METODOLOGIA PARA O DESENVOLVIMENTO DE UM SISTEMA DE AQUISIÇÃO DE DADOS E DE SUPERVISÃO

J.M. FARINES*, R.S. DE LEMOS**, L.F. BORGES DE ANDRADE***

SUMÁRIO

Neste artigo, é apresentada uma metodologia para o desenvolvimento da parte material e de programação de um sistema de aquisição de dados e supervisão, tomando como exemplo o desenvolvimento de um sistema de aquisição de dados para um gaseificador de madeira de uma usina piloto para produção de metanol a partir da madeira. As várias técnicas de análise dos requisitos, projeto e testes utilizadas durante este desenvolvimento serão mostradas e discutidas.

ABSTRACT

In this paper, is presented a metodology for development of hardware and software for data aquisition and supervision system. The various phases are described, based to an example of aquisition data system development for a wood gasification process to produce methanol. For these phases, corresponding techniques are shown and examined.

* Engenheiro (INPT-1972); Doutor-Engenheiro (INPT-1979); controle de processos, sistemas de aquisição e supervisão, software em tempo real, C.A.D. para sistemas de controle - Laboratório de Controle e Microinformática - Depto. Engenharia Elétrica - CP 476 - UFSC 88000 - Florianópolis - SC.

** Engenheiro (UFSC-1983); controle de processos, sistemas de aquisição e supervisão, hardware - Laboratório de Controle e Microinformática - Depto. Engenharia Elétrica - CP 476 - UFSC - 88000 - Florianópolis - SC.

*** Engenheiro (UFSC-1983); controle de processos, sistemas de aquisição e supervisão; Laboratório de Controle e Microinformática - Depto. Engenharia Elétrica - CP 476 - UFSC - 88000 - Florianópolis - SC.

INTRODUÇÃO

A diminuição dos custos da microinformática e das tecnologias adjacentes (em particular, para a transmissão de dados), a necessidade de uma melhoria na qualidade dos produtos e de uma maior economia de energia e a exigência de uma flexibilidade e de uma confiabilidade cada vez maiores levaram a uma crescente automatização dos processos industriais. Entre as atividades industriais automatizadas de maior importância hoje, se encontram a monitoração, a supervisão e o controle das plantas ou de parte delas.

O processo de monitoração, supervisão e controle segue, na maior parte dos casos, a filosofia da descentralização ou da distribuição das funções. Em consequência, se torna necessária a existência de unidades de aquisição de dados, supervisão e controle repartidas em diferentes pontos de uma planta ou de um sistema e em comunicação entre si. Outrossim, algumas destas unidades podem, quando utilizadas autonomamente, desempenhar também um papel importante na fase de análise, projeto e teste de uma planta piloto ou parte desta.

Neste trabalho, é apresentada uma metodologia para o desenvolvimento da parte material (hardware) e de programação (software) de um sistema de aquisição de dados e supervisão que deve funcionar autonomamente sobre uma planta piloto; entretanto, procurar-se-á, no decorrer do desenvolvimento deste sistema, levar em conta a sua futura introdução num sistema distribuído.

O desenvolvimento de um sistema de aquisição de dados e supervisão para um gaseificador de madeira instalado na usina piloto da CESP em Rio Claro servirá para exemplificar os vários aspectos desta metodologia. A figura 1 representa o diagrama-bloco deste gaseificador que permite a transformação da madeira num gás de síntese a ser utilizado posteriormente na produção de metanol. A gaseificação se faz pela passagem de uma corrente elevada entre os eletrodos e o processo é controlado pela injeção de vapor na zona de reação. Estudos de rendimento e análise do comportamento do gaseificador estão sendo realizados para melhor conhecer o processo e otimizar o projeto.

