

GKS-VW : UMA PROPOSTA PARA UTILIZAÇÃO DE JANELAS NO GKS

Fernando Elyas Nóbrega Nasser (*)

Yanira d'Arcanhy Nóbrega Nasser (**)

Rubens Nascimento Melo (**)

(*) CEPEL - Centro de Pesquisas de Energia Elétrica
Divisão de Processamento de Dados
Caixa Postal 2754
Rio de Janeiro - RJ 21941

(**) PUC/RJ- Pontifícia Universidade Católica
Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
Rio de Janeiro - RJ 22453

SUMÁRIO

Neste trabalho é proposta uma extensão para o pacote gráfico padrão GKS (Graphics Kernel System) de forma a permitir a utilização de janelas com o mínimo de alterações possível na interface padronizada. É definido o conceito de Estações Virtuais e sugerido o nome GKS-VW (Graphics Kernel System - Virtual Workstations) para denominar o pacote estendido. O trabalho analisa uma solução existente para o padrão "Core" e os problemas existentes para utilização do GKS em um ambiente com janelas. Também são comentadas a implementação e utilização dessa facilidade em alguns ambientes.

1. TERMINOLOGIA

É importante esclarecer que o termo "janela" é aqui utilizado primariamente para denominar uma porção da superfície física de exibição destinada a conter uma determinada imagem, quer esta imagem esteja visível ou esteja encoberta, parcialmente ou totalmente, por outras janelas. No entanto, o uso deste termo no sentido adotado pelo GKS, ou seja, a área no sistema de coordenadas mais próximo do usuário que será mapeada por uma transformação, também seja empregado. Quando o contexto não fizer clara a distinção, esta será enfatizada.

2.MOTIVAÇÃO

Este trabalho surgiu da necessidade de se fornecer um ambiente de janelas necessário à implementação do interpretador de diálogos proposto e desenvolvido por Russo [1], para o projeto EITIS [2], utilizando o pacote gráfico padrão GKS. Este interpretador faz uso de três janelas cujo tratamento ocorre de forma bastante independente. Em geral, a divisão da tela em áreas é muito utilizada para estruturar a informação apresentada, agrupando espacialmente os diferentes tipos de informação, a fim de facilitar sua percepção e entendimento [3]. Desta forma, três componentes básicos costumam estar presentes: a área de trabalho, a área de menus e a área de "prompt" ou "feedback" [4]. Outras divisões, inclusive com a sobreposição (e obscurecimento) de áreas, também são comuns.

Uma das dificuldades encontradas para se trabalhar com telas estruturadas no GKS é a necessidade de gerenciar o espaço de coordenadas normalizadas (NDC) de forma a refletir a estruturação da tela, uma vez que uma única e uniforme transformação ("Workstation Transformation") mapeará todas estas áreas na superfície de exibição. A sobreposição com obscurecimento não pode ser representada neste modelo. Além disso, redesenhos indesejáveis de toda a tela serão efetuados quando for necessário redesenhar uma destas áreas. A versão 2.0 do GKS/PUC possui uma otimização que permite preencher rapidamente uma porção da tela com a cor de fundo através da primitiva "fill area". Esta solução, no entanto, acrescenta uma preocupação adicional ao programador, além de forçá-lo a causar a execução das primitivas que geram a porção da figura alterada em lugar de simplesmente ativar ou desativar a visibilidade de um segmento ou deixar que uma regeneração implícita ocorra. A independência desejada entre as estruturas exibidas somente seria obtida no GKS utilizando-se Estações de Trabalho diferentes para cada uma delas.

Outra deficiência é que, ao estruturarmos as informações exibidas ao usuário, estamos agrupando logicamente porções de gráficos ou texto. Neste caso seria bastante útil se pudssemos agrupar os segmentos da mesma forma. Este nível de estruturação adicional não é possível no GKS, a não ser entre diferentes

Estações de Trabalho.

As considerações acima indicam que o GKS possui os mecanismos adequados às facilidades desejadas, porém condicionado ao uso de mais de um dispositivo gráfico para exibição, o que nem sempre é economicamente viável ou produz ambiente de trabalho adequado às necessidades. No entanto, se várias Estações de Trabalho puderem ser definidas compartilhando a superfície de exibição de um único dispositivo gráfico, as características de estruturação e independência serão obtidas a um custo mínimo.

