

**2° CONGRESSO
NACIONAL DE
AUTOMAÇÃO
INDUSTRIAL**
CONAI

SESSÕES TECNOLÓGICAS

ANAIIS

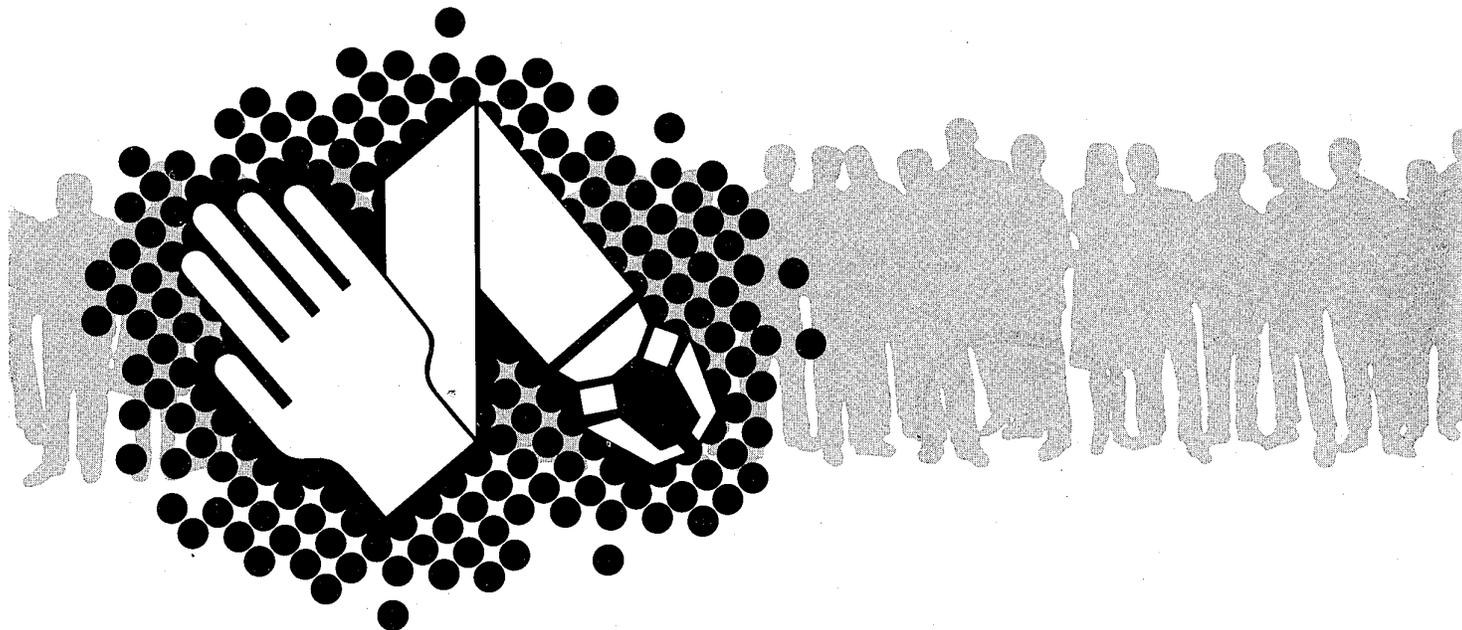
338.06
C749

TEMA
AUTOMAÇÃO E SOCIEDADE

REALIZAÇÃO SUCESU S.P.

PALÁCIO DE CONVENÇÕES DO ANHEMBI São Paulo SP

25 a 29 de NOVEMBRO 1985



2° CONGRESSO NACIONAL DE AUTOMAÇÃO INDUSTRIAL

CONAI

SESSÕES TECNOLÓGICAS

ANAIIS

EDITORES

ARYLDO GENTIL RUSSO
MOACIR MARTUCCI JR.

TEMA

AUTOMAÇÃO E SOCIEDADE

REALIZAÇÃO SUCESU S.P.

PALÁCIO DE CONVENÇÕES DO ANHEMBI São Paulo SP

25 a 29 de NOVEMBRO 1985

MOSAICO-E : Um Ambiente Interativo para o Projeto Modular de Programas.