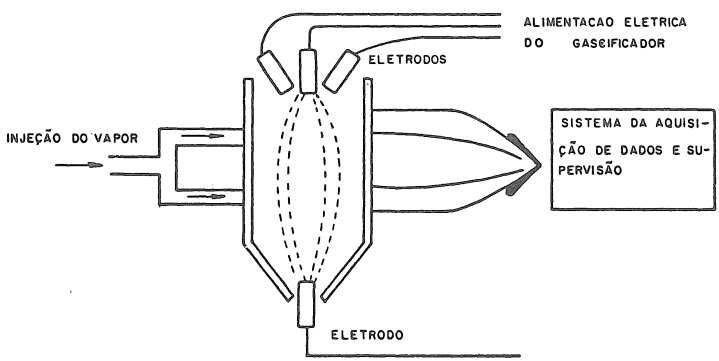


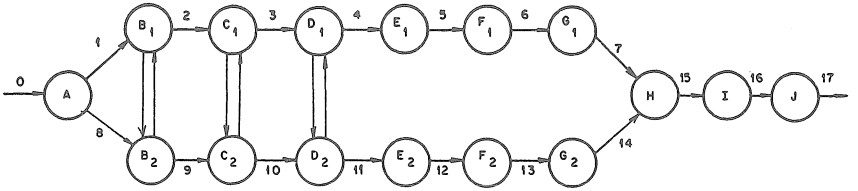
Figura 1: Diagrama-bloco da planta piloto

METODOLOGIA GERAL DE DESENVOLVIMENTO DO SISTEMA

A importância da escolha de uma metodologia para o desenvolvimento das partes material e de programação de um sistema se deve principalmente a melhor qualidade do produto final e da sua documentação, a racionalização das várias etapas do desenvolvimento e do gerenciamento destas e a diminuição dos custos de desenvolvimento e de manutenção decorrentes.

Na figura 2, são apresentadas as várias fases do desenvolvimento de um sistema e o seu encadeamento. As fases principais são as seguintes:

- Planejamento e especificações divididas em:
  - definição do sistema (A)
  - planejamento do hardware (B1) e do software (B2)
  - análise de requisitos do hardware (C1) e do software (C2)
- Desenvolvimento propriamente dito com as seguintes etapas:
  - projeto preliminar e detalhado do hardware (D1) e do software (D2)
  - implementação: montagem da parte material (E1) e codificação (E2)
  - teste de unidade e de integração do hardware (F1) e do software (F2)
  - teste de validação do hardware (G1) e do software (G2)
- Instalação do sistema:
  - "assemblagem" final do hardware e do software (H)
  - teste do sistema total (I)
  - instalação e teste de aceitação no local (J)
- Manutenção



pedido do usuário: 0
especificações informais do hardware (1) e do software (8)
plano de hardware (2) e de software (9)
especificações funcionais formais do hardware (3) e do software (10)
diagrama-bloco da parte material (4); módulos descritos em pseudo-código (11)
placas montadas com esquemas (5); programa codificado com documentação (12)
placas testadas (6); módulos de programa testados e integrados (13)
hardware validado (7) e software validado (14)
hardware e software "assemblados" (15)
sistema pronto para entrega (16)
sistema instalado e aceito (17)

Figura 2: Fases do desenvolvimento

Cada uma destas fases necessita das informações obtidas nas fases anteriores e transmitidas na forma de documentação. O desenvolvimento do sistema segue dois caminhos principais: o do hardware e o do software muitas vezes realizados independentemente. Entretanto, no caso de sistemas utilizando microcomputadores, é necessário a comunicação entre estes dois caminhos, principalmente nas etapas de planejamento e análise de requisitos ; isto permitirá diminuir a complexidade durante a fase de "assemblagem" final e evitar erros irreversíveis. As informações vindo de um caminho funcionam como restrições para o outro no decorrer de todo o desenvolvimento do sistema.

Revisões e procura de erro ("walkthrough") são realizadas no final de cada fase e em especial na fase de planejamento e especificações, com a participação do usuário.

