

# PUC

---

Série: Monografias em Ciência da Computação  
Nº8/88

EXTENSÕES PROPOSTAS SOBRE O MONITOR PARA APOIO À  
PROGRAMAÇÃO EM LÓGICA

Jeferson Ferreira Soares  
Paula Ypiranga dos Guarany's

Departamento de Informática

---

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO  
MARQUÊS DE SÃO VICENTE, 225 – CEP 22453  
RIO DE JANEIRO – BRASIL

PUC/RJ - Departamento de Informática

Série: Monografias em Ciência da Computação, nº8/88

Editor: Paulo A. S. Veloso

Setembro de 1988

EXTENSÕES PROPOSTAS SOBRE O MONITOR PARA APOIO À  
PROGRAMAÇÃO EM LÓGICA \*

Jeferson Ferreira Soares  
Paula Ypiranga dos Guaranys

\* Projeto parcialmente financiado pelo CNPq e pela SID Informática.  
Apresentado pelo prof. Sérgio Eduardo Rodrigues de Carvalho.

Responsável por publicações:

Rosane Teles Lins Castilho  
Assessoria de Biblioteca, Documentação e Informação  
PUC/RJ-Depto. de Informática  
Rua Marquês de São Vicente, 225 - Gávea  
22453 - Rio de Janeiro, RJ  
BRASIL

## RESUMO

Este trabalho descreve uma proposta de alteração na especificação do sistema "MPL - Um Monitor para Apoio à Programação em Lógica", cuja apresentação é feita em [Guar 88].

Existem dois objetivos principais para as modificações propostas: a definição de um sistema de interface homem-máquina que possibilite o uso confortável e eficiente do sistema por usuários com graus de experiência distintos, e a introdução de um dispositivo de apoio ao mecanismo de herança de domínios de parâmetros.

**PALAVRAS-CHAVES:** interface dirigida por menus, herança de domínios de parâmetros, mecanismo de explicação, plano de ação.

## ABSTRACT

This work describes a proposal for the alteration of the specification of the system "MPL - An aid Monitor for Logical Programming" [GUAR 88].

The two main goals for the proposed modifications are: the definition of a human-machine interface system to facilitate the comfortable and efficient use of the system by users with different levels of expertise, and the introduction of a help tool to the domain inheritance mechanisms of the parameters.

**KEYWORDS:** menu-driven interface, parameter domain inheritance, explanation, action plan.

## SUMARIO.

LISTA DE FIGURAS .....	11
1. INTRODUÇÃO .....	1
2. FUNCIONALIDADE / INTERFACE .....	4
2.1 O Mecanismo de Interação com o Usuário .....	4
2.2 A Interação Sistema X Usuário .....	5
2.2.1 Edição .....	5
2.2.2 Execução .....	11
2.2.3 Operações com Arquivo .....	11
2.2.4 Sair .....	14
3. EXPLICAÇÃO .....	15
3.1 A Utilização da Função: "Explicação" .....	16
3.2 Representação de Dados e Funcionamento .....	19
3.2.1 A Representação Interna .....	19
3.2.2 O Funcionamento da Explicação .....	21
3.3 A Reutilização de Históricos de Herança .....	22
4. CONCLUSÃO .....	23
REFERENCIAS BIBLIOGRAFICAS .....	24

## LISTA DE FIGURAS.

- 1 - TELA DE APRESENTAÇÃO.
- 2 - TELA DO MODO DE EDIÇÃO.
- 3 - TELA DAS FUNÇÕES DISPONÍVEIS AO MODO DE EDIÇÃO.
- 4 - TELA DE EXIBIÇÃO DE UMA CONSULTA AOS DOMÍNIOS DOS PARÂMETROS DE UMA REGRA.
- 5 - TELA DEMONSTRATIVA DE UM PEDIDO DE EXPLICAÇÃO SOBRE DOMÍNIOS DOS PARÂMETROS DE UMA REGRA.
- 6 - TELA DE EXIBIÇÃO DO HISTÓRICO DA HERANÇA DO DOMÍNIO DE UM PARÂMETRO.
- 7 - TELA DEMONSTRATIVA DA SELEÇÃO DE UM PASSO NO HISTÓRICO DA HERANÇA DO DOMÍNIO DE UM PARÂMETRO.
- 8 - ESTRUTURA INTERNA DA REPRESENTAÇÃO EXISTENTE PARA OS DOMÍNIOS DOS PARÂMETROS DE UMA REGRA.

