

APOIO POR COMPUTADOR À CRIAÇÃO DE ESPECIFICAÇÕES OPERACIONAIS

Maritza Rodríguez
Paulo Cesar Masiero
ICMSC-USP

Resumo

Neste trabalho apresenta-se uma ferramenta construída para apoiar o desenvolvimento de especificações operacionais, baseando-se nos quatro primeiros passos do método JSD (Jackson System Development). Mostra-se como o método foi modelado e a arquitetura do sistema. A especificação criada é armazenada em formato interno em uma base de dados, estando disponível para obtenção de relatórios gerais e relatórios de consistência, validação e referências cruzadas.

Abstract

This paper presents a tool to support the development of operational specifications. This tool is based on the first four steps of the JSD method. It is shown the method data model and its architecture is described. A specification developed using this tool is stored in a data base and can be retrieved in many ways: validation reports, general reports, cross reference reports, etc.

1 Introdução

Ferramentas de apoio por computador ao desenvolvimento de software, designadas pela sigla CASE (Computer Aided Software Engineering), vêm sendo desenvolvidas nos últimos anos de maneira bastante acentuada, visando a facilitar e aumentar a produtividade dos profissionais encarregados do desenvolvimento de software.

JSD (Jackson System Development) é um método para especificação e projeto de sistemas de software e tem como objetivo principal o desenvolvimento de sistemas mais confiáveis e facilmente manuteníveis, ao mesmo tempo que permite a criação de uma especificação concisa, não ambígua e legível. JSD tem recebido bastante atenção na literatura técnica especializada por suas características únicas [Ja83, Ca86]. Em [Za84], por exemplo, Zave destaca o método como sendo o mais evoluído e que melhor se enquadra dentro da abordagem operacional, pois JSD tem procedimentos e diretrizes bem desenvolvidas para suporte ao desenvolvimento de especificações operacionais. A especificação criada por JSD pode ser, em tese, executada por um ou mais processadores.

Os aspectos mencionados no parágrafo anterior motivaram o desenvolvimento de um Ambiente para Desenvolvimento de sistemas baseado na Abordagem Operacional (Projeto ADAO), cujo objetivo é dar apoio a várias fases do ciclo de vida de desenvolvimento de software, desde a especificação até a implementação. Este trabalho descreve uma ferramenta automatizada, denominada Analisador de Especificações Operacionais (AnEspO), que é um dos módulos componentes do ambiente ADAO. Uma descrição mais detalhada do Projeto ADAO encontra-se em [Ms89a].

Alguns esforços têm sido relatados na literatura sobre ferramentas para dar apoio por computador ao uso do método JSD. Cameron, em [Ca86], menciona um editor de diagramas de estrutura com capacidade para gerar textos em COBOL e para simulação do mecanismo de "suspende-e-continua" de JSD. Ambrósio, em [Am88], descreve uma ferramenta para apoio automático à especificação JSD, denominada JSD-TOOL, e outra para simulação da especificação. As duas ferramentas não estão integradas, mas poderiam ser. O tipo de interface e o armazenamento das informações numa base de dados são *as principais diferenças entre o AnEspO e o JSD-TOOL.

2 Revisão sucinta do método JSD

Esta seção tem o objetivo de fazer uma apresentação sucinta de JSD para os que não estão familiarizados com o método, usando como exemplo um sistema de controle de fitas em um Video Clube. Apenas alguns pequenos trechos da especificação desse exemplo serão mostrados. A especificação completa do exemplo pode ser encontrada em [Ms89b] e a descrição completa do método JSD em [Ja83]. O exemplo será utilizado nas seções seguintes para ilustrar os módulos do AnEspO.

O método é composto de seis passos, agrupados em três fases principais: A fase de modelagem, na qual as entidades relevantes são selecionadas e definidas, bem como suas ações; a fase de construção da rede de processos, na qual a especificação é completada com a interconexão dos processos que modelam o comportamento das entidades e dos processos funcionais que recuperam e formatam as informações requeridas pelos usuários do sistema; e, a fase de implementação, na qual os processos são adaptados aos processadores existentes e os dados organizados e ajustados à memória disponível.

Consideremos o exemplo do Video Clube. Entidades relevantes nesse contexto seriam FITA, MEMBRO, RESERVA, etc. Na figura 1 encontra-se uma descrição parcial da entidade FITA. Cada entidade é definida pelo conjunto de ações que executa no mundo real e as ações são identificadas por um conjunto de atributos.