DEFINIÇÃO DO SISTEMA

Nesta fase, foram estabelecidas as especificações informais do sistema em contato com o usuário, no caso os engenheiros da usina piloto da CESP. No final, foi elaborado um documento contendo:

- informações sobre o processo físico a supervisionar: gaseificador de madeira, neste caso;
- descrição do sistema atual de medição e supervisão e das suas limitações: no caso, não automatizado, lento, sujeito a erros de leitura, armazenamento em planilhas de difícil uso, etc.;
- especificações informais do sistema a ser desenvolvido: no caso um sistema microcomputadorizado de aquisição de dados com armazenamento em disco flexível, com capacidade de supervisão e monitoração em tela de diversas grandezas (temperaturas, potência elétrica, vazão de gás, etc.) e com alguns parâmetros pré-fixados (tempo de medição, precisão, etc.).

Nesta fase, foram também levantadas, as restrições de custo de investimento do usuário, as expectativas e o grau de experiência dos futuros utilizadores do sistema e os impactos da sua instalação.

Ainda nesta fase, a partir das especificações funcionais informais definidas pelo usuário e da análise das restrições impostas tanto do ponto de vista econômico quanto tecnológico e de recursos humanos, é feito o estudo da realizabilidade do sistema a partir do levantamento das alternativas de solução para o sistema. Uma destas será escolhida e proposta após este exame.

No caso do sistema proposto as restrições são relacionadas:

- ao econômico: custo limitado;
- à disponibilidade de recursos: 2 engenheiros e 2 técnicos de nível médio; sistema microcomputador para desenvolvimento sem as possibilidades gráficas previstas para o projeto;
- ao tecnológico: sistema operacional tipo CP/M; linguagem de fácil aprendizagem pelo u-

suário;

- aos requisitos operacionais: previsão de uma recuperação rápida do sistema no caso de falha na rede de alimentação; previsão de uma futura interligação com outros microcomputadores; interface homem/máquina simples; facilidades para a manutenção do hardware (sinalização), etc.

As especificações funcionais do sistema proposto como solução são contidas num documento descrevendo cada função numa linguagem natural e contendo também a alocação de cada uma das funções ao hardware ou ao software. As principais funções são enunciadas a seguir:

condicionamento dos sinais; filtragem; isolação; amplificação/redução; conversão analógica-digital; entrada de dados não automatizados pelo teclado; cálculo dos valores médios das medidas; armazenamento em disco; supervisão: alarmes na ultrapassagem dos limites de temperatura e de potência elétrica; visualização dos perfis de temperatura nos vários eixos do corpo do gaseificador e das outras medidas indicativas; interface com o usuário com possibilidade de configuração; relatório na impressora; calendário, etc.

PLANEJAMENTO DO HARDWARE E DO SOFTWARE

Ao visto das funções anteriormente alocadas ao hardware e ao software, faz-se uma estimação dos custos, dos recursos humanos envolvidos em cada uma das fases do desenvolvimento, dos recursos materiais e de programação, e do planejamento no tempo do desenvolvimento do sistema (hardware e software).

Para o sistema de aquisição de dados e supervisão, o planejamento no tempo do desenvolvimento do hardware e do software previu uma duração de 22 semanas para cada um; entretanto, o início do desenvolvimento do software ficou posterior ao do hardware por 12 semanas.

ANÁLISE DE REQUISITOS

A partir dos documentos anteriores, são analisados os requisitos do sistema tanto do ponto de vista do hardware como do software e é elaborado um documento contendo as especificações formais. Para tal, utilizam-se as técnicas de análise estruturada (Ross-77), (De Marco-79). O uso exclusivo de uma linguagem natural, nesta fase, é desaconselhado por gerar documentos que podem conter um detalhamento excessivo, ambiguidades, contradições, referencias antecipadas e características não expressas (Meyer-85). A representação gráfica sob forma de um diagrama de fluxo de informação, além de ser de fácil uso, permite evitar estes problemas. Neste diagrama, as bolhas representam os processos de transformação (ou as funções a serem realizadas), as setas indicam as informações de entrada e saída de cada função; os dispositivos externos são representados por caixas e o armazenamento de informações por dois traços paralelos.

