

SAES - Sistema de Apoio à Especificação de Sistemas

Carlos Miguel Tobar Toledo

IC - Centro Tecnológico para Informática
CP 61662, 13081 Campinas - SP - Brasil

Claudia Bauzer Medeiros

DCC - IMECC - UNICAMP
CP 6065, 13081 Campinas - SP - Brasil

RESUMO

O trabalho apresenta um protótipo de um ambiente para suportar a especificação automatizada de sistemas de software. O ambiente está baseado no método ANA-RE, desenvolvido a partir do SADT, de forma a estendê-lo e simplificá-lo. O protótipo implementado suporta a definição e manipulação de diagramas SADT e ANA-RE sob forma textual.

ABSTRACT

This paper presents a prototype of a software specification environment. The environment is based on the method ANA-RE, developed from SADT, so as to extend and to simplify its capabilities. The prototype allows the automatic definition and updating of SADT and ANA-RE diagrams in textual form.

1. INTRODUÇÃO

Uma especificação de requisitos deve exercer várias funções durante o desenvolvimento de software, dentre as quais destacamos:

- descrever o que o sistema faz, de acordo com os requisitos do usuário, permitindo-lhe fazer revisões e acompanhar o processo de desenvolvimento;
- descrever o que a equipe deve produzir para permitir a gerência do trabalho de desenvolvimento [Boeh77];
- ser uma referência completa a respeito do comportamento externo do sistema [Parn82], reduzindo problemas de coordenação entre programadores durante a implementação;
- ser o elemento contra o qual são feitas as validações da especificação de projeto [Wae83].

- ser a base necessária (mas não suficiente) para uma boa previsão de cronograma, recursos e custos do desenvolvimento;
- ser o elemento básico (referencial) para o desenvolvimento do plano de testes e aceitação [Parn82].

Existem várias classificações de trabalhos sobre análise e especificação de requisitos, como [Frée84, Roch87, Geha86, Leit87, Ceri86, Blan83]. O diagrama a seguir relaciona algumas das siglas das linguagens, métodos e metodologias citadas nesses trabalhos, juntamente com o ano de sua criação ou o ano em que o trabalho correspondente foi publicado.

USE

Jsd

TAXIS

SARA

BSM

Ssa

IDEF

Isac

TAGS

SDS

MSG84

SREM

PLEASE

Warnier-Orr

PSL/PSA

TEDIUM

Hipo

SADT

PAISLEY

SXL

PDL

SOFT.FACT.

GIST

FDM

1973 1975 1977 1979 1981 1983 1985 1987

Dentre os métodos mais citados o SADT [Ross85] se destaca, sendo considerado apropriado para especificação de requisitos de um problema, por, dentre outros, [Booc84, Ceri86, Borg85, Blan83, Yada89 e Leit87]. Este último faz um estudo taxonomônico envolvendo mais de 120 títulos, em cuja bibliografia o SADT é o mais frequentemente referenciado.

Apesar de sua popularidade, o SADT apresenta vários problemas, inclusive do ponto de vista de dificuldade de treinamento de usuários. O SADT não cobre de maneira satisfatória os seguintes tópicos: estrutura de controle, detalhes de procedimentos, detalhes de entrada e saída e de

Considerando a necessidade cada vez maior de dar tratamento adequado à análise de requisitos e tendo em vista a aceitação do SADT na comunidade de Engenharia de Software, decidiu-se desenvolver um método que aliasse as qualidades do SADT a mecanismos que permitissem seu suporte automatizado. O método resultante, ANA-RE, elimina vários dos problemas existentes no SADT.

O ANA-RE é suportado pelo ambiente SAES, cujo protótipo é descrito neste artigo. Através dele, é possível dar suporte automatizado à geração e manipulação de diagramas ANA-RE e SADT para especificação de requisitos.

O restante deste artigo está dividido da seguinte forma: a seção 2 apresenta em linhas gerais os componentes do método ANA-RE; a seção 3 descreve o ambiente SAES e as características do protótipo implementado; a seção 4 contém exemplos de sua utilização; e, por último, a seção 5 contém conclusões e sugestões para pesquisas futuras.