A fase de modelagem prossegue criando-se um diagrama de estrutura para cada entidade. Os componentes no nível mais baixo da estrutura hierárquica são as ações das entidades. O diagrama de estrutura modela cronologicamente a ocorrência

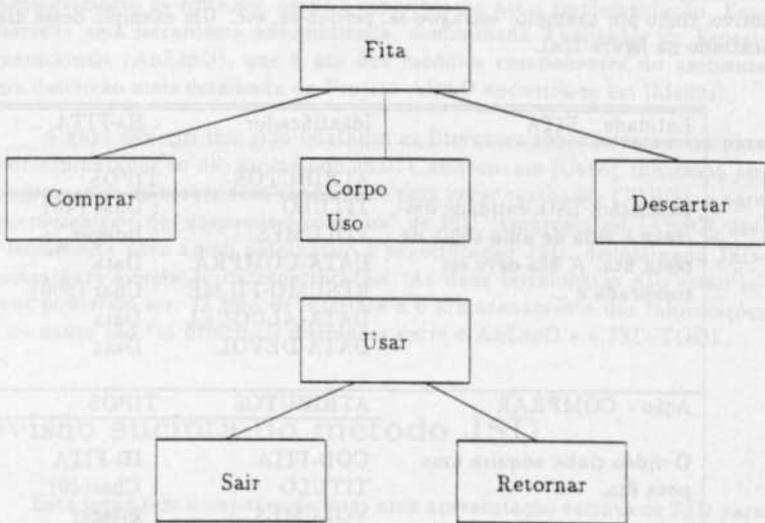
das ações. Por exemplo, uma instância particular de fita deve primeiro ser comprada para então poder ser utilizada no Video Clube. As utilizações ocorrem repetidamente, como seqüências de saídas e retornos da fita. Ao final a fita é descartada por algum motivo, como por exemplo: estragou-se, perdeu-se, etc. Um exemplo desse diagrama é mostrado na figura 1(a).

Entidade : FITA	Identificador	: ID-FITA
	ATRIBUTOS	TIPOS
Descrição: Esta entidade descreve a vida de uma cópia de certa fita. A fita deve ser comprada e ...	TITULO	: Char (40)
	VOLUMES	: Integer
	DATA-COMPRA	: Data
	RESUMO-FILME	: Char (200)
	DATA-ALUGUEL	: Data
	DATA-DEVOL	: Data
	...	
Ação : COMPRAR	ATRIBUTOS	TIPOS
O video clube adquire uma nova fita.	COD-FITA	: ID-FITA
	TITULO	: Char(40)
	VOLUMES	: Integer
	DATA-COMPRA	: Data
	RESUMO-FILME	: Char(200)
Ação: SAIR	COD-FITA	: ID-FITA
Uma fita é retirada do acervo... etc...	MEMBRO	: ID-MEMBRO
	DATA-ALUGUEL	: Data

Tabela 1: Descrição parcial da entidade FITA

A fase de construção da rede de processos consiste da criação de um diagrama, denominado Diagrama de Especificação do Sistema. Esse diagrama é composto das entidades relevantes ao problema (representadas por retângulos rotulados com nomes com sufixos = 0) que se conectam a processos de modelagem (com o mesmo nome das entidades, mas com sufixo maior que zero). Cada ação da entidade no mundo real é enviada como entrada para o processo de modelagem correspondente, que simula internamente o evento ocorrido externamente no mundo real.

O modelo em rede do sistema é acrescido dos processos funcionais, isto é, dos processos que retiram informação de dentro do sistema, passando-as para os usuários na forma de consultas, relatórios, etc. A figura 2 mostra um pequeno trecho



(a)

FITA seq

Comprar;

Corpo-Uso itr

Usar seq

Sair;

Retornar;

Usar fim

Corpo-Uso fim

Descartar;

FITA fim

(b)

