

# Um Ambiente Interativo para Definição de Semântica

Lucas Martins do Amaral  
Roberto da Silva Bigonha

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO  
INSTITUTO DE CIÊNCIAS EXATAS  
UNIVERSIDADE FEDERAL DE MINAS GERAIS

## Abstract

Este artigo apresenta a arquitetura do sistema para definição de semântica LDS, bem como os principais componentes da interface desenvolvidos utilizando-se o Xt Intrinsics baseado no sistema X Window rodando no sistema operacional UNIX. A programação de interfaces no ambiente X Window através da sua biblioteca Xlib ou através do *toolkit* Xt Intrinsics é rapidamente discutida. Aspectos da integração de um ambiente interativo ao sistema operacional UNIX são abordados, uma vez que o sistema xlds fora projetado para trabalhar com programas que utilizam a linha de comando.

[**keywords**]: interface homem-máquina, X Window, Xt Intrinsics, semântica denotacional, LDS, xlds, *widgets*, *gadgets*, *Literate Programming*.

## 1 Introdução

O sistema xlds (*Legible Denotational Semantics*) é resultado do trabalho do grupo de Linguagens de Programação da UFMG no sentido de se obter um ambiente interativo para definição e teste de semântica formal de linguagens de programação [4, 2, 6, 8]. O baixo grau de legibilidade e a grande dificuldade para se entender, formular ou validar descrições denotacionais de semântica de linguagens de programação devem-se principalmente à maneira em que são apresentadas. Tradicionalmente, definições denotacionais consistem em quatro partes: sintaxe abstrata, domínios semânticos, equações semânticas e equações auxiliares. Estas partes de domínios sintáticos e semânticos, seguida por um grupo de equações definindo funções semânticas e auxiliares. Para definições de pequeno porte, esta organização funciona bem, mas para definições de linguagens de porte realista, o método é inadequado. O principal problema para entender, e também para formular, uma definição semântica formal de grande porte, estruturada monoliticamente, está relacionado com o fato de os métodos formais serem inerentemente complicados e de o leitor dever ter controle sobre um volume excessivo de informação para entender qualquer fragmento de definição.

Outro problema relacionado com o estilo usual de definições denotacionais é a dificuldade de manutenção e depuração. Como domínios podem ser utilizados em qualquer equação, uma alteração em sua especificação pode acarretar mudanças em toda a definição. Note-se também que definições formais de linguagens de interesse prático apresentam, além da *ilegibilidade*, baixo nível de *confiabilidade* e de *correção*, devido, principalmente à complexidade dos métodos formais, que enseja incidência de erros de especificação.

O sistema xlds é resultado de pesquisas realizadas para solucionar tais problemas através de:

- mecanização do processo de validação de definições;
- uso de mecanismos de abstração para implementar recursos de modularização em linguagens funcionais;
- produção de definições formais legíveis para não-especialistas em semântica;
- uso do computador para projeto de linguagens de programação, através da automatização da produção do compilador a partir da definição formal da linguagem.

A filosofia de programação adotada se baseia na programação erudita ou legível — *Literate Programming* — preconizada por Donald Knuth [7]. Desta forma, a linguagem fonte do sistema se constitui na combinação da linguagem para definição de semântica e macros  $\text{\LaTeX}$ . O sistema xlds foi dividido em duas partes: interface homem-máquina [2] e rotinas da parte computacional da aplicação [6, 8]. Em linhas gerais, as funções executadas pelo sistema, no que se refere a parte computacional, são:

- teste da consistência do código fonte;
- geração de documentação  $\text{\LaTeX}$ ;
- teste e depuração da definição semântica;
- geração de compiladores; e
- execução de programas.

Para que tais funções sejam alcançadas pelo usuário é necessário que a interface homem-máquina do sistema xlds ofereça recursos para edição de textos  $\text{\LaTeX}$  para manipulação de definições semânticas compostas por vários módulos<sup>1</sup> e para seleção de arquivos. Arquivos e textos são a matéria prima básica do ambiente xlds.

Devido ao sistema para definição de semântica ser um ambiente interativo, há no código da aplicação uma parte dedicada à interface. A interface homem-máquina do sistema xlds fornece todo suporte para que o usuário do sistema o opere com facilidade e, principalmente, alcance as funções fornecidas pela aplicação sem dificuldades. Dentre as funções que a interface do sistema xlds fornece estão:

- janelas de diálogo;
- menus para seleção de funções do sistema;
- editor de textos com facilidades para edição da linguagem de definição de semântica e de macros  $\text{\LaTeX}$  (acentuação em português);

---

<sup>1</sup>definições semânticas compostas por vários módulos são denominadas projetos.

- resposta semântica;
- verificação léxica e sintática das ações de interação;
- manipulação de arquivos do sistema operacional UNIX, no que diz respeito a acesso de arquivos e diretórios.