## 1. INTRODUÇÃO.

A versão do sistema MPL descrita em [Guar 88] tem como alvo o usuário iniciante. Esta orientação é fundamental na determinação do tipo de interface disponível naquela versão do sistema.

Buscou-se guiar o usuário em dois níveis distintos de atuação: no que diz respeito ao crescimento do programa sendo construído é imposta uma estratégia descendente de construção; sobre o aspecto do tratamento sintático do mesmo optou-se por permitir que apenas uma operação de edição esteja ativa a cada vez.

O usuário deve explicitar a operação desejada e o ponto do texto no qual deseja operar. A partir daí apenas este trecho do programa estará disponível para edição até que a operação corrente seja completada. Se o resultado da operação implica em construção sintática inválida, o mesmo é rejeitado e o usuário é notificado. Caso este resultado gere alguma inconsistência em relação ao conteúdo anterior do texto, há dois procedimentos possíveis:

- a) Rejeição e notificação do usuário.
- b) Detecção e marcação do trecho inconsistente para futura correção pelo usuário.

Esta forma de interação induz um processo de trabalho no qual cada ação sobre o texto do programa é realizada isoladamente e tem suas consequências, no nível sintático, imediatamente avaliadas e informadas ao usuário.

O esquema de uso descrito acima mostra-se bastante adequado para o usuário iniciante e ainda inexperiente na construção do

tipo de aplicações suportadas pelo MPL. O principal ônus desta maneira de trabalhar é o surgimento de um grande número de passos intermediários, dados na realização de cada operação. Isto torna o sistema massante para o usuário mais experiente que, normalmente, pode e prefere trabalhar com mais liberdade, cometendo poucos erros detectáveis pelo sistema de apoio à construção de aplicações.

Uma alteração interessante sobre a versão atual do MPL é, portanto, a mudança de seu esquema de interação com o usuário no nível da edição de textos. O objetivo é atingir-se um tipo de interação que seja um meio termo entre as necessidades de atendimento a usuários de diferentes graus de experiência. Para os iniciantes é importante manter disponíveis os mecanismos de direcionamento da construção e detecção de inconsistências e erros sintáticos nos programas. Para os usuários mais experientes é interessante prover uma interface que lhes permita trabalhar com o menor número possível de passos intermediários. O novo esquema de interface é discutido no capítulo 2.

Uma importante função automática disponível no MPL é a que implementa os mecanismos de herança de domínios de parâmetros.

Para complementar o apoio dado por esta função criou-se um mecanismo auxiliar, que tem por objetivo "explicar" ao usuário o processo de herança de domínios sofrido por um parâmetro qualquer de relação.

O capítulo 3 especifica o comportamento e a maneira de utilização do dispositivo de "explicação" de herança de domínio



de parâmetros. A estrutura de dados usada para a representação interna das "explicações" e a possibilidade de reutilização das mesmas em diferentes sessões de trabalho, sobre uma mesma aplicação, são também comentadas neste capítulo.

O capítulo 4 apresenta a conclusão do trabalho.

## 2. FUNCIONALIDADE/INTERFACE.

A seguir é descrita a nova proposta de interface para o MPL. Informações detalhadas sobre as funções disponíveis no sistema e eventualmente citadas nesta descrição podem ser obtidas em [GUAR 88].

### 2.1 - O Mecanismo de Interação com o Usuário:

Dentre as alternativas disponíveis de mecanismos de interação com o usuário optou-se pelo uso de menus interativos do tipo "pull-down". A principal justificativa para a escolha é o fato de que com esta técnica foi possível manter a estratégia e as funções de monitoramento convenientes para usuários iniciantes. Além disso pode-se oferecer um esquema simples e rápido de escolha de funções, satisfatório também para os usuários mais experientes.

