



I.C.M.S.C.

UNIVERSIDADE DE SÃO PAULO  
CAMPUS DE SÃO CARLOS  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS DE SÃO CARLOS

UMA VISÃO GERAL DO MÉTODO JSD

PAULO CESAR MASTIERO

Nº3

Notas Didáticas do ICMSC - USP

**UMA VISÃO GERAL DO MÉTODO JSD**

**PAULO CESAR MASTERO**

Nº3

# UMA VISÃO GERAL DO MÉTODO JSD

PAULO CESAR MASIERO

ICMSC-USP

## 1. INTRODUÇÃO

O método JSD - Jackson System Development [1] surgiu como uma evolução do método JSP - Jackson Structured Programming [2]. Este último é voltado para o projeto de programas, enquanto JSD tem o objetivo de auxiliar na especificação e projeto de Sistemas de Informação, bem como sistemas de tempo-real.

A especificação de um Sistema de Informação em JSD é constituída por uma rede de processos sequenciais. Os processos são de dois tipos: de modelagem e funcionais. Os primeiros modelam o comportamento das entidades do mundo real envolvidas no sistema, enquanto os outros geram as funções úteis para os usuários.

O método é composto de seis passos, agrupados em três fases principais: A fase de modelagem, na qual os processos são selecionados e definidos, a fase de construção da rede de processos, na qual a especificação é completada com a interconexão dos processos, e a fase de implementação, na qual os processos são adaptados aos processadores existentes e os dados organizados e ajustados à memória disponível.

Na fase de modelagem escolhem-se as Entidades e Ações relevantes para o sistema considerado e ordenam-se cronologicamente as Ações das Entidades. Essa fase define o ambiente do problema do usuário, independentemente das funções a serem executadas dentro desse ambiente. A construção da rede de processos é realizada construindo-se um modelo inicial e acrescentando-se a seguir a especificação das funções requisitadas pelos usuários. Na fase de implementação (re)avaliam-se os aspectos de articulação temporal do Sistema e desenvolve-se o projeto de implementação, na forma de uma hierarquia de programas controlados por um programa principal escalonador dos demais.

Os seis passos são os seguintes:

### I. Fase de Modelagem

1. Escolha das Entidades e Ações
2. Estruturação Cronológica das Ações

### II. Fase de Construção da Rede de Processos

3. Construção do Modelo Inicial
4. Adição de Funções

### III. Fase de Implementação

#### 5. Articulação Temporal dos Componentes do Sistema

#### 6. Projeto da Implementação do Sistema

O objetivo deste trabalho é fazer uma introdução ao método JSD. Isso é feito resumindo-se os seus principais conceitos e componentes e usando-se um exemplo bastante pequeno e simplificado. Em cada uma das três seções seguintes são apresentadas as três fases principais do método e a parte da solução do exemplo correspondente à fase.

## 2. A FASE DE MODELAGEM

O sistema a ser analisado deve ser estudado pelo analista de sistema visando a percepção de quais são as entidades de interesse no ambiente real do problema. As entidades são melhor percebidas pelas ações que elas executam ou sofrem no mundo real. As ações são eventos que afetam o ambiente e mudam a situação de cada entidade. Por exemplo, no ambiente de um banco, CLIENTE seria uma entidade relevante e algumas de suas ações seriam: ABRIR A CONTA, DEPOSITAR, SACAR E FECHAR A CONTA. O Sistema deve manter 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 tais como: nome do cliente e saldo. As ações de DEPOSITAR e SACAR alteram o saldo do cliente, por exemplo.

Vamos considerar como exemplo um videoclub de médio porte. Os membros se inscrevem mediante o pagamento inicial de uma módica quantia e o preenchimento de uma ficha cadastral. Após isso estão aptos a retirarem quantas fitas quiserem, desde que uma de cada vez, pagando um preço fixo diário por fita retirada. O objetivo do Sistema de Informação a ser projetado é controlar o pagamento dos clientes e a situação de cada fita.

Após a análise mais detalhada desse sistema, as entidades consideradas como relevantes são MEMBRO E FITA. A tabela 1 mostra a lista dessas entidades, de suas ações e dos atributos associados às ações.