Interfaces homem-máquina não devem somente apresentar informações ao usuário ou colhê-las. É necessário que a própria interface do sistema faça testes quanto à validade das operações léxicas e sintáticas realizadas pelo usuário [2]. Na Seção 3, será apresentada a organização da interface do sistema xlds quanto a objetos de interação. A seguir, serão apresentados alguns dos princípios utilizados para o projeto da interface do sistema.

## 2 Programação de Interfaces

A programação de interfaces pode ser feita utilizando-se ambientes interativos para definição de interfaces através de manipulação gráfica, sistemas de janelas como o sistema X Window onde o usuário trabalha com regiões retangulares na tela ou *toolkits* que fornecem um nível de abstração baseado em objetos de interação do mundo real. A programação de interfaces utilizando-se bibliotecas que implementam a funcionalidade de um sistema de janelas é bastante laboriosa. Isto, porque tais bibliotecas fornecem rotinas que realizam tarefas de baixo nível, exigindo que o programador trabalhe com eventos e janelas. Uma outra forma de se desenvolver interfaces para determinado sistema de janelas é através da utilização de *toolkits*.

O X Toolkit, também conhecido como X Intrinsics ou Xt, [3] é o *toolkit* desenvolvido para programação de interfaces para o sistema X Window. O código de determinada interface baseado no Xt é aproximadamente um quinto do código escrito utilizando-se a biblioteca do sistema de janelas X Window. É possível fazer uma analogia entre programar interfaces utilizando-se o Xt ou a biblioteca do X11 e escrever programas na linguagem Assembly ou na linguagem C.

Um programa escrito utilizando a biblioteca X11 trabalha com janelas e eventos. Quando baseado no Xt, todo programa utiliza abstrações de mais alto nível, denominadas *widgets*. Um *widget* é definido como sendo uma janela que possui semântica. *Labels, menus, scroll-bars, pushbuttons, dialog boxes, text fields*, etc são widgets, pois determinam que uma área do vídeo de um computador pode receber eventos que têm significados pré-estabelecidos. Uma variação de widgets é denominada *gadgets*. Gadgets ou *lightweight widgets* foram introduzidos pela Open Software Foundation [5] como uma forma de se economizar memória, pois eles não possuíam suas próprias janelas estando vinculados a um widget. Programar aplicações utilizando o Xt não significa abrir mão da funcionalidade fornecida pela biblioteca Xlib. Assim como ocorre na relação entre a linguagem C e Assembly, onde código Assembly pode ser escrito em um programa C, rotinas da Xlib podem ser usadas quando se programa utilizando o Xt.

### 3 A Interface do Sistema xlds

A filosofia que regeu o projeto da interface do sistema xlds se baseou em três pontos principais: relevância das informações, facilidade de operação e resposta semântica:

1. *Relevância das informações* determinou que somente as informações estritamente necessárias para o diálogo em que o usuário estivesse engajado seriam apresentadas.
2. *Facilidade de operação* determinou que as ações de diálogo deveriam ser feitas, sempre que possível, utilizando-se como dispositivo de interação o *mouse*. Além disto, *hot-keys* ou *accelerators*, segundo o vocabulário X, deveriam estar disponíveis para funções de notória importância. *Accelerators* são importantes quando o usuário está utilizando o teclado e deseja iniciar alguma ação de diálogo sem retirar a mão do teclado.
3. *Resposta Semântica* determinou verificação da validade de dados de entrada e informação para o usuário do estado em que a aplicação se encontra. Quando qualquer evento representando erro é detectado, uma janela apresentando a informação do motivo ou tipo de erro é mostrada. Tal objeto de interação utiliza recursos sonoro e visual para obter atenção do usuário. Além disto, todo acesso às funções da aplicação é restringido até que o usuário responda à mensagem de erro. Quando processamento da parte computacional da aplicação é feito, o *cursor* da aplicação é alterado para o formato de um relógio a fim de informar ao usuário o estado do sistema.

Hierarquicamente, a interface do sistema xlds é composta de algumas janelas denominadas na terminologia X *pop-up shells*. Tais janelas podem ser mapeadas e manipuladas por um sistema gerenciador de janelas. Dentre as pop-ups do sistema está a janela principal da aplicação, a partir da qual toda funcionalidade do sistema é alcançada. A janela principal é permanentemente visível, uma vez inicializado o sistema.

#### 3.1 Janela Principal

A partir da janela principal toda funcionalidade do sistema xlds é alcançada. Esta janela possui um menu principal contendo comandos para manipulação de arquivos, manipulação de projetos, formatação de textos, compilação de definições semânticas, execução do código gerado a partir de determinada definição, procura de expressões léxicas, consulta a arquivos gerados pelos comandos anteriores e escolha de opções para o funcionamento do ambiente.