Figura 1: Diagrama de Estrutura da Entidade FITA

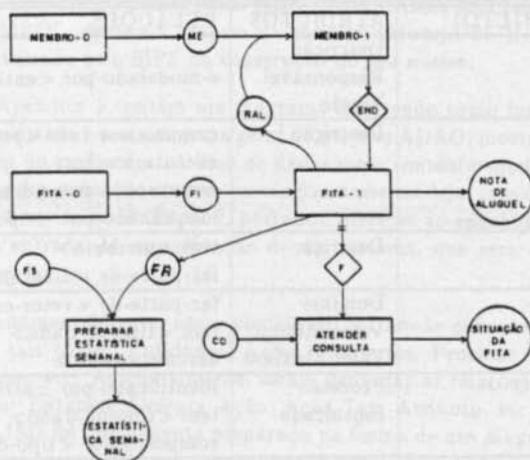


Figura 2: Diagrama de Especificação do Sistema

do diagrama em rede do exemplo do Video Clube. Aparecem nele as entidades FITA e MEMBRO, com seus respectivos processos de modelagem. Mostram-se também duas funções, uma consulta à situação da fita (Disponível ou Alugada) e outra gerando uma estatística mensal da quantidade de fitas alugadas por um membro.

Na figura 2 os processos se comunicam por seqüências de dados, como F1, FA e ME, e nesse caso a iniciativa de enviar a seqüência de dados está no processo onde ela se origina. JSD permite também que um processo verifique o conteúdo (ou estado) das variáveis de outro processo, sem alterá-lo, e essa consulta é denominada "consulta ao vetor de estado" do processo.

Na fase de implementação, controe-se um diagrama hierárquico a partir do Diagrama de Especificação. Os vetores de estado de cada processo são separados em arquivos e acrescenta-se ao sistema um processo escalonador para receber os dados de entrada e executar o procedimento adequado.

3 O Modelo de Dados do Método JSD

Todas as ferramentas que compõem o Projeto ADAO utilizam o núcleo da base de dados do Sistema Integrado para Produção de Software (SIPS), para armazenar sua informação. Esse núcleo está formado por um Interpretador Meta, uma Base de Dados Meta e uma Base de Dados Sistema, complementados por um conjunto de rotinas

ENTIDADE (OBJETO)	ATRIBUTOS	RELAÇÕES
SISTEMA	Descrição Responsável Versão	é-modelado-por <entidade>
ENTIDADE	Descrição	comunica-se com <processo-modelagem> executa <ação> estruturada-por <diagrama-estrutura> identificado-por <tipo-de-dados>
AÇÃO	Descrição	tem <atributo> faz-parte-de <diagrama-estrutura>
ATRIBUTO	Domínio Valor Mínimo Valor Máximo	faz-parte-de <vetor-estado> tem <tipo-de-dado> assume <valor>
VETOR ESTADO	Processo-consultado	Identificado-por <atributo> tem <tipo-de-dado> composto-por <tipo-de-dado>
TIPO DE DADO	Tipo Tamanho	Composto por <tipo-de-dado>
VALOR	Descrição	
DIAGRAMA DE ESTRUTURA	Texto Estrutura Gráfico Help	
PROCESSO FUNCIONAL	Cj. Instruções Especificação	consulta <vetor-estado> intercala <MIT> intercala <seq-dado-interna> gera <seq-dado-interna> gera <seq-dado-saída> recebe <MIT> recebe <seq-dado-interna>
PROCESSO MODELAGEM	Cj. Instruções Especificação	consulta <vetor-estado> gera <seq-dado-interna> gera <seq-dado-saída> recebe <seq-dado-interna> recebe <seq-dado-externa> intercala <seq-dado-interna>
MIT	Descrição	tem <tipo-de-dado>
SEQ. DADO EXTERNA	Descrição	tem <tipo-de-dado>
SEQ. DADO INTERNA	Descrição	tem <tipo-de-dado>
SEQ. DADO SAIDA	Formato Descrição	tem <tipo-de-dado>

Tabela 2 - Modelo de Dados do Método JSD

que manipulam a informação dessas bases [Si88]. Esse sistema será referenciado como "Base de Dados MRO", devido à origem do modelo (Modelo de Representação de Objetos [Tr89]), utilizado pelo SIPS na construção do seu núcleo.

O Apêndice 1 contém um diagrama ilustrando como funciona o núcleo da base de dados utilizada e a arquitetura geral do projeto ADAO, mostrando com mais detalhes os módulos do analisador. A base de dados meta contém o modelo de dados do método e a base de especificação contém instâncias desse modelo, isto é, especificações de sistemas alvo. A tarefa do AnEspO é, portanto, oferecer ao engenheiro de software facilidades para a entrada da especificação de um sistema, que será armazenado na "Base de Especificação".

