

# PUC

---

Série: Monografias em Ciência da Computação, No.10/91

REGISTRO DE DECISÕES E JUSTIFICATIVAS DE DESENHO EM SOFTWARE  
PROJETADO COM A METODOLOGIA JSD

Antônio F. Prado  
Carlos J. P. Lucena  
Júlio Cesar S. P. Leite

Departamento de Informática

---

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO  
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP-22453  
RIO DE JANEIRO - BRASIL

PUC RIO - DEPARTAMENTO DE INFORMÁTICA

Série: Monografias em Ciência da Computação, No. 10/91

Editor: Carlos J. P. Lucena

Maio, 1991

REGISTRO DE DECISÕES E JUSTIFICATIVAS DE DESENHO EM SOFTWARE  
PROJETADO COM A METODOLOGIA JSD \*

Antônio F. Prado  
Carlos J. P. Lucena  
Júlio Cesar S. P. Leite

\* Trabalho patrocinado pela Secretaria de Ciência e Tecnologia da  
Presidência da República.

**In charge of publications:**

Rosane Teles Lins Castilho  
Assessoria de Biblioteca, Documentação e Informação  
PUC Rio - Departamento de Informática  
Rua Marquês de São Vicente, 225 - Gávea  
22453 - Rio de Janeiro, RJ  
Brasil

Tel.:(021)529-9386

Telex:31078

Fax:(021)511-5645

E-mail:rosane@inf.puc-rio.br

**Abstract :**

Design decisions are present everywhere in the software process, but most of them is lost as the final artifact is delivered. This article describes an integration of Pott's model of design decisions to a process oriented description of Jackson System Development (JSD). The integration occurs in the context of JSD/PUC, a case tool for JSD.

**Keywords :**

Design history, JSD, Software Process, Design Decisions, Design Rational, CASE.

**Palavras-chave :**

História do Desenho, JSD, Processo de Software, Decisões de Desenho, Justificativa de Desenho, CASE.

# Registro de Decisões e Justificativas de Desenho em Softwares Projetados com a Metodologia JSD

Antonio Francisco do Prado  
Carlos J.P. Lucena  
Julio Cesar S. P. Leite

Departamento de Informática  
Pontificia Universidade Católica do Rio de Janeiro  
R. Marques de S. Vicente 225 Rio de Janeiro 22453  
Brasil  
Maio 1991

## Resumo

Muito esforço vem sendo realizado nos últimos anos, pela comunidade de engenharia de software, para compreensão do processo de desenvolvimento de software. Ambientes de suporte a metodologias vêm permitindo que o especialista pratique o desenho de software de forma análoga ao CAD. Quando se usa o suporte automatizado para o desenho, especialmente no caso do software, descobre-se naturalmente a importância da utilização de mecanismos que registrem as decisões de desenho adotadas como ponto de partida para o desenvolvimento de técnicas de reuso e manutenção de artefatos.

O presente trabalho mostra como o método genérico de Potts, para registro das deliberações de desenho, foi instanciado para um programa de processo no estilo de Osterweil que sistematiza a aplicação dos métodos da metodologia JSD. O programa de processo está incorporado no ambiente JSD/PUC. A estruturação da metodologia JSD em um programa de processo viabiliza o registro das decisões de projeto que ficam organizadas para permitir futuros trabalhos sobre reutilização e manutenção.

## 1 Introdução

Modelos prescritivos do processo de desenho de software, seguindo a noção genérica de desenho proposta por Maher [Maher 90]), podem ser obtidos organizando-se os métodos de algumas metodologias de desenvolvimento de software que:

- a) subsidiam a formulação do problema,
- b) subsidiam representação, e
- c) usam avaliação.

Observe-se que estamos traduzindo o termo "design" (já consagrado por diversas teorias [Bruns 86]) por desenho e não por projeto ou projeto lógico como foi feito, em boa parte da literatura existente em português. Desenho, mais do que uma fase do processo de desenvolvimento de software, é uma abordagem para a solução de problemas [Simon 69] que produz uma representação da solução a partir da qual é possível derivar uma implementação. A metodologia JSD [Jackson 83] permite o desenho de software assistido por computador, em particular, possibilitando que a estratégia de síntese por decomposição possa ser levada a cabo tanto a partir das formas ou objetos (Entidades/Ações) quanto das funções.

Modelos de desenho prescritivos, como os que podem ser construídos a partir dos métodos da metodologia JSD, podem ser um auxílio importante para o processo de desenvolvimento de software porque podem dar assistência ao desenvolvedor por explicitar a progressão do desenho e fornecer notações padrão. Aos métodos podem estar associadas heurísticas expressas na forma de procedimentos que fornecem orientações para auxílio à

tomada de decisões durante o desenvolvimento. As decisões baseiam-se em raciocínios que podem ser justificados. Conhecer esses raciocínios e suas justificativas (o "design rationale") pode ser de grande valia, pois isto permite [Lee 91] melhor compreensão: das questões envolvidas no desenho, do espaço de desenho (espaço de soluções do problema) e dos princípios subjacentes da interação homem máquina. A documentação dos raciocínios e suas justificativas (rationale) pode dar suporte à manutenção e re-desenho do artefato, ao reuso do conhecimento sobre desenho e à reflexão crítica durante o processo de desenho [Fischer 91].

Neste artigo propomos uma instanciação do modelo geral de registro de decisões de desenho de Potts [Potts 88] e [Potts 89] para o processo de desenho de software que se obtém quando se formula um programa de processo [Osterweil 87] que encadeia os métodos previstos na metodologia JSD [Jackson 83]. O programa do processo mencionado é o modelo prescritivo de desenho utilizado no ambiente JSD/PUC [Lucena 91]. O ambiente JSD/PUC é um ambiente acadêmico experimental construído sobre Motif/Unix [OSF/Motif 90] para estações SUN no qual são desenvolvidos experimentos sobre a natureza do processo de desenvolvimento de software, em geral, e sobre o processo de desenho, em particular. A adaptação do ambiente para permitir o registro de decisões e justificativas de desenho viabilizará alguns trabalhos em andamento sobre técnicas de reutilização e manutenção de desenhos. A solução encontrada vincula-se à associação do conceito de programa de processo à metodologia JSD.

## 2 Resumo do Método JSD

JSD [Jackson 83] é uma metodologia bastante difundida [Rodriguez 90], [Ambrosio 89] para especificar sistemas de software. Ela distingue duas importantes tarefas:

- O desenvolvimento da especificação do sistema, e
- O desenvolvimento da implementação.

Por sua vez a tarefa de desenvolvimento da especificação é subdividida nas seguintes etapas:

- Especificação do modelo da realidade, e
- Especificação das funções do sistema.

A especificação do modelo da realidade compreende :

- Descrição abstrata da realidade, e
- Descrição de um modelo concreto.