**Menu File:** Uma definição formal de semântica consiste em uma coleção de arquivos, contendo módulos ou parte da semântica sendo definida. O menu **file** dá acesso aos comandos para manipulação de arquivos permitindo: carga de qualquer arquivo através da exploração de diretórios utilizando-se apenas o *mouse*; salvar o arquivo que está sendo editado; salvar o arquivo que está sendo editado com outro nome; iniciar a

The screenshot shows a window titled "Xlds - Legible Denotational Semantics". The menu bar includes "File", "Project", "Weave", "Search", "Tangle", "Compile", "Run", "View", and "Options". The "Project" field is empty. The "Filename" field contains "/tmp/.nnt/home/pos/lucas/X11/Xlds/x.c". The editor shows the following code:

```

/*
** Xlds.c          -      main LDS file
** Created: 02/05/92
** by: Lucas Martins do Amaral
**
*/

#define      _XLDS_MAIN      ^      1
#include      "xlds.h"

#include      <stdio.h>
#include      <stdlib.h>

#include      <X11/Shell.h>

#include      <X11/Xaw/Box.h>
#include      <X11/Xaw/Command.h>
#include      <X11/Xaw/Label.h>
#include      <X11/Xaw/Paned.h>

static Widget exitDialog;

static void HardKill();

static XtActionsRec un_actions[] = {
{"hard_kill", HardKill}
};

```

Figure 1: Janela Principal

edição de um novo arquivo, especificando ou não o seu nome; carregar um dos nove últimos arquivos editados, selecionando-o através de uma lista denominada *history list*; obter informações sobre o estado e *set-up* do sistema e abandonar a aplicação. Quando o arquivo em edição foi modificado e o usuário não o salvou mas decidiu abandonar a aplicação, uma pop-up de confirmação será exibida. Esta pop-up permite que o usuário decida abandonar a aplicação sem salvar o arquivo editado, salvar o arquivo editado e abandonar a aplicação ou ainda cancelar a operação de abandono da aplicação e continuar trabalhando normalmente.

**Menu Projeto:** Os arquivos que formam o corpo de uma definição semântica podem ser reunidos em projetos, de forma a facilitar seu processamento pelo sistema xlds. As funções oferecidas para manipulação de projetos são: seleção de um projeto através da exploração de diretórios utilizando apenas o *mouse*; definição de um novo projeto de definição de semântica; edição do arquivo projeto carregado no sistema. A edição é feita utilizando-se apenas o *mouse*. Um painel é apresentado ao usuário, encontrando-se do lado esquerdo uma lista dos arquivos fontes que se localizam no diretório corrente (veja Figura 2). Todos os diretórios do sistema operacional podem ser explorados. No canto direito do painel é apresentada a lista dos arquivos fontes que compõe o projeto em edição. No centro do painel estão localizados quatro botões que implementam as funções de edição para inserção do arquivo selecionado em um diretório na lista dos arquivos que fazem parte do projeto. Função para retirada do arquivo selecionado na lista de arquivos do projeto em edição. Por fim, funções para salvar as modificações feitas ou abandoná-las e abandonar o projeto carregado no sistema.

As duas últimas funções do sistema só poderão ser ativadas quando houver um

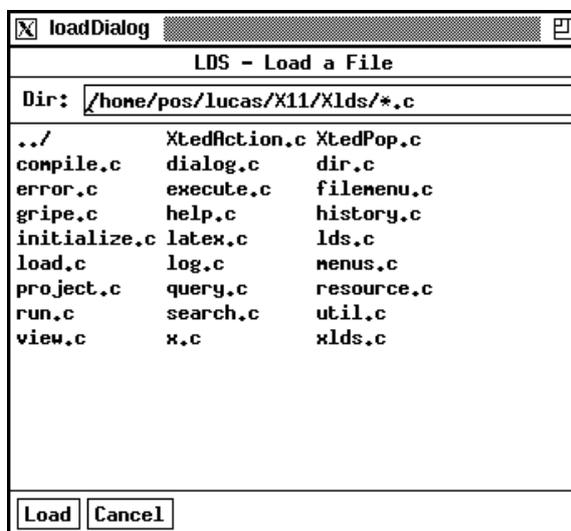


Figure 2: Pop-up para edição de projetos

projeto carregado no ambiente.