As ações sempre ocorrem numa certa ordem e elas definem o comportamento das entidades. No caso de um banco, o cliente deve primeiro abrir a conta e realizar um depósito inicial, podendo em seguida movimentar a conta quantas vezes quiser (isto é, movimentar iterativamente a conta). Cada movimento pode ser um depósito ou uma retirada. A ação final de todo cliente será sempre fechar a conta. A interação do cliente com o banco pode durar meses ou anos e o modelo dessa interação deve representar essa longa duração.

O Diagrama de Estruturação Cronológica, ou Diagrama de Estrutura das Entidades, mostra a ordenação no tempo das ações de uma entidade. O diagrama mostra sequência de ações (representadas por retângulos), itera

ção de ações (retângulo com asterisco no canto superior direito) e ações alternativas (retângulo com círculo no canto superior direito). Deve-se produzir um Diagrama de Estrutura Cronológica (DE) para cada entidade escolhida. A lista das entidades mostradas na tabela 1 dá uma visão estática das entidades e ações, ao passo que o DE oferece uma visão dinâmica do mesmo sistema. A entidade MEMBRO deveria ter uma ação final CANCELAR inscrição, omitida por simplificação.

ENTIDADE	AÇÃO	ATRIBUTO
MEMBRO	INSCREVER	identificação do membro nome do membro endereço, etc.
	RECEBER NOTA DE ALUGUEL	identificação do membro identificação da fita total devido data
	PAGAR	identificação do membro data pagamento valor do pagamento, etc.
FITA	COMPRAR	identificação da fita título volumes data da compra resumo do filme, etc.
	SAIR	id-membro id-fita data do aluguel
	RETORNAR	identificação da fita data da devolução
	DESCARTAR	identificação da fita data do descarte motivo

Tabela 1

As saídas requeridas dos Sistemas de Informação normalmente se referem a aspectos históricos do ciclo de vida das entidades. Por exemplo:  
Quando foi feita a última retirada nesta conta ?

A fita xyz está emprestada ?

Quantas fitas, em média, este cliente retira por mês ?

O sistema só pode atender com eficiência (e às vezes não consegue atender) aquelas consultas que se referem a entidades às quais um histórico de seu comportamento no mundo real é mantido, daí a relevância da modelagem dinâmica realizada através dos Diagramas de Estrutura Cronológica das ações das entidades. A figura 1 mostra os DEs para as entidades MEMBRO e FITA. O texto estruturado mostrado abaixo é uma forma equivalente para representação da estruturação cronológica das ações.

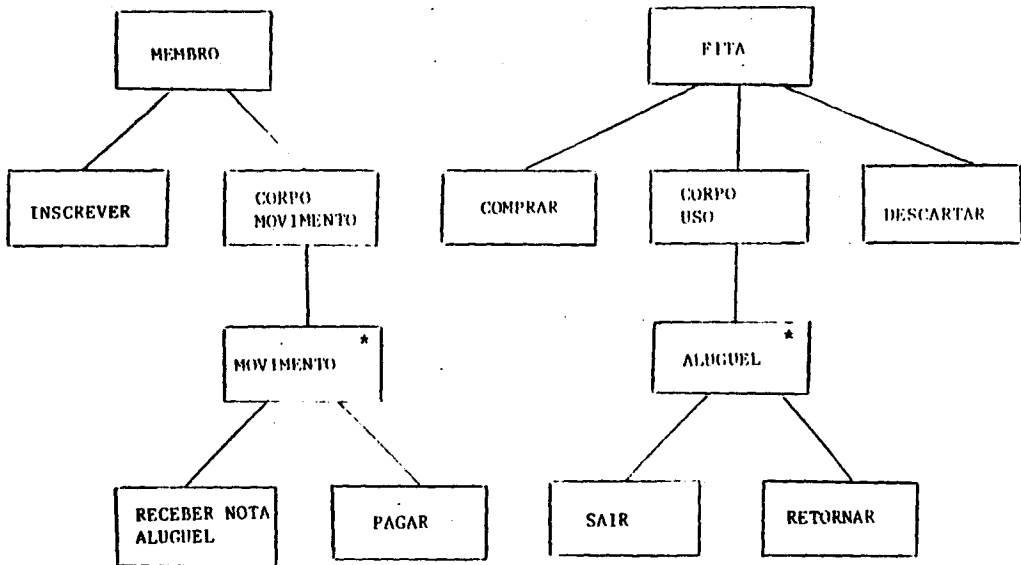


Figura 1

```

FITA sequência
  COMPRAR;
  CORPO-USO iteração
    ALUGUEL sequência
      SAIR;
      RETORNAR;
    ALUGUEL fim
  CORPO-USO fim
  DESCARTAR;
FITA fim
  