Para o caso do sistema de aquisição de dados e supervisão citado, o diagrama de fluxo de informação de uma parte do hardware é representado na figura 3. O refinamento deste dia

grama de fluxo se faz por explosão das bolhas num novo diagrama de fluxo referenciado abaixo da bolha explodida; desta forma, as especificações estão sendo apresentadas utilizando uma hierarquização que facilitará posteriormente o projeto.

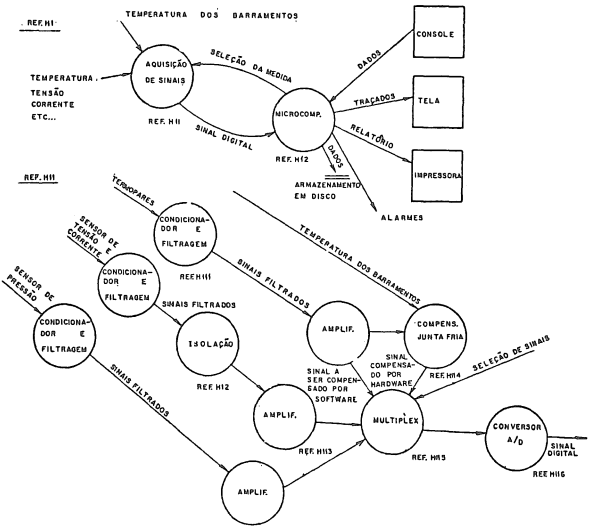


Figura 3: Diagrama de fluxo de informação para parte do hardware do sistema

Para o software, a técnica de análise de requisitos é a mesma; um exemplo desta está apresentado na figura 4.

Define-se também no caso do software, uma estrutura de dados e um dicionário de dados.No nosso exemplo, as grandezas a serem processadas, foram organizadas na forma de uma estrutura hierárquica com os campos temperatura, grandezas elétricas, pressão e características do gás que por sua vez foram também subdivididas; cada campo tem a sua forma própria de organização, por exemplo, no caso do sub-campo temperaturas do corpo do gaseificador segundo um espaço tridimensional (eixo, altura, tempo). Tal estrutura facilitou a proteção e o acesso aos dados a partir das partes do software que precisam deles.

O documento de especificações funcionais formais contém, além dos diagramas de fluxo e da estrutura de dados, uma descrição das várias funções contendo os seguintes ítens: objetivo; interfaces (entradas, saídas, funções em comunicação) e restrições. São definidas também as funções concorrentes e a ordem de prioridade destas.

No final desta fase, é realizada uma etapa de validação onde os vários estados do sistema e todos os caminhos de fluxo de informações são testados. No nosso caso, um diagrama

de estados foi suficiente para tal teste; em casos mais complexos, é necessário utilizar técnicas mais poderosas como a Rede de Petri, por exemplo. A forma estruturada desta fase de análise facilitou a escolha adequada dos módulos e a estruturação a nível de projeto e de implementação, como será apresentado a seguir.