As modernas interfaces com usuário muitas vezes almejam combinar gráficos com texto em uma mesma exibição ("display"). Embora isto possa ser obtido em alguma extensão utilizando-se a primitiva "Text", várias facilidades encontradas em um terminal que opera em modo texto são perdidas, tais como mudança de linha e rolagem automática, entre outras. Mesmo os terminais que possuem os dois modos de operação, texto e gráfico, perdem estas facilidades ao entrar em modo gráfico. Propomos também, neste trabalho, que possam ser criadas janelas com um atributo "texto", onde estas facilidades sejam simuladas.

3. ANÁLISE DE UMA IMPLEMENTAÇÃO PARA O CORE

Um sistema com objetivos semelhantes foi desenvolvido por Feldman e Kamran [5] para o pacote gráfico "Core" na forma de um conjunto de subrotinas a serem chamadas pelo programa aplicativo em lugar das rotinas correspondentes do sistema "Core". Estas subrotinas, por sua vez, foram implementadas com chamadas ao "Core" padrão. Este sistema, denominado SH-1, fornece o conceito de "Telas Lógicas" a serem exibidas compartilhando, sem sobreposição, a superfície de exibição de um dispositivo físico. Além disso, O SH-1 visa fornecer um nível adicional de estruturação para os segmentos do "Core", uma vez que estes passam a estar associados com "Telas Lógicas" que podem ser criadas em número arbitrário pelo programador da aplicação.

A grande desvantagem deste tipo de solução é que a interface padrão com a linguagem fica alterada. Não somente ocorre a inevitável inclusão de chamadas para o gerenciamento

das janelas, como também a substituição das chamadas normais do "Core" para lidar com segmentos e transformações. Desta forma, fica prejudicada a adaptação de programas para a inclusão de janelas e a portabilidade entre ambientes com janelas e ambientes com múltiplas estações físicas. Mais adiante no texto serão citados outros motivos pelos quais a inclusão do conceito de janelas não deve ser feito nesta camada do sistema gráfico.

4. O GKS EM UM AMBIENTE COM JANELAS

Existem dois tipos de objetivos básicos a serem atingidos por um Sistema de Janelas [6]. Um sistema pode destinar-se a atingir a um, a outro ou a ambos. O primeiro é o de multiplexar saídas de várias aplicações sob o controle de um único usuário em uma única superfície de exibição. Esta funcionalidade é fundamental em ambientes do tipo multi-tarefas onde o usuário consegue interagir ou, pelo menos, comandar a execução de diversos programas em seu terminal ou micro-computador. Um exemplo de Sistema de Janelas existente com este objetivo é o "Microsoft Windows". O segundo objetivo básico, o de interesse neste trabalho, é o de fornecer a uma única aplicação a possibilidade de simular, em porções da superfície de exibição, diversos dispositivos virtuais de saída em vídeo. Em termos do pacote gráfico GKS, Hopgood [6] sugere que a função do Sistema de Janelas seria a de multiplexar diversas estações GKS de uma única aplicação em uma única superfície de exibição. Estas considerações nos induziram à adoção do termo Estações Virtuais para denominar uma estação GKS simulada em parte da tela (ou seja, virtual). O GKS estendido com este conceito passaria a se denominar GKS-VW (Graphics Kernel System - Virtual Workstations). Um exemplo de Sistema de Janelas existente que atende a ambos os objetivos é o Sistema "X" do M.I.T. [7].

A primeira importante consideração a ser feita ao se pensar em combinar o pacote gráfico padrão GKS e um Sistema de Janelas é onde ocorrerá a interação entre estes dois sistemas [7]. Como a maior parte do GKS é dedicada a criar e manipular imagens em NDC, a interação acima deste nível causaria enorme duplicidade e seria incompatível com as aplicações GKS já

existentes. Assim, devemos assumir que a interação ocorra abaixo do nível NDC. O conceito de Estações Virtuais é compatível com esta conclusão. Segundo Hopgood [7], o Sistema de Janelas deve receber as saídas gráficas produzidas pelo GKS após a Transformação de Estação, podendo estas já estarem cortadas ou ser fornecido o retângulo de corte para que este seja realizado em conjunto com o que deve ser feito em relação ao contorno da janela (parte da superfície de exibição destinada à simulação desta estação).