```

Ambos os diagramas da figura 1 não apresentam exemplo de ações alternativas. Tomemos o caso da entidade MEMBRO. Da forma como foi modelado, um membro do videoclube sempre deve PAGAR a última NOTA DE ALUGUEL recebida, talvez no retorno da fita alugada. Com outras palavras: o MEMBRO não pode alugar novas fitas sem ter antes pago o aluguel anterior. Para mostrar um exemplo com ações alternativas, apresentamos o diagrama da figura 2, onde as ações RECEBER NOTA DE ALUGUEL e PAGAR são alternativas. Esse modelo da realidade relaxa a restrição anterior, isto é, um membro pode alugar várias fitas e ir pagando aos poucos. Entretanto, ele não muda a restrição de que o pagamento deve vir após o recebimento da nota de aluguel. Isso poderia ser feito através de uma nota informal acrescentada no diagrama.

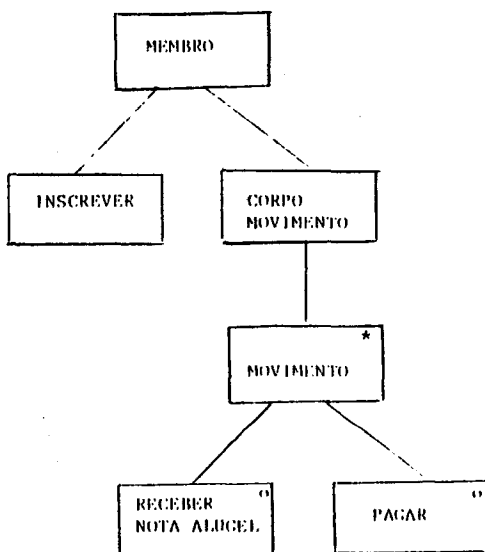


Figura 2

Devemos entender o DE, como referindo-se a um tipo de entidade, representando o comportamento de qualquer instância de entidade daquele tipo. Uma determinada fita, por exemplo, poderia ter a seguinte história: COMPRAR, SAIR, RETORNAR, SAIR, RETORNAR, DESCARTAR. Isso significa que essa fita, após dois alugueis, tornou-se imprestável e foi aposentada. Outra fita poderia ter uma história mais curta: COMPRAR, DESCARTAR significando que ela já foi comprada com problemas.

### 3. A FASE DE CONSTRUÇÃO DA REDE DE PROCESSOS

Nesta fase deve-se conectar o modelo iniciado na fase anterior com a parte do mundo real delimitada pelo sistema. Isso ocorre pensando-se

no modelo como um simulador do que ocorre no mundo real. Cada ação ocorrida na realidade é recebida pelo sistema (com algum atraso) e uma operação é alocada para processar essa ação, fazendo com que o sistema ajuste-se internamente ao que ocorreu no seu exterior.

Em termos técnicos, devemos alocar operações de leitura das ações num canal de comunicações que liga o mundo real e os processos que simulam o mundo real. Esse canal de comunicações será definido posteriormente e poderão ser um terminal de vídeo, uma leitora de cartões magnéticos, etc. Também deve ser decidido quais são as condições para o término das iterações e para a escolha de decisões alternativas.

A figura 3 mostra o modelo inicial da especificação do sistema. O sufixo 0 (zero) indica o processo no mundo real e o sufixo 1 (um) indica o processo de modelagem que simula o comportamento do processo no mundo real. Esse diagrama é denominado de Diagrama de Especificação do Sistema (DES). Os retângulos representam os processos do sistema e as circunferências as mensagens ou sequências de dados, no sentido indicado pela flecha.