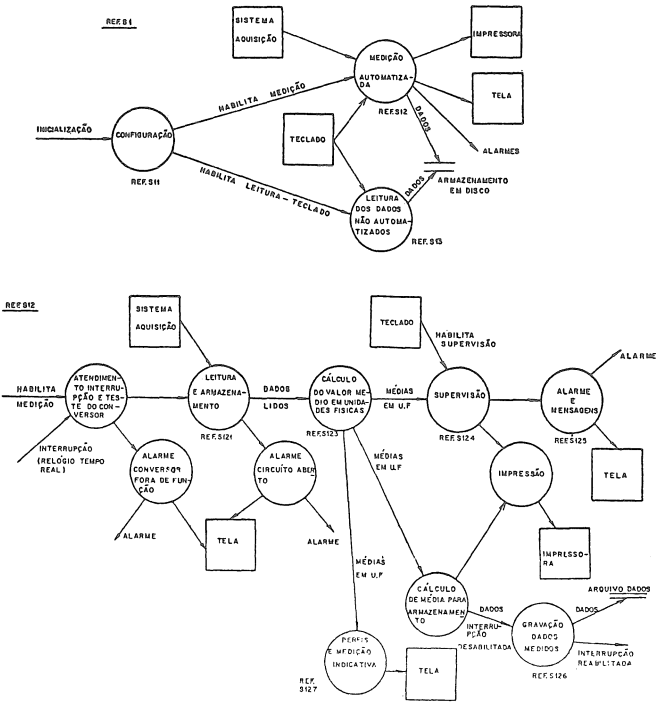


Figura 4: Diagrama de fluxo de informação para parte do software

PROJETO

1. Projeto Preliminar:

A partir dos resultados da fase anterior, os módulos e sub-módulos são definidos utilizando a técnica de projeto estruturado (Yourdon-79).

No projeto preliminar do hardware do sistema de aquisição de dados e supervisão, a partir dos diagramas de fluxo correspondentes e das restrições (como por exemplo, escolha de um barramento em função da futura interligação com outros micros, facilidades para a manutenção da parte material, etc.), procurou-se definir todos os módulos e sub-módulos da parte material. Para cada um destes, descreveu-se a função executada, os interfaces (entradas, saídas, módulos interligados), o procedimento de realização e as restrições;

estas informações são apresentadas na forma narrativa e na forma de uma representação gráfica onde cada bloco representa um módulo ou um sub-módulo.

No caso do software, o projeto preliminar baseado no diagrama de fluxo de informação visa obter uma estrutura hierárquica com as características de mais alta coesão e de mais baixo acoplamento possível dos módulos e sub-módulos.

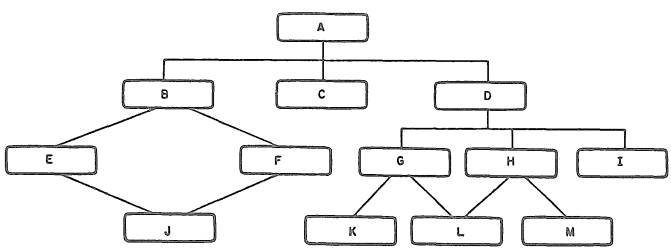
O primeiro passo do projeto consiste em detetar a existência de subsistemas concorrentes (i-e assíncronas) para o sistema descrito pelo diagrama de fluxo de informação. O projeto e a implementação de cada um deles serão feitos de forma independente, tendo sido atribuído também a cada um, um nível e prioridade. Por exemplo, os subsistemas "leitura dos dados não automatizados" e "medição automatizada" (figura 4) são processos concorrentes; atribui-se a maior prioridade ao subsistema "medição automatizada".

A seguir, cada subsistema é decomposto a partir do diagrama de fluxo em módulos e sub-módulos com as características já citadas, procurando também evitar que as informações de um módulo sejam acessíveis a outro, quando não for necessário (Parnas-72), isto é, minimizando o uso de variáveis globais (acessíveis em qualquer lugar do programa).

No final desta decomposição, se obtém uma estrutura hierárquica para cada subsistema. A seguir é realizada a revisão da estrutura do ponto de vista da minimização do acoplamento, do acréscimo da coesão, de possíveis redundâncias e da estruturação dos módulos entre si (fan-in, fan-out).

Cada módulo será também descrito num documento contendo: nome do módulo; objetivo deste; interfaces (entradas, saídas, módulos em comunicação: superordinados e subordinados, forma de chamada); dados globais acessados; estrutura de dados do módulo; algoritmo a ser implementado neste módulo; eventuais restrições.