Arndt von Staa
Credinê Silva de Menezes

Pontificia Universidade Católica do Rio de Janeiro
Departamento de Informática
Rua Marques de São Vicente, 225
CEP 22453 Rio de Janeiro - RJ

RESUMO: neste trabalho discute-se os problemas encontrados durante o projeto modular de programas e apresenta-se um sistema de apoio a esta atividade. Mosaico-E, ora em desenvolvimento na PUC-RJ, é um ambiente interativo que permite a edição de estruturas modulares em linguagem gráfica, além de permitir a especificação detalhada de cada módulo.

1. INTRODUÇÃO

Durante o desenvolvimento de um software, são criadas, alteradas, traduzidas e observadas diversas representações[1]. Uma representação é qualquer documento gerado durante as várias fases de desenvolvimento do sistema, por exemplo, especificação de requisitos, especificação funcional, dicionário de dados, diagrama de estruturas modulares, etc. Estando o sistema em funcionamento, faz-se necessário dar-lhe manutenção (corretiva ou evolutiva), e novamente suas representações voltam a ser manipuladas.

É usual hoje em dia, este conjunto de operações ser realizado com pouco controle global. É usual, ainda, controlar-se somente a correção sintática das representações, sendo que frequentemente, somente o código fonte é verificado.

Apesar dos esforços empreendidos pelas comunidades Acadêmica e Industrial, a crise do software[2] continua a se agravar. Dentre as causas fomentadoras desta crise, destacam-se:

- a) carência de metodologias que abordem o processo de desenvolvimento e manutenção de sistemas como um todo. Dificilmente se encontra um conjunto integrado de métodos que suporte derivação sistemática;

- b) a demanda por software quer em quantidade, quer em complexidade, continua a crescer;

- c) a escassez de ferramentas de apoio ao desenvolvimento e manutenção de sistemas. As poucas encontradas são, em geral dirigidas a fases específicas do projeto, gerando com isso representações isoladas, as quais se desatualizam com o decorrer do tempo, tornando-se pois inúteis.

Segundo [3], ao desenvolvermos um sistema, usamos diversas linguagens de representação as quais formam uma família de linguagens. Assim, usamos linguagens diferentes para: descrever o problema, especificar requisitos funcionais, especificar requisitos de qualidade, especificar em vários níveis o projeto e implementar o sistema. Tendo em vista que esse processo não é exclusivamente "top-down", o projetista manipula simultaneamente várias representações, em linguagens distintas. Como não há integração entre elas, essas representações são deterioradas ao longo do desenvolvimento e manutenção do sistema, levando a uma perda de qualidade.

Como resposta a essas necessidades, desenvolve-se hoje na PUC-RJ, um ambiente de desenvolvimento de software (projeto Mosaico)[4], o qual tem como objetivo proporcionar aos desenvolvedores um ambiente no qual

todo o processo de desenvolvimento e manutenção seja tratado de forma integrada, com um apoio maciço à manipulação integrada do conjunto de representações. Tal ambiente baseia-se na constatação de que um sistema é formado por suas representações, e que estas são fortemente interrelacionadas[1]. Um tal é essencialmente um sistema CAD/CAM voltado para a produção de software.

Projetos dessa envergadura, obviamente, demandam muito tempo. Sendo assim o projeto Mosaico será desenvolvido em várias etapas. A primeira delas, em andamento, corresponde à construção, de Mosaico-Estruturado, que vem a ser um ambiente para projetar programas através de modularização.

2. ESCOPO DE MOSAICO-E

Constata-se através da bibliografia específica da área [5, 6, 7, 8], que as principais metodologias para desenvolvimento de programas, hoje utilizadas, fazem uso maciço de linguagens gráficas. Nestas, a organização do projeto e de suas unidades são representadas através de figuras interconectadas, as quais são posteriormente detalhadas.

O detalhamento do projeto faz uso de outras linguagens complementares, tais como dicionário de dados, descrição de interfaces, descrição de assertivas, pseudo-código, código fonte, e outras, além da própria linguagem gráfica. O conjunto de representações gerado através destas linguagens, deve ser mantido consistente durante toda a vida do sistema, o que em geral é impraticável com o uso de recursos convencionais (lâpis e papel).