A metodologia JSD compreende 5 grandes fases:

- Entidade/Ação;
- Estrutura/Entidade;
- Modelo Inicial;
- Função; ~
- Cronologia; e
- Implementação.

O procedimento de desenvolvimento JSD é mostrado na figura 1, no diagrama árvore de estrutura do próprio JSD, onde nas folhas encontram-se as 5 fases (com seus métodos associados) da metodologia.

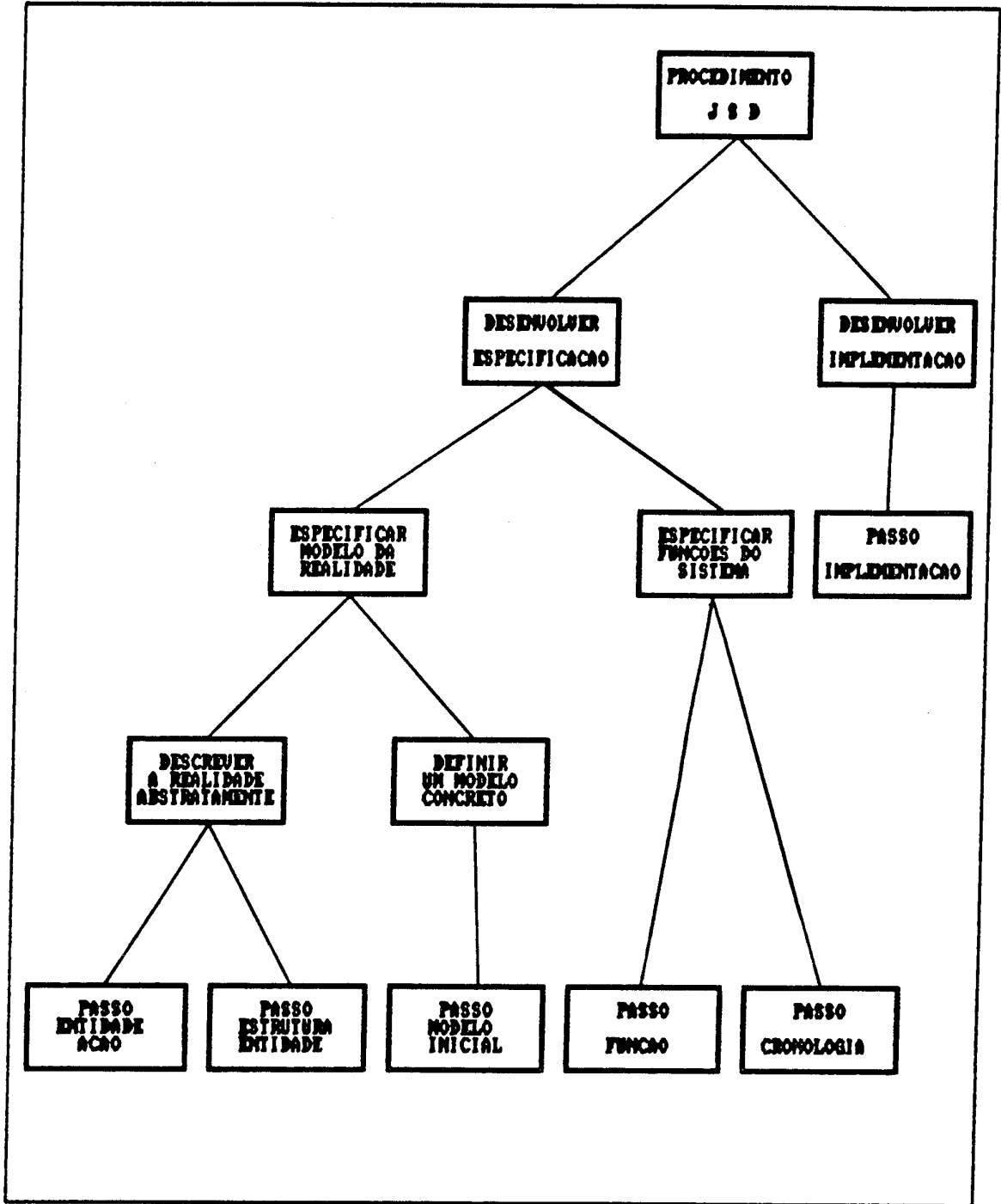


Figura 1

Resumidamente tem-se que na fase:

- Entidade/Ação: o desenvolvedor define a área do mundo real de interesse levantando as entidades e ações com as quais o sistema será concebido;
- Estrutura/Entidade: as ações sofridas ou executadas por cada entidade são arranjadas em sua ordem temporal. Esta execução ordenada no tempo é representada por um diagrama, o diagrama estrutura/entidade;
- Modelo Inicial: a descrição da realidade, em termos de entidades e ações é concretizado num modelo do processo e conexões entre a realidade e o mundo real. Os detalhes de cada processo são mostrados num texto estruturado, cujo esqueleto básico é a do diagrama estrutura da entidade;
- Função: adicionam-se os processos de funções que se conectam, de diferentes formas, com os processos do modelo ou mesmo entre si, com a finalidade de estabelecer a especificação funcional do sistema. Dentre outros propósitos da fase função, estão o de prover entradas e saídas do sistema, tratar erros e melhor elaborar o processo do modelo nível inicial que se comunica com o processo do modelo do mundo real.
- Cronologia: o desenvolvedor considera os aspectos do processo de escalonamento que podem afetar as saídas funcionais do sistema e as restrições de sincronismo dos processos que precisam ser respeitadas na implementação. Estas restrições podem mudar a especificação ou conduzir a uma especificação mais exata das conexões entre processos; e
- Implementação: o desenvolvedor considera qual "hardware" e "software" serão ou deveriam ser fornecidos para execução do sistema, e aplica técnicas de transformação e escalonamento, juntas com técnicas de definição de banco de dados.

### 3 Arquitetura do ambiente JSD-PUC

Para que se conheça o ambiente no qual se pretende incorporar procedimentos para fazer registro das decisões e justificativas do desenho, será dado inicialmente uma noção geral da sua arquitetura e possibilidades. O protótipo desenvolvido na PUC permite o desenvolvimento interativo de sistemas segundo o processo JSD. Este sistema tem vários recursos, onde é importante destacar as facilidades para elaboração de diagramas do método.

Em qualquer situação o desenvolvedor conta com um menu, mostrado nos topos das telas (ver figura 2), que permite seu retorno ao passo ou sub-passo anterior, cancelar ou confirmar as operações efetuadas na situação atual, exibida no canto superior direito e atualizada a cada mudança de situação, bem como encerrar o desenvolvimento no nível em que se encontra retornando ao menu inicial do ambiente.