No caso do sistema de aquisição de dados e supervisão, o subsistema "medição automatizada" tem a estrutura básica apresentada na figura 5 que poderá ser posteriormente refinada.



medição automatizada: A
medição das grandezas físicas: B
cálculo dos valores médios e transformações em unidades físicas: C
supervisão, monitoração e armazenamento: D

leitura do conversor e armazenamento na memória: E
alarme do conversor fora de função: F
atendimento de interrupção e teste do conversor: J
supervisão: G
cálculo de média para relatório e armazenamento: H
traçado dos perfis na tela: I
alarme e mensagem de ultrapassagem na tela: K
impressão: L
gravação de dados: M

Figura 5: Estrutura hierárquica do subsistema "medição automatizada"

2. Projeto Detalhado:

Em seguida, cada módulo de hardware e de software é detalhado por refinamentos sucessivos. No caso do hardware, cada bloco anteriormente definido é refinado introduzindo uma representação simbológica de cada um deles, baseada nas operações eletrônicas básicas tais como integração, comparação, soma, amplificação, etc. No caso do software, o refinamento consiste em reescrever a descrição do módulo feita na etapa anterior até chegar a um pseudo-código que utiliza ao mesmo tempo a sintaxe da programação estruturada com as palavras-chaves (IF-THEN-ELSE; DO WHILE; CASE OF; DECLARE; etc.) e a sintaxe e vocabulário da lingua natural.

IMPLEMENTAÇÃO: MONTAGEM E CODIFICAÇÃO

Nesta fase, são escolhidas as formas práticas de implementação dos módulos de hardware e software anteriormente detalhados; esta fase se caracteriza pela transformação dos blocos de hardware em esquemas contendo o detalhamento de todos os circuitos eletrônicos e na montagem posterior destes; no caso do software, ela corresponde a escrita do código correspondente à pseudo-linguagem da fase anterior.

A escolha da linguagem utilizada é função tanto das possibilidades oferecidas em termos de desenvolvimento (sistema operacional, compiladores, etc.) como das suas características (clareza, estruturação, compactação do código, tipo de entrada-saída, etc.) e das limitações impostas na definição do sistema (memória e tempo de cálculo, código otimizado, possibilidade de fácil modificação pelo usuário e outras exigências). Para o sistema de aquisição de dados e supervisão proposto, alguns módulos foram codificados em Assembly para cumprir com os requisitos de tempo do sistema (módulos ligados a aquisição de dados e ao traçado de gráficos na tela); os outros módulos foram escritos em Basic. O Basic foi escolhido por ser a única linguagem conhecida pelo usuário, permitindo assim uma evolução rápida do programa pela introdução de novos módulos por este. Entretanto as limitações conhecidas desta linguagem em termos de clareza, compactação, tempo de execução levaram a elaborar uma outra versão numa linguagem estruturada, Pascal.

TESTES

1. Testes de Unidade e Integração:

Esta fase deve seguir um plano detalhado elaborado nas fases anteriores (em particular na análise de requisitos e no projeto preliminar); este indicará a sequência de testes, o tipo destes e a forma de integração dos módulos.

No caso do hardware, cada bloco é testado separadamente, levando em conta principalmente critérios de precisão, imunidade ao ruído e faixa passante; em seguida integra-se vários destes blocos para realizar funções maiores.

No caso do software, cada subsistema concorrente será testado independentemente, módulo por módulo, fazendo no final a integração dos mesmos. Para cada módulo, verificar-se-ão as interfaces, a lógica do módulo (percorrendo todos os caminhos condicionais possíveis) e as condições limites destes. O teste de integração realizado a seguir tem dois objetivos:

- testar a estrutura principal do software e a comunicação entre os módulos, por um teste de cima para baixo onde os módulos são representados unicamente pelas suas interfaces;
- integrar os módulos de baixo nível para realizar a função definida por um módulo de nível superior.

Por exemplo, foram testados separadamente os módulos E e F da figura 5 (leitura do conversor e alarme do conversor fora de função) para serem integrados em seguida realizando a função medição (módulo B da figura 5).