Complementando o quadro acima, observa-se que algumas dessas derivações são sistemáticas, sendo passíveis portanto de tradução automática ou semi-automática assistida pelo projetista. Dentre estas podemos destacar: linearização da organização para gerar o esqueleto do pseudo-código, geração de código fonte em uma linguagem selecionada, derivação de dados para teste. A seguir apresenta-se uma pormenorização desses aspectos e indica-se que forma eles são abordados em Mosaico-E.

MOSAICO-E tem por objetivo simplificar a elaboração de sistemas, através de apoio à construção de organizações modulares. Este apoio compreende a eliminação das tarefas

mecânicas de registro da atividade criativa e apoio automatizado às transformações de representações e geração de casos teste. O resto desta seção é destinado ao esclarecimento dessas formas de apoio.

2.1 - Projeto Modular

O desenvolvimento do projeto modular de um programa, é uma tarefa experimental. Nesta, o projetista parte de uma arquitetura inicial, e, a partir de sucessivas expansões e modificações tenta obter um arquitetura satisfatória para resolver o problema. Durante este processo criativo é natural ocorrerem arquiteturas satisfatórias alternativas, as quais devem ser registradas para uma posterior avaliação, quando uma delas, ou até mesmo uma combinação destas será adotada como solução.

O uso de recursos inadequados (lâpis, papel, etc) tem sido um fator inibidor dessa atividade. A necessidade de redesenhar a estrutura do programa a cada nova alteração, desencoraja a prática da experimentação e reflexão, tendo como consequência a produção de programas de insuficiente qualidade.

MOSAICO-E proporciona ao projetista um ambiente interativo para criação, edição e armazenamento de estruturas modulares. Nele, as diversas estruturas alternativas obtidas pelo projetista podem ser registradas a parte, para usos posteriores, a critério do usuário.

Estruturas modulares em MOSAICO-E são criadas e editadas usando uma linguagem gráfica, que é sem dúvida alguma a maneira mais natural de registrar esse tipo de projeto. A linguagem gráfica adotada é semelhante à descrita em [7], na qual se restringe a derivação de filhos de um dado nodo de acordo com as regras de composição descritas no trabalho de Jackson[8]. Na figura 1, apresenta-se um resumo da citada linguagem.

Essa linguagem apresenta como característica uma versatilidade que nos permite seu uso para descrição de abstração de operações (programas e subprogramas)[7], estruturas lógicas de dados[8], abstração de dados[10] e conjunto de objetos correlatos (bibliotecas, "information hiding"[9]).

2.2 - Detalhamento do Projeto

Para complementar o Projeto de um programa, faz-se necessário adi-

cionar detalhes estruturais e individuais à arquitetura modular.

Detalhes estruturais são compreendidos pelo estabelecimento da comunicação entre os componentes do projeto. Para tanto o projetista estabelece quais os dados que serão fornecidos pelo módulo chamador ao chamado (parâmetros de entrada) e quais os que serão devolvidos pelo chamado ao chamador (parâmetros de saída).

Já no detalhamento individual, o projetista estabelece uma descrição rigorosa de cada componente da estrutura. Esta descrição é em geral composta dos seguintes aspectos:

- natureza do serviço - descrição informal da atividade realizada pelo componente.
- elementos de entrada - identificação de variáveis, arquivos e estruturas de dados a serem considerados como entrada para a execução do componente.
- observação de resultados - identificação de todos os arquivos, variáveis e estruturas de dados, os quais terão seus estados modificados pela execução do componente.
- condições iniciais - descrição informal das restrições impostas aos valores dos dados de entrada, arquivos e estruturas de dados, assim como da relação entre eles, para que o componente execute a sua tarefa.
- condições finais - descrição informal das restrições a serem observadas nos dados de saída, arquivos e estruturas de dados, quando do término da execução, para que o componente seja considerado correto.
- descrição comportamental - descrição algorítmica, em pseudo linguagem, das transformações a serem realizadas pelo componente, quando submetido à execução.

O conjunto de detalhamentos acima descrito, é suportado por MOSAICO-E através de editores específicos, os quais funcionam também de forma interativa. Tendo em vista a flexibilidade com a qual o sistema está sendo construído, permitirá sem maiores esforços a incorporação de novos itens de descrição.