Outra decisão importante é a de permitir ou não a subdivisão de janelas. Além da dificuldade de colocar este conceito no modelo do GKS, a necessidade de seu uso indica que se deveria estar trabalhando em um nível superior de controle, talvez utilizando um pacote padrão de janelas como o Sistema X, sob o qual seria executada uma parte da aplicação escrita em GKS.

Um outro aspecto a ser lembrado é a necessidade de conservação do "aspect ratio" de forma que, por exemplo, um círculo no NDC apareça como um círculo na superfície de exibição. Não deve ser responsabilidade do Sistema de Janelas atualizar a figura para ajustá-la a um novo tamanho de janela. Sendo assim, seria necessário retornar uma indicação à aplicação de que o tamanho da janela foi alterado para que esta providenciasse a adaptação da imagem exibida, através de redesenho. Até o momento não fomos levados a acreditar que esta facilidade seja tão fundamental a ponto de justificar os imensos problemas que este procedimento traria ao modelo GKS, de forma que se deixa a cargo do programador cuidar para que, ao redefinir janelas, as figuras não sejam distorcidas.

A partir do momento que podemos definir Estações Virtuais com a relação comprimento/largura que queremos, torna-se pouco útil definir a "workstation viewport" como uma subparte da tela da estação [7]. Também não vemos necessidade de retirar explicitamente esta facilidade, cabendo ao programador definir a "workstation viewport" como toda a tela virtual sob pena de desperdiçar parte da superfície de exibição.

O GKS possui vários modos de atualização de imagens. Em

alguns destes modos a atualização da imagem pode ser postergada mas deve ter sido realizada antes que uma interação ocorra. No modo BNIL isto implica em que a Estação Virtual na qual for solicitada a entrada esteja visível [7]. No modo BNIG o problema se torna complexo quando há saídas pendentes para as Estações Virtuais obscurecidas total ou parcialmente. As possíveis soluções seriam atualizar somente as Estações Visíveis e realizar atualizações nas porções visíveis das parcialmente obscurecidas. A geração de saída em Estações parcialmente obscurecidas deve ser cortada nos limites de todas as Janelas que as obscureçam. Outra solução seria a de trazer para o primeiro plano (sobrepondo as demais) as janelas a serem atualizadas, em ordem pelo número da Estação Virtual associada. Neste caso, o programador deve tomar cuidado com o número de redesenhos que pode ser causado. Outra forma, ainda, seria redefinir o modo BNIG como aplicável somente às Estações totalmente visíveis. É claro que, se adicionada a restrição de que as janelas não se sobreponham ("tiled windows"), as situações acima não ocorrem.

Os dispositivos físicos de entrada, tais como "mouse" e teclado, terão que ser compartilhados por diversas Estações Virtuais para implementar seus dispositivos lógicos de entrada. Hopgood [6] lembra que os dispositivos lógicos dos tipos "LOCATOR", "STROKE" e "PICK" estão fortemente relacionados com as janelas, ao contrário dos dispositivos "VALUATOR", "CHOICE" e "STRING" que podem ser realizados sobre qualquer porção da superfície de exibição.

5. UTILIZAÇÃO DAS ESTAÇÕES VIRTUAIS

São muitas as funções de um Gerente de Janelas [8]. Nem todas estas funções são necessárias ou mesmo podem ser enquadradas no modelo criado pelo pacote gráfico padrão GKS para o programador. Na verdade, desejamos apenas estender o GKS de forma a poder utilizar algumas facilidades de janelas com o mínimo de alterações possível na interface padronizada. As Estações Virtuais, a menos das de tipo "texto" e de terem de compartilhar a tela de um único dispositivo físico, se comportam como qualquer Estação de Trabalho real. Desta forma,

aproveitamos as facilidades de estruturação de informações gráficas por Estação do GKS para estruturar o conteúdo de informações por janela. A interface padronizada fica também mantida, pois os comandos padrão do GKS funcionarão nas Estações Virtuais da mesma forma que o fariam em Estações de Trabalho ordinárias. A única exceção ocorre com relação às Estações Virtuais do tipo "texto" que simulam um terminal alfanumérico e somente aceitam as diretivas específicas:

`Read_Text_From_Virtual_Workstation` e

`Write_Text_To_Virtual_Workstation`.