Fundamentalmente, nesse modelo encontram-se definidos os objetos relevantes em JSD, tais como: Entidades, Ações, Atributos, Processo de Modelagem, Sequências de Dados, etc. Adicionalmente, foram definidas as relações entre esses objetos, por exemplo: Entidade Executa Ação, Ação Tem Atributo, etc. O Modelo de Dados do método JSD foi inicialmente preparado na forma de um diagrama Entidade-Relacionamento para, posteriormente, ser mapeado para a linguagem usada pelo Interpretador Meta para a geração da definição do método JSD na Base de Dados Meta. A Tabela 1 mostra as entidades, atributos e relações modeladas.

Várias entidades e relacionamentos modelados não fazem parte explicitamente de JSD como Entidade, Ação, Processo, etc. . mas são necessários para permitir o armazenamento de informações relevantes para o ambiente em desenvolvimento. Por exemplo, a entidade SISTEMA do diagrama, foi incluída para armazenar informações correspondentes à identificação do sistema alvo e a entidade VALOR é importante para a definição de consistências sobre os dados de entrada do sistema.

4 Arquitetura do Sistema AnEspO

O Analisador de Especificações Operacionais permite ao engenheiro de software entrar com a especificação de um sistema na Base de Dados Sistema. Para realizar esta tarefa, o AnEspO seguiu as diretrizes dos quatro primeiros passos do método JSD.

O sistema AnEspO está dividido em cinco módulos principais: Tipos de Dados, Entidades-Atributos, Estruturação Cronológica, Modelo Inicial e Rede de Processos. Esses módulos formam o menu principal da ferramenta. Adicionalmente, está sendo desenvolvido um sexto módulo que envolve relatórios gerais das informações armazenada na base de especificação.

Da forma como foi desenvolvido, o AnEspO evita que informações inconsistentes sejam armazenadas na base de dados, como por exemplo uma ação não

relacionada a nenhuma entidade. No caso de erros que não puderem ser checados automaticamente, existe a alternativa de emitir relatórios que ajudem ao engenheiro de software a encontrá-los antes de passar para as fases seguintes do Projeto, quando se pretende gerar um protótipo do sistema especificado.

Nas seções posteriores descreve-se cada um desses módulos, relacionando-os com os conceitos envolvidos nos diferentes passos do método JSD.

* ADM * INSERIR ATRIBUTOS Entidade=Fita

SISTEMA : videoclube

Nome : preco de compra

Tipo : value

Continuo
Discreto
Alfanumerico
* Retornar *

ESCOLHA O DOMINIO

Figura 3 - Inserção de um domínio de dados

4.1 Tipos de Dados

Embora os tipos de dados não apareçam explicitamente em nenhuma fase do método JSD, eles foram incluídos como parte da especificação do sistema porque eles são essenciais para que a especificação operacional armazenada possa ser executada automaticamente.

Os tipos de dados participam de vários relacionamentos, e o sistema permite registrar que os atributos, os vetores de estado e as seqüências de dados têm um certo tipo de dado. Como princípio básico o sistema exige que o tipo de dados tenha sido definido antes de ser utilizado, a menos de um caso: a definição de atributos chaves, onde permite-se que na definição de uma ação, o tipo do atributo chave seja declarado posteriormente.

O AnEspO fornece alguns tipos de dados primitivos, como por exemplo Inteiros, Booleanos, Valores Monetários, Caracteres, Datas e Endereços, mas o usuário também tem opção de criar novos tipos de dados, definindo o seu nome e sua composição em função de outros tipos já existentes. Os tipos de dados definidos pelo usuário podem ser divididos em duas categorias: Tipos de Dados Simples, definidos diretamente em função de um tipo de dado já existente e Tipos de Dados Compostos, definidos em função de outros tipos de dados (simples ou compostos), como numa estrutura de registro. A Figura 3 mostra uma tela onde se está inserindo um domínio de dados para um atributo de tipo value.

4.2 Entidades - Ações - Atributos

Em JSD, o sistema a ser desenvolvido deve ser estudado visando a percepção de quais são as entidades de interesse no ambiente real do problema. A especificação deve conter um registro da situação de cada entidade e de como ela se altera a cada ação executada. Esse registro é mantido por diversos atributos relacionados à entidade. Dentro do AnEspO, o módulo Entidades-Ações-Atributos permite ao engenheiro de software a entrada das entidades, ações e atributos do sistema. Esses elementos são armazenados na base de dados da especificação como entidades separadas, mas relacionadas entre si.

