

# ARCoPAS: um Ambiente para Recuperação Cooperativa do Projeto Arquitetônico de Sistemas

Maria Cláudia Reis Cavalcanti    Marcos Roberto da Silva Borges

yoko@nce.ufrj.br

mborges@nce.ufrj.br

NCE e COPPE Sistemas  
Universidade Federal do Rio de Janeiro  
Caixa Postal 2324 Rio de Janeiro, RJ  
CEP 20001-970

## Resumo

*Este artigo apresenta um protótipo de ambiente para apoiar o processo de Engenharia Reversa, mais especificamente a Recuperação do Projeto Arquitetônico de sistemas. As ferramentas existentes para apoiar este processo pouco investem na participação humana. Ao contrário destas ferramentas, o protótipo ARCoPAS alia a extração automática da estrutura modular de um sistema, à extração cooperativa do conhecimento dos especialistas sobre o mesmo. Esta aliança possibilita a recuperação de um documento mais completo, enriquecido por informações que só os especialistas possuem.*

## Abstract

*This paper presents a prototype of an environment to support the Reverse Engineering process, more specifically the Architectural Design Recovery of systems. Few existing tools that support such process allow human participation. Different from these tools, the ARCoPAS prototype joins the automatic extraction of the modular structure of a system, to the cooperative extraction of the specialists knowledge about it. This union allows the recovery of a more complete document, rich in informations that only the specialists know.*

**Palavras Chave:** Engenharia Reversa, Recuperação de Projeto, CSCW, Hipertexto.

## 1. Introdução

A fase de manutenção de software é muito prejudicada pelos altos custos envolvidos no trabalho com o software sem documentação. Para atacar este problema alguns investem no processo de desenvolvimento de software, utilizando metodologias que apoiam todo o processo e que geram uma documentação adequada. Outros investem no processo reverso, isto é, na extração de documentação a partir do software já desenvolvido.

A realidade de 80% dos softwares existentes consumirem grandes recursos em sua manutenção [Lafu90], exige uma atitude imediata. A solução daqueles que investem no processo de desenvolvimento, geralmente envolve a construção de um novo sistema e o descarte do sistema antigo. Entretanto, a atual valorização do reaproveitamento, tem conquistado cada vez mais adeptos à Reengenharia de Software, que em geral, envolve uma atividade de Engenharia Reversa seguida por alguma atividade do processo de desenvolvimento de software.

As ferramentas desenvolvidas até hoje para apoiar a Engenharia Reversa, pouco investem na participação humana. A maioria destas ferramentas são analisadores estáticos automáticos [Lafu90] que geram relatórios, como por exemplo o fluxograma ou a estrutura modular dos programas. No entanto, os especialistas no sistema que se quer "reverter", detêm informações muito enriquecedoras, que estas ferramentas totalmente automáticas não conseguem extrair. Assim sendo, faz-se necessário o investimento em ferramentas que possibilitem a interação com o conhecimento humano.

Chikofsky e Cross [ChCr90] identificaram a Recuperação de Projeto como uma atividade de Engenharia Reversa. Este artigo apresenta o protótipo ARCoPAS, cujo objetivo é a Recuperação do Projeto Arquitetônico de sistemas. Diferentemente da maioria dos trabalhos nesta área, o enfoque dado pelo ARCoPAS ressalta a importância da participação da equipe envolvida com o sistema, propondo um processo semi-automático e cooperativo. A próxima seção descreve o ambiente em que baseou-se a construção do ARCoPAS. Na quarta seção, descrevemos o ARCoPAS e as ferramentas que o compõem. Finalmente, na quinta seção, comentamos os resultados obtidos a partir da utilização do ARCoPAS.

## 2. Recuperação de Projeto

Neste trabalho optamos pelo investimento no processo reverso, acreditando ser esta abordagem, a mais direta no ataque aos problemas dos softwares em manutenção. Mais especificamente, focalizamos a atividade de Recuperação de Projeto de software. Segundo Biggerstaff, *"A Recuperação de Projeto recria abstrações de projeto a partir de uma combinação de código, documentação de projeto existente (se disponível), experiência pessoal, e conhecimento geral sobre o problema e o domínio da aplicação."* [Bigg89]

A partir desta definição, idealizamos um ambiente adequado à tarefa de Recuperação da documentação relativa ao Projeto Arquitetônico de sistemas. O enfoque adotado visa recuperar esta documentação a partir do código fonte e do conhecimento dos especialistas no software. No ambiente proposto a tarefa de Recuperação, divide-se em duas etapas: automática e cooperativa. Em ambas as etapas conta-se com o apoio de ferramentas automáticas. Para a primeira etapa é necessária a utilização de um analisador de códigos fonte. A segunda etapa apóia-se em ferramentas de hipertexto e trabalho cooperativo.

São dois os documentos recuperados: o Projeto Arquitetônico propriamente dito, e um documento contendo as informações trocadas entre os especialistas. Para representar o Projeto Arquitetônico utilizamos a notação "HyperText Design Notation" (HTDN), enquanto que para capturar o conhecimento dos especialistas utilizamos o modelo de argumentação "Issue-Based Information System" (IBIS).

### 2.1. Representando o Projeto Arquitetônico

Entre as notações existentes para representar o Projeto de um sistema, existem as notações gráficas e textuais. O gráfico da Estrutura Modular ou Diagrama de Estrutura, desenvolvido para apoiar o método de Projeto Estruturado, é uma das mais utilizadas. Mas, embora a notação gráfica permita uma visualização imediata, facilitando a compreensão do sistema, ela é incompleta pois, por motivos de simplicidade, omite informações tais como detalhes sobre a interface entre módulos. Já a notação não gráfica, embora de difícil visualização, permite uma descrição completa do Projeto.

Ghezzi et al. em [Ghez91], apresentam uma notação textual chamada "Textual Design Notation" (TDN). A notação TDN comporta informações formais e informais (veja a figura 1). A informação formal pode ser obtida a partir de uma análise do sistema em si, enquanto que a informação informal, como os comentários e a descrição do módulo, provém do conhecimento dos especialistas.

