

## Promoção e Organização

Comissão Especial de Engenharia de Software  
da Sociedade Brasileira de Computação

Departamento de Informática  
da Universidade Federal de Pernambuco

## Comitê Organizador

Roberto Spolidoro (SEI)

Eratóstenes Araújo (CNPq)

Décio Fonseca, Judith Kelner e Silvio Meira (UFPE)

## Comitê Executivo

Ivanilda Mendes, Ivoneide Ribeiro, Maria José Gonçalves e Maria Lília Freitas (UFPE)

Edda Lück (Agência Luck)

## Patrocínio e Apoio

BANORTE – Banco Nacional do Norte

BSM – Banorte Sistemas e Métodos

CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico

DEC – Digital Equipment Corporation

FINEP – Financiadora de Estudos e Projetos

IBM – International Business Machines

Hering Nordeste

Itautec Informática

Sid Informática

Scopus Tecnologia

Sudene – PNUD/Banco Mundial

4917-3

capa duna

Daltro

6/11/89

## Comitê de Programa

Roberto Bigonha, Dep. de Ciência da Computação, UFMG

Marcos Borges, Núcleo de Computação Eletrônica, UFRJ

Rogério Drummond, Dep. de Ciência da Computação, UniCamp

Décio Fonseca, Dep. de Informática, UFPE

Júlio Leite, Dep. de Informática, PUC-Rio

Raul Martins, Centro Científico, IBM

Silvio Meira, Dep. de Informática, UFPE (Presidente)

Daltro Nunes, Dep. de Informática, UFRGS

Tadao Takahashi, Projeto ETHOS

## Palestras Convidadas

*Strategic Directions in Software Engineering: Past, Present and Future.*  
Prof. Peter Freeman, George Mason University (on leave from Univ. of California, Irvine).

*On Supporting De-centralized Large-Scale Software Processes.*  
Dr. Pankaj Garg, Hewlett-Packard, USA.

*A Reutilização de Software sob o Ponto de Vista de uma Teoria Geral de Metáforas.*  
Prof. Carlos Lucena, PUC-Rio.

UFPE  
BIBLIOTECA

## Nota do Editor

Esta Terceira Edição do Simpósio Brasileiro de Engenharia de Software conta com 23 artigos técnicos e 1 tutorial, selecionados dentre 65 submissões. No Simpósio também serão demonstradas 6 ferramentas de software escolhidas pelo Comitê de Programa.

Pela primeira vez no SBES os trabalhos foram submetidos a uma revisão pelos Autores, a partir dos comentários e modificações sugeridas pelos Revisores. O resultado final foi uma sensível melhoria na qualidade dos textos aqui impressos. Para que tal fosse possível, foi fundamental a presteza e qualidade do trabalho dos Autores, Revisores e do Comitê de Programa.

Em que pese todas as dificuldades — parte das quais inerentes ao processo em que o nosso País se encontra — acreditamos que este III SBES representa o começo da consolidação do evento e da área de Engenharia de Software como uma das mais importantes da Computação no Brasil.