**Menu Weave:** O menu denominado **Weave** tem por finalidade extrair o texto  $\LaTeX$  da definição de semântica. As funções oferecidas por este menu são: extração do código fonte  $\LaTeX$  do arquivo carregado no editor de textos do ambiente; extração do código fonte  $\LaTeX$  dos arquivos do projeto carregado no sistema que estiverem desatualizados (este comando representa **Make**); extração do código fonte  $\LaTeX$  de todos arquivos do projeto carregado no sistema (este comando representa **Build**); formatação do texto  $\LaTeX$ . O usuário pode selecionar ou editar arquivos prólogo e epílogo para serem inseridos no texto  $\LaTeX$  no início ou final, respectivamente. Esta função compreende a extração do código  $\LaTeX$ , análise sintática para verificação de correção do código fonte e por fim inserção dos arquivos prólogo e epílogo para formatação final do texto utilizando um arquivo *stub*. A formatação do texto  $\LaTeX$  é feita utilizando-se o comando do sistema UNIX especificado no recurso da aplicação `latexCommand`; visualização da saída gerada após a formatação do texto, utilizando o comando do sistema UNIX especificado no recurso da aplicação `viewCommand` e impressão do texto formatado utilizando o comando do sistema UNIX especificado no recurso `printCommand`.

Para todas três opções o usuário pode especificar parâmetros que serão passados para as ferramentas do sistema operacional UNIX que executarão as tarefas requisitadas.

**Menu Tangle:** A geração de compiladores a partir de definições semânticas consiste na compilação de definições semânticas e pode ser feita através de três funções que são alcançadas através do menu denominado **Tangle**. Estas funções são: geração do

compilador para o módulo LDS carregado no editor de textos do ambiente; execução de **MAKE** no projeto carregado no ambiente. Neste caso, apenas aqueles módulos que necessitam ser compilados para geração do compilador o serão e execução do comando **BUILD** no projeto carregado no ambiente. O comando **BUILD** corresponde a compilação incondicional de todos os módulos que compõem o projeto.

As funções para **MAKE** e **BUILD** de projetos só poderão ser ativadas se houver um arquivo projeto carregado no ambiente.

Uma vez gerado um compilador a partir de determinada definição semântica é possível testá-lo. O comando denominado **Compile** determina que o compilador gerado pelo sistema compile arquivos fonte escolhidos pelo usuário através da exploração de diretórios.

**Menu Search:** Os comandos de pesquisa permitem que o usuário procure por: a definição de qualquer símbolo em um projeto ou no arquivo carregado no editor de textos do sistema e qualquer expressão regular utilizando o comando **grep** do sistema operacional UNIX. Se alguma seqüência de caracteres que satisfaça a expressão regular for encontrada, uma pop-up será exibida. Através desta pop-up o usuário pode selecionar uma seqüência. O arquivo que contém a seqüência selecionada é automaticamente carregado no editor de textos do ambiente e o cursor do editor é posicionado na linha em que a seqüência foi encontrada.

**Menu View:** Durante a execução das funções do sistema vários arquivos são gerados. Alguns arquivos contêm informações importantes e podem, portanto, ser analisados pelo usuário, caso ele deseje. Estes arquivos podem ser carregados através dos comandos do menu **View**, sendo apresentados em modo *read-only*, não sendo possível editá-los. As funções apresentadas são: exibição da gramática gerada a partir da definição semântica; exibição dos conflitos existentes na gramática e detectados durante a análise sintática; exibição da tabela LR(0) produzida a partir da gramática extraída da definição semântica; exibição do arquivo **LOG** produzido pelo formatador de textos **T<sub>E</sub>X**e exibição de qualquer outro arquivo. Neste caso, o usuário seleciona o arquivo desejado através da exploração de diretórios.

**Menu de Opções:** Por fim, há o menu de opções que permite que o usuário selecione a forma de operação do ambiente. Neste menu encontram-se as seguintes funções: **back-up** representa um *toggle* que determina se o sistema deve ou não gerar arquivo de back-up quando o arquivo em edição for salvo. A mensagem apresentada neste toggle varia conforme seu estado; **showcoordinates** é outro *toggle* que determina se o usuário deseja ou não que as coordenadas do cursor dentro do arquivo em edição sejam exibidas; **about** é uma opção que apresenta o nome dos autores da ferramenta e a instituição onde ela foi desenvolvida e **gripe** é um comando que exhibe uma pop-up para edição de uma mensagem que será enviada aos autores da ferramenta. Esta opção tem por finalidade estabelecer um canal de acesso entre os usuários da ferramenta e seus autores, para que comentários e críticas sejam feitos.

Logo abaixo do menu principal, está localizada a área para exibição do nome do arquivo projeto carregado no sistema seguida da área para exibição do nome do arquivo em edição. A área que apresenta o nome do arquivo projeto carregado no sistema é editável. Quando o usuário digita o nome de um arquivo neste widget, o sistema verifica se o nome dado tem a mesma extensão do recurso `projectExtension` (veja Seção 4). Se não tiver, tal extensão é acrescentada ao nome digitado e o projeto carregado anteriormente no sistema é abandonado. O novo projeto corresponde ao nome dado. Caso esse nome corresponda ao nome de arquivo projeto já existente, este arquivo é carregado. Caso contrário um novo arquivo projeto é criado.