2. DESCRIÇÃO SUCINTA DO ANA-RE

Tendo em vista os problemas já apontados no SADT, foi feito um estudo sistemático do método [Tole89], de forma a eliminar redundâncias e inconsistências de representação, padronizar a notação e permitir seu suporte automatizado. O resultado do estudo foi a proposta de um novo método de especificação de requisitos, o ANA-RE (ANálise e especificação de REquisitos), que além de simplificar o SADT permite o tratamento formal de um universo maior de problemas. O ANA-RE possui 31 componentes em comparação com os 40 do SADT. Apenas 16 componentes do SADT foram mantidos, sendo os demais condensados, modificados ou eliminados.

Além disto, o ANA-RE criou 7 componentes não disponíveis no SADT, que visam permitir:

- . apresentação de detalhes a respeito de entradas, saídas e comunicação com o usuário, basendo-se em componentes similares existentes em diagramas DFD [Gane79];
- . apresentação de detalhes padronizados que complementem as informações de cada diagrama;
- . diferenciação entre documentos gerados por projetistas e por comentaristas; além de,
- . cobertura de detalhes de estrutura de controle e o tratamento de problemas de tempo-real.

Qualquer diagrama SADT pode ser especificado em ANA-RE de forma mais compacta e padronizada. Há no entanto diagramas ANA-RE que não têm correspondente SADT.

A análise dos 40 componentes do SADT e a definição dos componentes do ANA-RE é extensa, não podendo ser apresentada aqui. Para maiores detalhes sobre a definição do ANA-RE, sugere-se a leitura de [Tol89].

O quadro a seguir contém uma comparação entre os componentes do SADT e os do ANA-RE. Esta seção supõe que o leitor já tenha tido contato com o SADT, podendo assim avaliar as modificações introduzidas.

	Aspecto ANA-RE	Notacao ANA-RE	Aspecto SADT	Notacao SADT
1	caixa		caixa	
2	seta		seta	
3	interface		interface	
4	substituído por observação ou número-c		mecanismo e chamada	
5	nome		nome	
6	rotulo		rotulo	
7	passagem		passagem	
8	restricao		restricao	
9	nao tem similar		seta omitida	
10	desvio e juncao		desvio e juncao	
11	desvio e juncao		para-todos	

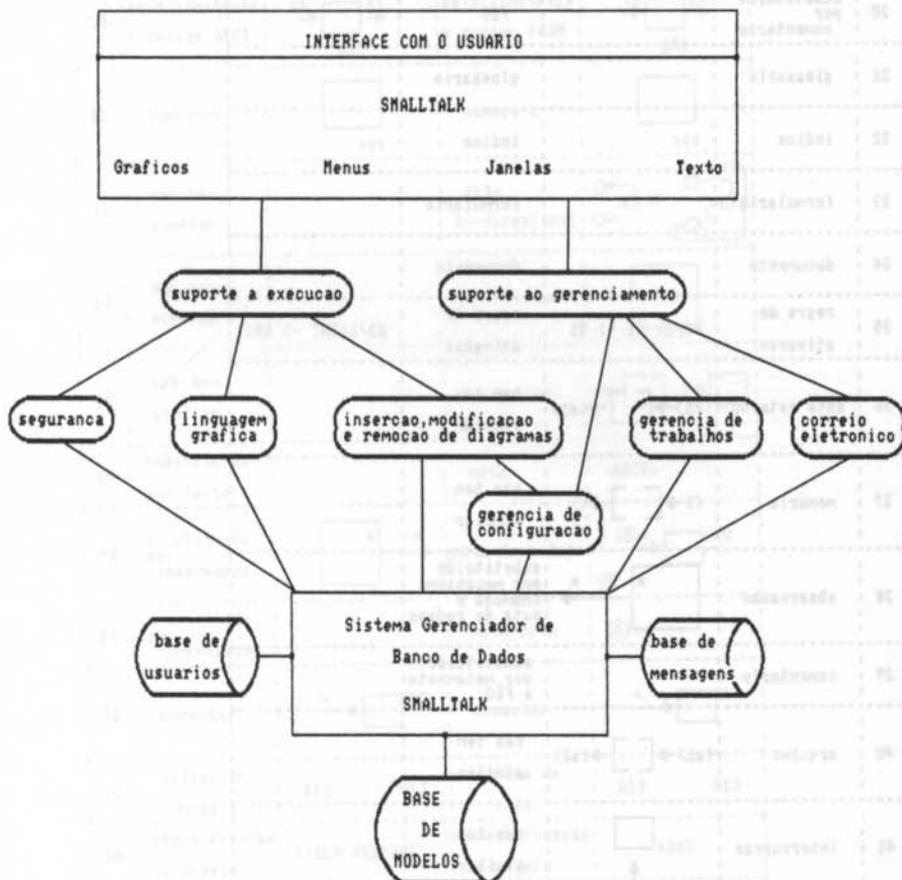
Aspecto ANA-RE	Notacao ANA-RE	Aspecto SADT	Notacao SADT
12 uniao e separacao		uniao e separacao	
13 desvio-ou e juncao-ou		desvio-ou e juncao-ou	
14 seta limitrofe e codigo SECI		seta limitrofe e codigo ICOM	
15 numero-c		numero-c	
16 nao tem similar		seta bi-direcional	
17 nao tem similar		seta mista	
18 nao tem similar		tunneling	
19 substituido por texto		nota	NOTE:
20 substituido por observacao		nota de rodape	
21 substituido por comentario		meta-nota	
22 conector		conector	
23 codigo de nodo	A13 D13	codigo de nodo	A13 D13
24 identificacao de modelo	SIGLA' RELEASE'	identificacao de modelo	NOME
25 codigo de interface	A1.3codigoSECI	codigo de interface	A1.3codigoICOM
26 nao tem similar		codigo de nota	A1.301-A1.413