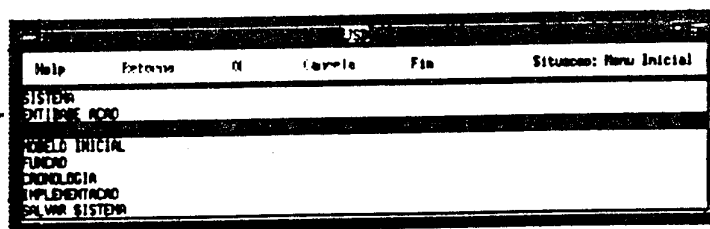


Figura 2



Na fase Modelo Inicial o desenvolvedor pode incorporar os processos do mundo real, correspondentes às entidades, interagindo com o ambiente com telas como a da figura 3, para escolher a entidade que deseja modelar, o tipo de conexão entre processos dos modelos e a posição relativa destes processos no diagrama.

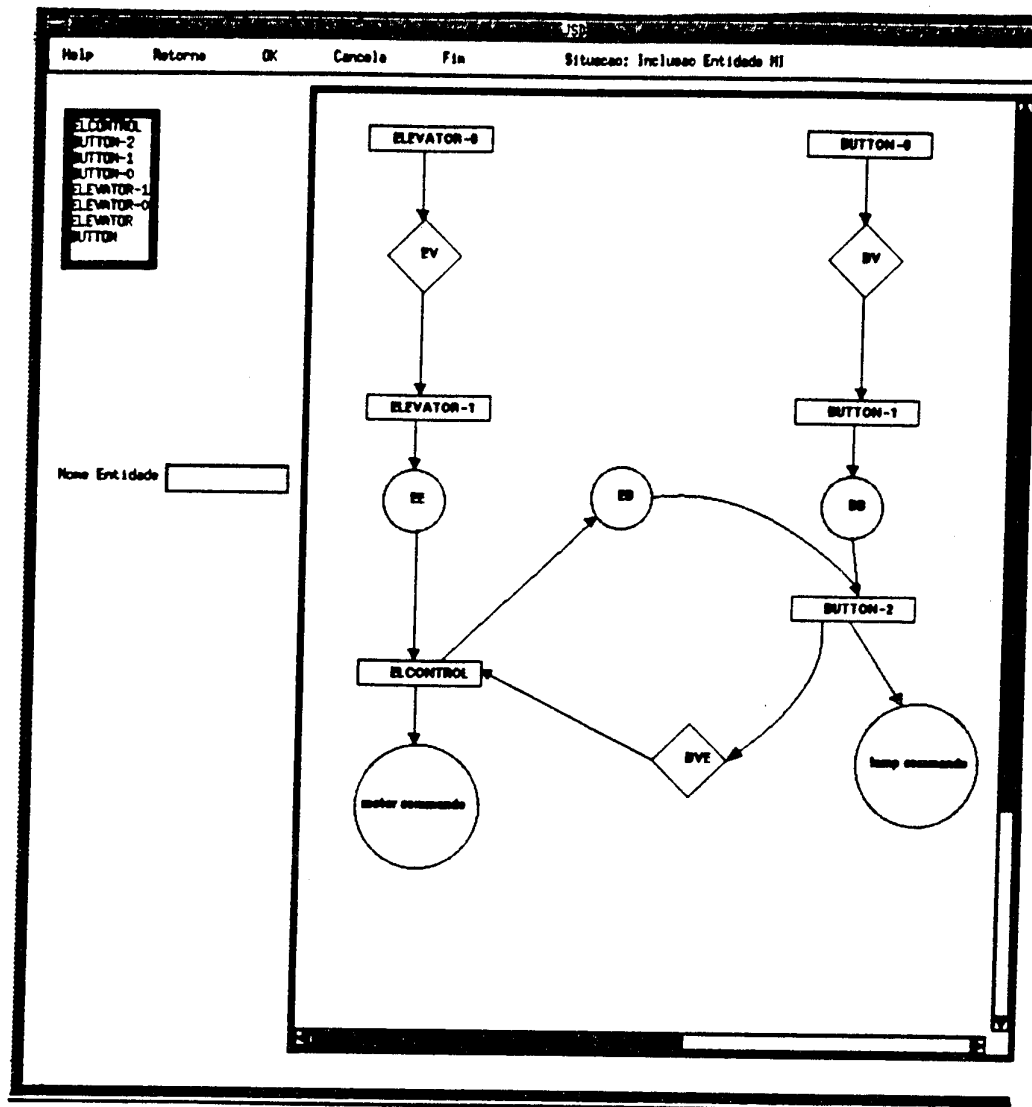


Figura 3

Para especificar detalhes do processo do mundo real, não mostrados nos diagramas, o desenvolvedor escreve um texto, estruturado automaticamente com base no diagrama Estrutura/Entidade, conforme mostra a figura 4.