Abaixo da área para exibição do nome do arquivo projeto encontra-se a área para exibição do nome do arquivo em edição. Se o usuário desejar editar um arquivo existente em algum diretório no sistema operacional, basta digitar o nome do arquivo nesta área. Toda verificação quanto a existência e acesso ao arquivo será feita, antes que qualquer ação ocorra. O sistema garante que o nome apresentado nesta área sempre será o nome do arquivo em edição.

Abaixo da área de apresentação do nome do arquivo em edição encontram-se informações e comandos para alteração do estado do editor de textos. Os comandos determinam modos de edição:

1. Modo `TeX` determina expansão da seqüência `TeX` correspondente para palavras acentuadas em português.
2. Modo de inserção ou *overwrite* determina se os caracteres digitados irão deslocar os caracteres a direita da posição do cursor para direita ou irão ser substituídos.

Nesta mesma parte da janela principal é informada sobre a posição do cursor dentro do arquivo em edição em termos de linha e coluna. Se o usuário selecionar a opção para não exibição de coordenadas no menu principal, os widgets relativos a esta função não mais serão visíveis.

**Editor de Textos:** O último widget pertencente à janela principal é o de edição de textos. Todos os comandos de edição podem ser personalizados pelo usuário. Os comandos pré-estabelecidos pelo ambiente correspondem aqueles do editor de textos `ted` [1]. Através do editor de textos é possível estabelecer o intervalo de tempo para que alterações no texto sejam salvas automaticamente, fazer procura e substituição de uma seqüência de caracteres no texto, inserir um arquivo na posição em que o cursor se encontra e deslocar o cursor para uma determinada linha do arquivo.

O editor de textos utilizado é baseado no widget `AsciiText` do conjunto Athena. Várias novas ações, bem como modificações de outras já existentes foram desenvolvidas para que o editor funcione de forma semelhante ao editor de textos `ted`. A idéia consistiu em fornecer a mesma interface com o usuário. As ações modificadas foram àquelas relativas a inserção de caracteres, pois o editor de textos `ted` permite modos de inserção/*overwrite* e `TeX`não-`TeX` para inserção de caracteres acentuados segundo a língua portuguesa. Algumas ações para apagar caracteres à esquerda ou a direita, também foram modificadas.

## 3.2 Janelas Pop-up

As pop-up shells definidas para a interface da aplicação são: janela principal; pop-up de erros; pop-up para carga de arquivos e exploração de diretórios; pop-up para edição de projetos; pop-up para disparo de processos; pop-up para exibição de mensagens; pop-up para edição de *mail*; pop-up para confirmação de eventos; pop-up para lista de arquivos já editados; pop-up para informações sobre o *set-up* do sistema; pop-up para formatação de textos; pop-up para exibição de arquivos; pop-up para determinação do intervalo de tempo para gravações automáticas do arquivo em edição; pop-up para inserção de arquivos no editor; pop-up para busca e substituição de palavras no editor e pop-up para deslocamento para determinada linha no texto em edição.

Cada pop-up possui uma série de widgets que fornecem semântica de botões, rótulos, áreas para edição de textos, listas de strings, viewports, etc. Cada instância de um widget de uma interface recebe um nome, além de possuir uma classe. O nome do widget serve para referências específicas aquele widget, enquanto a classe referencia todos widgets pertencentes a ela. O nome e a classe de widgets são utilizados para especificação de recursos dos mesmos (veja Seção 4). A descrição da composição de cada pop-up é necessária para que usuários possam personalizar a aplicação.

**Pop-up de Erros:** A pop-up de erros é ativada sempre que algum erro ou condição anormal for detectada. Uma mensagem explicando qual o motivo da sua ativação é apresentada. Para chamar a atenção do usuário do ambiente de definição de semântica LDS quando algum error ocorrer recurso sonoro, *beep*, é utilizado. Quando está pop-up é ativada, todas as outras funções do sistema ficarão desativadas e somente serão reativadas assim que o usuário responder à pop-up de erro.

**Pop-up para Execução de Comandos:** Todo comando para execução de comandos simples como compilação de módulos, **make**, **build**, impressão da saída  $\LaTeX$ , visualização do texto formato e mudança do nome do arquivo a ser salvo utiliza esta pop-up. Desta forma economiza-se memória e utiliza-se uma mesma função no código fonte para execução de tarefas semelhantes. A pop-up de execução possui uma área para edição de texto.