O texto estruturado do processo FIIA-1, mostrado em seguida, completa a especificação inicial, com a alocação das operações executáveis para leitura de fluxo de dados. Pode ocorrer que entre uma leitura e outra, decorra um longo tempo e nesse caso o processo de modelagem fica parado, esperando o aparecimento de uma mensagem no canal de comunicação.

```
FIIA-1 seq
  Ler FIT;
  COMPRAR; Ler FIT;
  CORPO-USO itr (enqto ação = SAIR)
    ALUGUEL seq
      SAIR; Ler FIT;
      RETORNAR; Ler FIT;
    ALUGUEL fim
  CORPO-USO fim
  DESCARTAR;
FIIA-1 fim
```

Se o sistema fosse implementado neste ponto, ele estaria simulando o comportamento do videoclube no mundo real, sem produzir nenhuma saída. Todas as saídas que podem ser produzidas pelo sistema dependem da modelagem inicial. Em JSD, as funções do sistema são os processos que geram as saídas a serem utilizadas pelos usuários. Tem-se, então, os processos de modelagem, que simulam as entidades no mundo real e os processos funcionais, que produzem as saídas (ou funções) do sistema.

Em seguida deve-se completar o modelo inicial do sistema (figura 3) com os processos funcionais. Há dois tipos principais de funções: funções interativas, que interagem com os processos de modelagem, gerando outras entradas para o sistema e funções que geram as saídas do sistema.





Figura 3

Continuando com o exemplo do videoclube, vamos imaginar que os usuários solicitaram os três relatórios listados abaixo:

1. Produzir automaticamente as notas de aluguel.
2. Informar a situação de cada fita, isto é, se está fora ou se está disponível para alugar.
3. Emitir um relatório semanal com a estatística das fitas alugadas durante a semana.

Vamos considerar primeiro a produção automática de notas de aluguel. Quando uma fita sai da fitoteca, assinalando que um aluguel está em curso, o processo FITA-1 deve enviar uma mensagem referente à ação RECEBER NOTA DE ALUGUEL, para o processo MEMBRO-1. Este, ao receber a mensagem, deve registrar os dados referentes a essa ação, tais como: valor a pagar, data do aluguel, etc. Dessa forma, a operação RECEBER NOTA DE ALUGUEL deixa de passar pelo canal de comunicações que liga o mundo real ao processo FITA-1 (MEM, na figura 3) e passa a ser gerada automaticamente por FITA-1, após cada ação SAIR.

Em seguida, FITA-1 deve consultar o registro histórico de MEMBRO-1, para saber o nome e endereço que deverá constar da nota de aluguel. O registro do histórico de uma entidade está armazenado no que chamamos de vetor de estado da entidade. A consulta ao vetor de estado é representada por um losango no Diagrama de Especificação do Sistema, com a flecha na direção do processo que realiza a consulta.

O processo que gera de fato a nota de aluguel poderia ser um processo funcional adicionado ao DES, mas neste caso, por ser uma simples impressão, decidimos embutir essa função dentro do processo de modelagem FITA-1. A figura 4 mostra a situação do DES após a inclusão dessa função. O texto estruturado do processo de modelagem FITA-1 toma então o formato apresentado abaixo. Observe a inclusão da consulta ao vetor de estado de

MEMBRO-1, do envio da mensagem REC(RECEBER NOTA DE ALUGUEL) ao processo MEMBRO-1, e do processo funcional "EMITIR-NOTA-DE-ALUGUEL" embutido dentro de FITA-1.

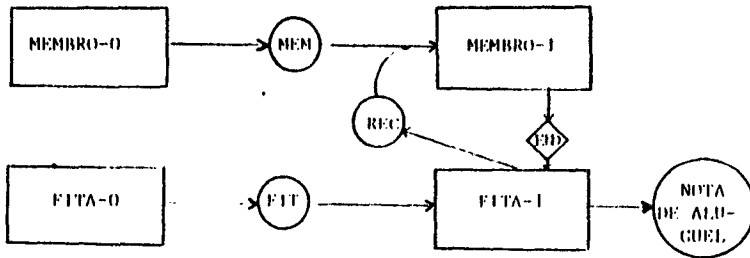


Figura 4

```

FITA-1 seq
  Ler FIT;
  COMPRAR; Ler FIT;
  CORPO-USO itr (enqto açao - SAIR ou RETORNAR)
  ALUGUEL seq
    SAIR seq
      ObterVE MEMBRO-1;
      EMITIR-NOTA-DE-ALUGUEL seq
        Enviar nome, endereço, etc.,
        para a saída NOTA DE ALUGUEL;
      EMITIR-NOTA-DE-ALUGUEL fim
      Enviar REC;
      Ler FIT;
    SAIR fim
    RETORNAR; Ler FIT;
  ALUGUEL fim
  CORPO-USO fim
  DESCARTAR;
FITA-1 fim
  