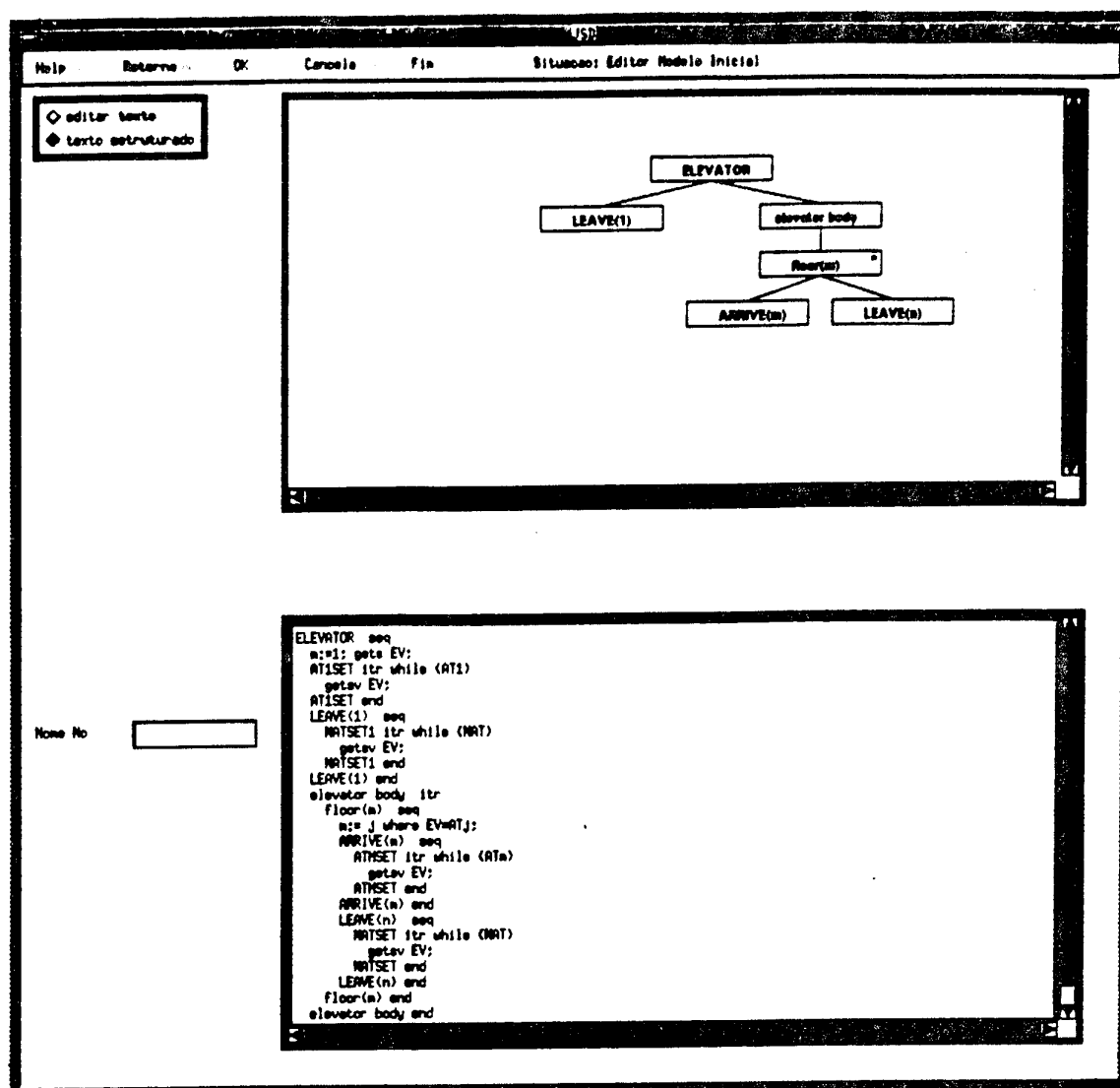


Figura 4

Continuando a especificação do sistema o módulo Funções permite a inclusão de outros processos necessários para completar a modelagem, conforme prevê a metodologia. Além das conexões por vetores de estados as filas de mensagens podem ser conectadas por merge fixo ou merge variável.

O passo Cronologia é coberto por um editor de texto que possibilita ao desenvolvedor especificar os aspectos de escalonamento e sincronismo dos processos, que devem ser levados em consideração na fase de implementação.

O ambiente cobre os 5 passos da metodologia JSD, porém não possui mecanismos para registrar as decisões e justificativas adotadas pelo desenvolvedor.

#### 4 Visão Geral do modelo de Potts

O modelo geral proposto por Potts [Potts 89] segue, essencialmente as idéias do sistema IBIS (Issue Based Information System [Kunz 70]) desenvolvido originalmente com o propósito de representar as atividades de argumentação do "designer". Uma variação mais recente, o sistema gIBIS [Conklin 91] usa uma ferramenta de hipertexto para a discussão exploratória de políticas. O objetivo de IBIS/gIBIS é capturar os problemas de desenho, alternativas para soluções, avaliação de alternativas e as resoluções (commitments) firmes ou tentativas tomadas durante o processo de tomada de decisões na atividade de desenho.

O modelo de Potts [Potts 89] pretende representar deliberações de desenho. O modelo estende o modelo IBIS permitindo a representação da história de derivação de um artefato de desenho, inclusive de um artefato de desenho de software. Começa-se com um artefato abstrato (como um plano), associa-se a ele questões (issues) que surgem ao se procurar tornar o plano mais concreto e associa-se a cada uma das questões as alternativas consideradas, algumas das quais conduzem a um plano mais concreto até que se obtenha um plano suficientemente concreto para ser implementado. Justificativas podem ser associadas a alternativas.

Mais especificamente o modelo genérico de representação de métodos de desenho proposto por Potts [Potts 89] pode ser representado (ver figura 5) através de 5 tipos de entidades, as quais constituem 5 classes básicas: Artefato, Questão, Passo, Posição e Argumento.

A inter-relação entre essas entidades dá lugar a oito relacionamentos binários: Revisa, Sugere, Responde a, Cita, Contribui para, dá Suporte, Rejeita e Modifica.

Artefatos são os documentos específicos de um método e podem ser textuais, gráficos ou de outra forma adequada para manter as informações do desenho.

Um passo do desenho modifica ou cria um artefato. As representações dos raciocínios e decisões de desenho são expressadas pelas Questões, Posições e Argumentos. Uma posição é uma resposta candidata à Questão. Uma decisão é feita quando uma posição é tornada verdadeira (selecionada verdadeira), resolvendo uma questão.

Resumidamente tem-se que:

- Passos: modificam ou criam um novo artefato;
- Questões: são sugeridas pelas execuções dos Passos; revisam os Artefatos;
- Posições: respondem as Questões; contribuem para os Passos, uma vez que o Passo é executado quando uma série de Posições foram selecionadas; e
- Argumentos: dão suporte às Posições; rejeitam as Posições e citam Artefatos.

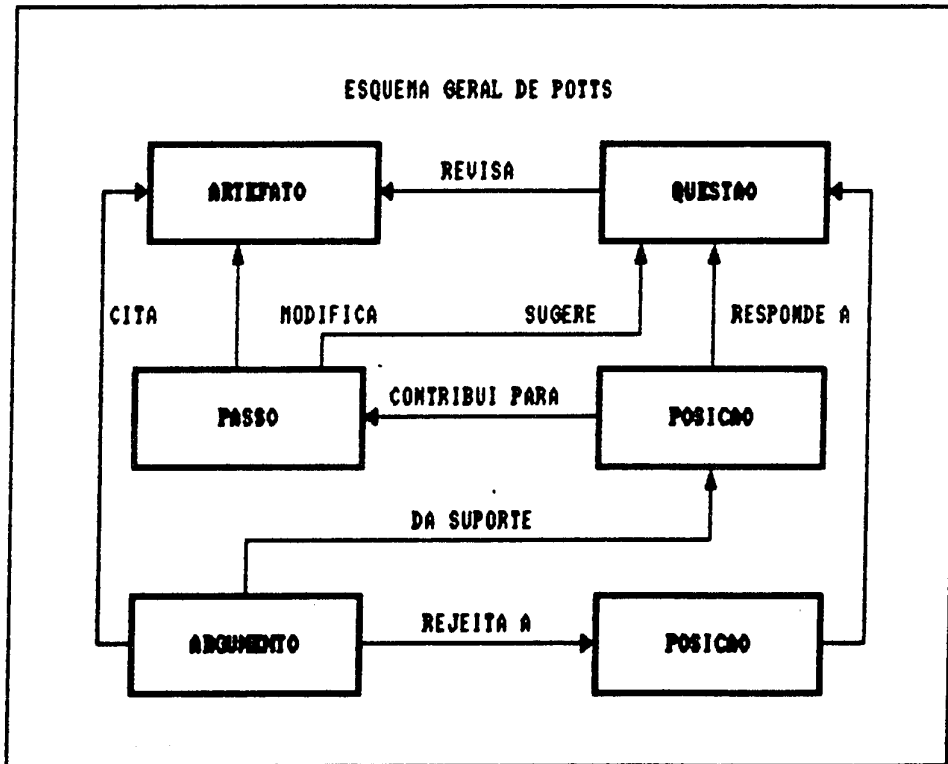


Figura 5

## 5 Adequação do modelo Potts ao método JSD

Para registrar as decisões e raciocínios do desenvolvedor, propõe-se uma customização do modelo genérico de Potts [Potts 89 ] para o caso JSD, conforme os itens a seguir:

- A classe Passo será customizada segundo os passos do Metodo JSD, conforme hierarquia mostrada na figura 6;
- A classe Artefato será customizada para os documentos constantes do desenho desenvolvido no ambiente. Na hierarquia mostrada na figura 7 tem-se os documentos onde são mantidas as informações do desenho. O Universo de informações contém documentos não produzidos pelo método, isto é documentos ou informações pertencentes ao mundo real. Os artefatos podem ser referenciados nas decisões e justificativas;
- A classe Questão será customizada segundo as verificações de desenho, baseadas em heurísticas do método JSD, na forma de checklist, conforme hierarquia mostrada na figura 8;
- A classe Posição será customizada pelas decisões tomadas quando as questões são levantadas;
- A classe Argumento, será customizada com o registro das justificativas adotadas para suportar ou rejeitar as posições.

A classe Passo foi implementada no protótipo pelas 5 fases de desenvolvimento do desenho. Da mesma forma a classe Artefato foi implementada pelos documentos constantes: a Lista de Entidades e Ações, os Diagramas de Estruturas das Entidades , os Textos Estruturados , os Diagramas de Especificação do Desenho (Rede) e opcionalmente documentos do Universo de Informações, que são documentos referenciais.

Dentro do conceito de hierarquia de especialização de classes [Potts 89] estas entidades podem ter novos níveis formando uma hierarquia de classes. Assim, por exemplo, a entidade Passo tem o sub-passo Entidade/Ação, que por sua vez tem o sub-passo Incluir Entidade e assim por diante.