	Aspecto ANA-RE	Notacao ANA-RE	Aspecto SADT	Notacao SADT
27	ponto	A122.4E1	ponto	A122.4I1
28	layout em escada		layout em escada	
29	texto		texto	
30	substituido por comentario		FEO	
31	glossario		glossario	
32	indice		indice	
33	formulario		formulario	
34	documento		documento	
35	regra de ativacao	A3/2: E1 -> S1	regra de ativacao	A3/2:ENT -> SAI
36	ente externo	$\langle e1 \rangle \rightarrow [] \rightarrow \langle e1 \rangle$	nao tem similar	
37	memoria	$() \rightarrow [] \rightarrow ()$	nao tem similar	
38	observacao		substituido por mecanismo chamada e nota de rodape	
39	comentario		substituido por meta-nota e FEO	
40	arquivo	$\langle a1 \rangle \rightarrow [] \rightarrow \langle a2 \rangle$	nao tem similar	
41	interrupcao		nao tem similar	
42	descricao		nao tem similar	

3. O AMBIENTE SAES E SUA IMPLEMENTAÇÃO

O SAES é uma proposta de ambiente que se destina a auxiliar o processo de análise, definição e especificação de requisitos, automatizando o método ANA-RE. O ambiente visa oferecer facilidades para permitir a definição, especificação e verificação de requisitos de sistemas de software.

O diagrama a seguir mostra os componentes do SAES e a forma de interação com o usuário.



Ambientes automatizados para desenvolvimento de software dependem de suporte adequado por parte de bancos de dados (ou, mais recentemente, bancos de dados orientados a objetos, às vezes denominados bases de objetos). Uma análise dos 23 trabalhos publicados no 3º Simpósio de Ambientes de Desenvolvimento de Software [ACM89] mostra que 14 trabalhos ressaltam ser necessário o suporte de uma base de dados adequada para automatização de um ambiente, permitindo a integração de seus dados.

Uma taxonomia de ambientes de desenvolvimento de software classifica-os em quatro tipos [Dart87]: ambientes centrados em linguagem, orientados por estruturas, ambientes de "tool kits" e ambientes baseados em métodos.

O SAES pertence à última categoria, dando suporte ao ANA-RE e ao SADT. Assim como os demais ambientes automatizados que suportam métodos para especificação de requisitos, o SAES também está centrado no suporte de um banco de dados, aqui chamado banco de modelos.

A automatização do ANA-RE (e do SADT) implica a automatização da linguagem gráfica e da técnica para análise estruturada e projeto. A automatização deve permitir busca e localização de informações (especificações de requisitos), navegação através da árvore hierárquica de diagramas, emissão de documentação padronizada, gerência de configuração e verificação de consistência entre diagramas de um mesmo modelo. A definição do ANA-RE simplificou vários conceitos do SADT e permitiu a especificação automatizada de diagramas e sua verificação de consistência.

O sistema gerenciador de banco de dados, a interface com o usuário, a linguagem gráfica e o módulo de inserção, modificação e remoção de diagramas compõem o núcleo que suporta o método ANA-RE. A linguagem gráfica não foi implementada, sendo a interação com o usuário realizada sob forma textual.

