogo,
4. as anotações feitas sobre parâmetros da mesma classe do da decisão em qualquer contexto,
5. as anotações feitas sobre parâmetros distintos submetidos à um contexto análogo e
6. as anotações feitas sobre a cadeia de subparâmetros.

Se houver mais de uma anotação de cada tipo, ADD-GHS mostra primeiramente as anotações feitas pelo autor da requisição e depois as de outros autores. Além disso, ele as ordena de acordo com a semelhança entre o contexto de design e o contexto no qual foi feita a anotação. Para tanto, ele usa uma função avaliadora que faz uma comparação entre os valores do parâmetro, dos subparâmetros, histórico e número de cálculos da situação atual com a da situação da anotação. O valor desta função serve para ordenar as anotações segundo sua relevância. ADD-GHS apresenta, então, uma lista ordenada das anotações para que o usuário selecione quais deseja ver.

No caso do designer fornecer informações adicionais sobre as anotações que deseja recuperar, ADD-GHS adota um procedimento similar ao do caso anterior. Uma diferença é que caso o usuário tenha selecionado uma classe diferente da do parâmetro no campo Classe, as anotações destas classes também são inseridas na lista. Além

disso, são eliminadas da lista as anotações que não atendam aos requisitos do usuário. Por fim, a função avaliadora é modificada para também levar em conta o grau de satisfação de requisitos de cada anotação.

5.5.2 - Recuperação de Anotações Coletivas

O processo de recuperação de anotações coletivas é semelhante ao de anotações específicas. Durante a explicação de um parâmetro, deve-se sinalizar para o usuário que existem anotações coletivas relacionadas com o parâmetro em questão que podem contribuir para o seu entendimento e permitir que ele as consulte.

A qualquer momento, também, o usuário pode querer obter informações sobre as anotações coletivas. Neste caso, ele pode recuperar anotações de acordo com o seu interesse. O processo de recuperação das anotações coletivas consiste no preenchimento opcional dos campos do Contexto, Significado e Comentários. Caso ele não forneça alguma informação não é possível recuperar as anotações coletivas.

5.5.3 - Recuperação de Anotações Extras

O processo de recuperação de anotações extras é mais restrito que o de anotações coletivas. Como ela não possui nenhuma associação com os parâmetros e classes da base de conhecimento, só é possível recuperar anotações extras através de uma consulta direta a base de dados. Deste modo, quando o usuário invoca o mecanismo de recuperação ele deve definir as características das anotações extras que ele deseja recuperar. Aqui, também, caso ele não forneça alguma informação não é possível recuperar as anotações extras.

6 - Conclusão e Trabalhos Futuros

Neste trabalho foi apresentada uma classificação das diferentes abordagens utilizadas para captura e recuperação do rationale de design. Adicionalmente, foi definida uma estrutura de comparação entre estas diferentes abordagens. A partir da análise comparativa das abordagens, foi constatado que a abordagem baseada em modelos é capaz de produzir design rationale mais rico e expressivo que as demais. Contudo, ela sozinha não é suficiente para atender todas as necessidades do usuário de design rationale. Além disso, pode-se verificar uma certa complementaridade entre as abordagens histórica, argumentativa e baseada em modelos. Concluiu-se, portanto, que a adoção de uma representação capaz de combinar estes três tipos de abordagem, tendo como base a abordagem por modelos, produziria um ganho em termos de poder expressivo do rationale.

Em decorrência desta conclusão, foi proposta uma extensão do sistema ADD [Gar92] (*Active Design Document*) de modo a incorporar a este sistema um mecanismo que permite capturar *design rationale* através da combinação de um modelo formal do processo de design, com o histórico do processo e com argumentos informais obtidos através de anotações estruturadas. As grandes vantagens desta extensão, chamada de ADD-GHS, sobre o modelo de ADD é permitir a documentação do conhecimento informal e oportunístico e proporcionar a formação de uma rede ortogonal de informações que possibilita capturar relações não causais. Por outro lado, a adoção de anotações estruturadas permite que o uso desta informação seja muito mais ampliado que em modelos puramente argumentativos. Além disso, a combinação com o modelo formal de ADD diminui substancialmente a carga de trabalho que o designer deve realizar para documentar seu *rationale*.

Identificamos, ainda, que ADD-GHS cumpre papéis linguísticos que permitem expandir a linguagem de expressão de ADD, ora atuando como extensão expressiva a nível da própria linguagem, ora atuando a nível de meta linguagem.