2.3 - Apoio Automatizado

O projeto físico de um programa é obtido a partir da sua estrutura lógica. Nesta fase, o projetista seleciona o conjunto de módulos correlacionados, para comporem as unidades físicas de programação. Aqui também se verifica um grande esforço experimental, onde o apoio adequado possibilitará e encorajará a criação de diferentes combinações até que se atinja a ideal. MOSAICO-E proporciona ao projetista um modo ameno de declarar que módulos lógicos farão parte de um módulo físico e solicitar a derivação da estrutura física correspondente. De forma inversa, dada uma estrutura física podemos fazer desempacotamentos e obter a estrutura lógica correspondente.

A seleção do conjunto adequado de casos teste, é fator decisivo na garantia da qualidade do programa obtido. Sem o devido apoio automatizado, esta seleção em geral é incompleta, enfadonha e dispendiosa. A existência de apoio automatizado é portanto altamente desejável. Seguindo a metodologia proposta em [11] para geração de casos teste, MOSAICO-E auxilia o projetista a planejar o conjunto de dados de teste a partir de uma análise da estrutura modular do programa.

Estruturas modulares obtidas usando derivação descendente, são caracterizadas por ausência de transformações de dados nos níveis superiores. Nestas, toda a transformação de dados é atribuída aos módulos que não possuem derivação, cabendo aos demais a tarefa de gerenciar a sequência de ativações. Usando a estrutura modular e o dicionário de dados para projetos deste tipo, MOSAICO-E produz a linearização em pseudo-código. Uma versão futura, permitirá a tradução semi-automática do pseudo-código para diferentes linguagens de programação à escolha do usuário.

3. ESTRUTURAS ORGANIZACIONAIS

O registro e manipulação dos componentes de um projeto requer a existência de uma base de software que permita adições, modificações e recuperações de forma controlada.

Para suportar o projeto MOSAICO (geral) e em particular o MOSAICO-E, optou-se pela criação de um modelo conceitual flexível, a onde se possa descrever diferentes tipos de linguagem de representação.

Em MOSAICO-E, uma estrutura organizacional é um conjunto de nodos (de tipos variados) interconectados, em forma de grafo dirigido, onde se admite mais de uma conexão entre um par de nodos. Cada nodo de uma estrutura pode possuir refinamentos através de uma outra estrutura.

Cada linguagem define que tipos de nodos são admitidos e as restrições a serem observadas nas interconexões. Observe-se que isto permite definir uma linguagem L por composição de diferentes sublinguagens Li's, da seguinte forma: uma linguagem Li define que tipo de nodos e interconexões compõem a estrutura principal. Como esta usa diferentes tipos de nodos que podem admitir refinamentos, não obrigatoriamente seguindo Li, podemos ter então várias linguagens L2, L3, etc para definir como pode ser refinado cada nodo.

A adoção do modelo acima descrito permite a criação de um ambiente adaptativo onde teremos uma metalinguagem que permitirá ao usuário descrever sua linguagem específica. Em MOSAICO-E, teremos uma versão preliminar ajustada especificamente para a linguagem apresentada na seção 2.

A forma de exibição das linguagens acima referidas, ficará condicionada à existência de editores específicos, pois uma solução de edição gráfica geral é inadequada. MOSAICO-E possuirá um editor específico provendo a visualização gráfica de estruturas modulares, e operando sob controle de uma linguagem de comando amarrada a essa linguagem de representação.

É importante observar que a modelagem obtida é mais geral que os requisitos de MOSAICO-E e serve de base para versões posteriores. Desta forma, além das estruturas modulares que é nosso objetivo no momento, este modelo de dados permitirá a descrição e processamento de outras estruturas organizacionais, tais como: Diagramas de Entidades e Relacionamentos, Diagrama de Fluxo de Dados, Rede de Petri, etc.