A TDN descreve a estrutura modular do sistema, ou seja, ela identifica e descreve cada

```
Módulo Detecta-Radar
Usa Monitora-Frequência, Controla-msg
Interface
    var Cod-radar: integer;
    type freq-comum: array (1..10) of real;
    <comentários sobre a interface>
Implementação
    <descrição do módulo>
É-composto-por Inicializa, Transf-Freq
fim Detecta-Radar
```

Figura 1: Notação TDN

módulo do sistema, e provê a relação *usa* entre eles. O conjunto de descrições de todos os módulos completa a documentação do sistema. Entretanto, se o número de módulos for razoavelmente grande, apenas a representação da relação *usa* é insuficiente. Além desta relação, a TDN provê também a relação *é-composto-por*, que representa a estrutura hierárquica do sistema conhecida como *árvore conceitual*. Nesta árvore, os módulos do sistema estão agrupados e representados por outros módulos de nível mais alto, conhecidos como *clusters*.

O meio eletrônico oferece muitos recursos para a representação de um documento. O enfoque de hipertexto é um exemplo em que a utilização destes recursos é bem sucedida, permitindo uma melhor visualização de informações. Neste trabalho optou-se por representar o Projeto Arquitetônico utilizando uma variação da notação TDN na forma de hipertexto, que denominamos HyperText Design Notation (HTDN).

Sistema: **CONTROLE DE RADAR**

Cluster: **SENSORES**

descrição: Módulos responsáveis pelo controle dos sensores.

Módulo: **DETECTA\_RADAR**

descrição: recebe sinais detectados pelo radar

interface: var cod\_radar: integer; type freq\_comum: real

usa: **MONITORA\_FREQ**, **CONTROLA\_MSG**

Módulo: **MONITORA\_FREQ**

descrição: monitora a frequência recebida

usa: **MOSTRA\_MSG**

usado por: [excursão pelos módulos: *Detecta\_Radar*]

Módulo: **CONTROLA\_MSG**

descrição: controla as mensagens a enviar

usa: **MOSTRA\_MSG**, **SOA\_ALARME**

Assim como a notação TDN, a HTDN representa a estrutura modular (relação *usa*) e a árvore conceitual (relação *é-composto-por*). Como é possível notar na figura 2, a estrutura modular é representada como na notação TDN, mas com a diferença de que cada módulo na lista da relação *usa*, é na verdade um botão que aciona um link para a posição no texto onde está definido aquele módulo. Além disso, é possível navegar nos dois sentidos da

Figura 2: Notação HTDN. Os nomes de módulo em negrito são links que transportam o leitor para a posição no texto onde estes módulos estão definidos.

relação *usa*, buscando os módulos que usam determinado módulo. Ao lado da expressão *usado por* há um botão que gera, dinamicamente, uma excursão pelos pontos do texto em que estão as definições dos módulos que usam o módulo em questão. Quanto à árvore conceitual, esta pode ser visualizada através da endentação de seus níveis.

## 2.2. Etapa Automática

Em geral, a documentação mais confiável que se tem de um sistema é o seu código fonte. Algumas organizações mantêm outras documentações além do código, mas dificilmente garantem a sua atualização em relação ao mesmo. Tomando a situação mais comumente encontrada, consideramos o código fonte como a única documentação confiável e disponível, para proceder a etapa automática. Esta etapa envolve quatro tarefas: a identificação dos módulos do sistema, o estabelecimento das relações existentes entre estes módulos, a extração de sua interface e descrição, e a geração destas informações na notação HTDN. O ambiente de programação, a linguagem usada e os hábitos dos programadores do sistema, são fatores que influenciam fortemente as três primeiras tarefas. Já a quarta tarefa depende apenas da conclusão das outras três.

Uma mesma organização pode apresentar diversos ambientes de programação, cada qual com suas próprias diretrizes e características. O ambiente e a linguagem de programação são os responsáveis pela organização dos fontes e pela identificação do que se pode considerar como

módulo. Em geral, os módulos de um sistema correspondem aos arquivos de programas fonte que o compõem, e a relação entre eles é determinada pelo "uso" que um programa faz do outro (relação *usa*), isto é, quando encontramos no módulo Detecta\_Radar uma "chamada" para o módulo Monitora\_Frequência, isso significa que Detecta\_Radar *usa* Monitora\_Frequência, e que portanto, existe um relacionamento entre estes dois módulos.

A identificação da relação *é-composto-por* depende da forma de *clusterização* existente, que por sua vez varia de acordo com o ambiente de programação. Os sistemas costumam apresentar alguma forma de *clusterização*, gerada segundo algum critério, seja este ideal ou não. Entretanto, existem algoritmos de *clusterização*, como o proposto por Schwanke e Platoff [ScP189], que *clusterizam* um sistema, gerando a sua árvore conceitual automaticamente.

A interface entre módulos é aquilo que se precisa saber para *usar* um módulo. Normalmente, os parâmetros esperados e retornados de e por um módulo são identificados como sua interface. Em relação à descrição dos módulos, se o objetivo é obter algum texto em linguagem natural, vamos depender da qualidade e utilização de comentários nos fontes. Quanto mais descritivos forem os nomes de variáveis e módulos, e quanto mais comentários elucidativos acompanharem a descrição do programa, melhor será o resultado do trabalho de Recuperação de Projeto.

Uma vez identificados os módulos, seus relacionamentos, interfaces e descrições, o que nos resta é organizar os dados obtidos no formato da notação adotada (HTDN). Para tanto, é necessário contar com uma ferramenta de hipertexto que ofereça uma interface para importação de textos comuns. Algumas ferramentas de hipertexto fornecem uma linguagem de marcação de texto ("mark-up language"), com a qual é possível adaptar um texto comum e depois, transportá-lo automaticamente para dentro do ambiente da ferramenta. Desta forma, os links necessários à representação da estrutura modular, e os níveis necessários à representação da árvore conceitual podem ser criados ainda nesta etapa.

### 2.3. Etapa Cooperativa

A etapa automática dificilmente consegue extrair dos códigos fonte uma descrição completa de cada módulo, baseando-se somente nos comentários encontrados. O enriquecimento da descrição de cada módulo depende do apoio dos especialistas no sistema, que em geral são os membros da equipe responsável por sua manutenção. O envolvimento deste grupo de pessoas caracteriza a segunda etapa como cooperativa.

