ICONUS - Sistema Gerenciador de Interfaces Gráficas

Adriana Figueiredo
Gilson Leal Gusmão
Jaime Tsuruta
João Batista Gomes de Freitas
João Camargo Neto
Marcelo Pinto da Luz
Miguel Argollo Junior
Regina Thienne

Projeto Fábrica de Software - PFS

Consórcio CTI - Banco do Brasil - Embrapa

Rodovia SP-340 km 105,4 - 13081 - Campinas - SP

Palavras Chaves: ambiente integrado para produção de software, editor, interface gráfica, objetos

Resumo

Este trabalho descreve o sistema gerenciador de interfaces gráficas Iconus, apresentando sua arquitetura, funcionalidade, detalhes de implementação e sua função em um ambiente integrado para produção de software. Na parte final do trabalho é mostrado um exemplo de uma interface criada utilizando o Iconus. Direções futuras também são reportadas.

1. Introdução

Com o grande desenvolvimento de estações de trabalho com monitores de alta resolução e o baixo custo advindo deste desenvolvimento, as interfaces gráficas assumiram um papel de grande relevância no desenvolvimento de uma cada vez maior gama de aplicativos. Por outro lado a implementação e a evolução de um sistema nos padrões comumente usados, isto é, um sistema de interface fortemente amarrado a rotinas gráficas e ao controle de todas as interações necessárias a uma interface gráfica, torna praticamente inviavél a sua manutenção e reaproveitamento em aplicações diferentes daquela para a qual foi inicialmente projetada , tanto do ponto de vista de custo como de esforco requerido.

Em um ambiente de produção de software onde existe uma demanda de linhas de composição orientadas para domínios específicos, e imprescindível uma forma industrial para a geração e evolução de interfaces homem-máquina.

O sistema de gerenciamento de interfaces gráficas - Iconus, é um sistema que suporta o desenvolvimento e a evolução de interfaces gráficas, facilitando a criação de interfaces através da manipulação direta, e tornando tanto o desenvolvimento como a evolução da interface independente da aplicação permitindo, desta forma o reuso de uma interface em outra aplicação e,

definição de múltiplas interfaces para uma mesma aplicação. Além do mais, com Iconus, uma interface pode ser rapidamente prototipada, antecipando a sua validação.

Iconus foi desenvolvido no Projeto Fábrica de Software, para proporcionar a integração externa entre as diferentes ferramentas dos ambientes criados. Uma descrição mais detalhada do Iconus e de outros sistemas que compoem o PFS, serão apresentados no II Workshop Internacional do PFS, que será realizado em marco de 1990, em Campinas, SP.

2. Visão Geral do Sistema

As figuras seguintes mostram a arquitetura de Iconus. O sistema utiliza o modelo de objetos que é suportado por um gerente de objetos chamado OM1. O suporte gráfico, desenhos na tela, janelas e aquisição de eventos, é provido pelo pacote gráfico X Window.

A figura 1 mostra o modelo do Iconus para o tempo de execução de uma aplicação. O sistema é uma camada entre o usuário e a aplicação, escondendo de ambos, a manipulação de objetos e a interação com o suporte gráfico. A figura 2 mostra o sistema do ponto de vista do projetista de interfaces. O editor provido pelo sistema é a ferramenta que permite o projetista de interfaces definir uma interface.

Uma interface criada utilizando o Iconus é composta por 5 classes de objetos :

- Objetos de apresentação : Objetos gráficos que permitem a interação com o usuário. Iconus fornece um conjunto destes objetos que estão arranjados em uma taxonomia hierárquica, onde atributos e métodos são compartilhados através de herança. Janelas, icones, menus e figuras geométricas são exemplos de objetos de apresentação. Objetos de apresentação podem ser compostos, e desta maneira podem ser combinados entre si, formando objetos mais complexos.

- Gerente de Diálogo : E uma máquina de estado finito que permite o projetista definir o fluxo de controle de uma aplicação.

- Gerente de Eventos : Objeto que interage com o X Window para a aquisição de eventos causados pelo usuário.

- AIF (Application InterFace) : Objetos que fazem a comunicação entre a interface e a aplicação uma vez que estes estão separados.

- Interface : Objeto automaticamente criado pelo sistema e que contem todos os outros objetos definidos pelo projetista.