```

Vamos agora considerar a consulta à situação da fita. Quando o usuário deseja realizar uma consulta, deve informar seu desejo ao sistema através de algum sinal. Num sistema manual, a própria existência da ficha externa da fita no painel, indica que ela se encontra disponível para aluguel. Num sistema automatizado, o sinal seria um fluxo de dados vindo externamente ao sistema e indicando o desejo de ativar o processo que realiza a consulta. Este consulta o vetor de estado do processo de modelagem FITA-1 e retorna a mensagem apropriada. A figura 5 mostra esse trecho do DES.

O DES deve ser interpretado como modelando uma única instância de cada entidade, por isso não há necessidade de se preocupar com a identificação de qual fita se deseja consultar: isso será feito automaticamente na fase de implementação. As barras paralelas entre o processo FITA-1 e o vetor de estado F indicam que, ao longo da vida útil do processo funcional ATENDER-CONSULTA, ele vai consultar várias vezes o estado de FITA-1. O texto estruturado de ATENDER-CONSULTA é então:

```

ATENDER-CONSULTA itr
PROC-GRUPO-CO seq
  Ler CO;
  ObterVE FITA-1;
  Imprimir Situação da Fita
PROC-GRUPO-CO fim
ATENDER-CONSULTA fim

```

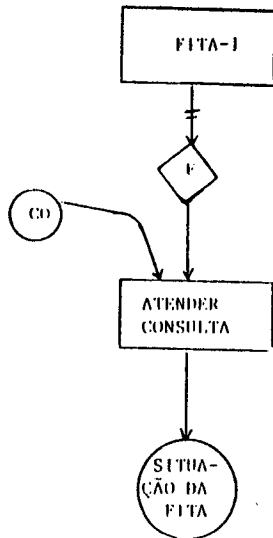


Figura 5

ATENDER-CONSULTA é um processo bastante simples. Mas o leitor atento deve estar estranhando a colocação da instrução "Ler CO" e o fato de que o processo nunca acaba. De fato, a idéia da especificação apresentada acima é modelar toda a vida útil do processo, que fica parado a maior parte de seu tempo esperando por uma mensagem do tipo CO e quando recebe executa rapidamente as demais instruções e retorna ao estado de latência esperando pelo aparecimento do próximo CO. Cameron chama esse tipo de processo de "processo de longa duração" [3]. Posteriormente, durante a implementação, esse processo deve mudar, tornando-se um procedimento de um outro de mais alto nível, recebendo CO como um parâmetro, e a interação na realidade é a abstração de cada consulta efetuada pelo usuário.

A terceira função, geração de estatísticas semanais, difere das demais porque ela deve se comunicar com o processo de modelagem FITA-1 através de uma sequência de dados (Figura 6). Isso ocorre porque é FITA-1 quem toma a iniciativa de comunicar que uma fita foi retirada ou devolvida. Só ele possui essa informação. Ao receber o fluxo de dados ALU, o processo funcional PREPARAR-ESTATÍSTICA-SEMANAL acumula os totalizadores

apropriados. Quando, pelo mesmo canal, aparece a informação de que é um fim de semana (FS), a estatística é enviada para o mundo real através de uma mensagem de saída. É necessário acrescentar uma instrução de gravação do fluxo de dados ALU no texto estruturado do processo de modelagem FITA-1.

A intercalação de FS e ALU representada pela junção de suas setas indicam que o processo preparar Estatística Semanal recebe os dois fluxos por um mesmo canal de entrada, na ordem em que aparecerem. FS indica que os totais acumulados devem ser emitidos e reinicializados para os próximos ALUs. A diferença entre a comunicação por fluxo de dados, nesta função e por vetor de estado, na anterior, deve-se ao fato de que nesta, a decisão de enviar o fluxo está em FITA-1 ao passo que na anterior a decisão de consultar está no processo funcional.