Assim como ocorre na fase de Projeto durante a engenharia progressiva, o processo reverso também exige que os membros da equipe discutam entre si. Esta interação ajuda os projetistas, ou recuperadores de projeto, a gerar um documento de Projeto de melhor qualidade. O registro desta discussão também é uma documentação importante, pois mais tarde, poderá facilitar o entendimento de como se originou o documento de Projeto.

O modelo de argumentação IBIS - "Issue-Based Information System" - foi projetado para capturar as discussões de Projeto [YaCo90, CoBe87]. A sua forma natural e intuitiva de expressão [ReEI91], torna-o fácil de utilizar e flexível para aplicação em outros contextos. Por este motivo, nós o escolhemos para apoiar a equipe de manutenção durante as discussões de recuperação de Projeto. O modelo IBIS compreende três elementos: questão<sup>1</sup>, posição e argumento, que correspondem respectivamente, aos problemas em discussão, às possíveis soluções para os problemas, e às opiniões favoráveis ou não às soluções levantadas. Uma

<sup>1</sup>Tradução do termo em inglês "issue".

questão pode ter várias posições para resolvê-la, e cada posição de uma questão pode, por sua vez, ter um ou mais argumentos favoráveis ou não.

Os documentos, Projeto Arquitetônico e Discussão de Projeto, estão fortemente relacionados, portanto, ambos precisam estar o mais próximo possível. Para manter esta proximidade, seria ideal que a mesma ferramenta de hipertexto que armazena o documento de Projeto, seja capaz de armazenar o documento da Discussão. Os sistemas de hipertexto são conhecidos por seu grande potencial de conexão entre documentos, fazendo crer que é possível prover esta proximidade. Porém, isto dependerá da forma escolhida para a implementação do modelo IBIS e das facilidades oferecidas pelo sistema de hipertexto escolhido.

### 3. O Protótipo ARCoPAS

O ARCoPAS (Ambiente para Recuperação Cooperativa de Projeto Arquitetônico) é a implementação de um protótipo de Ambiente para a Recuperação Cooperativa do Projeto Arquitetônico de Sistemas. Como foi comentado anteriormente, o ambiente idealizado divide a tarefa de recuperação em duas etapas: automática e cooperativa. Para apoiar a primeira etapa, foi desenvolvida uma ferramenta chamada ExEM (Extrator da Estrutura Modular). Ela é a responsável pela análise dos fontes: identificando os módulos e as suas interrelações, e gerando um texto com a estrutura modular no formato adequado para etapa seguinte.

Para apoiar a etapa cooperativa, utiliza-se duas ferramentas: o FolioViews<sup>2</sup> e a itIBIS. O FolioViews é um sistema de hipertexto, através do qual é possível visualizar e complementar o documento de Projeto, assim como "discutir" sobre a sua complementação. Para capturar esta discussão segundo o modelo IBIS, utiliza-se a ferramenta itIBIS, implementada sobre o FolioViews. A figura 3 apresenta uma esquema global do ARCoPAS, identificando suas etapas e ferramentas de apoio.

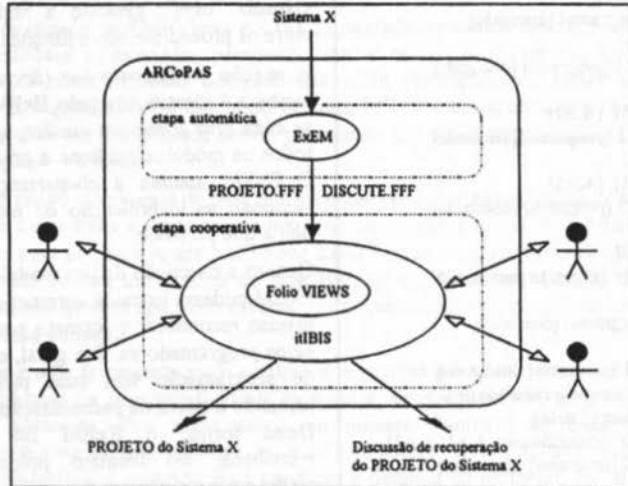


Figura 3: Esquema do ambiente ARCoPAS.

<sup>2</sup>Folio Views é marca registrada da Folio Corporation.

### 3.1. A ferramenta ExEM

O Extrator de Estrutura Modular (ExEM) é a ponte entre os fontes do sistema e o hiperdocumento. É ele o responsável pela identificação dos módulos e relacionamentos do sistema, assim como pela geração de um hipertexto com a estrutura modular e a árvore conceitual do mesmo utilizando a notação HTDN.

Em uma primeira versão, o ExEM foi desenvolvido para analisar fontes escritos na linguagem de programação Turbo Pascal. Um sistema típico escrito nesta linguagem, é composto por um fonte do tipo *program* e diversos fontes do tipo *unit*. Para extrair a estrutura modular de um sistema deste tipo é preciso estabelecer qual a granularidade desejada. De acordo com o granularidade escolhida, um módulo pode ser desde um arquivo fonte até uma variável. No entanto, como o objetivo deste trabalho é prover a recuperação do Projeto Arquitetônico, buscamos um nível de abstração alto. O ExEM identifica como módulos, os procedimentos e funções (públicos e privados) dos fontes do tipo *unit*, e os fontes do tipo *program*. Para obter o nível de abstração ideal, o ExEM desconsidera variáveis e procedimentos internos a outros procedimentos.

Os procedimentos e funções que compõem uma *unit*, podem *usar* uns aos outros. Para descobrir a relação *usa* entre eles, o ExEM varre cada um, procurando por referências aos demais. Para determinar a relação *usa* entre módulos de diferentes *units* e, entre estes e o *program*, o ExEM baseia-se na cláusula "uses". Nesta cláusula há uma lista das *units* que podem ser *usadas* pela *unit* ou pelo *program* que as declarou. O fato de um *unit* A poder usar a *unit* B, significa que os procedimentos e funções que compõem a *unit* A podem usar os procedimentos e funções públicos que compõem a *unit* B. Entretanto, nem todos os procedimentos e funções são realmente usados.

O ExEM desdobra as listas declaradas na cláusula "uses", gerando a real relação *usa* entre os procedimentos e funções.