É necessário, no entanto, adicionar comandos para a gerência de janelas, ou seja, das Estações Virtuais. Colocando estes comandos no código de inicialização e de finalização de programas, obtém-se, de forma bastante transparente para o restante do código, o efeito da utilização de janelas. Para que isto seja possível, é necessário que o programador tenha feito uso de variáveis para designar as Estações de Trabalho utilizadas, ou seja, usado "DIALOG", "MENU", "GRAPH" em lugar de 4, 5 ou 6, por exemplo. É claro que interfaces mais elaboradas podem ser desenvolvidas controlando as janelas ao longo da programação gráfica. Neste caso, como estaremos utilizando plenamente as facilidades fornecidas pelo uso de janelas, a migração para um ambiente sem as mesmas (com múltiplas estações físicas) pode requerer algum cuidado. Se estes cuidados forem tomados, a compatibilidade pode ser obtida até mesmo dinamicamente através de caminhos alternativos do programa.

As principais funções de controle de Estações Virtuais a serem incorporados ao GKS em sua versão GKS-VW são:

`Create_Virtual_Workstation` - Esta função associa parte da superfície de exibição (especificada em DC) de um determinado Tipo de Estação (WKTYPE) a uma Estação de Trabalho Virtual. Sugerimos que os números de Estações de Trabalho Virtuais sejam alocados dentro um intervalo pré-definido de Identificadores de Conexão (CONID) para aquele Tipo de Estação, correspondendo a entradas em uma Tabela de Estações Virtuais associada a este Tipo. Algumas características básicas, como o tipo da janela ("texto" ou "gráfica"), moldura, título e cor de fundo, devem ser fornecidos como parâmetros.

Delete_Virtual_Workstation - Libera a entrada na Tabela de Estações Virtuais e, conseqüentemente, o Identificador de Conexão para uso posterior. A Estação de Trabalho Virtual deve ter sido fechada previamente.

Redefine_Virtual_Workstation - Permite restabelecer a porção da superfície de exibição do dispositivo físico destinada a conter a janela correspondente à Estação Virtual.

Set_Virtual_Workstation_Visibility - Faz com que a janela correspondente à Estação Virtual se torne visível na tela ou que não seja exibida. Inicialmente as Estações Virtuais criadas estão "não visíveis".

Algumas outras funções poderiam ser criadas no GKS-VW para alterar atributos de Estações Virtuais e, principalmente, para gerenciar janelas sobrepostas. Baseados no fato simplificador de que estamos trabalhando em um ambiente com várias janelas mas com apenas um programa em GKS-VW, podemos supor que o foco desejado esteja relacionado com a sequência de comandos GKS do fluxo de execução deste programa. Desta forma, são trazidas para o primeiro plano (sobrepondo as demais) as janelas correspondentes às Estações Virtuais para as quais seja comandada regeneração explícita ou seja necessária regeneração implícita, ou que o modo de atualização requeira ação imediata. Assim, utilizando a função "Update-Workstation" e trabalhando adequadamente com os modos ASAP, BNIL, BNIG e ASTI, além dos números das Estações (prioridade), o programador poderá controlar a sobreposição de janelas. O funcionamento dos modos de adiamento passa a ser definido como:

ASAP - causa a colocação da janela em primeiro plano (se não estiver) quando qualquer função de saída for direcionada à Estação a ela associada.

BNIL - faz com que isto ocorra quando a primeira função de entrada for direcionada a esta Estação.

BNIG - causa a colocação em primeiro plano, se necessário, e a atualização de todas as janelas correspondentes às Estações de Trabalho Virtuais com este modo estabelecido, em ordem de Identificador, antes da execução de uma diretiva de entrada destinada a qualquer Estação.

ASTI - significa, para Estações Virtuais, "assim que a janela estiver totalmente visível".