A escolha é imediata nos casos de menus com um só nível hierárquico de opções. O cursor tem movimentos para a direita e esquerda, no caso de listas de opções exibidas horizontalmente e para cima e para baixo no caso de listas de opções verticais. Posiciona-se o cursor sobre a opção desejada, geralmente marcada com técnica de verso-reverso de tela, caracterizando-se assim a escolha.

Quando há mais de um nível hierárquico de opções a ativação de uma alternativa dispara uma nova lista de escolhas que terá o mesmo tratamento descrito acima.

A qualquer momento uma tecla de escape leva o usuário para o nível hierárquico imediatamente anterior.

## 2.2 - A Interação Sistema X Usuário:

Na tela inicial, figura 1, o usuário tem as seguintes opções de uso:

- . edição,
- . execução,
- . arquivos,
- . saída.

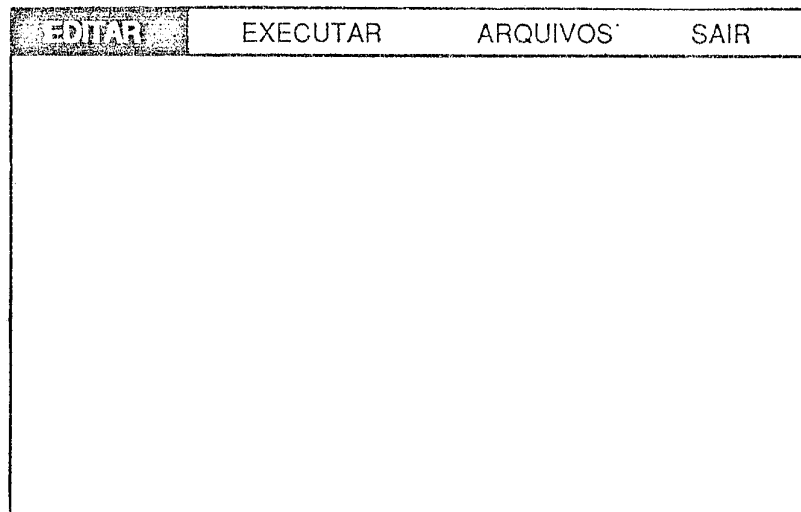


FIGURA 1

### 2.2.1 - Edição:

A opção "edição", figura 2, deixa disponível o modo de edição do sistema.

O usuário escolhe, a partir de um diretório, nomes de regras específicas a serem editadas. As regras são divididas em dois blocos: "definidas" e "por definir".

EDITAR		EXECUTAR	ARQUIVOS	SAIR
POR DEFINIR		DEFINIDAS		
jogada-computador		jogada		
jogada-usuário				

FIGURA 2

Regras "definidas" são aquelas para as quais já há pelo menos uma cláusula descrita no programa.

Estão no bloco das "por definir" as regras para as quais não há cláusulas definidas até o momento, mas já há pelo menos uma chamada em algum ponto do programa.

Para cada um dos blocos há um conjunto de funções específicas disponíveis.

Escolhido o nome da regra a ser editada as funções disponíveis são apresentadas na janela de um menu do tipo pop-up (Fig. 3).

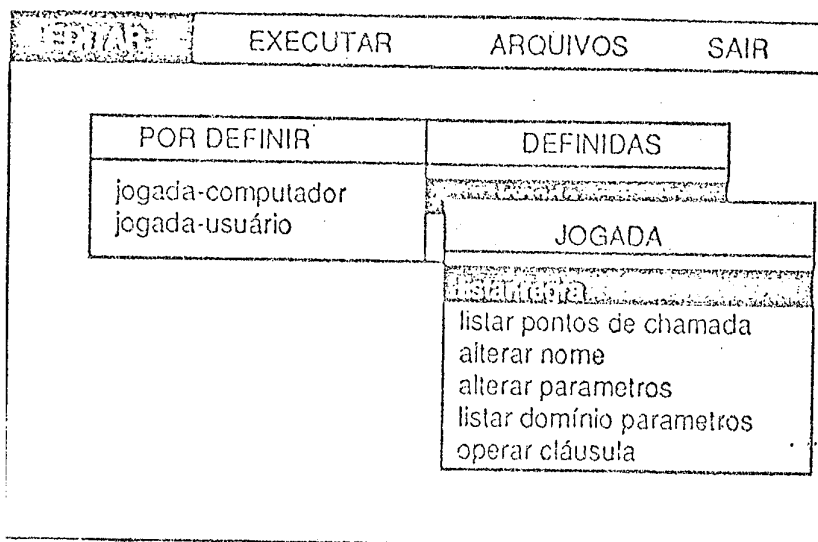


FIGURA 3

Para regras "definidas" são apresentadas as seguintes opções:

. **listar a regra:**

Localiza a regra desejada e exibe o texto do programa a partir da mesma.

. **listar pontos de chamada:**

Localiza todas as cláusulas do programa onde haja chamada para a regra escolhida exibindo-as uma a uma mediante solicitação do usuário.

. **alterar nome:**

Com o texto da regra exibido na tela normal de edição o usuário especifica, em uma janela de diálogo, o novo nome desejado para a mesma. Após ser consistido, este novo nome

passa a substituir todas as ocorrências do nome anterior da regra, tanto na definição quanto nos pontos de chamada.

. **alterar parâmetros:**

Possibilita alterações na lista de parâmetros de uma regra. São permitidas alterações na aridade e no domínio dos parâmetros das regras. O texto do programa, a partir da regra desejada, é exibido na tela normal de edição e uma janela especial, também de edição, é aberta apresentando o nome da regra e a lista de domínios dos parâmetros da mesma. O usuário pode fazer as alterações que deseja sobre esta lista, que será consistida pelo sistema e em caso de ausência de problemas, substituirá a anterior. Nos casos em que a nova lista de parâmetros contenha inconsistências sintáticas uma mensagem de erro é enviada e a operação é anulada.

. **listar domínios de parâmetros:**

O texto da regra é exibido na tela normal de edição e uma janela auxiliar apresenta o nome da regra e a lista de domínios de parâmetros da mesma, para consulta do usuário (Fig. 4).

. **operar cláusula:**

Ativa o menu com as seguintes opções de funções a nível de cláusulas:

- **incluir cláusula:**

Possibilita a definição de mais uma cláusula em uma regra já existente. Nesta opção o sistema lista a regra completa e em uma janela de diálogo,

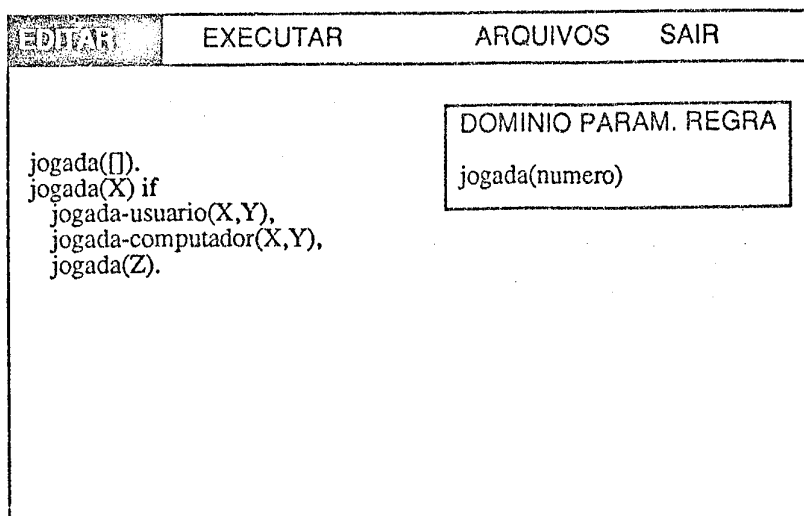


FIGURA 4

requisita ao usuário a posição que ocupará a nova cláusula, em relação às anteriores. A partir deste ponto o usuário fornece a nova cláusula seguindo o padrão de trabalho utilizado na tela normal de edição.

**- excluir cláusula:**

Determina a retirada de uma cláusula completa de uma regra após escolha por parte do usuário. A posição relativa da cláusula na regra será fornecida em uma janela de diálogo pelo usuário.