A definição de uma interface utilizando Iconus se resume na criação de instâncias das 4 primeiras classes de objetos. O objeto interface é gerado automaticamente pelo sistema.

3. Edição e Utilização de uma Interface

A utilização de Iconus é feita em duas fases distintas:

a) definição da interface e,

b) execução da aplicação com a interface previamente definida.

3.1 Fase de Definição

Nesta fase, o projetista de interface utiliza o editor provido pelo sistema para criar os objetos que compõem uma interface. O editor possui uma parte textual e uma gráfica. Na textual são definidas instâncias do gerente de diálogo, gerente de eventos, aif e objetos de apresentação. A

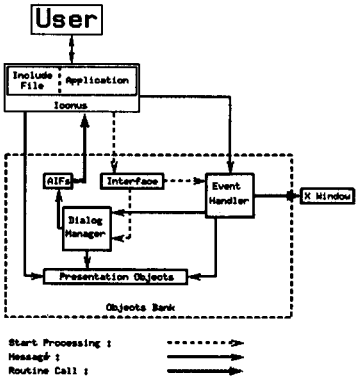


Figura 1

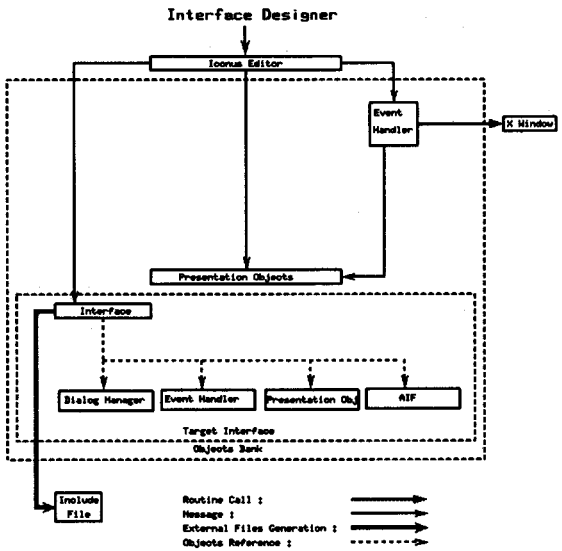


Figura 2

parte gráfica permite a visualização dos objetos de apresentação criados. Após a edição de uma interface, o editor gera os arquivos includes a serem utilizados pela aplicação e um arquivo a ser utilizado caso seja necessário recuperar a interface.

3.2 Fase de Execução da Aplicação

Para gerar o executável da aplicação, esta esta tem que ser previamente compilada com os arquivos includes gerados pelo editor na fase anterior, e ligada à biblioteca do Iconus. Em tempo de execução, o fluxo de controle da aplicação é o seguinte:

a) O objeto Interface inicia o processo enviando mensagens para que o gerente de diálogos e gerente de eventos se ativem.

b) O gerente de diálogos solicita ao gerente de eventos um evento causado pelo usuário. O gerente de eventos obtem esta informação do X Window, descobre qual objeto de apresentação sofreu o evento e retorna ao diálogo as informações obtidas.

c) O diálogo consulta suas tabelas, executa a ação previamente definida pelo projetista e retorna ao item b.

Este ciclo continua ate que o diálogo receba um evento pré-definido para finalizar a execução da aplicação.

4. Exemplo

A figura 3 mostra um exemplo de uma interface criada utilizando o Iconus. Foram utilizados objetos de apresentação tais como menus, janela de trabalho onde é feito um desenho, labels, setas e outros. Em um trecho do diálogo desta aplicação está definido que toda vez que o usuário selecionar a opção imprimir do menu, é exibido uma janela de diálogo onde são obtidas informações necessárias para a impressão do desenho. Estas informações serão passadas para a aplicação através de um objeto AIF, previamente definido e identificado pelo projetista da interface.

Esta figura também ilustra uma outra funcionalidade fornecida por Iconus. Como não é possível ao projetista saber com antecedência quantos nós (oval com label dentro) serão criados pelo usuário, Iconus oferece o mecanismo de prototização. Neste caso, o projetista cria em tempo de definição de interface um protótipo do objeto. Em tempo de execução da aplicação, este protótipo serve como base para a criação de novos objetos que serão manipulados pelo usuário.