4. CARACTERÍSTICAS DO SISTEMA

MOSAICO-E é um ambiente interativo para a criação e manipulação de estruturas organizacionais em um ambiente agradável ao uso. Para atingir este objetivo, MOSAICO-E é dotado das seguintes características:

- i- possibilita a edição de estruturas organizacionais, o que compreende:
 - a. criar e manter a estrutura geral;
 - b. detalhar cada nodo da estrutura através de:
 - b1. explosão em outra estrutura;
 - b2. definição da interface de um módulo com os demais módulos ativadores;
 - b3. descrição da realização através de textos (por exemplo, pseudo-código, código, texto descritivo, relação de operações associadas[8]);
 - b4. definição de assertivas de entrada e saída;
 - b5. definição do tipo de controle (sequência, seleção iteração).
- ii- registra e recupera todos os fatos de um projeto lógico, utilizando para isso uma "Base de Software"[11] apoiada no modelo de dados previsto na seção 3.
- iii- oferece uma interface amena ao usuário, que permite:
 - a. navegar em uma estrutura organizacional através de teclas de movimentação de cursor ou posicionamento em listas de componentes que satisfaçam uma dada condição;
 - b. explorar um módulo através do acesso aos seus diversos componentes (explosão ou descrição);
 - c. editar todos os tipos de representação de uma estrutura organizacional.
- iv- produzir, quando solicitado, a impressão de representações selecionadas, usando impressora adequada.

5. PARADIGMA DE ARMAZENAMENTO

Com o objetivo de facilitar o manuseio das representações de um projeto, em ambiente de microcomputador, devemos ter em conta que este tipo de equipamento, em geral, faz uso de memória secundária de baixa capacidade de armazenamento (diskette de 5 1/4"). Em resposta a isto, MOSAICO-E usa o seguinte paradigma:

- 1- cada unidade de projeto usa um caderno de folhas eletrônicas, o qual se realiza através de um arquivo do sistema operacional;
- 2- cada representação de um projeto usa uma folha eletrônica de tamanho ilimitado; desta forma temos a noção de tipo de folha;
- 3- um projeto pode usar um ou mais cadernos, dependendo do seu número de unidades de projeto;
- 4- é permitido, em um caderno, fazer referências a folhas de um outro caderno;
- 5- existem cadernos especiais, contendo unidades de uso geral;
- 6- para garantir um bom desempenho na manipulação de representações, MOSAICO-E implementa o conceito de memória virtual;
- 7- o armazenamento e recuperação de informações baseado em valores de atributos, será realizado através de uma implementação do conceito B-TREE+.

Os fatos contidos em uma folha são tornados visíveis através de janelas. A janela visualiza um volume limitado de fatos, onde esta limitação depende das características físicas do equipamento de visualização (vídeo). Através de teclas de posicionamento o usuário pode posicionar estas janelas em qualquer lugar sobre a folha eletrônica.

6. EPILOGO

Neste trabalho discutiu-se a natureza do processo de projetar programas utilizando a metodologia de modularização. Em particular, abordaram-se os problemas com os quais o projetista de sistemas se defronta quando utiliza recursos convencionais para criação e registro de estruturas alternativas. Neste contexto, descreveram-se as características de uma ferramenta de apoio a esse processo, que ora desenvolvemos na FUC/RJ.

A construção de ferramentas para apoiar o projeto de sistemas nas engenharias tradicionais, já é de grande difusão em nossos dias. Em engenharia de software, por ser uma disciplina em formação, ainda existem poucas ferramentas. A razão principal é o alto investimento e o tempo necessário para construir um software desta natureza, em contraste com o alto grau de instabilidade das metodologias em vigor o que torna estas ferramentas obsoletas em pouco tempo.

Para contornar este problema, o projeto que ora desenvolvemos enfatiza flexibilidade. Esta característica, que é central em MOSAICO-E, possibilitará a evolução da ferramenta de acordo com a evolução das metodologias. O ponto chave do projeto está na concepção de sua base de software, discutida na seção 4. Esta permitirá a especificação natural de outras linguagens, assim como a evolução de uma já existente.

As diversas fases do processo de desenvolvimento caracterizam-se por realizarem transformações sobre as representações geradas na fase anterior. Algumas dessas transformações são totalmente automatizáveis e outras necessitam da participação do projetista, sendo no entanto passíveis de apoio automatizado. Construindo ferramentas que levem esses aspectos em consideração evita-se a necessidade de duplo registro de informações, fato este que como se sabe tende a provocar deturpações. Em Mosaico-E isso explorado através da geração de casos testes e linearização de pseudo-código, através de ferramentas que atuam sobre a base de software do projeto, a qual garante a integridade das representações.

7. REFERENCIAS BIBLIOGRAFICAS

[1] Freeman, P. ; Staa, A.v. - Towards a Theory of Software Engineering, submetido para publicação, 1984.