**- alterar cláusula:**

Permite ao usuário introduzir modificações na cláusula a nível de assertiva. Inicialmente é

exibido o texto da regra e o usuário seleciona o a cláusula a sofrer a alteração, como nos casos anteriores. Depois é exibida apenas a cláusula em questão. Em seguida as operações possíveis com as assertivas serão apresentadas em uma outra janela de menu. Estas operações são:

. incluir assertivas:

O procedimento é semelhante ao de incluir uma cláusula. Neste caso o usuário fornecerá apenas uma assertiva de cada vez.

. excluir assertiva:

Processo semelhante ao de exclusão de cláusula.

- listar domínios dos parâmetros:

Exibe, em uma janela à esquerda da tela, a cláusula em estudo. À direita é apresentada uma tabela que relaciona, para cada ocorrência de parâmetro, seu nome e domínio. Os domínios podem ter sido herdados ou definidos. A existência de inconsistências sobre domínios é constatada pelo usuário com o auxílio desta tabela.

As opções de edição disponíveis para regras "por definir" são, basicamente, as mesmas das "definidas". A função "listar a regra", naturalmente, não faz sentido neste caso. Em troca tem-se a função "definir regra" que, através de diálogo, solicita ao usuário a definição da primeira cláusula de regra em questão.



### 2.2.2 - Execução:

Esta função tem como objetivo fornecer ao usuário informações para que este possa preparar as pré-condições necessárias para o teste de uma determinada regra.

Uma vez ativada a função "execução" uma janela de edição é aberta para que o usuário forneça o nome da regra a ser testada. Feito isto é montada e exibida, uma tabela com as seguintes informações sobre a regra:

- a) Nomes de todos os descendentes da mesma, diretos e indiretos, para os quais ainda não haja definição. São os "descendentes por definir".
- b) Nomes de todos os descendentes (incluindo a própria regra) em cuja definição haja inconsistência quanto ao domínio de parâmetros: "descendentes inconsistentes".
- c) Nomes de todos os fatos, instanciados pela própria regra, ou por algum descendente seu, necessários para que o teste possa ser realizado: "fatos necessários".

### 2.2.3 - Operações com Arquivo:

Com a seleção da opção "arquivos" será apresentado ao usuário um menu com as seguintes alternativas:

ler:

Abre-se uma janela pedindo o nome do arquivo a ser lido. A seguir, o arquivo indicado, contendo um programa

fonte codificado em TURBO PROLOG, é lido de disco e colocado disponível para edição e monitoramento.

Vale ressaltar que o programa citado não precisa, necessariamente, ter sido gerado pelo MPL. Entretanto, ao ser colocado disponível para edição as informações necessárias ao seu monitoramento são geradas automaticamente.

#### gravar e continuar:

Esta função, ao ser ativada, abre uma janela de edição para que o usuário especifique o nome do arquivo no qual o programa editado será gravado. O nome dado será sempre complementado com a extensão ".PRO".

Antes da gravação o usuário determina ainda, sempre através de escolha em menus, se deseja que as relações que compõem seu programa sejam organizadas em ordem alfabética na versão gravada do mesmo.

E realizada a gravação e o MPL retorna ao ponto em que estava antes da ativação desta função.

#### gravar e sair:

A ativação desta função produz um comportamento semelhante ao da opção "gravar e continuar". A diferença é que antes da realização da operação de gravação, é oferecida ainda a possibilidade de geração (ou atualização) de um arquivo denominado "ARQUIVO DIAGNOSTICO" contendo algumas das informações que retratam o estado atual do programa em relação aos itens

definições ajustadas primeiro, o número de alterações sobre as regras é minimizado e o número de regras prontas para teste é maximizado em cada rodada de correções.

A concepção dos programas continua tendo uma orientação descendente. Os ajustes e correções sobre uma dada configuração do grafo semântico são feitos da base para o topo, se forem seguidas as sugestões do plano de ação gerado.

A geração deste arquivo é automática e seu nome é composto pelo nome principal do arquivo que contém o programa fonte seguido da extensão "DGN".