O protótipo do SAES foi implementado em Smalltalk e pode ser executado em micros do tipo PC-compatível. A razão da escolha foi o fato de ser baseada no paradigma de objetos, que permite, de forma natural, a implementação direta do projeto do banco de dados.

O ambiente Smalltalk pode ser encarado como um gerenciador de base de objetos, visto que através dele é possível manipular objetos sem a preocupação de como se dá o seu armazenamento, a sua recuperação ou qual a estrutura física dos dados. Desta maneira, para sistema gerenciador de banco de dados foi usado o Smalltalk acrescido da implementação da modelagem dos relacionamentos dos componentes de diagramas e modelos ANA-RE. Assim, os objetos modelados representam os componentes de diagramas e modelos ANA-RE, permitindo a manipulação direta dos objetos e sua integridade na definição de objetos e sua "explosão".

A interface com o usuário foi desenvolvida aproveitando ao máximo a biblioteca do Smalltalk de classes e respectivos métodos com relação à manipulação de menus e janelas, o mesmo podendo ocorrer com gráficos e textos.

O módulo inserção, modificação e remoção de diagramas pode ser acionado diretamente pelo usuário (tratamento de diagramas como um todo) ou pela linguagem gráfica (manipulação dos componentes de um diagrama, no protótipo representados por texto).

O sistema SAES, além de suportar o ANA-RE, prevê facilidades de correio eletrônico (para comunicação dentro de uma equipe), ferramentas para suporte à gerência dos trabalhos e mecanismos que garantem segurança em termos de permissão de acesso a diagramas. Além disso, estão previstos comandos de ensino ('teach' ou 'help') para usuários esporádicos e uso opcional de janelas para múltiplas manipulações de diagramas.

O SAES foi desenvolvido considerando a futura implementação de simulação de modelos visando um suporte para experimentação dos requisitos especificados.

A implementação resultou em, aproximadamente, 2000 linhas de código Smalltalk, nas quais constata-se uma forte reutilização de software no que diz respeito a estruturas de dados, interface com o usuário e primitivas (conversão, seleção, iteração, etc).

O Smalltalk prevê abstração de dados, o que permitiu reutilização de software e intrinsecamente ofereceu o conceito de relacionamento de generalização, através das classes e subclasses.

A interface implementada utiliza menus que permitem a definição de um modelo, tendo como base a identificação única dos diversos componentes em um actigram que podem ser interligados através de setas. O uso de menus tem a vantagem de permitir que o protótipo seja utilizado por usuários não especializados, em oposição a uma linguagem de especificação de diagramas.

A visualização da estrutura dos diagramas é possível através da exibição dos identificadores dos componentes e de símbolos que indicam os respectivos contextos. O resultado pode ser considerado como uma imagem do actigram descrito em uma linguagem declarativa, que prejudica muito o entendimento se comparada a uma interface gráfica.

A implementação verifica a consistência dentro de um diagrama e entre níveis de um conjunto de diagramas. Por exemplo, a solicitação para decompor uma caixa tem como resultado a verificação da existência de setas associadas à futura caixa-pai (consistência interna). As interfaces da caixa-pai são herdadas pelo diagrama-filho não podendo ser modificadas ou removidas quando da edição do diagrama (consistência externa).

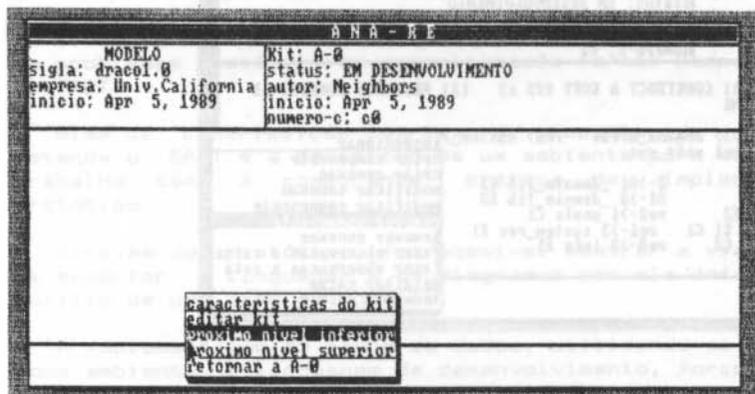
A seção a seguir mostra um exemplo de utilização do protótipo implementado.