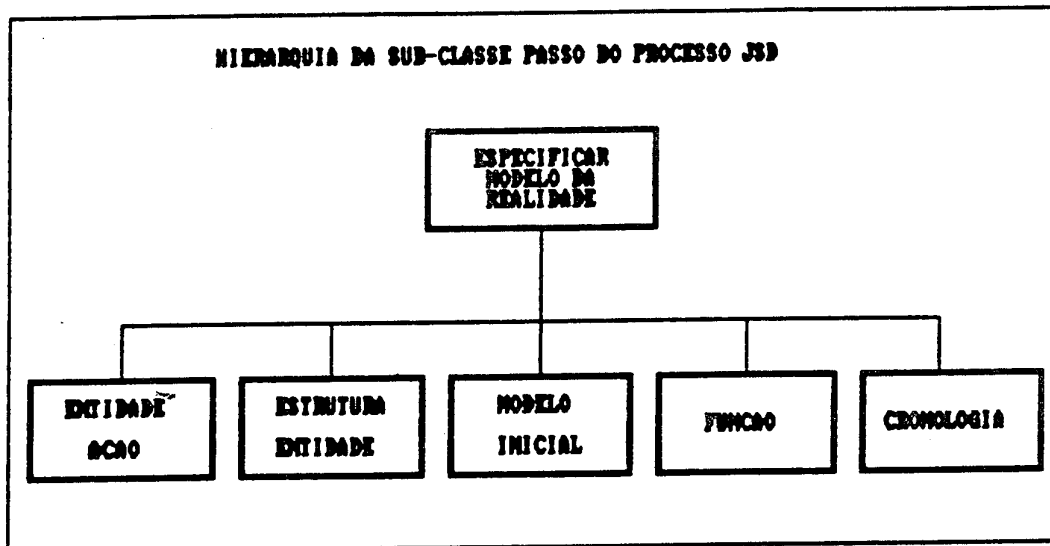


Figura 6

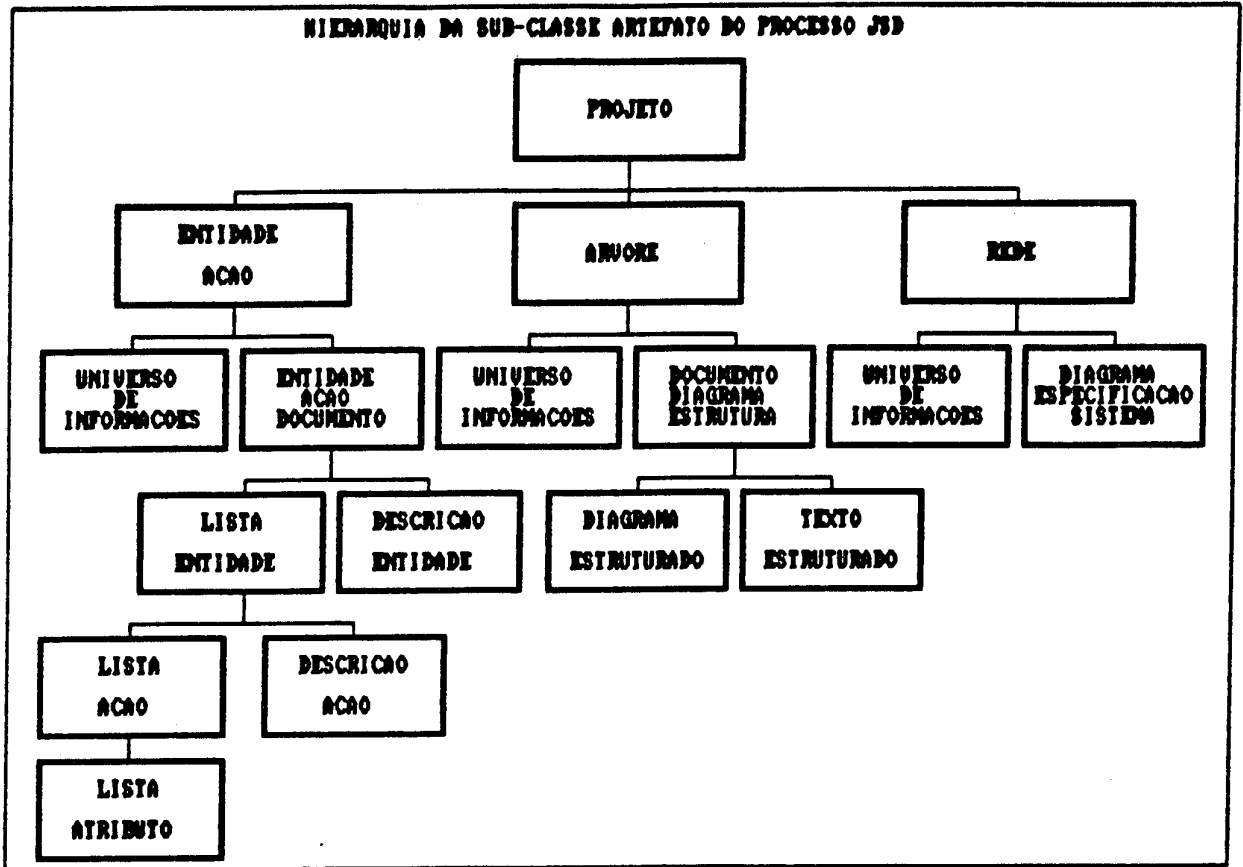


Figura 7

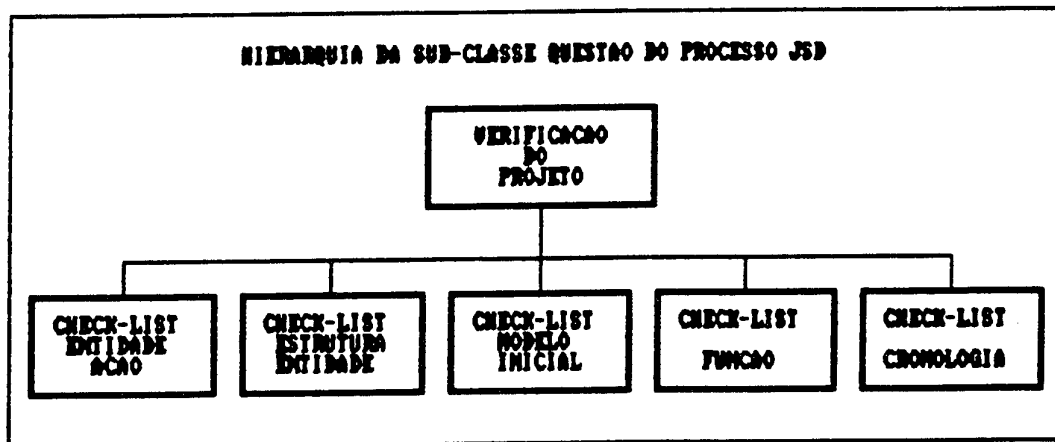


Figura 8

Considerando que o ambiente prototipado já dá suporte às classes Passo e Artefato, através das fases e documentos, respectivamente, da metodologia JSD, é necessário acrescentar ao ambiente o tratamento para as classes Questão, Posição e Argumento, que possibilitarão o registro das decisões e justificativas do desenho, conforme será visto a seguir.

## 6 Integração do estilo procedimental com o declarativo no caso JSD/PUC

A integração dos estilos procedimental e declarativo pode ser feita instanciando-se o modelo genérico(declarativo) de Potts na forma de um programa de processo para o método JSD onde os relacionamentos binários entre as entidades Artefato, Passo, Questão, Posição e Argumento são absorvidos pelas ações Apresentar Questão, Obter Posição, Obter Argumento, Citar Argumento, Alterar Artefato e Sugerir Nova Questão.

Uma descrição genérica deste programa do processo que registra o histórico do desenho é apresentado na figura 9, onde, nas folhas do diagrama, encontramos as ações básicas que compõem o modelo de Potts [Potts 89].

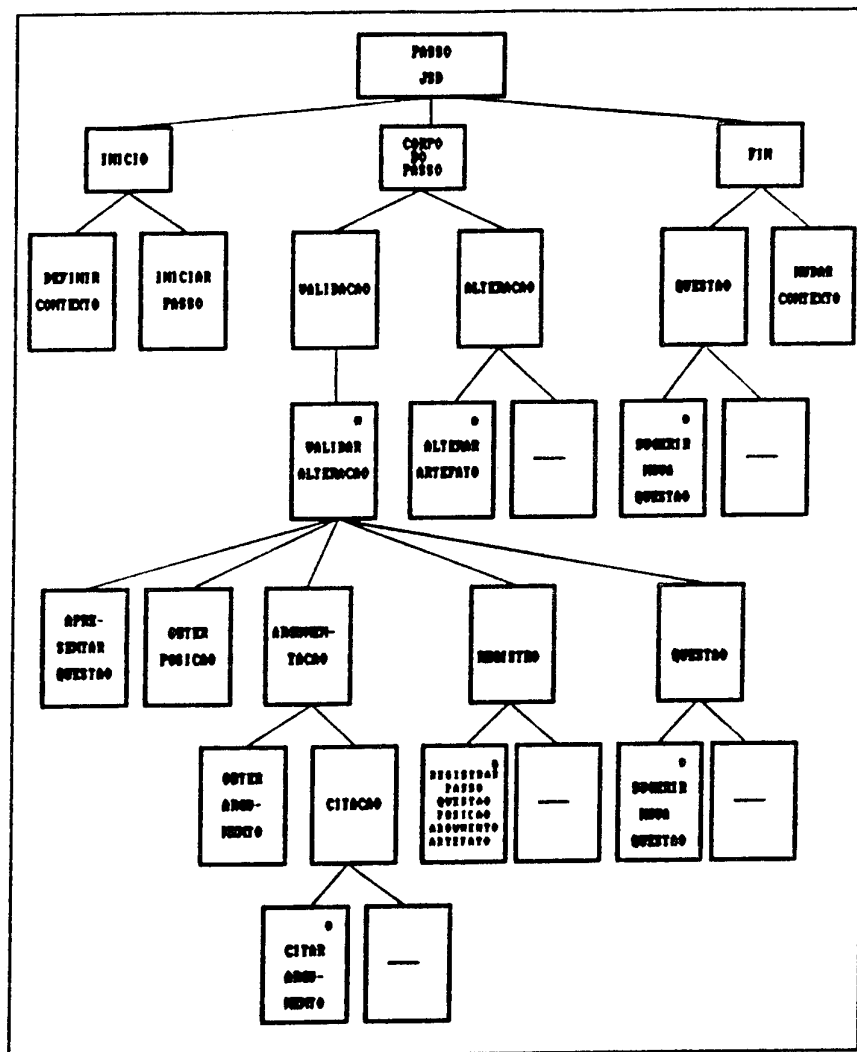


Figure 9

Para cada passo do processo JSD que introduzem alterações nos artefatos há que se passar por uma fase de validação baseada em heurísticas e normas de desenho, quando o desenvolvedor toma as decisões e apresenta as justificativas pela adoção dessa ou daquela decisão. Inicialmente, orientado pelo procedimento JSD, o desenvolvedor define exatamente o contexto no qual a alteração será realizada. Uma mudança de contexto encerra a alteração, a qual, pode eventualmente sugerir nova Questão a ser tratada em outro contexto.

A validação compreende a apresentação de questões, a obtenção de posição sobre a questão, normalmente sim ou não, a argumentação ou seja a justificativa da resposta e opcionalmente o registro da quintupla: passo, questão, posição, argumento e artefato, no histórico do desenho. A justificativa pode citar ou não o argumento, fazendo referência a artefatos(gerados pelo método ou pertencentes ao Universo de Informações). O processo se repete caso nova questão seja sugerida. Uma vez que os dados tenham sido validados o próximo passo será a alteração do artefato pela inclusão, modificação ou exclusão de dados.