#### 2.2.4 - Sair.

São oferecidas duas possibilidades no menu de opções de saída do ambiente do MPL:

. abandonar edição:

O ambiente é desativado sem que o arquivo em edição seja gravado em disco.

. gravar e sair:

Procedimento igual ao descrito no item 2.2.3, "gravar e sair", acima.

monitorados pelo MPL.

Estas informações são:

- a) dados sobre partes do programa prontas para teste.
- b) um plano de ação sugerindo atividades a serem desempenhadas pelo usuário com o objetivo de completar o programa e corrigir inconsistências, detectadas nos domínios de parâmetros.

A estratégia utilizada para a montagem da sequência de atividades sugeridas é: percorre-se o grafo semântico do programa em profundidade. Para cada nó, representando uma relação do programa, é sugerido que sejam definidos seus descendentes e em seguida, resolvidas as inconsistências de domínios de parâmetros existentes.

Justifica-se a escolha desta estratégia da seguinte maneira: o percorrimento escolhido implica em começar o tratamento da definição e correção das regras a partir do nível das folhas. Assim, dada uma certa configuração do grafo semântico do programa, busca-se obter, em primeiro lugar, o maior número possível de regras básicas (nos níveis mais baixos do grafo) definidas e livres de inconsistências. Gradualmente as regras situadas nos níveis mais altos do grafo são também tratadas. Como as regras básicas têm suas

### 3 - EXPLICAÇÃO.

A "explicação" é feita através da descrição de todos os passos percorridos durante o processo de herança de domínio do parâmetro. São indicados, para cada passo do processo: a origem do domínio herdado e o sentido da herança ocorrida.

Um parâmetro herdeiro é um parâmetro que herda um domínio vindo de um outro parâmetro. Define-se a origem do domínio como sendo o ponto do programa em que é definido o parâmetro cujo domínio será atribuído ao parâmetro herdeiro. As heranças podem ocorrer em um de três sentidos possíveis:

a) conclusão de cláusula ---> assertiva

Ativada quando há, em alguma assertiva de uma cláusula, algum nome de parâmetro que tenha domínio indefinido e que também ocorra na conclusão da cláusula, tendo nesta um valor definido para seu domínio.

b) assertiva ---> assertiva

Ativada quando o mesmo nome de parâmetro ocorre em duas ou mais assertivas de uma mesma cláusula tendo ao menos uma ocorrência com domínio definido e uma com domínio indefinido. Para haver herança, neste caso, o parâmetro em questão não pode ocorrer, com domínio definido, na conclusão da cláusula.

c) assertiva ---> conclusão

Ativada quando o nome de parâmetro ocorre com domínio definido em alguma assertiva e com domínio ainda indefinido na conclusão da mesma cláusula.

E importante observar que, para cada um dos três sentidos de herança mencionados, as ocorrências de domínios de parâmetro com valor definido podem ter sido especificadas diretamente pelo usuário ou estabelecidas anteriormente através do mecanismo de herança.

No caso de haver mais de uma possibilidade de solução para um passo de herança é utilizado o sistema de prioridades especificado em [Guar 88].

### 3.1 - A Utilização da Função: "Explicação":

O mecanismo de "Explicação" estará disponível nos seguintes momentos, durante a interação do usuário com o MPL:

a) sempre que o usuário solicitar a exibição da tabela de DOMÍNIOS DE PARÂMETROS (ver operação "listar domínios de parâmetro) de uma dada regra ou de alguma cláusula específica de alguma regra.

b) após a definição de uma cláusula de uma regra, caso o sistema solicite a confirmação do valor do domínio de algum parâmetro da conclusão da mesma.

Uma vez realizada qualquer das consultas acima será apresentado ao usuário um menu com as opções: "continuação" e "explicação".

Na primeira delas o sistema volta a apresentar a tela inicial de trabalho exibindo em seu topo o menu principal.

Se a consulta realizada foi sobre os domínios dos parâmetros

de uma regra será exibido um menu que apresenta como opções de escolha os domínios destes parâmetros (ordenados por sua posição relativa na definição da regra), figura 5.