Outra característica da extensão proposta é que ela permite recuperar anotações em contextos similares àquele no qual ela foi capturada. Para tanto, foi desenvolvida uma estrutura de classificação para os parâmetros do modelo formal de ADD, segundo o domínio de design, que indica como esta classificação deve ser implementada para os diferentes domínios de aplicação. Esta classificação, juntamente com o modelo formal, o histórico de ADD e as informações fornecidas pelo usuário, tornam o mecanismo de anotações estruturadas capaz de:

- identificar a relevância da anotação - para identificar dentre as anotações, quais as que são relevantes naquele contexto para o enfoque desejado;
- manipular as relações categóricas hierárquicas entre os parâmetros no domínio - para habilitar os processos de generalização e especialização superficial das anotações;
- manipular as relações causais representadas no modelo formal de ADD - para habilitar os processos de generalização e especialização profunda das anotações;

Embora não tenhamos garantias quanto a adequação total da proposta de ADD-GHS, no caso de sistemas mono-agente, o modelo de ADD nos dá evidências positivas concretas a respeito da viabilidade da representação utilizada. O mesmo não pode ser dito com relação a sistemas multi-agente. Neste caso, como empregamos uma representação bastante genérica, somente uma análise de casos gerados pelo uso de ADD-GHS poderá determinar o grau de adequação da proposta e quais as alterações que devem ser feitas.

Uma versão preliminar de ADD-GHS já foi implementada no sistema ADDProc e se encontra operacional. ADDProc é um sistema multi-agente [GV96] que objetiva apoiar e documentar o design de processo em plataformas de petróleo. Embora a versão implementada de ADD-GHS contenha apenas parte do que foi proposto aqui — as representações das anotações são um pouco simplificadas e específicas, e o mecanismo de inferência não foi implementado — ela serve como uma demonstrativo da viabilidade da proposta. Além disso, a análise do uso desta versão nos fornecerá indicadores a respeito da expressividade da parte implementada de ADD-GHS.

Nosso próximo passo nesta pesquisa será implementar a versão completa de ADD-GHS. Em seguida, realizaremos um conjunto de testes que nos permitirão avaliar a adequação da nossa proposta. Estes testes objetivam analisar:

1. a interação de ADD-GHS com os usuários - verifica se os usuários conseguem captar a funcionalidade de ADD-GHS e entender como usá-lo na captura e recuperação da anotação.
2. a expressividade da representação utilizada - a taxa de uso, durante a captura da anotação, da opção de palavras chave nos campos foco e objetivo em comparação aos outros campos nos dá indicação da adequação da representação empregada.
3. a adequação do mecanismo de recuperação - verifica se ADD-GHS recupera a informação desejada pelo usuário e existente na base de anotações.

4. a taxa de utilização de ADD-GHS - verifica o quanto o usuário faz uso de ADD-GHS em comparação com o uso do modelo paramétrico de ADD.

No futuro, o mecanismo de anotações estruturadas poderá, ainda, ser utilizado como fonte de realimentação do modelo de ADD. Com a crescente utilização deste mecanismo, muito conhecimento vai ser incorporado ao ADD sob a forma de anotações. De tempos em tempos, será conveniente fazer uma análise destas anotações para que se possa identificar o que precisa e pode ser atualizado no modelo paramétrico de ADD. Nesta análise pode-se usar, inclusive, o mecanismo de reconhecimento de padrões morfológicos para se identificar padrões culturais de design. Portanto, o mecanismo de anotações estruturadas funcionará também como fonte de aquisição de conhecimento para o ADD.

Nessa mesma linha, entendemos que a super-utilização deste mecanismo pode implicar numa baixa adequação do modelo paramétrico de ADD. Caso esta super-utilização seja constatada, deve-se analisar a possibilidade de se refazer por completo o modelo paramétrico subjacente de ADD.

7 - Referências

[Alb93] - Alberts, L.K. *"YMIR: an Ontology for Engineering Design"*. **PhD Thesis**, University of Twente, May, 1993.

[BSZ90] - Baudin, C., Sivard, C. & Zweben, M. *"Recovering Rationale for Design Changes: A Knowledge-based Approach"*. **Proceedings of the IEEE International Conference on Systems, Man and Cybernetics**, pp. 745-749, Los Angeles, 1990.

[CB89] - Conklin, J. & Begenman, M.L. *"gIBIS: A Tool for All Reasons"*. **Journal of the American Society for Information Science**, May, pp. 200-213, 1989.

[DD95] - Dorst, K. & Dijkhuis, J. *"Comparing paradigms for describing design activity"*. **Design Studies**, Vol. 16, No. 2, April, 1995.

[FKS95] - Finger, S., Konda, S. & Subrahmanian, E. *"Concurrent design happens at the interfaces"*. **AIEDAM**, 9, pp. 89-99, 1995.

[FLMM91] - Fischer, G., Lemke, A., McCall, R. & Morch, A.I. *"Making Argumentation Serve Design"*. In Moran, T. & Carrol, J. (eds.), **Design Rationale: Concepts, Techniques and Use**, Lawrence Erlbaum, 1996.