A relação *é-composto-por* (árvore conceitual) também é identificada pelo ExEM. Considera-se cada *unit* como um *cluster*, que representa todos os módulos públicos e privados da *unit*. O ExEM assume a *clusterização* existente, incluindo na identificação de cada módulo, a *unit* a que pertence.

Quanto à descrição de um módulo, para que o ExEM pudesse extrai-la automaticamente, seria preciso reconhecer o formato padrão utilizado pelos programadores. Em geral, cada ambiente de programação tem suas próprias regras, tornando a tarefa de padronização muito difícil. Desta forma, o ExEM foi obrigado a reconhecer um formato próprio. Caso o módulo possua alguma descrição, esta deve ser encontrada na linha imediatamente após a declaração do mesmo, iniciando-se com a seguinte sequência de caracteres: '{*descrição*:' . Não encontrando a descrição neste formato,

Sistema S
Cluster A [programa] [discussão]
descrição:
Módulo A.A1 [programa] [discussão]
descrição:
usa: [B.B1], [B.B2]
Módulo A.A2 [programa] [discussão]
descrição:
usa: [B.B1], [A.A3]
Módulo A.A3 [programa] [discussão]
descrição:
usa: [B.B2]
é usado por: [excursão para A.A2]
Cluster B [programa] [discussão]
descrição:
Módulo B.B1 [programa] [discussão]
descrição: reverte o conteúdo de S
interface: var S: string
é usado por: [excursão para A.A1, A.A2]
Módulo B.B2 [programa] [discussão]
descrição:
é usado por: [excursão para A.A1, A.A3]

Figura 4: Descrição do sistema S na notação HTDN



o ExEM extrai os comentários encontrados no trecho correspondente ao módulo para facilitar o trabalho na etapa cooperativa.

A interface de um módulo são os parâmetros necessários para permitir a sua utilização por outro módulo. O ExEM encontra a interface de cada módulo logo após a declaração dos mesmos, não sendo necessário fazer qualquer transformação.

Para ilustrar o funcionamento do ExEM, tomemos como exemplo um sistema S. Sejam A e B duas *units* pertencentes ao sistema S. Sejam A1 e A2 dois procedimentos públicos de A, A3 um procedimento privado de A, e B1 e B2 dois procedimentos públicos de B. A figura 4 apresenta o arquivo gerado pelo ExEM com a descrição dos módulos identificados a partir dos fontes analisados, usando a notação HTDN. Na descrição de cada módulo podem haver quatro tipos de botão: programa, usa, excursão e discussão. O botão *programa* transporta o leitor para o ambiente de programação Turbo Pascal, trazendo para edição o programa fonte correspondente àquele módulo. Os botões do tipo *usa* correspondem aos nomes de módulos que estão em negrito. Ao acionar o botão B.B2 que consta na lista de módulos usados pelo módulo A.A1, o leitor é levado à descrição do módulo B.B2. O botão *excursão* cria dinamicamente uma excursão pelos módulos que são usados por aquele módulo. E finalmente, o botão *discussão* transporta o leitor para a área em que lhe é permitido discutir sobre a descrição daquele módulo.

### 3.2. A ferramenta FolioVIEWS

A ferramenta de hipertexto escolhida é um co-editor [Foli93] chamado FolioViews, versão 3.0. O FolioViews armazena e gerencia seus hipertextos em arquivos denominados Infobases. Uma Infobase é composta por registros, que constituem a menor unidade de informação manipulável. Os registros podem ser agrupados e/ou associados a níveis hierárquicos, permitindo a organização e estruturação da informação na Infobase.

Uma grande facilidade do FolioViews é a sua capacidade de efetuar buscas na Infobase. Cada palavra da Infobase é indexada, permitindo maior eficiência e rapidez. As consultas ou "queries" geram uma excursão pela infobase, com passagem por todas as ocorrências encontradas. A organização do conteúdo de uma Infobase em grupos e níveis acrescenta à facilidade de busca um poder ainda maior, permitindo que as consultas possam se restringir a determinado grupo ou nível.

O FolioViews provê cinco tipos de link, que permitem ao leitor navegar dentro de uma ou mais Infobases. Os *jump links* transportam o leitor de um ponto a outro da Infobase. Os *pop-up links* mostram uma pequena janela onde pode haver textos ou gráficos. Os *object links* abrem uma janela onde haverá um gráfico do tipo Bitmap ou Windows Metafile. *Program links* iniciam a execução de uma aplicação externa. E finalmente, há os *Query links*, que provocam a execução de uma consulta.

O FolioViews é uma ferramenta multi-usuário. Ele suporta até 125 usuários autores acessando uma única Infobase ou cópias individuais da mesma, simultaneamente, para editar, anotar ou marcar seu conteúdo. Por outro lado, um número ilimitado de usuários pode usar simultaneamente uma mesma Infobase, no modo de leitura. Uma Infobase pode estar ou não disponível para múltiplos usuários. Esta condição é definida no momento da abertura de uma Infobase, quando o usuário pode torná-la privada ao desligar a opção multi-usuário ("multi-user"). Quando múltiplos usuários estão acessando e modificando uma Infobase simultaneamente, o FolioViews "prende" o registro em alteração ao usuário que o está alterando, proibindo os demais usuários de alterá-lo.

A cópia individual de uma Infobase ("shadow file") permite ao usuário ter um acesso particular à mesma. Nesta cópia o usuário pode fazer anotações e marcações, mudar estilos e editar o conteúdo da Infobase sem precisar compartilhar estas alterações com os demais usuários da mesma. Cada Infobase pode ter um número ilimitado de cópias individuais, permitindo que um grupo de pessoas trabalhe sobre uma mesma Infobase simultaneamente, e mantenha em separado suas anotações pessoais.

Para fins de segurança, o FolioViews provê um controle de acesso às suas Infobases. Além de permitir a associação de senhas a usuários individuais ou a grupos de usuário com relação a uma Infobase, é possível ainda, definir que direitos terão os usuários de cada senha. Por exemplo, é possível negar o direito de edição e permitir o direito de anotação e marcação do texto, a determinado grupo de usuários de uma Infobase.