**Pop-up para Carga de Arquivos:** A carga de arquivos é feita através da pop-up de carga de arquivos (veja Figura 3) que apresenta uma lista dos arquivos de determinado diretório. Somente os arquivos que possuírem determinada extensão e não forem executáveis e os subdiretórios serão mostrados nesta lista. A extensão dos arquivos é determinada inicialmente por recursos da aplicação conforme o tipo de arquivo a ser carregado. Se o arquivo a ser carregado for um arquivo fonte LDS, somente os arquivos com a extensão determinada pelo recurso `sourceExtension` serão apresentados nesta pop-up. Todavia, se o usuário desejar selecionar arquivos com outra extensão, basta digitar a máscara de extensão na área de edição do nome do diretório. A mudança de diretórios através do uso do *mouse* exige a passagem por todos diretórios intermediários

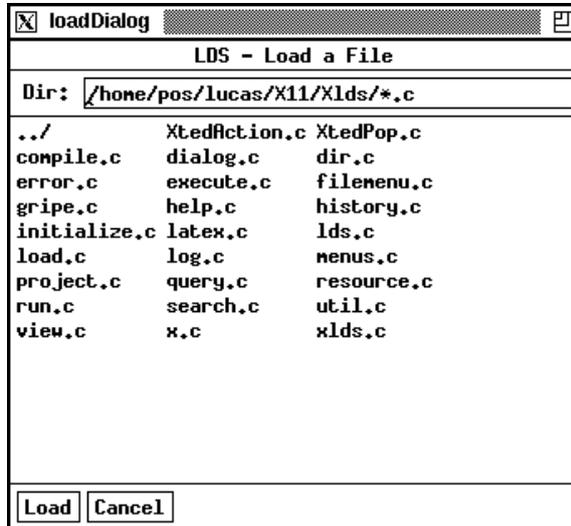


Figure 3: Pop-up para Carga de Arquivos

entre fonte e destino. Se o usuário desejar deslocar de um diretório para outro sem passar por diretórios intermediários basta digitar o nome do diretório destino na área de edição do nome do diretório. Foram criadas duas ações para edição de nomes de diretórios e arquivos usadas neste widget. A primeira permite que o nome do diretório seja apagado e que o usuário possa digitar o nome do diretório desejado a partir da raiz do sistema de diretório. A combinação de teclas associada a isto é **Ctrl Y**. A outra ação permite que o nome do **HOME** diretório do usuário seja inserido, estando o caracter `~` associado a esta ação. Dentro da área de exibição dos nomes de arquivos a seleção de um arquivo é feita através de um *click* do botão 1 do *mouse*, enquanto dois *clicks* consecutivos do botão 1 no mesmo arquivo representam seleção e carga. Um *click* no botão 3 do *mouse* representa mudança para o diretório pai.

**Pop-up para Formatação de Textos  $\LaTeX$ :** Toda definição semântica é escrita através de macros  $\LaTeX$  e construções da linguagem SDL [4]. Os macros e texto para formatação através da ferramenta  $\TeX$  são usados para geração da descrição textual da definição semântica. Para que está descrição textual seja gerada é necessário executar o comando de formatação no menu **Weave** do ambiente xlds. A formatação de textos envolve escolha de algumas opções como o nome de arquivos prólogo e/ou epílogo que serão inseridos antes e depois, respectivamente, do texto  $\LaTeX$  gerado pelo sistema. O usuário determina qual o arquivo prólogo ou epílogo será usado através de um menu com as opções de não se usar arquivo algum, selecionar um arquivo já existente ou editar um arquivo. Os arquivos prólogo têm a extensão determinada no recurso `prologExtension` e os arquivos epílogo devem ter a extensão determinada no recurso `epilogExtension`. A escolha do arquivo prólogo não é dependente da escolha do arquivo epílogo, e vice-

versa. Além da escolha destes arquivos, o usuário pode definir uma string de opções que será passada para o formatador de textos.

**Pop-up para Lista de Arquivos Editados:** Durante a edição de projetos ou mesmo durante uma seção de edição é muito comum o usuário desejar editar algum arquivo que já esteve carregado no editor anteriormente. A fim de acelerar o processo de carga de arquivos, o sistema mantém uma lista dos últimos nove arquivos editados durante a seção corrente de trabalho. Para que o usuário carregue qualquer arquivo que já esteve carregado anteriormente no editor basta ativar a pop-up sobre a história de edição e selecionar o nome de um dos arquivos lá apresentados.

**Pop-up de Informações:** Existem várias opções e variáveis no sistema que podem ser controladas pelo usuário através de comandos no menu principal ou no editor de textos. É interessante, entretanto, que o usuário possa saber o estado das principais opções e variáveis do sistema em um único local. A pop-up de exibição de informações acionada através do menu denominado **File** cumpre este papel. Nesta pop-up existe o botão **Update** utilizado para atualização das informações apresentadas na pop-up. Desta forma, é possível mantê-la constantemente ativada e quando for necessário consultar as informações nela contidas, basta ativar este botão para atualização dos dados.