4. EXEMPLO DE USO DO PROTÓTIPO DO SAES

O exemplo de modelo SADT apresentado neste anexo, elaborado por Neighbors e citado em [Free87], descreve em detalhes a utilização pretendida do contexto DRACO.

Para exemplificar a iteração de uma sessão de definição com o SAES, são apresentadas cópias das telas exibidas, juntamente com cópia dos originais SADT de Neighbors.

As telas a seguir mostram o início da criação do primeiro nível de um diagrama (kit A-0) e o resultado de sua criação.



O menu do kit permite navegar para cima ou para baixo na estrutura de kits, editando o kit e modificando suas informações.

ANA - RE	
MODELO Sigla: dracol.0 empresa: Univ. California inicio: Apr 5, 1989	Kit: A-0 status: EM DESENVOLVIMENTO autor: Neighbors inicio: Apr 5, 1989 numero-c: c0
COMPONENTES (e2) Organization (e3) Areas (e1) User (e4) Domain_base [0] CREATE SOFT SYS ci *	
REFERENCIAS	
CONEXOES e1->0 system_req E1 e2->0 goals C1 e3->0 demand_areas C2 e4->0 info C3 0->e1 soft_sys S1	
inspecionar criar componente criar conexao modificar conexao modificar componente remover componente remover conexao fazer observacao a caixa fazer observacao a seta detalhar caixa terminar edicao	

A tela a seguir corresponde ao diagrama SADT AO (apresentado em seguida) totalmente definido, inclusive pela indicação de detalhamento das caixas 2 e 3.

ANA - RE	
MODELO Sigla: dracol.0 empresa: Univ. California inicio: Apr 5, 1989	Kit: A0 status: EM DESENVOLVIMENTO autor: Neighbors inicio: Apr 5, 1989 numero-c: c1
COMPONENTES (01) separacao [3] CONSTRUCT A SOFT SYS c3 [2] RESEARCH DOMAINS c2 (1) DETERMINE DOMAINS	
REFERENCIAS re4) info re3) demand_areas rel) system_req re2) goals (rel) soft_sys	
CONEXOES 01->3 Draco_lib C1 2->01 domain_lib S1 3->rel soft_sys S1 01->2 domain_lib E3 3->2 experience S2 E2 re2->1 goals C1 1->2 problem_domain S1 C1 rel->3 system_req E1 re3->1 demand_areas C2 re4->2 info E1	
inspecionar criar componente criar conexao modificar conexao modificar componente remover componente remover conexao fazer observacao a caixa fazer observacao a seta detalhar caixa terminar edicao	

A notação 1 -> 2 problem domain S1 C1 representa a seta que liga a caixa 1 à 2, sendo a primeira (e única) saída da origem (caixa 1) e sendo o primeiro (e único) controle do destino (caixa 2), além disto indica que o rótulo da seta é **problem domain** (no diagrama SADT o rótulo é bem mais extenso).

A janela inferior da tela é utilizada para exibição de mensagens de erro, quando ocorre violação de integridade.

USED AT	AUTHOR: James M. Neighbors PROJECT: Draco 1.0	DATE: 15-OCT-80 REV: 3	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT
NOTES: 1 2 3 4 5 6 7 8 9 10						


```

graph TD
    C1[C1 organizational goals] --> D[ ]
    C2[C2 areas where there is a demand for many similar systems] --> D
    D --> A[determine domains of interest]
    A --> B[ ]
    B --> C[research the domain (DRA04)]
    C3[C3 information about different problem domains] --> C
    B --> D1[ ]
    D1 --> E[construct a software system (DRA03)]
    I1[I1 system requirements] --> E
    E --> O1[O1 software system]
    E --> B
    D2[ ] --> C
    D2 --> E
    D2 --> O1
    D2 --- L[library of domain analysis reports and domains which have been described to Draco]
    D2 --- R[Draco library of domains]
    D2 --- S[experience with building systems in a domain]
    
```

NODE	A0	TITLE	CREATE SOFTWARE SYSTEMS	NUMBER	DRA02
------	----	-------	-------------------------	--------	-------

5. CONCLUSÕES E EXTENSÕES

Este trabalho apresentou o ambiente SAES que representa uma ferramenta de apoio para a elaboração de modelos que permitem a análise, definição e comunicação de requisitos de sistemas de software. É o suporte do método ANA-RE, que aumenta o espectro de problemas passíveis de especificação formal comparativamente ao SADT.