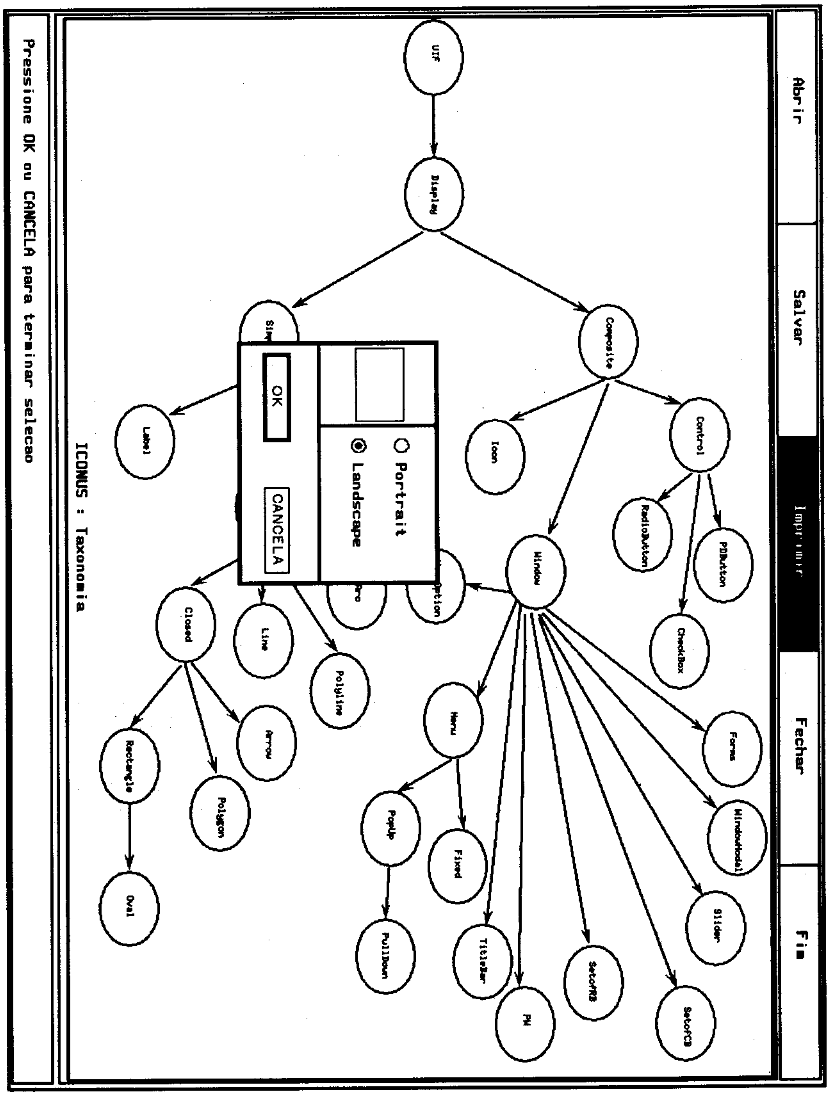
5. Trabalhos relativos

Situando o Iconus entre outros sistemas de gerenciamento de interfaces existentes pode-se notar uma forte tendência no sentido de enfatizar o desenvolvimento de interfaces gráficas de manipulação direta que permitem sua implementação paralelamente ao aplicativo.

O George Washington User Interface Management System (GWUIMS) da George Washington University é um gerenciador de interfaces gráficas desenvolvido segundo o modelo de objetos. O GWUIMS apresenta uma clara separação entre as classes de responsabilidade pela análise a nível léxico, sintático e semântico tanto das ações do usuário quanto das necessidades de feedback. Embora no Iconus quase toda a responsabilidade pela análise semântica das ações esteja suportada pelo gerente de diálogo e rotinas da aplicação, exitem classes de objetos de apresentação que incorporam um certo conhecimento de suas características e podem executar ações completas sem a intervenção da aplicação ou do gerente de diálogo.