Tomando por exemplo o caso do sub-passo Incluir Entidade do passo Entidade/Ação, este programa é instanciado conforme está mostrado na figura 10, para registrar as decisões e justificativas do desenvolvedor ao incluir uma entidade no desenho.

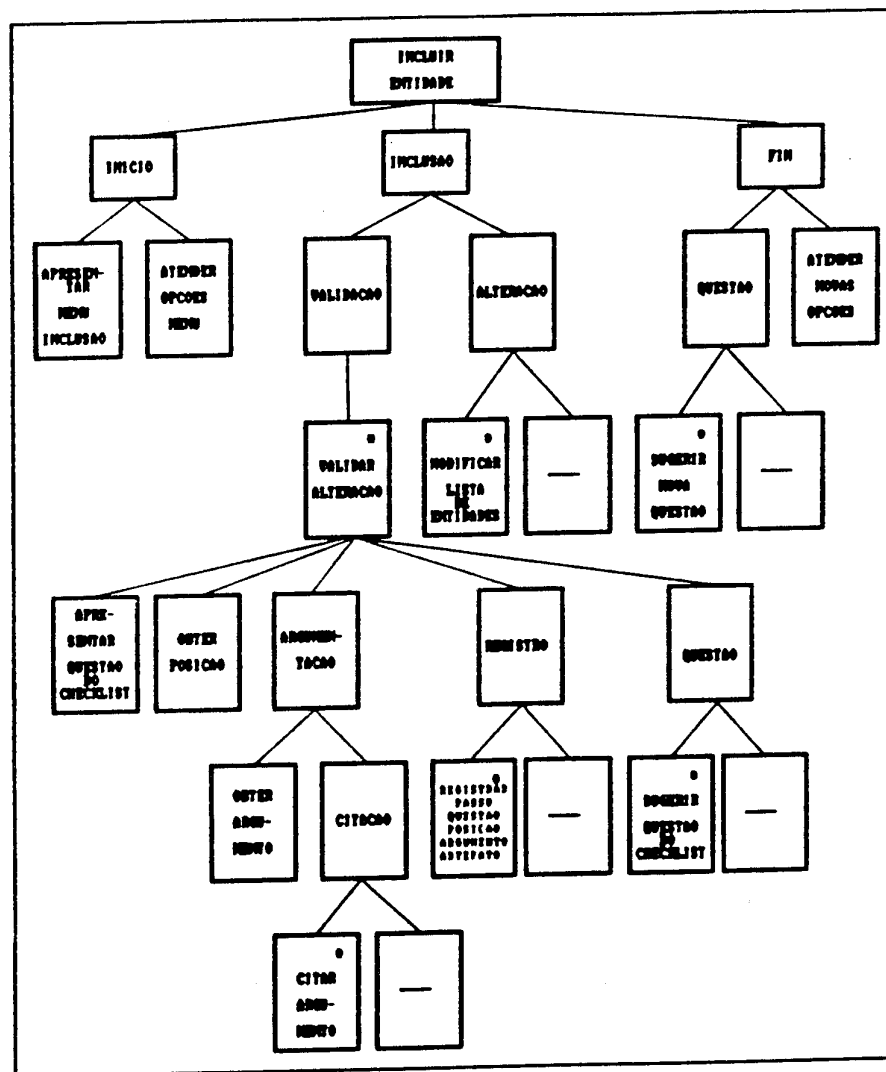


Figure 10

Conforme se pode ver, uma vez obtida a entidade a ser incluída esta é validada submetendo-a as questões constantes do checklist para o passo Incluir Entidade e, caso a entidade seja válida, esta será incluída na lista de entidades.

De uma forma geral, em qualquer dos passos, o método JSD contém heurísticas na forma de checklist para a classe questão que podem ser validadas através de um programa de processo semelhante ao que acabamos de apresentar.

Assim por exemplo, no passo Estrutura/Entidade, há que se validar a ocorrência de entidades que emergem de outras entidade ou entidades que executam vários papéis, antes da criação o diagrama Estrutura/Entidade.

Por fim, convém mencionar que algumas verificações de consistência do método, como por exemplo a existência de uma ação no diagrama Estrutura/Entidade que não consta na lista Entidade/Ação ou, a existência do diagrama de uma entidade que não consta da lista de entidades já são feitas automaticamente, dispensando a validação por parte do desenvolvedor.

## 7 Exemplos

Para caracterizar melhor a adequação proposta para o método geral de registro de decisões de Potts dois episódios descrevem como é feito o registro das decisões de desenho no Ambiente JSD.

Considere o primeiro caso, mostrado na figura 11, onde se deseja incluir uma entidade LEITOR. Tem-se então:

- Passo: Incluir Entidade;
- Questão: É um nome dentro dos limites do desenho?;
- Artefato: Lista de Entidades e Documento que cita o nome LEITOR;
- Posição: Sim ;
- Argumento: Justificativa de que LEITOR contém um conjunto ordenado de ações que se deseja modelar.

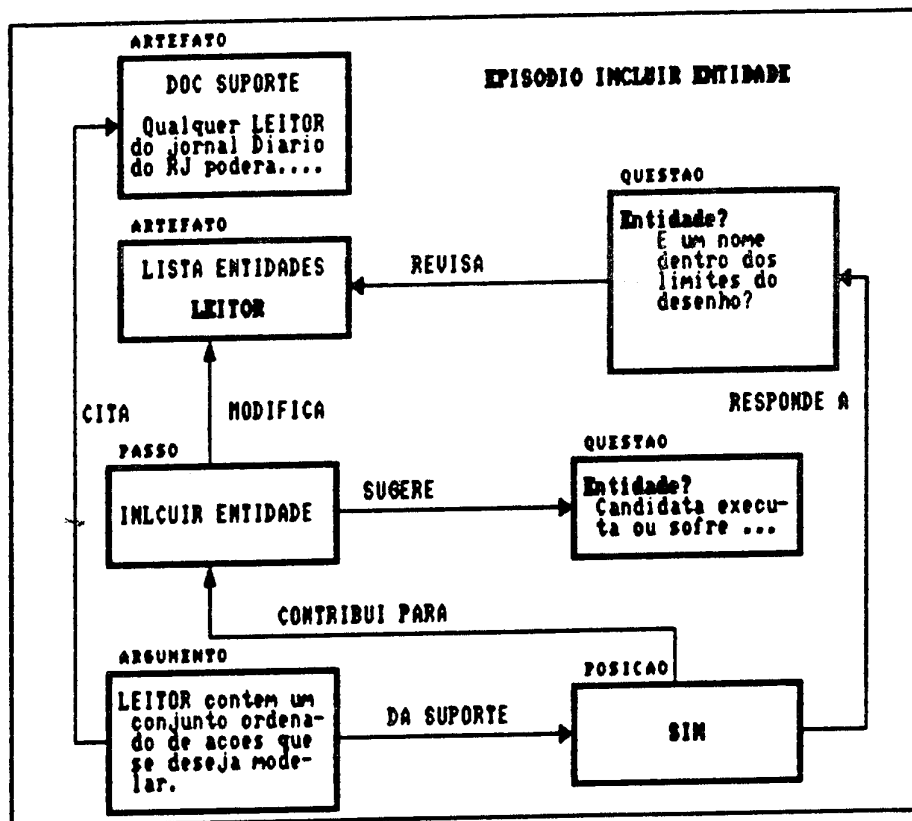


Figura 11



Conforme se pode ver a entidade LEITOR foi incluída na Lista de Entidades e a resolução desta Questão sugere uma nova Questão sobre a validade da candidata com relação as ações sofridas ou executadas, e, então, o ciclo se repete.