**Pop-up de Mensagens:** Certos comandos do ambiente de definição de semântica geram uma série de informações que devem ser apresentadas ao usuário. Quando se compila um programa que contém erros, é função do compilador gerar mensagens de erros que sejam inteligíveis para o ser humano. Quando da execução do comando para procura de símbolos em arquivos, uma lista de mensagens contendo o nome do arquivo onde o símbolo foi encontrado e o conteúdo e número da linha do arquivo deve ser apresentada para o usuário. Para estes casos, é interessante que o usuário seja capaz de, ao percorrer a lista de mensagens, selecionar uma mensagem. A seleção da mensagem têm como semântica a carga do arquivo que a gerou no editor de textos do ambiente xlds e o posicionamento do cursor em frente ao símbolo que está relacionado à mensagem. Ela possui quatro comandos para deslocamento através da lista e outro para desativação da pop-up. Os comandos para percorrer a lista permitem que se posicione no primeiro ou último elemento da lista ou que, caso seja possível, se desloque para o elemento anterior ou posterior. Se o usuário selecionar através do *mouse* uma mensagem na lista, o arquivo que a gerou será automaticamente carregado no editor. A cada instante os widgets de comando desta pop-up estarão habilitados ou não a receber eventos, caso o item selecionado na lista assim determine (ex. se o item selecionado na lista for o primeiro, o comando **First** e **Previous** estarão desabilitados).

**Pop-up de Controle:** Certos eventos dentro de um sistema são considerados críticos e, portanto, devem ter tratamento especial. Considere o caso do recebimento do sinal

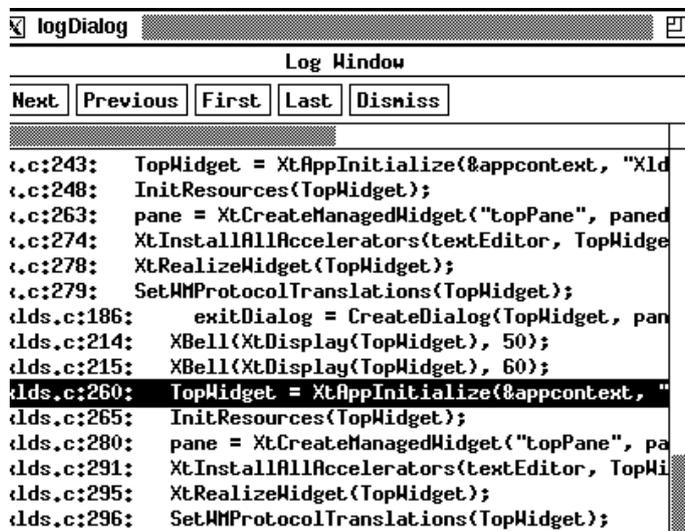


Figure 4: Pop-up de Mensagens

para finalização da seção de trabalho quando o arquivo em edição sofreu alterações e não foi salvo. A pop-up de controle será imediatamente ativada informando ao usuário que o arquivo foi alterado, mas não foi salvo. O usuário terá somente três opções de ação: salvar o arquivo e terminar a seção de trabalho, descartar as alterações feitas no arquivo e terminar a seção de trabalho ou cancelar a operação de término da seção. Todas as outras funções do sistema estarão temporariamente desativadas.

**Pop-up para Edição de Mensagens:** O objetivo principal de todo sistema interativo é facilitar o alcance da funcionalidade da aplicação pelo usuário. No entanto, o usuário em geral só tem contato com a ferramenta após ela estar pronta. A fim de obter a opinião e comentários dos usuários do sistema de definição de semântica, sua interface homem-máquina foi projetada com um comando que estabelece um canal de comunicação entre os usuários da ferramenta e seus projetistas. O comando **Gripe** ativa uma pop-up para edição de textos, onde o usuário poderá manifestar sua opinião. Uma vez escrito os comentários, o usuário pode selecionar o comando **Deliver** para que a mensagem seja automaticamente enviada para os projetistas da ferramenta. Caso o usuário decida por não enviar qualquer comentário, uma vez ativada esta pop-up, basta selecionar o comando para cancelar a operação.

### 3.3 ICCCM

Muitas aplicações X Window são terminadas quando o usuário, através do gerenciador de janelas, requisita a remoção de uma pop-up. Na maioria dos casos, o término da aplicação não era a semântica desejada pelo usuário. O principal prejuízo deste tipo de

comportamento está no fato de informações importantes serem perdidas, como aconteceria com as modificações feitas no arquivo carregado no editor de textos, caso este não tenha sido salvo. O sistema X Window através do documento denominado **Inter-Client Communication Conventions Manual** — **ICCCM** estabelece protocolos em que aplicações podem participar e que gerenciadores de janelas devem seguir. Entre os vários protocolos estabelecidos está o que determina que se uma aplicação deseja participar do protocolo de remoção de pop-ups, o gerenciador de janelas deve passar o controle do término da aplicação para a aplicação. A interface do sistema xlds foi projetada para participar do protocolo de remoção de pop-ups, evitando o término da aplicação quando o usuário desejar remover qualquer pop-up através do gerenciador de janelas. Sempre que o sistema xlds receber o evento de remoção de pop-ups, a pop-up de confirmação será exibida para que o usuário informe se deseja terminar a aplicação sem salvar o estado do sistema, terminar a aplicação após salvar o estado do sistema ou não terminar a aplicação.