GROW (Graphical Object Workbench) também um sistema de gerenciamento de interfaces gráficas de manipulação direta implementado segundo o modelo de objetos, permite a

Figura 3


Abrir
Salvar
Imprimir
Fechar
Fim
ICONUS : Taxonomia
Pressione OK ou CANCELA para terminar selecao
Portrait
Landscape
OK
CANCELA
UIF
Display
Composite
Control
Icon
Label
Window
RadioButton
PBButton
CheckBox
Forms
WindowModal
Slider
SetofCB
SetofRB
PW
TitleBar
Menu
PopUp
Fixed
PullDown
Closed
Line
Polyline
Arrow
Rectangle
Polygon
Oval

composição de vários objetos em unidades mais complexas o que é bastante semelhante ao conceito de composição fornecido por Iconus. Esta característica proporciona ao projetista da aplicação uma grande flexibilidade na definição dos objetos que formarão a interface uma vez que possibilita a criação de objetos específicos a um determinado aplicativo. Uma outra característica do sistema GROW é a possibilidade de definir relações entre atributos permitindo desta forma a dependência gráfica entre objetos de apresentação. Esta importante funcionalidade não foi implementada na atual versão do Iconus e fica a cargo da aplicação se seu uso é necessário.

O sistema de gerenciamento de interfaces da Universidade de Alberta não segue o modelo de objetos e além disso, a interface criada é fortemente acoplada ao código da aplicação, dificultando sua reusabilidade. Mas, uma grande vantagem deste sistema é que são fornecidas ao usuário, 3 maneiras de se definir o diálogo, tornando o sistema bastante flexível neste aspecto. No sistema da Universidade de Alberta o diálogo pode ser definido através de "event handler", diagrama de transição e gramática livre de contexto. No Iconus, um diálogo é definido através de um diagrama de transição mas, esta abordagem se mostrou ineficiente para interfaces relativamente complexas uma vez que o alto número de transições torna difícil a compreensão do diálogo como um todo.

6. Direções Futuras

Em futuras evoluções do sistema a parte textual do editor será eliminada, tornando a definição dos objetos que compõem uma interface totalmente gráfica. O gerente de diálogos precisa ser revisto e uma idéia é permitir diálogos concorrentes. Novos objetos de apresentação também devem ser fornecidos, aumentando desta forma, o domínio de utilização do Iconus.

Existem pontos que significam a evolução imediata do sistema, tais como, suportar definição de cores, objetos tridimensionais, "views" e a relação de conectividade entre objetos de apresentação, fornecendo desta forma o conceito de dependência gráfica.

7. Conclusão

Iconus foi desenvolvido com caraterísticas que suportam a definição de interfaces gráficas para vários domínios de aplicação. Com o uso do modelo de objetos, Iconus pode ser facilmente incrementado, podendo ser criadas novas classes de objetos ou ser especializadas as já existentes, sem alterações na sua arquitetura.

Uma grande vantagem de se ter o desenvolvimento da aplicação independente da definição da interface é aumento na produtividade de software através do reuso de interfaces.

Essa ferramenta e básica para qualquer atividade de produção de software ajudando em atividades tais como desenvolvimento de produtos novos, composição de componentes, controle de qualidade e gerência de configuração.

Dentro do PFS, Iconus ja' foi utilizado na implemetação do editor da linguagem gráfica para geração de aplicativos no domínio comercial (BAG) e de um editor gráfico utilizado na preparação de documentos.

8. Referências

1. **Paul S. Barth**
An Object-Oriented Approach to Graphical Interfaces - GROW: *ACM Transition on Graphics, Vol. 5, No. 2, Abril 1986, Pag. 142-172.*

2. **Marja-Riita Koivunen e Martti Mantyla**
Hut Windows: An Improved Architecture for a User Interface Management System *IEEE Computer Graphics & Applications, Janeiro 1988, Pag. 43-52.*

3. **W. Buxton, M. R. Lamb, D. Sherman e K. C. Smith**
Towards a comprehensive User Interface Management System *Computer Graphics Vol. 17, No. 3, Julho 1983, Pag. 35-42.*

4. **John L. Silbert, William D. Hurley e Teresa W. Bleser**
An Object-Oriented User Interface Management System *Computer Graphics, Agosto 1986, Pag. 259-268.*

5. **Mark Green**
The University of Alberta User Interface Management System *ACM - San Francisco, Vol. 19, No. 3, 1985, Pag 205-213.*

6. **Brad A. Myers**
User-Interface Tools: Introduction and Survey *IEEE Software, Janeiro 1989, Pag. 15-23.*