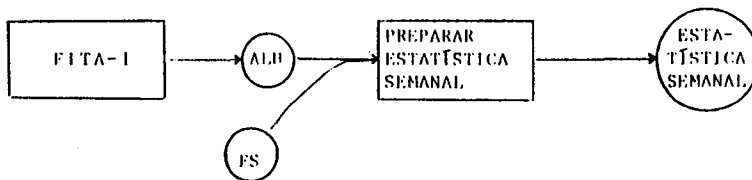


Figura 6

O Diagrama de Especificação do Sistema, mostrado na figura 7 reúne as três funções especificadas e adicionadas ao modelo inicial. Pode ocorrer que a conexão dos processos de modelagem com o mundo real seja feita através de vetor de estado. Isso ocorre quando o processo de modelagem apenas inspeciona as variáveis do mundo real, como no caso de um arquivo que já existe e é mantido por um outro Sistema de Informação.

#### 4. A FASE DE IMPLEMENTAÇÃO

O primeiro passo da fase de implementação é verificar a articulação temporal dos componentes do sistema, isto é, considerar mais cuidadosamente as restrições de tempo. As restrições especificadas devem ser documentadas como anotações complementares do Diagrama de Especificação do Sistema, sem nenhuma preocupação mais formal.

No exemplo apresentado, uma preocupação a ser considerada poderia ser a emissão das notas de aluguel, que deveria ocorrer num intervalo de tempo suficientemente pequeno para não irritar o membro do clube, mas como todos sabemos, essa restrição pode ser considerada como facilmente atendível por qualquer Sistema de Informação do porte deste. Uma preocupa

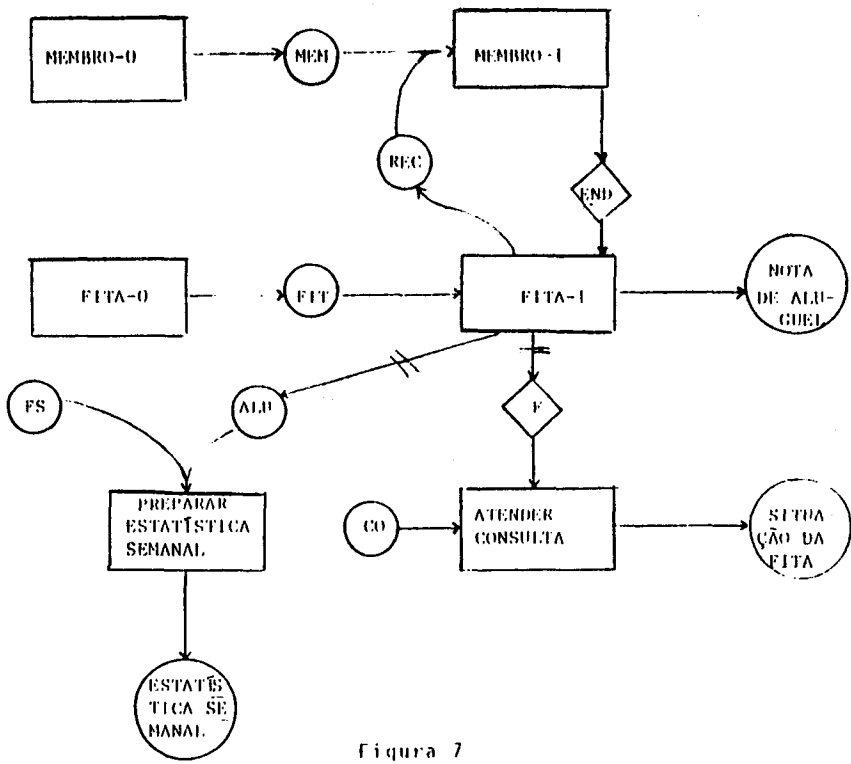


Figura 7

ção mais séria é causada pela emissão do relatório de estatísticas que pode demorar muito e bloquear o sistema por um tempo inaceitavelmente longo. Esse problema pode ser resolvido com a implementação desse processo como um programa isolado que roda no final de semana (processamento por lotes). Outras soluções mais sofisticadas poderiam ser imaginadas, como por exemplo a interrupção da execução desse processo sempre que ocorresse alguma outra ação.

A seguir devemos cuidar da transição da especificação para a implementação. Um sistema especificado segundo os princípios das fases 1 e 2, como o sistema de controle de fitas de videoclubes, poderia, teoricamente, ser executado diretamente. Para isso bastaria termos um micro para cada fita e para cada membro (instâncias das entidades MEMBRO e FITA), com uma cópia de cada processo executando nesses micros durante toda a existência de cada uma dessas instâncias.