## 4 Recursos

Sistemas interativos são utilizados por pessoas diferentes. A interface de tais sistemas deve permitir que cada usuário defina aspectos visuais de acordo com o próprio gosto. Um usuário pode preferir que o fundo da janela principal seja da cor verde, outro deseja que todos os nomes de comandos sejam escritos em uma língua diferente daquela utilizada pelo projetista da interface, etc.

O X Intrinsic Toolkit definiu recursos<sup>2</sup> como sendo o nome de um campo em alguma estrutura de dados. Através de recursos, o usuário de um sistema interativo pode personalizar sua interface. Recursos também fornecem uma interface uniforme para especificação e consulta de atributos de widgets. É função do projetista da interface estabelecer quais recursos poderão ser definidos pelo usuário. Os recursos que estabelecem a disposição espacial dos widgets são em geral fixos, pois estes recursos estabelecem o formato da interface, o qual não deve variar de usuário para usuário.

A maior parte dos recursos em uma interface dizem respeito a atributos ou características dos widgets. Aplicações também podem declarar seus próprios recursos para campos de uma estrutura de dados definida para a personalização de seus aspectos funcionais. Um editor de textos pode definir recursos para controlar se *back-up* será feito, qual a extensão será dada para os arquivos *back-up*, se o usuário do sistema é um especialista ou está utilizando-o pela primeira vez. O sistema xlds pode ser personalizado através de um arquivo de recursos (arquivo ASCII) ou através de opções da linha de comando.

---

<sup>2</sup>do inglês: *resources*

## 5 Avaliação

A utilização de qualquer aplicação computacional exige dois tipos de conhecimento do usuário: conhecimento sobre a tarefa e conhecimento nível de conhecimento sobre o sistema exigido do usuário, pois através de sua interface homem-máquina, usuários não especialistas em sistemas operacionais e máquinas podem exercitar seu conhecimento sobre semântica e linguagens de programação rapidamente e com grande facilidade.

É interessante notar o reduzido número de aplicações deve principalmente ao fato do sistema UNIX já possuir uma interface determinada para suas aplicações. Programas devem assumir como entrada e saída arquivos. Desta forma, pode-se criar novos programas no sistema UNIX através da simples combinação dos programas já existentes — utilização de *pipes* e redirecionamento de entrada e saída. Assim, pode-se entender o reduzido número de aplicações interativas adotada pelo sistema UNIX. Grande parte dos aplicativos do sistema operacional tem ganhado uma interface gráfica que na maioria dos casos nada mais representam que uma nova maquiagem. O sistema xlds foi projetado para rodar no sistema UNIX como um sistema interativo.

## 6 Conclusão

A interface do sistema xlds foi projetada independentemente das rotinas da aplicação garantindo possibilidade de efetuar alterações em do sistema xlds pode ser utilizada em outros sistemas cuja matéria prima de trabalho seja arquivos e textos. O fato do editor de textos do sistema funcionar da acordo com o editor **ted** [1] não implica em perda de funcionalidade quanto ao editor de textos do sistema X Window. O editor do sistema xlds se baseia no *widget* de edição de textos do X Window. Uma forma de se integrar aplicações do sistema UNIX em um

## References

- [1] Lucas M. Amaral. TED — Um Editor de Textos Portável. In *V Simpósio Brasileiro de Engenharia de Software*. Sociedade Brasileira de Computação, UFMG, UFOP, October 1991.
- [2] Lucas M. Amaral. Interface homem-máquina do ambiente de definição de semântica lds. Master's thesis, UFMG, 1992.
- [3] Paul J. Asente and Ralph R. Swick. *X Window System Toolkit*. Digital Press/Prentice-Hall International Edition, 1990.
- [4] R. S. Bigonha. *A Denotational Semantics Implementation System*. PhD thesis, UCLA, 1981.
- [5] OSF Open Software Foundation. *Motif Style Guide*. Prentice-Hall, 1989.

- [6] José L. Silva Jr. Linguagem de definição e geração de analisadores sintáticos em semântica denotacional legível. Master's thesis, UFMG, 1993.
- [7] Donald E. Knuth. Literate programming. *The Computer Journal*, 27(2), 1984.
- [8] Wallace A. Rodrigues. Compilação e otimização de uma linguagem para definição de semântica denotacional. Master's thesis, UFMG, 1993.