■ DADO ■ CONSULTAR ENTIDADES

SISTEMA : videoclube

Nome : Fita

Descrição: fitas disponíveis no video clube para serem alugadas

Identificador: id-fita

Continuar Retornar

Figura 4 - Consulta aos Atributos de uma Entidade

Pela forma hierárquica como o módulo Entidades-Ações-Atributos foi

especificado, garante-se a consistência da informação que ele manipula, no que se refere à dependência entre entidades, ações e atributos que o método sugere. Por exemplo, para inserir uma ação, o usuário deve escolher primeiro a entidade à qual ela pertence (usando mouse ou teclas de setas). Se essa ação já existe na base de especificações, ou seja, trata-se de uma ação compartilhada por outra(s) entidade(s), o sistema alerta o engenheiro de software, mostrando-lhe essa informação e faz com que ele confirme se a inserção a ser feita está correta. Esse controle implícito garante que não possam ser criadas ações ou atributos que não estejam ligados diretamente a uma entidade.

A Figura 4 mostra uma tela do AnEspO, onde se faz uma consulta aos atributos de uma entidade, mostrando-se também o seu atributo identificador.

4.3 Estruturação Cronológica

Em JSD, a estruturação cronológica mostra a ordenação no tempo das ações de uma entidade: basicamente, é um diagrama que mostra a seqüência de ações, iteração de ações, e ações alternativas. Para cada entidade deve haver um Diagrama de Estrutura (DE). No AnEspO, este módulo permite a entrada do diagrama de estruturação cronológica, utilizando uma gramática simples. Essa gramática está baseada nos conceitos de seqüência, seleção, e iteração, utilizando a notação de Jackson.

Atualmente a entrada deste diagrama é feita de forma textual. Planeja-se o desenvolvimento de um editor gráfico para diagramas de estrutura. A gramática utilizada é semelhante à que Velasco usa no JSD-TOOL [Ve86]. A estrutura inserida é checada sintaticamente e, com a adição de novos módulos ao projeto, os nomes de entidades e ações são conferidos na base de dados de especificação para no final armazenar um texto livre de erros.

4.4 Modelo Inicial

No AnEspO este módulo está relacionado com a fase de construção do modelo inicial no método JSD. Cada ação ocorrida na realidade é recebida e processada pelo processo de modelagem correspondente.

O objetivo do módulo "Modelo Inicial", dentro do sistema AnEspO, é permitir ao engenheiro de software inserir o modelo inicial do sistema na base de especificação. Para realizar essa atividade, o sistema pede ao usuário para escolher a entidade que deseja modelar e gera automaticamente os nomes dos processos do mundo real (sufixo 0) e do sistema (sufixo 1) e em seguida interage com o projetista para saber o tipo de conexão entre esses processos (seqüência de dados ou vetor de estado).

4.5 Rede de Processos

Continuando com o Diagrama de Especificação do Sistema, o módulo REDE DE PROCESSOS do AnEspO permite a inserção de outros processos de modelagem do sistema, assim como todos os demais processos que geram as saídas a serem produzidas pelo sistema, chamados em JSD de processos funcionais.

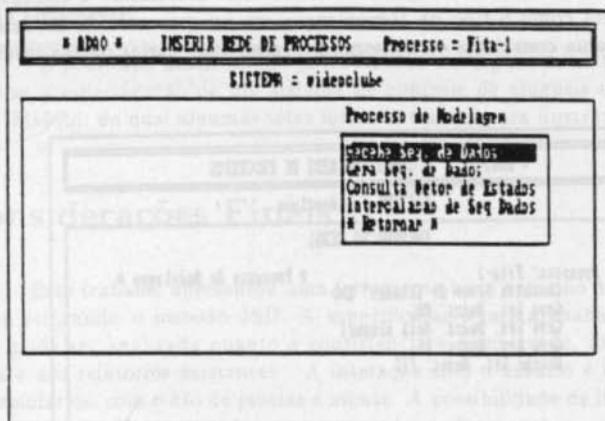


Figura 5 - Processo de Modelagem recebe seqüência de dados

Para cada processo de modelagem e funcional no Diagrama de Especificação do Sistema, no AnEspO indicam-se todos os fluxos de seqüências de dados e/ou vetores de estados que entram e saem dele. Os fluxos de entrada são identificados como **recebe seqüência de dados** Os fluxos de saída são identificados como **gera seqüência de dados** As inspeções aos vetores de estados são denominadas **consulta vetor de estado** Dependendo do tipo de processo que está sendo inserido (Modelagem ou Funcional), o AnEspO pede a informação correspondente a cada caso.