A principal preocupação da implementação deve ser compartilhar os processos e os dados entre os processadores disponíveis (quase sempre um). Por isso, necessitamos de um programa para fazer a articulação entre o mundo real e o sistema, aceitando as entradas do mundo real e ati-

vando os processos apropriados, que dessa forma se transformam em subrotinas do programa articulador (ou escalonador).

Se tomarmos a decisão de implementar o sistema com o processamento imediato de todas as transações, com exceção do processo "PREPARAR ESTATISTICA SEMANAL", que seria executado isoladamente no final de cada semana, a especificação do programa articulador será aquela apresentada abaixo. A ativação de "PREPARAR ESTATISTICA SEMANAL" pode ser efetuada pelo operador, a partir de instruções escritas preparadas previamente, ou então a ativação pode ser feita pelo próprio sistema através de codificação na linguagem de comandos do sistema operacional.

```
ARTICULADOR seq
  Ler transação;
  CORPO-ARTICULADOR itr (até fim da execução)
    INVOCAÇÃO sel (transação=INSCREVER ou PAGAR)
      Ativar MEMBRO-1
    INVOCAÇÃO alt (transação=COMPRAR, SAIR, RETORNAR, ou
      DESCIAR)
      Ativar FITA-1;
    INVOCAÇÃO all (transação=CONSULTA)
      Ativar ATENDER-CONSULTA;
    INVOCAÇÃO fim
  Ler Transação;
  CORPO-ARTICULADOR fim
ARTICULADOR fim
```

O processo FITA-1 deve processar transações para muitas instâncias de fitas, por isso as variáveis relativas a esse processo devem ser separadas num arquivo de vetores de estado da entidade fita, escolhendo-se o método de acesso mais adequado. O processo FITA-1 é invertido, isto é, torna-se uma subrotina do articulador e a operação "Ler FIT" passa a receber a sua entrada através do articulador. A operação "Enviar REC" passa a gravar um registro num arquivo intermediário contendo detalhes dos aluguéis, para posterior processamento de lotes.

O processo MEMBRO-1 é transformado de forma semelhante ao processo FITA-1. O processo "ATENDER CONSULTA" também transforma-se numa subrotina do articulador, sem necessidade de criação de um arquivo de vetores de estado, porque a operação "ObterVE FITA-1" tem acesso direto ao arquivo de FITA-1. Por decisão de projeto, o articulador deve se encarregar de abrir e fechar, bem como realizar os acessos aos vetores de estado de MEMBRO-1 e FITA-1, passando-os aos processos respectivos, que trabalham em áreas da memória. Essas operações não aparecem no texto estruturado do articulador, mostrado acima. Uma outra alternativa seria a distribuição dos acessos e das operações de abertura e fechamento de arquivos de vetores de estado pelas subrotinas, evitando a passagem dos vetores de estado como parâmetro nas chamadas do articulador. Razões de eficiência devem ser consideradas na tomada dessa decisão.

O processo "PREPARAR ESTATÍSTICA SEMANAL" é processado isoladamente e a operação "ler ALU" é codificada como uma leitura do arquivo intermediário criado pelo processamento diário descrito acima. As operações de abertura e fechamento do arquivo intermediário são alocadas aos locais apropriados do programa. Como a impressão dos resultados só ocorre ao final do processamento de todos os registros, a indicação de fim de semana (FS) é simulada pela condição de fim do arquivo intermediário ALUGUEL. Para ser mais preciso, a condição de fim de arquivo é uma das formas de informar ao programa que não há mais nenhum fluxo de dados ALU para ser processado. O indicador de FS foi detectado, na realidade, pelo operador do sistema, ao colocar em execução o programa PREPARAR ESTATÍSTICA SEMANAL.

As decisões de implementação são documentadas na forma de um Diagrama de Implementação do Sistema (DIS). O DIS do exemplo discutido nesta seção é mostrado na figura 8. Nesse diagrama, a ativação das subrotinas é mostrada como linhas paralelas e o fluxo de dados de fora para dentro do sistema e vice-versa, bem como o acesso a arquivos, é mostrado por linhas simples com indicação de sentido.