[2] Boehm, B. W. -Software and Its Impact: a quantitative assessment; in Datamation, 1973.

[3] Staa, A.v ; Freeman, P. - requisitos de linguagens de especificação, submetido para publicação, 1984.

[4] Staa, A. v. - Projeto Mosaico, 1985.

[5] Gane, C. ; Sarson, T. - Structured Analysis

[6] De Marco, T. - Structured Analysis and System Specification, Prentice-Hall, 1979.

[7] Yourdon, E.; Constantine, L. - Structured Design, Prentice-Hall, 1979.

[8] Jackson, M. - Principles of Program Design, Academic-Press, 1975.

[9] Parnas, D.L. - "On The Criterion to Be Used in Decomposing Systems in Modules", in CACM, dec/72, pp 1053-1058.

[10] Booch, G. - "Object Oriented Design", in Tutorial on Software Design Techniques, 4th edition, 1983.

[11] Staa, A. v. - Um critério de Seleção de Casos Teste, in Anais do XII SEMISH, julho/85.

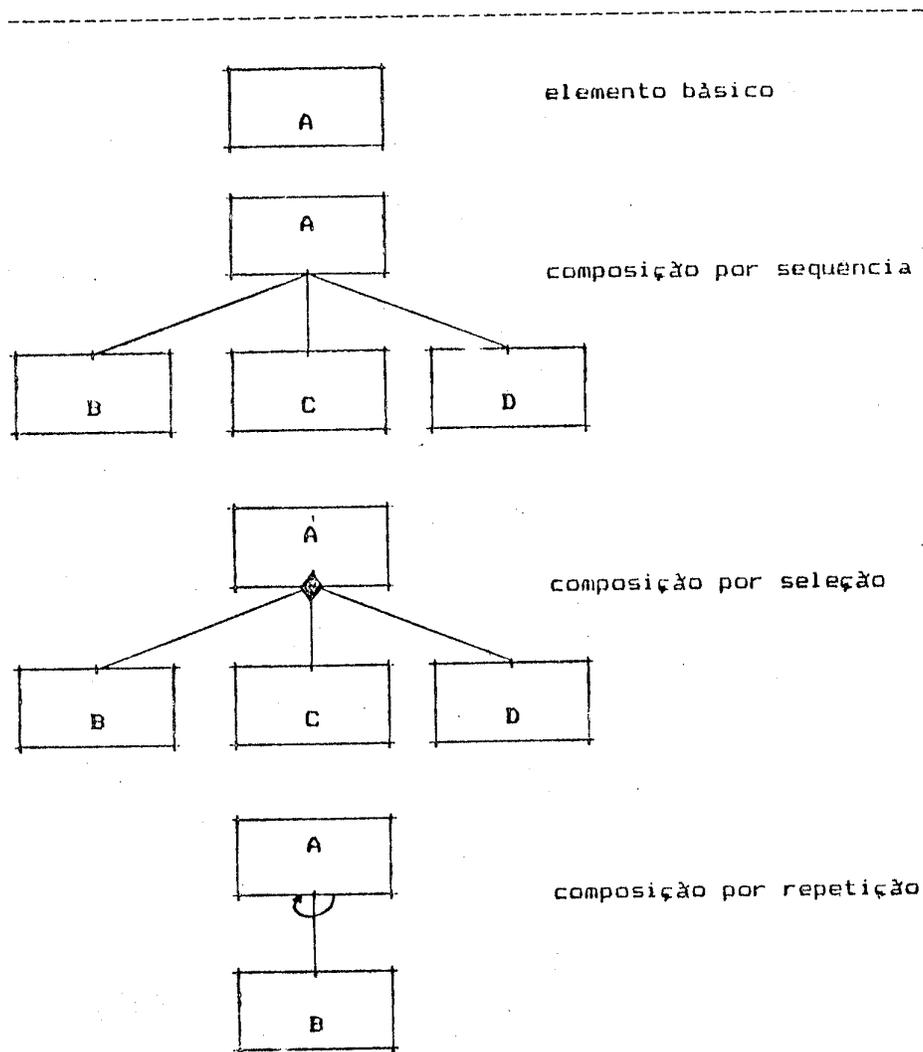


FIG 1 - Linguagem Gráfica para Estruturas Modulares