6. CONSIDERAÇÕES SOBRE IMPLEMENTAÇÃO

Uma vez que no sistema GKS o componente responsável pelo gerenciamento dos recursos de um dispositivo físico é o "Driver" correspondente, e que a implementação de janelas é bastante dependente deste dispositivo, decidimos criar "Drivers" dotados da capacidade de janelas. Desta forma, aproveitamos integralmente as funções do GKS e maximizamos o aproveitamento dos recursos do "hardware" ou do ambiente de implementação. Outro motivo para se implementar janelas neste nível de "Driver" de dispositivo é o de facilitar o compartilhamento que deve ser feito dos dispositivos físicos de entrada da Estação.

As principais funções adicionadas a este "Driver" são a de gerenciar a Tabela de Estações Virtuais associada ao seu Tipo de Estação, manter controle da situação de janelas sobrepostas e de sua atualização ou redesenho, além da simulação de terminais tipo "texto" (quando esta facilidade for oferecida). No mais, este funciona como um "Driver" normal, recebendo, no entanto, comandos para diversas Estações de Trabalho (Virtuais), que são acrescidos do fator de escala, da translação e do retângulo de corte necessários ao enquadramento na janela. Como, em várias implementações, o retângulo de corte é recebido pelo "Driver" juntamente com os comandos de saída gráfica, se este tiver sido especificado para a Estação será escalonado e utilizado em substituição ao da janela.

O ambiente ideal para a implementação de um "Driver" com janelas é sobre um Sistema X ("X-Windows") [7]. Neste caso, todas as facilidades necessárias já estariam disponíveis no Servidor X ("X-Server") da estação gráfica que se deseja utilizar, tornando o "Driver" apenas uma interface entre a modalidade de utilização através do GKS e o modo de uso do Sistema X. Algumas facilidades adicionais tais como ícones associados às Estações Virtuais e técnicas de eco mais elaboradas nos dispositivos tipo "CHOICE" podem ser conseguidos neste caso. A figura 6.1 mostra a situação do GKS-VW em um ambiente com Sistema X. Esperamos que um ambiente como este venha a estar disponível para o projeto EITIS futuramente.

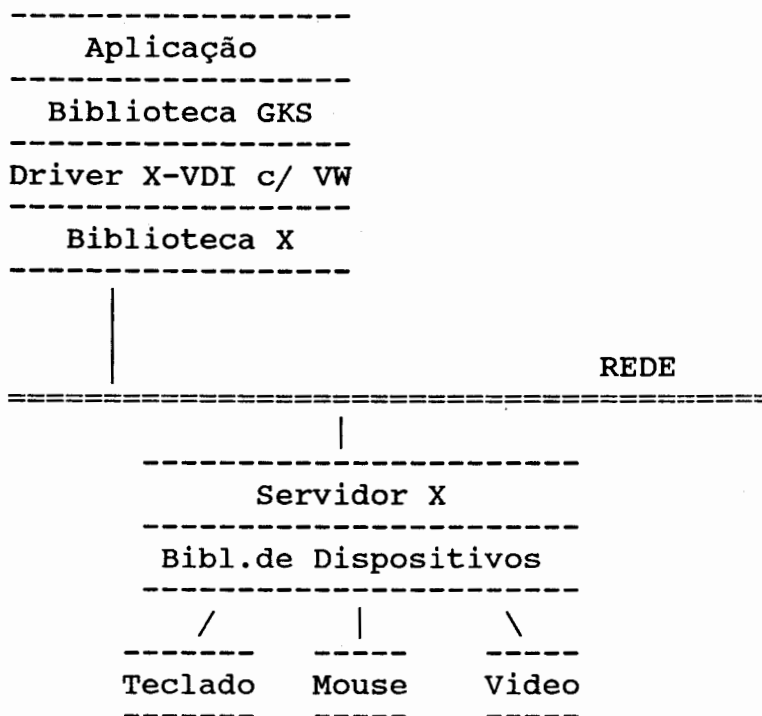


Fig. 6.1 GKS-VW em ambiente com Sistema X

O ambiente atual do projeto EITIS consiste em microcomputadores tipo IBM PC com Sistema Operacional DOS. Embora, até o momento, o desenvolvimento tenha sido feito, inclusive de uma implementação do GKS, em Turbo Pascal, decidiu-se começar a criar a infraestrutura para a migração para ambientes mais poderosos.