S. R. L. M.

# Índice

<i>Strategic Directions in Software Engineering: Past, Present and Future</i> .....	1
Prof. Peter Freeman, George Mason University (on leave from UC Irvine). (Palestrante Convidado)	
LegoShell, Linguagem de Computações .....	2
R. Drummond (UniCamp).	
Simulação de Especificações LOTOS Usando Linguagens Funcionais .....	17
C. Ferraz, P. Cunha, S. Meira (UFPE).	
Experience in Parallelizing Large Application Programs .....	31
M. J. Teixeira (CEPEL-Eletróbrás).	
> TUTORIAL: A Linguagem CHILL .....	42
M. Pereira (CPqD-Telebrás).	
O Uso de Pontos de Vista na Elicitação de Requisitos .....	71
J. Leite (PUC-Rio).	
Especificação de Sistemas de Informação de Escritórios: A Metodologia T. A. & A .....	86
J. Pallazo, N. Hoppen (UFRGS).	
Representação Gráfica para LOTOS .....	98
M. T. Moura, P. Cunha (UFPE).	
Meta-Modelos para Ferramentas Genéricas Integráveis ao Ambiente SIPS .....	113
F. Akhras, M. Costa (CTI).	
<i>On Supporting De-centralized Large-Scale Software Processes</i> .....	123
Dr. Pankaj Garg, Hewlett-Packard. (Palestrante Convidado)	
• Linda: Uma Linguagem de Autoria Automática para Sistemas de Hipertexto .....	124
J. Kelner, A. Cavalcanti, A. Pardo (UFPE).	
• Um Editor Híbrido (Texto e Diagramas) Orientado por Estruturas (do Tipo Grafo e Árvore) ..	137
R. Price, E. Favero (UFRGS).	
• Hipertexto: O Projeto do Sistema H .....	152
S. Meira, J. Kelner, E. Albuquerque, J. Martins, A. Melo, A. Vasconcelos (UFPE).	

Protótipos Funcionais a Partir de Especificações em VDM .....	171
P. Borba, S. Meira (UFPE).	
Especificação de Um Programa de Matrículas Usando VDM .....	185
L. Ribeiro, D. Nunes (UFRGS).	
Métodos Formais para Desenvolvimento de Software Orientado a Objetos: um Estudo de Caso, o Projeto GARDEN .....	201
R. Martins, G. Annaruma, L. Carneiro, A. Lima, P. Molinari, E. Quintana, R. Stern (IBM).	
SAES - Sistema de Apoio à Especificação de Sistemas .....	214
C. Toledo, C. Bauzer (UniCamp).	
Desenvolvimento de Algoritmos Aproximativos por Aterramento: Especificação Formal .....	228
L. Toscani (UFRGS), P. Veloso (PUC-Rio).	
Modelando a Determinação de Potenciais DU-Caminhos Através da Análise de Fluxo de Dados .....	239
M. Chaim (UniCamp), J. Maldonado (USP), M. Jino (UniCamp).	
*A Reutilização de Software do Ponto de Vista de uma Teoria Formal de Metáforas .....	252
Carlos Lucena (PUC-Rio), J. Silva (USP). (Palestrante Convidado)	
*Uma Ferramenta para Auxílio no Reuso de Software .....	253
M. M. Teixeira (CTA).	
*Mecanismo de Gerenciamento de Versões e Configurações do A_HAND .....	269
E. Victorelli, G. Magalhães, R. Drummond (UniCamp).	
*Um Sistema de Gerenciamento de Projetos de Software .....	281
L. Mendes (SulAmérica)	
Modelagem de Dados de um Ambiente de Desenvolvimento de Software .....	296
J. Machado, R. Price (UFRGS).	
Uma Formalização em IMC: o Modelo DODM .....	311
M. Pimenta, C. Heuser (UFRGS).	
Especificação de um Gerenciador Orientado a Objetos .....	325
M. Máccêdo, D. Fonseca (UFPE).	
Aspectos de Epistemologia do Software .....	340
G. Câmara (INPE).	



Strategic Directions in Software Engineering:  
Past, Present and Future

Peter Freeman

George Mason University<sup>1</sup>  
School of Information Technology  
Fairfax, Virginia, 22030, USA

Abstract

Software Engineering was created in the late 1960's in response to strategic needs, and will continue to be needed as a strategic technology by many organizations for the foreseeable future.

Despite 20 years of effort, the current situation in software engineering is problematical at best. To aid our understanding of where to head in the future, this paper analyses some of the (apparent) strategic directions of the past and present before suggesting several for the future.

Our purpose here is to provide an outline of where we have been in software engineering, where we are now, and where we should be headed; it is not our intention to provide a comprehensive survey of the content of the field.

We will do this by describing some representative activity in several areas, assessing the strategies that they implicitly represent, and suggesting strategies for the future. This analysis of strategies should prove illuminating both to those who wish to use software engineering and its results and to those whose interest is in creating new software engineering technology.