No decorrer desta fase, são registrados em documento os resultados dos testes anteriormente planejados; este será de grande utilidade na fase de manutenção do hardware e do software.

2. Teste de Validação:

A validação da parte de hardware e de software é feita de forma independente verificando se os requisitos de cada uma delas são cumpridos, isto é, se todas as funções previstas são realizadas com as características desejadas (como por exemplo, a precisão no caso do hardware).

"ASSEMBLAGEM" E TESTE DO SISTEMA TOTAL

Os frequentes contatos efetuados no decorrer das várias fases do desenvolvimento de hardware e de software facilitaram a "assemblagem". Nesta fase, serão testadas principalmente as partes do sistema que dependem ao mesmo tempo do hardware e do software como por exemplo a parte do sistema que realiza a aquisição de dados e a que permite a leitura a partir do teclado dos dados não automatizados.

Uma vez a "assemblagem" acabada, é realizado o teste final do sistema: em laboratório e

em seguida no próprio local da instalação final deste. Neste teste são verificados o cumprimento de todos os requisitos funcionais do sistema. a compatibilidade deste com a planta. o desempenho deste em operação e a conformidade da documentação que acompanha o sistema. Uma vez esta fase de testes terminada, o sistema é considerado aceito e pode ser utilizado normalmente pelo usuário.

CONCLUSÃO

Neste trabalho foi apresentada uma metodologia para o desenvolvimento de um sistema de aquisição de dados e supervisão. Ela foi testada com sucesso num sistema a ser instalado num gaseificador de madeira de uma usina piloto de produção de metanol.

Esta metodologia contribui para se obter um produto final de melhor qualidade, mais confiável e melhor documentado e em consequência com maiores possibilidades de modificação e evolução. Constatou-se também algumas dificuldades na aplicação desta metodologia que deverão ser levadas em conta em trabalhos futuros. A fase de definição do sistema deverá ser mais aprofundada, apesar das dificuldades de comunicação com o usuário. Os possíveis atrasos no fornecimento de material deverão ser melhor avaliados e levados em conta no planejamento no tempo. Em caso da existência de um grande número de subsistemas concorrentes, é necessário utilizar as técnicas de análise e validação apropriadas tais como por exemplo as Redes de Petri.

Deve-se notar que o uso da estruturação a nível de análise, projeto e implementação melhorou sensivelmente a qualidade do produto e o tempo envolvido no desenvolvimento.

Um fator de melhoria na "assemblagem" do sistema deveu-se ao fato que, por razões de escassez de recursos humanos, os projetistas de hardware tiveram também envolvidos no trabalho de desenvolvimento do software e vice-versa. Entretanto, em caso de desenvolvimento de sistemas maiores, tal possibilidade desaparece e só pode ser substituída por contatos regulares e organizados entre as duas equipes: de hardware e de software.

BIBLIOGRAFIA

- DE MARCO T. - "Structured Analysis and System Specification" - Prentice-Hall, 1979.
- GOMAA H., LUI J., WOO P. - "The Software Engineering of a Microcomputer Application System" - Software: Practice and experience, vol.12, pp. 309-321, 1982.
- MEYER B. - "On Formalism in Specifications" - IEEE Software, vol.2, nº1, pp.6-36, jan. 1985
- NETO J.J. - "Introdução a Engenharia de Software e Metodologia de Programação" - 1º CONAI - Tutorais, pp. 119-146, jul. 1983.
- PARNAS D.L. - "On the Criteria to be used in Decomposing a System into Modules" - Communications ACM, pp. 1053-1058, dez. 1972.
- PRESSMAN R.S. - "Software Engineering: A Practitioner's Approach" - Mac-Graw-Hill-1982
- ROSS D., SCHOMAN K. - "Structured Analysis for Requirements Definition" - IEEE Trans. on Software Engineering, vol.3, nº1, jan.1977, pp.6-15.
- YOURDON E., CONSTANTINE L. - "Structured Design" - Prentice-Hall, 1979.