Neste módulo foram adotados alguns princípios básicos com respeito aos marcadores de intervalo de tempo e as seqüências de dados de saída. Assim, os marcadores de intervalo de tempo (MIT) poderão ser do tipo CO (consulta), FS (fim de semana), FM (fim do mês), etc.. ou outros, criados pelo próprio usuário.

As seqüências de dados de saída que correspondem aos relatórios do sistema alvo, terão opção de escolher o tipo de formato com o qual serão mostrados na tela pelo protótipo a ser gerado. Esses formatos são de dois tipos: vertical e horizontal.

O formato vertical será utilizado principalmente para responder a consultas no terminal de vídeo, enquanto que o formato horizontal será mais utilizado para emitir relatórios via impressora. No primeiro, os campos aparecerão dispostos horizontalmente, isto é, cada campo seguido de seu conteúdo, numa linha. No segundo, haverá um cabeçalho, contendo o título dos campos e em seguida aparecerão as várias linhas de detalhe.

Como exemplo, a Figura 5 contém telas mostrando a escolha de uma seqüência de dados como o tipo de comunicação em um processo de modelagem e a Figura 6 mostra uma consulta a esse mesmo processo após várias outras inserções.

```

  * ADO *      CONSULTAR REDE DE PROCESSOS
  SISTEMA : videoclube

  INFORMACAO GERAL

  Processo: Fita-1      * Processo de Modelagem *
  Consulta Detor de Estado: EMO
  Gera seq. Dados: NAL
  Gera seq. Dados: Nota aluguel
  Gera seq. Dados: ALI
  Recibe seq. Dados: FIL

  Continuar  Retornar
  
```

Figura 6 - Conexões de um processo

4.6 Relatórios

Como foi mencionado anteriormente, o sistema AnEspO coloca à disposição do engenheiro de software alguns relatórios, que podem servir como documentação do sistema, assim como para verificação de consistência da base. Inicialmente, foram projetados quatro tipos de relatórios: Entidades e Ações, Ações e seus Atributos, Resumo da Rede de Processos e Tipos de Dados não definidos (correspondentes a tipos de dados que são identificadores de entidades). Outros relatórios estão podendo ser facilmente programados e acrescentados à ferramenta.

4.7 Comentários Adicionais

A implementação do AnEspO foi realizada utilizando a linguagem de Programação C e um "tool kit" para manipulação de janelas, em um ambiente de microcomputador tipo PC/DOS. A interface conta com uma padronização de operações, com opções de inserir, consultar, modificar e eliminar informação da base. Na medida em que a informação é inserida, ela pode aparecer em seguida num menu de escolha, evitando que se tenha que digitá-la novamente.

O protótipo desenvolvido já foi testado em alguns casos completos. Um deles refere-se à especificação de um sistema de controle de aluguéis de fitas em um vídeo clube [Ms89b], do qual algumas telas foram extraídas para ilustrar este trabalho.

5 Considerações Finais

Este trabalho apresentou uma ferramenta para apoio ao desenvolvimento de sistemas utilizando o método JSD. A especificação é armazenada em uma base de dados e pode ser analisada quanto a consistência e completude, através das telas de consulta e dos relatórios existentes. A interação com o usuário é feita através de menus e formulários, com o uso de janelas e mouse. A possibilidade de interação direta, utilizando editores gráficos poderá ser acrescentada no futuro. A capacidade da base de dados depende apenas da memória secundária disponível no microcomputador.

O objetivo principal do projeto é conseguir a geração automática de um protótipo do sistema, a partir da especificação operacional. Para isto, um editor de programas orientado por sintaxe está sendo desenvolvido, para que se possa detalhar a especificação dos processos de modelagem e funcionais até o ponto em que possam ser interpretados. Além desse, um outro módulo, para gerar e executar o protótipo está sendo projetado. Detalhes adicionais deste módulo podem ser encontrados em [Ga90].