O FolioViews oferece uma linguagem de marcação de textos que permite a importação de textos comuns para o formato de suas Infobases. A ferramenta ExEM gera arquivos para a criação de duas Infobases no ambiente do FolioViews: Projeto e Discute. Para cada Infobase a ser criada, a ExEM gera dois arquivos, um com a informação que constituirá a Infobase (.FFF) e outro com a definição dos estilos (formatos), níveis e objetos utilizados na estruturação desta informação (.DEF). Uma vez dentro do ambiente do FolioViews, no momento em que o usuário solicitar a "abertura" do arquivo PROJETO.FFF utilizando o tipo "Folio Flat File", a conversão é disparada automaticamente. Ao final da conversão, o usuário passa a manipular um arquivo chamado PROJETO.NFO que corresponde à Infobase com o documento de Projeto na notação HTDN. A figura 5 mostra, na forma textual, como fica um trecho da Infobase gerada a partir dos arquivos PROJETO.DEF e PROJETO.FFF.

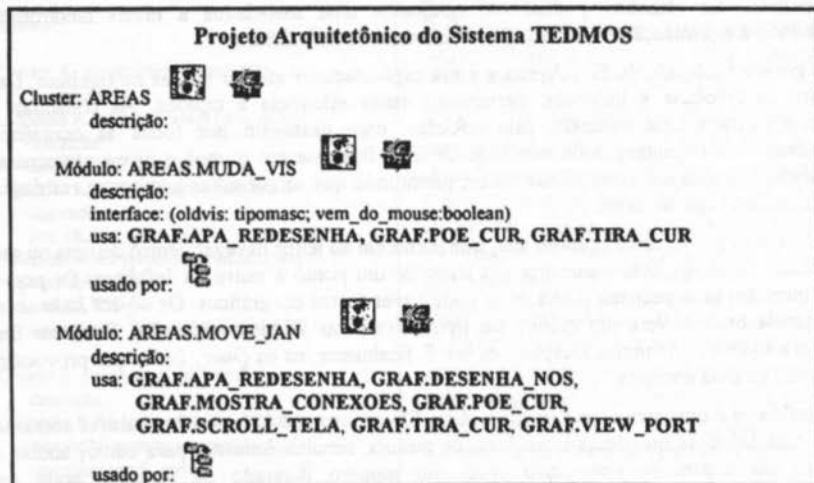


Figura 5: Trecho extraído da Infobase PROJETO.NFO gerada para o sistema TEDMOS<sup>3</sup>.

Os ícones que aparecem na figura 5 são botões que acionam os seguintes links previamente preparados pela ExEM:

<sup>3</sup>O sistema TEDMOS é uma ferramenta CAD, desenvolvida pelo Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro.





- *program link* que dispara o editor de Turbo Pascal da Borland com o fonte correspondente ao módulo ou cluster em que o usuário está posicionado;



- *query link* com uma consulta restrita a Infobase DISCUTE, para encontrar a área de discussão sobre o cluster ou módulo em que o usuário está posicionado;



- *query link* com uma consulta restrita ao nível usa, para obter os registros com os módulos que usam o módulo em que o usuário está posicionado.

Ainda na figura 5, nota-se que a lista de módulos usados está em negrito. Esta evidênciação se deve ao fato de que o nome de cada módulo na lista é na verdade um *query link* com uma consulta que leva à definição correspondente àquele módulo na mesma Infobase.

O FolioViews é muito parecido com os editores de texto conhecidos, o que torna a sua utilização bastante confortável para os integrantes do grupo de Recuperação. Mas além da edição, o FolioViews oferece diversas outras facilidades relativas à sua característica de hipertexto. O bom aproveitamento destas facilidades exige que a equipe que irá utilizá-lo receba um treinamento adequado.

### 3.3. A ferramenta itIBIS

A ferramenta itIBIS, que significa "indented text issue-based information system", foi desenvolvida para permitir o uso independente de hardware ou software, do modelo IBIS. A representação por endentação da estrutura hierárquica deste modelo, pode ser implementada em qualquer editor de textos [YaCo90]. Esta facilidade de implementação foi o principal motivo da escolha desta ferramenta.

Na itIBIS usa-se os códigos I, P, AS e AO para significar respectivamente os elementos Questão ("issue"), Posição, Argumento de Suporte e Argumento de Objeção. Os símbolos

"?", "A" e "-" são utilizados para significar respectivamente, pendência, aprovação/resolução e rejeição. Para identificar o proponente de uma questão, posição ou argumento, extendemos a itIBIS, precedendo cada um destes elementos com o nome do seu proponente. Um exemplo de utilização da itIBIS pode ser visto na figura 6.

\* I (Maria): Qual a notação de argumentação a ser utilizada?  
 - P (ShHa94): DRL  
     AO (Mário): mais complicada de usar  
     AO (Mário): difícil implementação  
 ? P (ShHa94): QOC  
     AO (Mário): pouco conhecida  
 \* P (CoBe87): IBIS  
     AS (YaCo90): fácil de implementar  
     AS (ReE191): simplicidade de uso

Figura 6: Exemplo de utilização da itIBIS.



Infobase PROJETO, utiliza-se uma outra Infobase denominada DISCUTE, onde deverá ocorrer a discussão. A Infobase DISCUTE é criada a partir dos arquivos DISCUTE.DEF e DISCUTE.FFF, gerados pela ferramenta ExEM. Para implementar a itIBIS, foram definidos no arquivo DISCUTE.DEF três níveis: Questão, Posição e Argumento. Cada um destes níveis tem um estilo característico, que permite a visualização da sua hierarquia.

Para iniciar a discussão sobre a complementação do documento de Projeto, uma primeira Questão surge, a partir da análise dos códigos fonte, quando um módulo não possui uma descrição no formato esperado: "Qual a descrição deste módulo?". A ferramenta ExEM é

capaz de encontrar nos códigos fonte, comentários que aparecem nas linhas correspondentes a cada módulo. Estes comentários são a primeira Posição para solucionar a Questão levantada. Como um Argumento a favor desta Posição, sabe-se que os comentários relativos a um módulo podem ajudar bastante na sua descrição. Por outro lado, um Argumento contra esta mesma posição é o fato dos comentários estarem fora de uma apresentação adequada. Este princípio de discussão é lançado pela própria ferramenta ExEM, gerando no arquivo DISCUTE.FFF, os registros necessários a criação da Infobase DISCUTE.