O exemplo da figura 12 mostra o episódio de excluir a entidade SESSÃO da Lista de Entidades.

Semelhantemente é levantada a Questão sobre a validade da entidade com relação as ações que ela executa ou sofre num espaço de tempo significativo ordenado. Com o Argumento de que "todas as ações de SESSÃO são ações da entidade JUÍZES, já selecionada", a entidade SESSÃO é excluída. Novamente sugere-se uma nova Questão sobre a existência da entidade SESSÃO no Modelo Inicial(MI), uma vez que esta não pode ser retirada da lista de entidades e permanecer no MI.

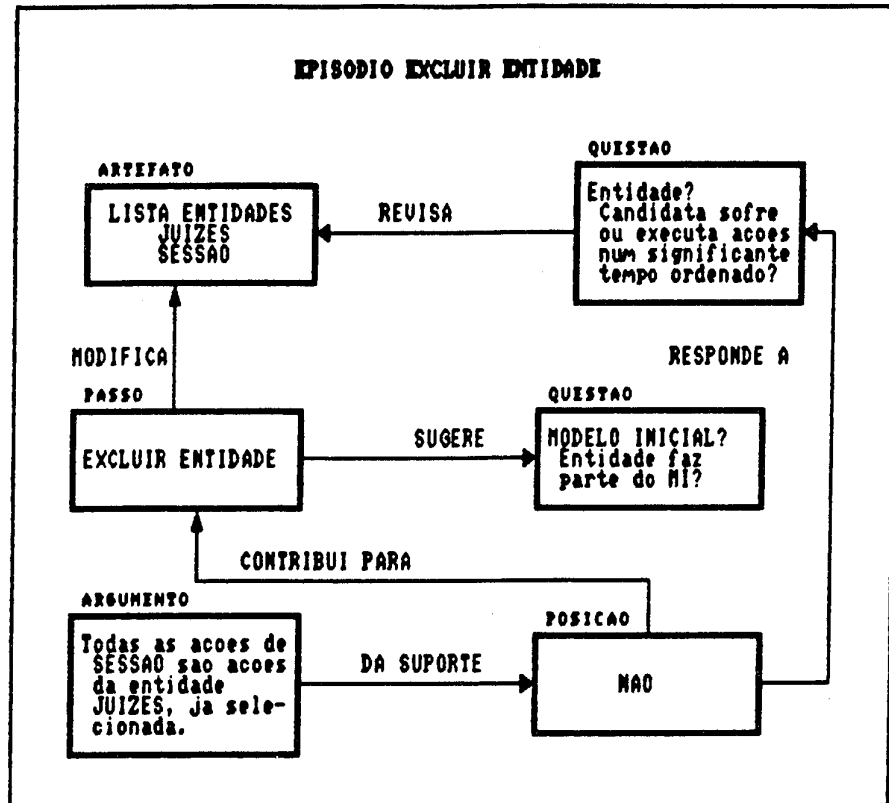


Figura 12

## 8 Conclusões

O presente trabalho explorou, no contexto do ambiente acadêmico experimental, JSD/PUC [Lucena 91], a possibilidade do registro de decisões e justificativas de desenho através do modelo genérico de Potts [Potts 89]. O modelo de Potts pôde ser aplicado porque a arquitetura do ambiente utilizado baseou-se em um programa do processo [Osterweil 87] que sistematizou a aplicação dos métodos que constituem a metodologia JSD [Jackson 83]. Os autores estudam no momento o desenvolvimento de técnicas de reutilização e manutenção de desenhos que exploram as representações estendidas para o JSD aqui propostas.

## 9 Referências Bibliográficas

- [Ambrosio 89] Ambrosio, Ana Maria, Velasco, Flavio R.D. Um Sistema para Execução de Especificações JSD, III Simpósio Brasileiro de Engenharia de Software, 1989.  
 [Bruns 86] Bruns, G.R.; Gehart, S.L.; "Theories of Design: An Introduction to the Literature", Tech. Report, Microelectronics and Computer Technology(MCC), STP-068-86, 1986.

- [Conklin 91] Conklin, Jeff; Yakemovic, K.C.; "A Process-Oriented Paradigm for Design Rationale", Human-Computer Interaction (a ser publicado), 1991.
- [Fischer 91] Fischer, G.A.; Lenke, R.; Morch, M.; "Making Argumentation Serve Design", Human-Computer Interaction (a ser publicado), 1991.
- [Jackson 83] Jackson, M.A.; "System Development", Prentice Hall, 1983.
- [Kunz 70] Kunz, W.; Rittel, H.; "Issues as Elements of Information Systems", Center for Planning and Development Research, Univ. of California, 1970.
- [Lee 91] Lee, V.; Lai, KoY; "Whats in Design Rationale", MIT CCS Tech. Report 118, 1991.
- [Lucena 91] Lucena, Carlos J.P.; Gheiner, Mario; Fernandes, J. Rodrigues; Prado, A. Prado; "JSD/PUC: Um Ambiente de Software Experimental para Estudo do Processo de Automatização do Desenvolvimento de Software", Monografia, Depto de Informática, PUC/RJ, 1991.
- [Maher 90] Maher, M.L.; "Process Models for Design Synthesis", AI Magazine, Winter, 1990.
- [OSF/Motif 90] "Motif Programmer's Reference", Open Software Foundation, Prentice-Hall, 1990.
- [Osterweil 87] Osterweil, L.; "Software processes are software too", Proc. 9th Int. Conf. Software Eng, IEEE Comp. Soc. Press, 1987.
- [Potts 88] Potts, Colin; Bruns, Glenn; "Recording the Reasons for Design Decisions", IEEE Comp. Soc. Press, 1988.
- [Potts 89] Potts, Colin; "A Generic Model for Representing Design Methods", ACM, 1989.
- [Rodriguez 90] Rodriguez M.; Masiero, P. Cesar; "Apoio por Computador a Criação de Especificações Operacionais", IV Simpósio Brasileiro de Engenharia de Software, SBC, 1990.
- [Simon 69] Simon, H. A.; "The Sciences of the Artificial", Cambridge, Mass.; MIT Press; 1969.