<sup>1</sup>On leave from University of California, Irvine

On Supporting De-centralized  
Large Scale Software Processes

Pankaj K. Garg  
HP Laboratories  
1501 Page Mill Road  
Palo Alto, CA 94303-0971  
Ph: (415)857-4709  
E-mail: garg@hplabs.hp.com

Abstract

The life cycle of large scale software systems requires the cooperative effort of several people over extended periods of time. Automated support for the communication and coordination involved in this cooperative effort requires the management (creation, storage, printing, and usage) of large volumes of information.

In the past, several models for managing this information have been proposed. Hypertext is a flexible, general purpose, non-linear model for storing textual information. It is flexible in the sense that readers can augment the organization of a hypertext for subsequent browsing. It is general purpose in the sense that no restriction is put on the type of information contained in a hypertext node. The thesis of my work is that software engineering information should be managed using a hypertext based approach. Previous models of software engineering information do not provide the flexibility and generality of a hypertext model.

We have designed and implemented an *Intelligent Software Hypertext System* (Ishys), for management of software engineering information. In Ishys, Basic Templates (BTs) are nodes of a hypertext which can have point-to-point and node-to-node links. Forms are sequences of BTs which can be used to model software life cycle documents. Compositions are arbitrary collections of BTs. Projects represent partitions of the hypertext such that each project has the same forms. Tasks are responsibilities of agents which use or produce forms. Tasks are composed of actions which use or produce BTs. Actions are composed of primitive actions which are commands provided through Ishys. In a general user mode, users create information in Ishys by instantiating BTs. In a super user mode, users define the projects, forms, BTs, tasks, actions, and primitive actions. Structured communication about BTs is supported through *annotations* attached to different points within BTs.

To illustrate the flexibility and generality of the hypertext based approach we show how a hypertext can support generalizations, specializations, revisions, and aggregations.

To illustrate the power of hypertext systems in supporting collaborative work, we discuss how a hypertext system can support two models of node compositions: (1) the assembly line model, and (2) the exploratory assembly model.

A Reutilização de Software sob o Ponto de Vista  
de Uma Teoria Formal de Metáforas

Carlos J.P. Lucena

Departamento de Informática  
Pontifícia Universidade Católica - Rio de Janeiro

Jose Reinaldo Silva

Escola Politécnica  
Universidade de São Paulo

RESUMO

O presente trabalho propõe um tratamento do problema da reutilização de software que, ao mesmo tempo, fornece uma explicação para os conceitos básicos envolvidos e um mecanismo para promover a reutilização. O enfoque se baseia em uma teoria formal de metáforas (e analogias).

Há dois níveis de reutilização a considerar: o de idéias e conhecimentos e o de artefatos e componentes. Essa é a questão da reutilização da especificação versus a reutilização do código. Trabalhamos segundo a hipótese de que especificações formais são uma versão apropriada do projeto (desenho) de um software. Concentramo-nos neste nível de reutilização porque na maioria dos domínios de problema complexos o código em si não é reutilizável, enquanto as idéias gerais de um projeto o são.

Uma teoria formal de metáforas tem as seguintes características:

- (i) metáforas usam termos de um domínio fonte para se referir a objetos que pertencem a um domínio possivelmente diferente, chamado domínio objeto;
- (ii) os domínios fonte e objeto não precisam, necessariamente, ser diferentes ou disjuntos;
- (iii) uma metáfora opera transferindo, coerentemente, um conjunto de restrições estruturais de um domínio fonte para um domínio objeto;
- (iv) a adequação de uma metáfora é determinada, pelo menos em parte, pela quantidade de estrutura que é coerentemente transferida do domínio fonte para o domínio objeto.

O trabalho apresenta um exemplo de aplicação da teoria de metáforas ao problema de reutilização da especificação de um software. Exploram-se, também, os aspectos computacionais do novo método procedendo-se a uma análise dos algoritmos adequados para o seu uso prático.