Da mesma forma como é feito para o arquivo PROJETO.FFF, os arquivos DISCUTE.FFF e DISCUTE.DEF são transformados na Infobase DISCUTE.NFO, onde se dará a discussão sobre o documento de Projeto. A figura 7 mostra, na forma textual, como fica um trecho desta Infobase.

**Discussão sobre o Projeto Arquitetônico do sistema TEDMOS**



Cluster: AREAS  

*I (ExEM): Qual a descrição deste cluster?*  
 ? P (ExEM): Esta pode ser a descrição do cluster:  
 ( Funcoes gerais de manipulacao de areas. )

*AS (ExEM): Comentários encontrados no arquivo fonte, no trecho correspondente a este cluster.*

*AO (ExEM): Texto mal escrito e impróprio para ser usado na descrição do cluster.*


Módulo: AREAS.APAGA\_CEL  

*I (ExEM): Qual a descrição deste módulo?*  
 ? P (ExEM): Esta pode ser a descrição do Módulo:  
 ( Apaga a celula atual. )  
 ( Foi originado pela divisao do procedimento Zera\_Cel )  
 ( Adaptacao para ser utilizada com menu do mouse )

*AS (ExEM): Comentários encontrados no arquivo fonte, no trecho correspondente a este módulo.*

*AO (ExEM): Texto mal escrito e impróprio para ser usado na descrição do módulo.*

Figura 7: Trecho extraído da Infobase DISCUTE.NFO gerada para o sistema TEDMOS.

Os ícones  é um *query link* com uma consulta restrita a Infobase PROJETO, para encontrar o módulo correspondente a discussão em que o usuário está posicionado;

Este princípio de discussão deve incentivar a equipe de manutenção do sistema, a iniciar a etapa cooperativa da Recuperação do Projeto Arquitetônico. A partir deste momento a equipe passa a utilizar a itIBIS, dentro do ambiente do FolioViews, para dar prosseguimento à discussão. A principal preocupação que os membros da equipe devem ter, é com a utilização correta dos níveis questão, posição e argumento, ao acrescentar os elementos IBIS de mesmo nome.

A facilidade de consultas do FolioViews traz muitas vantagens, porém impõe uma limitação na implementação da itIBIS. Os símbolos ?, \* e - não são reconhecidos como palavras a consultar. Tornando sem efeito, por exemplo, uma busca sobre as questões que já foram

resolvidas. Por isso, fomos obrigados a substituir estes símbolos pelos caracteres alfabéticos "d", "s" ou "r", respectivamente.

#### 4. Observações e Resultados

Após a implementação do protótipo ARCoPAS, efetuamos uma análise do mesmo, comparando-o com o que se considera um ambiente ideal para a Recuperação de Projeto. Iniciamos também, um experimento, através do qual tem sido possível identificar as vantagens e deficiências apresentadas pelo protótipo. A próxima versão do ARCoPAS incorporará melhorias que deverão eliminar as deficiências encontradas.

##### 4.1. O Ambiente Ideal

Considerando as características desejáveis, identificadas por Fletton e Munro [FIMu88], para um ambiente de Recuperação de Projeto, podemos observar que o ARCoPAS satisfaz a muitas delas:

✓ **Incremental:** As ferramentas FolioViews e itIBIS encarregam-se de tornar o ARCoPAS incremental, permitindo que o conhecimento dos especialistas seja estruturada e assincronamente capturado. A estruturação da informação mantém a coerência durante o processo de Recuperação, evitando que os especialistas se dispersem. O assincronismo dispensa a necessidade de reuniões frequentes, minimizando a concorrência com as atividades mais urgentes dos especialistas.

✓ **Atualização Casual:** O ARCoPAS encontra-se no próprio ambiente de trabalho dos responsáveis pela tarefa de Recuperação de Projeto, possibilitando-lhes um fácil acesso. Mas além disso, o ARCoPAS proporciona o desenvolvimento de uma discussão em grupo, tornando o processo de obtenção da informação quase natural.

✓ **Garantia de Qualidade:** O ARCoPAS não provê esta característica, porém, a utilização de uma ferramenta de apoio à discussão sobre o Projeto Arquitetônico (itIBIS), é um passo importante na direção da melhoria da qualidade do mesmo.

✓ **Trabalho Cooperativo:** O ARCoPAS pode ser encarado como uma tecnologia de suporte ao trabalho cooperativo. Segundo a classificação de groupwares apresentada por Olso et al [Olso93], o ARCoPAS é ao mesmo tempo um sistema de apoio a reuniões e de co-autoria, pois integra uma ferramenta de apoio à discussão (itIBIS), e uma ferramenta de co-autoria (FolioViews).

✓ **Gerência de Configuração:** A facilidade de utilizar cópias de uma Infobase ("shadow files"), permite que se trabalhe em novas versões destes documentos. Porém, ao contrário do que se poderia esperar, estas cópias não podem ser incorporadas às originais. Na verdade, os "shadow files" são máscaras, que possuem o objetivo de proteger a Infobase original de alterações indesejáveis. Assim, por enquanto, o ARCoPAS deixa a tarefa de gerência de configuração por conta dos seus usuários.

✓ **Integração ao Código Fonte:** Os *program links* do Folio Views, permitem que os usuários do ARCoPAS tenham fácil acesso ao código fonte relacionado a documentação de Projeto. Uma vez dentro da documentação de Projeto, basta acionar um botão, que o ARCoPAS transporta o usuário para o ambiente de programação, onde este pode visualizar imediatamente, o código fonte correspondente.

✓ **Integração com ferramentas automáticas:** O ARCoPAS é um ambiente formado pela integração de duas ferramentas automáticas, o que lhe confere naturalmente esta característica. O Folio Views permite o intercâmbio de arquivos com diversas outras ferramentas, permitindo a importação de vários documentos e a incorporação destes ao documento de Projeto.

✓ **Ocultamento de Informação:** Esta é outra característica provida pelo ARCoPAS. A possibilidade de definição de níveis hierárquicos no Folio Views, permite que os usuários visualizem os documentos de Projeto e de Discussão, com maior ou menor abstração.