Além da contribuição teórica da proposta de um método que estende o SADT e a definição de um ambiente para suportá-lo, o trabalho tem a componente prática da implementação do protótipo.

Através do protótipo, foi possível mostrar a viabilidade de se suportar a linguagem e os diagramas por ela definidos com o auxílio de uma base de modelos.

A implementação do banco de dados, utilizando-se o Smalltalk como ambiente e linguagem de desenvolvimento, forçou a análise e definição dos limites de um gerenciador de banco de dados do tipo necessário para o ambiente SAES, caso se deseje um protótipo implementado a partir do paradigma de objetos.

Como reflexo da implementação, verificou-se que o paradigma de objetos deve ser explorado pois facilitou a definição e implementação dos elementos modelados.

Dentre as extensões consideradas, pretende-se ampliar o protótipo, de forma a permitir maior controle de integridade além da simulação de modelos. É necessário, também, utilizar uma interface gráfica para facilitar o desenho de diagramas ANA-RE.

BIBLIOGRAFIA

- [ACMB89] ACM SIGSOFT/SIGPLAN
"Proceedings of the ACM SIGSOFT/SIGPLAN Software
Engineering Symposium on Practical Software
Development Environments"
SIGPLAN Notices 24(2), 1989
- [Blan83] Blank, J. & Krijger, M.J.
"Software Engineering: Methods and Techniques"
Wiley-Interscience Pub. 1983
- [Boeh77] Boehm, B.W.
"Seven Basic Principles of Software Engineering"
Infotech State of the Art Report, 1977, pp. 77-113
- [Booc84] Booch, G.
"Object Oriented Design"
Tutorial on Soft. Design Techniques, [Free84],
pp. 420-436
- [Borg85] Borgida, A., Greenspan, S. & Mylopoulos, J.
"Knowledge Representation as the Basis for
Requirements Specifications"
IEEE Computer, 18(4), 04/85, pp. 82-91
- [Ceri86] Ceri, S.
"Requirements Collection and Analysis in
Information Systems Design"
Elsevier Science Pub., North-Holland, 1986,
pp. 205-214
- [Colt84] Colter, M.A.
"A Comparative Examination of Systems Analysis
Techniques"
MIS Quarterly, 03/84, pp. 51-66
referenciado em [Leit87]
- [Dart87] Dart, S.A. et alii
"Software Development Environments"
IEEE Computer, 11/87, pp. 18-28
- [Free84] Freeman, P. & Wasserman, A.I.
"Tutorial on Software Design Techniques"
IEEE Catalog no. EH005-5, 4th ed., 1984

- [Free87] Freeman, P.
"A Conceptual Analysis of the Draco Approach to
Constructing Software Systems"
IEEE Transac. on Soft. Eng'g, SE13 (7), 07/87
- [Gane79] Gane, C.P. & Sarson, T.
"Structured Systems Analysis: Tools and Techniques"
Prentice-Hall, Software Series, 1979
- [Geha86] Gehani, N. & McGettrick, A.D.
"Software Specification Techniques"
Addison-Wesley Pub., 1986
- [Leit87] Leite, J.C.S.P.
"A Survey on Requirements Analysis"
Tech. Report, University of California at Irvine,
06/87
- [Parn82] Parnas, D.L. & Clements, P.C.
"A Rational Design Process: How and Why to Fake It"
IEEE Transac. on Soft. Eng'g, SE12(2), 02/82,
pp. 251-257
- [Roch87] Rocha, A.R.C.
"Análise e Projeto Estruturado de Sistemas"
Editora Campos, 1987
- [Ross85] Ross, D.T.
"Applications and Extensions of SADT"
IEEE Computer, 18(4), 04/85, pp. 25-35
- [Toba89] Tobar, C.M.
"Um Método para Análise e Especificação de
Requisitos"
Tese de Mestrado IMECC UNICAMP, 1989
- [Wass83] Wasserman, A.I.
"Software Engineering Environments"
Advances in Computers, 22, pp. 109-161
- [Yada89] Yadav, S.B. & et ali.
"Comparison of Analysis Techniques for Information
Requirements Determination"
CACM, 31(9), 09/88, pp. 1090-1097