Referências

- [Am88] AMBRÓSIO, A. M. - *Um Sistema para Execução de Especificações JSD*, Dissertação de mestrado, Instituto de Pesquisas Espaciais, 1988.
- [Ca86] CAMERON, J.R. - *An Overview of JSD*. IEEE Transaction on Software Engineering, vol. SE-12, pp 222-240, February 1986.
- [Ca88] CAMERON, J. R. - *The modelling phase of JSD*. Information and Software Technology, vol. 30, no. 6, pp 373-383, July/August, 1988.

- [Ga90] GASPAROTTO, I. A. - *Proposta de um Executor de Especificações Operacionais*. Mini-dissertação apresentada para o exame de qualificação. ICMSC-USP, São Carlos, 1990.
- [Ja83] JACKSON, M. - *System Development*, Prentice-Hall, 1983.
- [Ms89a] MASIERO, P.C et al. - *Um Ambiente de Desenvolvimento baseado na Abordagem Operacional*, Notas do ICMSC-USP, No. 57. São Carlos, 1989.
- [Ms89b] MASIERO, P.C - *Uma Visão Geral do Método JSD*, Notas Didáticas do ICMSC-USP, No. 3, 1989.
- [Si88] - *SIPS : Manual de Referência*. III Escola Brasileira Argentina de Informática. Curitiba, 1988.
- [Tr89] TRAINA Jr., C. e SLAETS, J.F.W - *Um modelo de representação de objetos*. (Em preparação), 1989.
- [Ve86] VELASCO, F.R.D. - *JSD - Tool Reference Manual*. Technical Report 86-08. Wang Institute of Graduate Studies, Tyngsboro, MA., 1986.
- [Za84] ZAVE, P. - *The Operational Versus The Conventional Approach to Software Development*. Communications of the ACM, 27(2), pp 104-118. February 1984.

Referências

APÊNDICE A

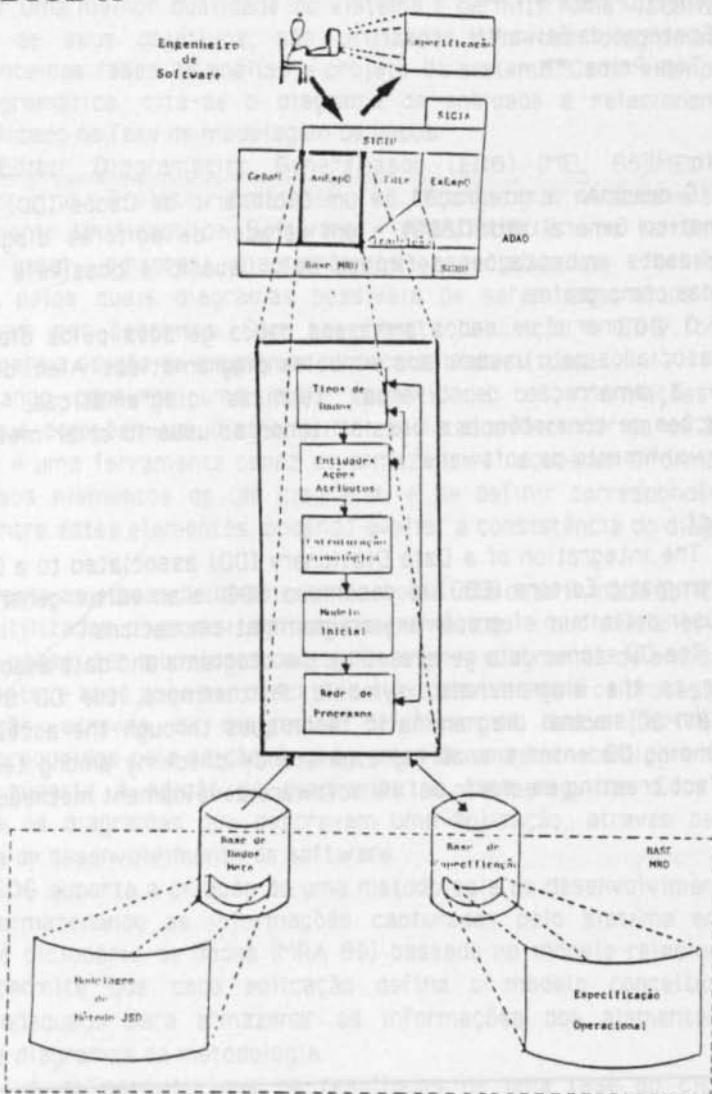


FIGURA A.1 - O AnEspO dentro do ambiente ADAO