O diagrama de Especificação do Sistema deve ser independente da implementação, de tal forma que se quiséssemos fazer uma implementação totalmente com processamento imediato (on-line), ele não mudaria em nada, apenas o Diagrama de Implementação mudaria: o procedimento para produzir as estatísticas semanais seria ativado pelo próprio articulador a partir do recebimento da entrada FS e seria ativado também por FIIA-1 sempre que ocorresse um aluguel.

## 5. CONSIDERAÇÕES FINAIS

O método JSD (Jackson System Development) parece ser uma alternativa interessante para o desenvolvimento de Sistemas de Informação. É um método que parte de conceitos próprios e geralmente não encontrados em outros métodos similares. Além disso, tem um certo rigor formal na especificação que, entretanto, não prejudica a sua praticidade.

O conjunto de diagramas e textos estruturados produzidos é conciso, mas de manutenção trabalhosa. Esse problema pode ser minimizado com o uso de ferramentas de apoio por computador. Acreditamos também que possa haver algumas dificuldades quanto ao treinamento de analistas de sistemas no uso do método, porque muitos conceitos tradicionais em processamento de dados recebem nomes diferentes em JSD (Exemplo: vetor de estado versus registro/parâmetro). Isso pode causar confusão e exigir um certo tempo para familiarização.

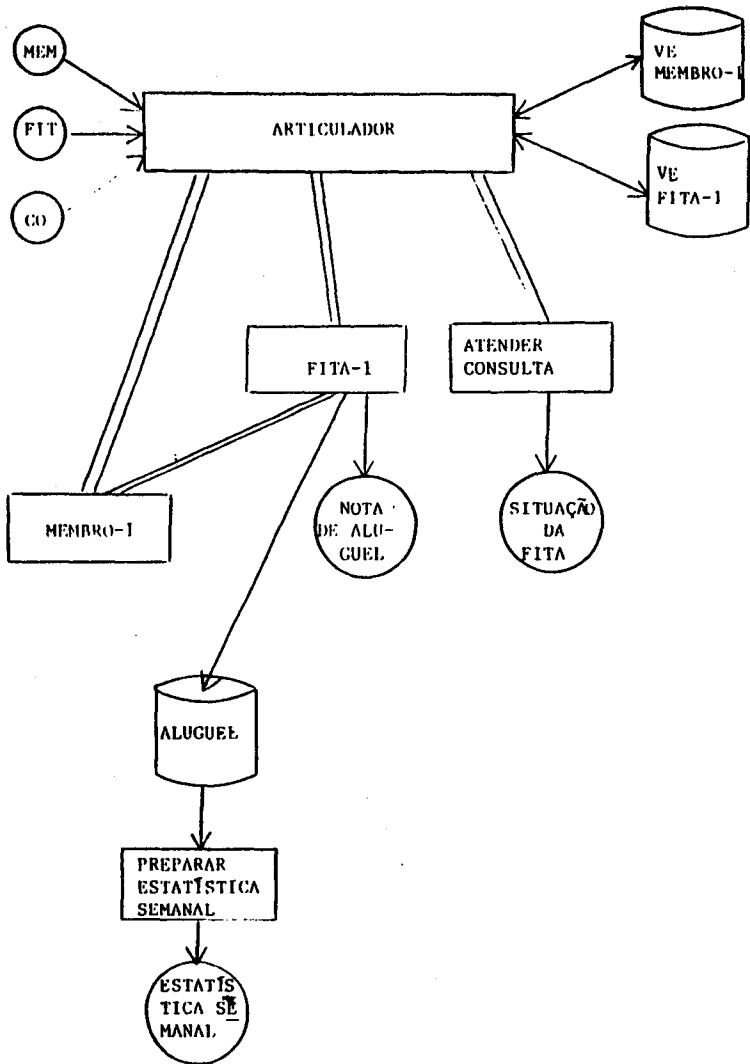


Figura 8



## **BIBLIOGRAFIA**

- [1] JACKSON, M. - System Development. Prentice Hall International, Inc., 1983.
- [2] JACKSON, M. - Principles of Program Design, Academic Press, 1975.
- [3] CAMERON, J.R. - An Overview of JSD, IEEE Transactions on Software Engineering, Vol. SE-12 (2), 1986.