[Gar92] - Garcia, A.C.B. *"Active Design Documents: A New Approach for Supporting Documentation in Preliminary Routine Design"*. **PhD Thesis**, Department of Civil Engineering, Stanford University, August, 1992.

[Ger90] - Gero, J.S. *"Design Prototypes: A Knowledge Representation Schema for Design"*. **AI Magazine**, pp. 26-36, Winter, 1990.

[GGF94] - Ganeshan, R., Garret, J. & Finger, S. *"A framework for representing design intent"*. **Design Studies**, 15, 1, January, 1994.

[GR96] - Gruber, T.R & Russel, D.M. *"Generative Design Rationale: Beyond the Record and Replay Paradigm"*. In Moran, T. & Carrol, J. (eds.), **Design Rationale: Concepts, Techniques and Use**, Lawrence Erlbaum, 1996.

[Gru91] - Gruber, T. *"Learning Why by Being Told What - Interactive Acquisitions of Justifications"*. **IEEE Expert**, pp. 65-75, August, 1991.

[GV96] - Garcia, A.C.B. & Vivacqua, A.S. *"The Use of Active Design Documents to Assist Conflict Mitigation in Concurrent Engineering"*. To appear in the **Third Conference on Concurrent Engineering Research and Applications (CE96/ISPE)**, Toronto, Ontario, Canada, 1996.

[IFVC93] - Iwasaki, Y., Fikes, R., Vescovi, M. & Chandrasekaran, B. *"How Thimgs Are Intended to Work: Capturing Functional Knowledge in Device Design"*. In **Proceedings of XIII IJCAI** - Chamberry, France, Vol. 2, pp. 1516-1522, 1993.

[KM94] - Kirsch, D. & Maglio, P. *"On Distinguishing Epistemic from Pragmatic Action"*. **Cognitive Science**, 18, pp. 513-549, 1994.

[KR70] - Kunz, W. & Rittel, H.W.J. *"Issues as Elements of Information Systems"*. **Working Paper 131**, Center for Planning and Development Research, University of California, Berkeley, CA, 1970.

[Lee91] - Lee, J. - *"Extending the Potts and Bruns Model for Recording Design Rationale"*. **Proceedings of the 13th International Conference on Software Engineering**, Austin, Texas, 1991.

[LWL89] - Lakin, F., Wambaugh, J. Leifer, L., Cannon, D. & Sivard, C. *"The eletronic design notebook: performing medium and processing medium"*. **Visual Computer: International Journal of Computer Graphics**, 5, 4, pp.214-226.

[McC91] - McCall, R.J. *"PHI: A Conceptual Foundation for Design Hypermidia"*, **Design Studies**, 12, 1, pp. 30-41, 1991.

[MM94] - McKerlie, D. & MacLean, A. *"Reasoning with Design Rationale: practical experience with Design Space Analysis"*. **Design Studies**, 15, 2, April, 1994.

[MYBM91] - MacLean, A., Young, R., Bellotti, V. & Moran, T. *"Questions, Options and Criteria: Elements of a Design Rationale for User Interfaces"*. **Humam-Computer Interaction**, 6, pp.201-250, 1991.

[PSL95] - Peña-Mora, F., Sriram, R.D. & Logcher, R. *"Conflict mitigation system for collaborative engineering"*. **AIEDAM**, 9, pp. 101-124, 1995.

[QG96] - Quian, L. & Gero, J.S. *"Function-Structure-Behavior Paths and Their Role in Analogy-Based Design"*. **Technical Report** qia-ger-fsb-aiedam.ps.gz, Key Centre of Design Computing, Department of Architectural and Design Science, University of Sidney.

[RG96] - Rosenman, M. & Gero, J.S. *"The Role of Functional Reasoning in Design"*. **Technical Report** ros-ger-role-aiedam.ps.gz, Key Centre of Design Computing, Department of Architectural and Design Science, University of Sidney.

[Shu96] - Shum, S.B. *"Design Argumentation as Design Rationale"*. In Kent, A. & Williams, J.G. (eds.), **The Encyclopedia of Computer Science and Technology**, New York, Marcel Dekker, Inc., 1996.

[Sou92] - de Souza, C.S. *"Augmenting Informativeness and Learnability of Items in Large Computer Networks"*. **Technical Report** No. CSLI-92-164, Center for the Study of Language and Information, Stanford University, May, 1992.

[Sou93] - de Souza, C.S. *"Communicative Indexing in Network Resource Discovery Systems"*. **Technical Report** No. CSLI-93-182, Center for the Study of Language and Information, Stanford University, December, 1993.

[Vis95] - Visser, W. *"Use of episodic knowledge and information in design problem solving"*. **Design Studies**, 16, 2, April, 1995.