Desta maneira, resolvemos passar a utilizar a implementação comercial do GKS, conhecida como GKS/PUC, desenvolvida pelo Grupo de Computação Gráfica da PUC/RJ [9]. Este GKS foi escrito em FORTRAN 77, embora possa ser chamado por programas escritos em linguagem "C" (usando o "bind" FORTRAN). Existem versões para vários computadores entre eles o IBM PC e compatíveis. É intenção deste Grupo reescrever o GKS/PUC em "C", ocasião e linguagem propícias para a implementação das facilidades de Estações Virtuais. No entanto, como isto depende da obtenção de recursos, optamos por realizar uma implementação simplificada, alterando a versão FORTRAN, apenas para atender ao Interpretador de Diálogos de Russo [1] e exemplificar a aplicação dos conceitos propostos.

Devido à linguagem em que foi escrito (FORTRAN) e à própria arquitetura do programa, que foi projetada para o máximo

aproveitamento das características desta linguagem, várias limitações foram impostas à implementação de janelas. Além de somente serem permitidas janelas não sobrepostas por questões de simplicidade (somente janelas "tiled" são aceitas), o número de janelas permitido ficou limitado em três (suficiente para o uso de Russo [1]). A identificação desejada para as Estações Virtuais através do Identificador de Conexão também não foi usada para evitar alterações no corpo do GKS/PUC, utilizando-se números vagos de Tipos de Estações (WKTYPE) para indicar as Estações Virtuais. Também foi necessário acrescentar a exigência de que a Estação de Trabalho que contém a superfície de exibição sobre a qual serão simuladas as Estações Virtuais esteja previamente aberta. Da mesma forma não foram consideradas Estações tipo "texto".

Embora a implementação das Estações Virtuais se concentre, principalmente, a nível de "Driver", as dificuldades encontradas em alterar uma implementação de GKS já existente concluímos que, para que esta proposta possa ser melhor avaliada, a inclusão das facilidades propostas neste trabalho deve ser prevista em tempo de projeto do sistema GKS. Esperamos que possam ser obtidos recursos para tanto em um futuro próximo.

7. AGRADECIMENTOS

Desejamos agradecer ao Grupo de Computação Gráfica da PUC/RJ, responsável pela implementação do GKS/PUC. Em especial, agradecemos a seu coordenador Prof. Marcelo Gatass e ao Srs. Albino J. Tavares, Paulo Roma Cavalcanti, Marcelo Maranhão e Paulo Francisco Sedrez pelas informações prestadas e inestimável ajuda para a execução deste trabalho.

REFERÊNCIAS BIBLIOGRAFICAS

- [1] Russo, Enio E. R. - "Um Sistema de Interpretação de Diálogos Gráficos" - Diss. de Mestrado - Dept. de Informática PUC/RJ - março 1988
- [2] Melo, R. N. - "EITIS: Um ambiente de ferramentas integradas para desenvolvimento de sistemas interativos" - Anais do I Simpósio Brasileiro de Engenharia de Software - outubro 1987
- [3] Foley, J. D.; Van Dam, A. - "Fundamentals of Interactive Computer Graphics" - Addison-Wesley 1982
- [4] Hearn, D.; Baker, M. P. - "Computer Graphics" - Prentice-Hall 1986
- [5] Feldman, M. B.; Kamran A. - "A Logical-Screen Handler Package (SH 1) for the GWU Core Graphics System" - Report GWU-IIST-81-18 - The George Washington University - August 1981
- [6] Hopgood, F. R. A. - "A Graphics Standards View of Screen Management" in "Methodology of Window Management", Chapter 9 - Springer-Verlag - EUROGRAPHICS 1986
- [7] Scheifler, R. W.; Gettys J. - "The X Window System" - ACM Transactions on Graphics, Vol. 5, No. 2 - April 1986
- [8] Nasser, Y. A. N. - "Um Gerente de Janelas para o EITIS" - Dissertação de Mestrado - Dpto. de Informática PUC/RJ - a ser submetida
- [9] Fundação Padre Leonel Franca - "Manual de Implementação do GKS/PUC" - julho de 1988