#### 4.2. Experimento

Para o primeiro experimento escolhemos o Sistema TEDMOS, que é uma ferramenta CAD para ensino e projeto de Circuitos Integrados [AnOI92], desenvolvida pelo Núcleo de Computação Eletrônica da UFRJ. Considerado como um sistema de médio porte, o TEDMOS possui mais de 57.000 linhas de código escrito em Turbo Pascal para funcionar em ambiente PC/DOS.

A Recuperação do Projeto Arquitetônico do sistema TEDMOS ainda encontra-se em andamento. Durante a sua fase inicial, pudemos observar o que se passou entre dois membros da equipe: Maria e Manoel. Maria é uma analista que participa temporariamente, da equipe do TEDMOS, e Manoel é o gerente desta equipe há alguns anos. Empenhada em conhecer o sistema, Maria utilizou o ARCoPAS para facilitar o seu estudo, enquanto dava início à recuperação do Projeto Arquitetônico do mesmo. Tanto para Maria quanto para Manoel, o primeiro contato com o ARCoPAS não foi precedido por um treinamento formal nas ferramentas que o integram. Após uma explicação rápida, ambos familiarizaram-se com as funções básicas do ambiente.

Observamos que entre as facilidades mais usadas do ARCoPAS, estão o acesso imediato ao código fonte, a navegação pela estrutura modular e o mecanismo de consulta. O botão que transporta o usuário para o código fonte, foi muito utilizado, confirmando a importância da integração entre os documentos do sistema. A navegação pela estrutura modular foi útil para orientar os estudos de Maria. No entanto, durante um dos passeios pela estrutura modular do sistema, Maria acusou problemas de desorientação. Estes problemas logo foram sanados pela facilidade "show trail" do FolioViews. Ao acionar o botão "show trail", situado na barra de botões, Maria pôde ver todo o caminho que havia percorrido na Infobase até então, retornando rapidamente ao ponto desejado.

O grande número de clusters do sistema TEDMOS (128) deixou Maria em dúvida quanto a escolha do cluster por onde iniciar seu estudo. Sabendo de antemão, que o CIRCO era um dos principais clusters do sistema, Maria o escolheu. Esta desorientação inicial, no entanto, mostra que a clusterização existente no sistema TEDMOS, não foi suficiente para gerar sua árvore conceitual automaticamente. Maria precisaria encontrar níveis intermediários de clusterização, entre o sistema e os clusters, de maneira a facilitar a sua compreensão do sistema e a escolha de um ponto de partida. Caso Maria houvesse solicitado a ajuda de Manoel, ele poderia criar os níveis intermediários, agrupando os clusters em "super clusters", segundo a sua experiência no sistema.

Maria já havia iniciado seu estudo, colocando algumas posições sobre a descrição dos módulos do cluster CIRCO, quando Manoel acessou a Infobase da Discussão. Com o intuito de ajudá-la, Manoel utilizou o mecanismo de consulta do FolioViews, para encontrar na Infobase, as posições em que Maria possuía dúvida. No entanto, a utilização deste mecanismo não é trivial, dificultando a ação de Manoel. Após uma breve consultoria a um usuário do FolioViews, Manoel pôde localizar as dúvidas de Maria.

Ao tentar sanar as tais dúvidas, Manoel precisou consultar arquivos de dados do sistema. Como a ferramenta ExEM só considerou os arquivos fonte do sistema, os arquivos de dados não constam nas Infobases como módulos do sistema. Assim, para alcançá-los foi preciso utilizar o botão de acesso ao ambiente de programação, pertencente a um módulo de programa fonte. Uma vez no Turbo Pascal, foi possível abrir e visualizar o arquivo de dados

correspondente. Após estar seguro de suas posições, Manoel as incluiu na Infobase de Discussão.

### 4.3. Melhorias Futuras

Futuramente, pretendemos tornar o protótipo ARCoPAS um ambiente mais completo e flexível, suprimindo as deficiências apresentadas. Uma melhoria prevista para breve, é o tratamento dos arquivos de dados. Além dos arquivos fonte (.PAS), a ferramenta ExEM passará a identificar como módulos, os arquivos de dados usados pelo sistema TEDMOS. A prioridade desta mudança deve-se à requisição feita pelos membros da equipe TEDMOS.

Outra importante melhoria refere-se à extração da árvore conceitual. Como foi observado, a ferramenta ExEM não consegue extrair mais do que três níveis de abstração. Precisamos investigar a possibilidade de encontrar dentro dos programas fonte, alguma pista que determine a existência de níveis intermediários de abstração, com o objetivo de gerar uma árvore mais "alta" e conseqüentemente, mais representativa.

Como foi comentado na sessão anterior, falta pouco para que o ARCoPAS se torne o ambiente ideal. Em uma próxima versão o ARCoPAS pretende prover a garantia de qualidade e o controle de versão dos documentos que gera. Para garantir a qualidade da árvore conceitual gerada, a ferramenta ExEM deverá sugerir mudanças e criar níveis na mesma, baseando-se em algoritmos de clusterização por similaridade [ScPI89]. O ARCoPAS deverá prover meios de fomentar a discussão em andamento, instigando a participação dos membros do grupo, aumentando a qualidade não só do objeto em discussão, como da própria discussão.

Para prover uma completa gerência de configuração, talvez seja necessário mudar a ferramenta de hipertexto (FolioViews). Porém, mesmo com esta ferramenta, podemos garantir a correspondência entre as versões da documentação de Projeto e dos códigos fonte, sendo necessário fazer apenas algumas adaptações no ARCoPAS.

Notamos também, a necessidade da definição de papéis para a tarefa de recuperação de Projeto. O mecanismo de segurança do FolioViews permite a identificação dos usuários, a formação de grupos e a restrição dos direitos de cada um. Esta facilidade possibilita uma fácil implementação de papéis. Entretanto, uma definição prematura, poderia prejudicar as pesquisas com relação ao aspecto social do grupo em estudo. A partir de experiências com a utilização do ARCoPAS, esperamos obter conhecimento suficiente para uma definição adequada dos papéis.