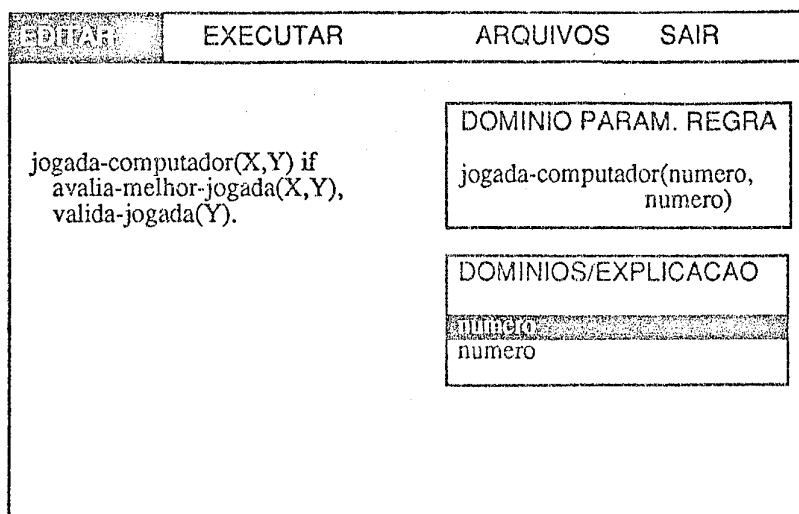


FIGURA 5

Para a opção escolhida neste menu será exibido o histórico com os passos de herança que determinaram o tipo de seu domínio (Figura 6).

Cada um desses passos, por sua vez, pode ser selecionado, através de menu do tipo pop-up. Neste caso a cláusula ou fato mencionado no passo escolhido será exibido em uma janela especialmente aberta para este fim. Veja figura 7.

Nas duas outras possibilidades de consulta citadas a cláusula em questão é apresentada ao usuário, na forma de menu do tipo "pop-up", para que seja determinado o elemento (conclusão ou

assertiva) no qual encontra-se o parâmetro a ser estudado. A partir daí o procedimento é igual ao descrito no parágrafo anterior.

EDITAR	EXECUTAR	ARQUIVOS	SAIR
HISTORICO/HERANÇA			
JOGADA-COMPUTADOR (número,número)			
REG-ORI/CLAU-ORI/ASSER-ORI/TIPO-HERANÇA			
-/-/-			

FIGURA 6

EDITAR	EXECUTAR	ARQUIVOS	SAIR
HISTORICO/HERANÇA			
JOGADA-COMPUTADOR (número,número)			
REG-ORI/CLAU-ORI/ASSER-ORI/TIPO-HERANÇA			
-/-/-			
PASSO SELECIONADO			
jogada (X) if jogada-usuário (X,Y), jogada-computador (X,Y), jogada (Z).			

FIGURA 7



### 3.2 - Representação de Dados e Funcionamento:

O mecanismo de explicação de heranças de domínios trabalha sobre um esquema especial de representação interna de informações.

#### 3.2.1 - A Representação Interna:

Em [Guar 88] mostra-se que o MPL usa uma representação interna baseada na estrutura de um grafo para a descrição das aplicações construídas por seu intermédio. Para servir ao mecanismo de explicação de heranças de domínios a estrutura do grafo de representação foi alterada.

A definição dos domínios de parâmetros mencionados em uma regra era feita através da lista ilustrada no quadro da figura 8, que apresentava o domínio de cada parâmetro:

```
TIPO-PARAMETROS(Nome-regra, [Dom-par1, Dom-par2, ...], Num-par).
```

FIGURA 8

No figura acima temos:

Nome-regra: nome da regra em cuja conclusão o parâmetro é mencionado.

Num-par: número total de parâmetros da conclusão da regra.

Dom-par: valor de domínio corrente do parâmetro.

Em função da alteração proposta esta representação passa a ser feita da seguinte forma:

TIPO-PARAMETROS(Nome-regra,[Lista-def1,Lista-def2,...],Num-par).

Cada parâmetro tem a sua "Lista-Def" correspondente, onde:

Lista-def: [Definição1,Definição2,...].

Definição: (Num-ver,Dom,Reg-orig,Clau-orig,Elem-orig,Pos-rel-par).

Os termos mencionados são:

Num-ver: Número que identifica uma definição de domínio para um parâmetro. Um parâmetro pode ter sua definição de domínio alterada ao longo da construção da aplicação a que pertence. Todas as versões de definição do domínio de um parâmetro são gravadas para possibilitar a montagem de seu histórico de heranças de domínio.

Dom: o valor de domínio do parâmetro.

Reg-orig: nome da regra em que se deu o processo de herança.

Clau-orig: cláusula específica de "Reg-orig" onde se dá o processo de herança.

Elem-orig: elemento de "Clau-orig" (pode ser a conclusão ou uma assertiva da cláusula) que contém o parâmetro origem.

Pos-rel-par: posição relativa do parâmetro que dá origem ao domínio herdado.

Nome-regra: nome da regra em cuja conclusão é mencionado parâmetro herdeiro.

Num-par: número total de parâmetros da regra mencionada em Nome-regra.

### 3.2.2 - O Funcionamento da Explicação:

A partir do disparo de um pedido de explicação sobre as heranças de domínio de um parâmetro "X", por exemplo, o procedimento seguido é o seguinte:

. Localiza-se, na estrutura de dados descrita em 3.2.1, a origem do domínio herdado por "X", isto é, o ponto do programa onde é definido o parâmetro "Y", cujo domínio é herdado por "X". Este é o primeiro passo na construção do histórico de heranças da explicação.

. A seguir localiza-se a origem do domínio de "Y", isto é, um parâmetro "Z", do qual "Y" herdou seu domínio. "Z" será o próximo parâmetro a ser analisado.

Este procedimento é repetido até que para um dado parâmetro analisado verifica-se que seu domínio não foi herdado. Este é um caso de domínio especificado diretamente pelo usuário. Assim encerra-se a montagem do histórico de "X".

As linhas do histórico de heranças exemplificado na figura 6 são montadas a partir dos passos do procedimento acima descrito.

Nos casos de inexistência de informação sobre herança de domínio de um parâmetro assume-se que este foi diretamente

especificado pelo usuário.

### 3.3 - A Reutilização de Históricos de Herança:

As informações necesssárias para a montagem de históricos de heranças estão sempre disponíveis no âmbito de uma sessão de uso do MPL.

Ao ativar as operações de gravação, mencionadas no item 2.2.3 deste trabalho, o usuário pode optar pela criação/atualização do "ARQUIVO DIAGNOSTICO" de sua aplicação. Neste caso também estarão sendo armazenadas as informações necessárias à construção das explicações de heranças. O usuário pode então recuperar estas informações para uma próxima sessão de uso do MPL.

A utilização de uma versão desatualizada do "ARQUIVO DIAGNOSTICO" implica em contruções de explicações inconsistentes.

#### 4 - CONCLUSÃO.

As alterações sobre a especificação do MPL apresentadas neste trabalho abordam duas frentes diversas de evolução do sistema: a nova proposta de interface trata o problema de tornar o sistema mais facilmente utilizado. A intenção é que as funções disponíveis no MPL sejam mais facilmente acessadas por usuários com graus de experiência diversos. Aumenta-se assim a capacidade de exploração das potencialidades do sistema por parte de seus usuários. A segunda alteração incorpora ao MPL a capacidade de registrar os resultados do funcionamento de um mecanismo automático presente no sistema. Com isso foi feita uma experiência de se oferecer um certo tipo de apoio automático à tarefa de desenvolvimento de aplicações no qual o usuário pode, se desejar, acompanhar os passos do sistema, através de uma representação explícita dos mesmos.

## REFERENCIAS BIBLIOGRAFICAS.

[Guar 88] - P. Y. Guarany, J. F. Soares, Um Monitor para Apoio à Programação em Lógica, Departamento de Informática, PUC/RJ, 1988.

[Schn 86] - B. Schneiderman, Designing the User Interface: strategies for Effective Human-Computer Interaction, 1st. Edition, Maryland, Addison-Wesley Publishing Company, 1986, 448.

[Such 87] - L. A. Suchman, Plans and Situated Actions, 1st. Edition, Cambridge, Cambridge University Press, 1987, 203.