O tratamento de uma única linguagem de programação (Pascal), é uma restrição que pretendemos eliminar. É nossa intenção, reprojeter a ferramenta ExEM, utilizando para isso o Projeto Arquitetônico recuperado da mesma. O próprio ARCoPAS será usado para o reprojeto da ExEM, mudando apenas o objetivo da discussão. Neste reprojeto, pretendemos prever o tratamento de outras linguagens tais como C e COBOL. Entretanto, nada impede que um outro protótipo da ExEM, dedicado a outra linguagem de programação, seja implementado a partir da reutilização do seu Projeto Arquitetônico. Enquanto a ExEM genérica não fica pronta, as equipes de manutenção interessadas poderão implementar por exemplo, uma ExEM<sub>C</sub> ou uma ExEM<sub>COBOL</sub>, adaptando o ARCoPAS às suas necessidades.

Para que uma equipe aceite utilizar uma nova ferramenta, é preciso contar com o apoio da gerência da equipe, e prover treinamento e suporte técnico a esta ferramenta [YaCo90]. Em nosso primeiro experimento, contamos com o apoio gerencial, porém não oferecemos um treinamento adequado, contando apenas com a facilidade de uso das ferramentas. Em futuras experiências, pretendemos planejar mais elaboradamente, a introdução do ARCoPAS.

## 5. Conclusão

Este trabalho apresenta o ARCoPAS como um protótipo de ambiente para a Recuperação do Projeto Arquitetônico de Sistemas. O desenvolvimento deste protótipo baseou-se na idealização de um ambiente que usa as tecnologias de Hipertexto e CSCW. Para implementar este ambiente, o ARCoPAS integrou harmonicamente três ferramentas: ExEM, FolioViews e itIBIS. Atualmente, a primeira versão do ARCoPAS está sendo utilizada para a recuperação do Projeto Arquitetônico de um sistema real. Este experimento tem possibilitado a identificação de melhorias, que serão incorporadas em uma próxima versão do ARCoPAS.

A introdução da característica cooperativa a uma ferramenta de apoio à Recuperação de Projeto traz uma importante contribuição para a Engenharia Reversa, já que poucas ferramentas a incluem. Primeiramente, viabilizamos a participação humana, possibilitando a captura do conhecimento dos especialistas (equipe de manutenção), e enriquecendo com isto, o produto recuperado. Em segundo lugar, a distribuição da responsabilidade desta tarefa entre os membros da equipe de manutenção, torna-a mais interessante e menos desgastante. Por último, a desobrigação de uma cooperação síncrona, pouco prejudica as principais tarefas desta equipe.

Os ambientes de apoio à Engenharia Reversa devem buscar a integração com ambientes de apoio à Engenharia Progressiva. Esta integração facilita a introdução de novas metodologias nos ambientes de manutenção e desenvolvimento de software. O protótipo ARCoPAS introduz uma metodologia de Reprojeto Cooperativo de sistemas. Da mesma maneira que o ARCoPAS permite que se registre a discussão sobre a Recuperação do Projeto, pode registrar também a discussão sobre o Reprojeto de sistemas. Uma vez terminada a recuperação do Projeto de um sistema, os membros da equipe já estarão bastante familiarizados com a utilização da ferramenta itIBIS. O Reprojeto deste sistema é uma consequência natural e indolor, pois esta mesma ferramenta poderá ser usada para capturar as decisões de Reprojeto.

Finalmente, o acompanhamento do uso do protótipo ARCoPAS, pode gerar informações interessantes para um estudo etnográfico sobre as equipes de manutenção de software e a tarefa de recuperação de Projeto. Os resultados obtidos com este estudo, poderão contribuir para o enriquecimento das pesquisas na área de CSCW. Através da realização de alguns experimentos, esperamos recolher material suficiente para a construção de um ambiente definitivo.

## Bibliografia

- [AnOl92] "A CAD system for teaching the design of VLSI Circuits - Status and Evolution" - M. L. Anido & C. E. Oliveira, PANEL'92 - 18th Latin American Informatics Conference, Las Palmas de G.C., Espanha, 1992;
- [Bigg89] "Design Recovery for Maintenance and Reuse" - T J Biggerstaff, Computer, pag 36-49, julho/1989;
- [Borl93] "Borland Pascal User's Guide" - Borland, 1993;
- [ChCr90] "Reverse Engineering and Design Recovery: A Taxonomy" - E. Chikofsky & J. Cross II, IEEE Software, janeiro/1990;
- [CoBe87] "gIBIS: A Hypertext Tool for Team Design Deliberation" - J. Conklin & M. Begeman, Hypertext'87, novembro/1987;
- [FiMu88] "Redocumenting Software Systems using Hypertext Technology" - N. Fletton & M. Munro, Proceedings of the Conference on Software Maintenance, 1988;



- [Foli93] "Folio VIEWS™ Personal Electronic Publishing Software" - Folio Corporation, 1993;
- [Ghez91] "Fundamentals of Software Engineering" - Carlo Ghezzi & Mehdi Jazayeri & Dino Mandrioli, Prentice Hall International editions, 1991;
- [Lafu90] "Panel on Software Re-Engineering" - G. Lafue, IEEE Proceedings da 12a Conf. Internacional de Engenharia de Software, 1990;
- [Olso93] "CSCW - Research Issues for the 90's" - J. Olson & S. Card & T. Landauer & G. Olson & T. Malone & J. Legget, User Interface Strategies'94, University of Maryland, dezembro/1993;
- [ReEl91] "rIBIS: a real-time group hypertext system" - G. Rein & C. Ellis, International Journal of Man-Machine Studies, vol. 34, 1991;
- [ScPl89] "Cross References are Features" - R. Schwanke & M. Platoff, Proceedings of 2nd International Workshop on Software Configuration Management, outubro/1989;
- [SeWaCh93] "Challenges to the Field of Reverse Engineering" - P. Selfridge & R. Waters & E. Chikofsky, proceedings da Working Conference on Reverse Engineering, Baltimore, MD, maio/1993;
- [ShHa94] "Argumentation-Based design rationale: what use at what cost?" - S. Shum & N. Hammond, International Journal of Human-Computer Studies, n.40, 1994;
- [YaCo90] "Report on a Development Project Use of an Issue-Based Information System" - B. Yakemovic & J. Conklin, Proceedings da CSCW'90, outubro/1990;