

5. CONSIDERAÇÕES GERAIS

Pelos resultados obtidos com o *software*, há indicações de que as premissas feitas referentes à ordem de extração parecem corresponder aos dados reais. Sabe-se que no processo de extração em que se usa KELEX-100 como agente quelante, o gálio é melhor extraído que alumínio e sódio (3), o que indica coerência nos resultados fornecidos pelo *software*.

A utilização do sódio nas comparações não correspondeu às expectativas iniciais, pois o composto gerado ficou muito diferente dos outros, não se devendo utilizar esse íon no decorrer do estudo. Foram também simulados complexos com outros agentes quelantes além do KELEX-100, cujos resultados tiveram as mesmas características dos acima expostos. Outro fator a ser observado é a grande discrepância em relação aos ângulos. Deve-se dar mais importância às distâncias de ligação nas análises. Para se ter uma idéia exata do quanto esses resultados são reais, é necessário realizar testes em laboratório, para comprovar ou não o que foi previsto.

5. BIBLIOGRAFIA

1. MOLECULAR SIMULATIONS, INC, User Manual CERIU Version 3.2 (1994).
2. MIHAYLOV, DISTIN, P.A. Gallium Solvent Extraction in Hidrometalurgy An Overview: Amsterdam. Hidrometalurgy, v. 28, p. 13-27, 1992.
3. BORGES, P.P., MASSON, I.O.C. Solvent Extraction of Gallium with KELEX-100 from Brazilian Weak Sodium Aluminate Solution. Rio de Janeiro: CETEM, 1993.

PAINEL 12

Ambiente Gráfico de Auxílio à Utilização de Modelagem Molecular

Patrícia Alves Dias de Rosa e Rio
Bolsista de Inic. Científica, Eng. Química, PUC/RJ

Peter Rudolf Seild
Orientadora, Químico Industrial, Ph.D.

1. INTRODUÇÃO

A Modelagem Molecular por computador tem a finalidade de simular estruturas e suas respectivas interações. Além disso visa a auxiliar os químicos em seus estudos e análises das estruturas moleculares.

Isso se dá através da utilização de sistemas computacionais - *hardware* e *software* - de alto nível e, conseqüentemente, de difícil compreensão para pessoas que não sejam usuários familiarizados com esse tipo de sistema.

Tendo em vista disseminar a cultura da Modelagem Molecular no CETEM, formou-se um Núcleo de Modelagem Molecular voltado para atender às seguintes tarefas básicas: suporte computacional ao usuário, desenvolvimento de uma interface gráfica para usuários, suportes químico ao usuário e aos aplicativos de Modelagem Molecular.

Dentro desse contexto, está em desenvolvimento uma interface gráfica de auxílio, que possibilita ao usuário da Modelagem Molecular utilizar os sistemas computacionais disponíveis, de

maneira fácil e transparente, através de telas gráficas orientadas por janelas e de um conjunto de textos explicativos.

Assim, pretende-se disponibilizar não só o uso de softwares acadêmicos e comerciais, por parte dos pesquisadores, como também, de manuais e cursos que serão desenvolvidos nesse sistema. De modo que, o usuário, sem deixar seu posto, possa ter acesso a informações e esclarecer dúvidas sobre modelagem molecular. O usuário poderá, também, utilizar os recursos do equipamento como cópia e transferência de arquivos, impressão e visualização tridimensional de estruturas moleculares.

Este trabalho teve como base teórica o estudo utilizado no desenvolvimento de uma interface gráfica para visualização de moléculas tridimensionais (1). Essa interface serve para visualizar moléculas construídas através do *software* SD, usado para modelagem de moléculas covalentes sob o aspecto acadêmico. O trabalho baseou-se em estudos sobre computação gráfica (2), utilizando a Graphics Library (GL) (3,4,5).

2. OBJETIVO

Este trabalho tem como objetivo criar uma interface gráfica para utilização de Modelagem Molecular. Através desse ambiente será possível ao usuário acessar todos os serviços do núcleo de Modelagem Molecular. A interface será auto explicativa, tornando possível o usuário utilizar com facilidade o computador, mesmo sem nenhum conhecimento de informática.

Ela terá, também, um sistema de segurança que permitirá ao usuário acessar somente a área pública disponível e os serviços do Núcleo. Seus arquivos não poderão permanecer gravados no computador, sendo copiados para disquete.

3. METODOLOGIA

Inicialmente, o trabalho concentrou-se no estudo aprofundado do sistema UNIX (6), o qual pretende entender como se processa a comunicação máquina-sistema operacional para que seja possível uma comunicação direta do programa com o sistema operacional.

Paralelamente, iniciou-se o estudo da *Open GL* - nova biblioteca gráfica da *Silicon Graphics*, na intenção de construir uma interface com apresentação de imagens. Para tanto, foram utilizados recursos de computação gráfica, juntamente com a ferramenta de programação *Motif* (7, 8, 9), que permite a geração de programas, utilizando-se a filosofia de janelas.

Cada serviço do núcleo será representado por um botão gráfico, ou seja, um botão com o desenho que representa o serviço. O nome do serviço aparecerá em uma barra de *status* no rodapé da janela, sempre que o cursor for posicionado sobre o botão equivalente ao serviço. E para se ter o acesso, será preciso efetuar dois pressionamentos consecutivos do botão direito do mouse do computador.

No caso do serviço selecionado ser um aplicativo de modelagem molecular, a interface comandará a execução do aplicativo. Quando o usuário finalizar o aplicativo, retornará à interface.

O processo de reconhecimento da localização do cursor é feito através da GL (3). O reconhecimento é dado pelo mapeamento do movimento do cursor do sistema de coordenadas do computador (*Window Coordinate System*) para o sistema de coordenadas real (*World Coordinate System*).

Para armazenar os dados referentes à movimentação do cursor, foram usados dois métodos: o de empilhamento (*polling*) e o de fila de entrada (*input queuing*).

O método de empilhamento é mais eficiente, uma vez que sempre quando ocorre uma mudança no estado do dispositivo,

uma entrada é feita nessa fila de entradas. Ela funciona juntamente com uma fila de eventos do tipo FIFO (*First In First Out*). A fila de eventos armazena cada possível estado do dispositivo de entrada e para cada estado um algoritmo correspondente. Dessa maneira, a cada mudança no estado do dispositivo de entrada, é feita uma pesquisa na fila de eventos, permitindo a execução de um determinado procedimento.

O método de empilhamento, por sua vez, é considerado muito trabalhoso, pois, sendo uma pilha, necessita de procedimentos do programador para armazenar o estado do dispositivo de entrada.

O próximo passo a ser seguido será o aprofundamento no estudo da linguagem OSF/Motif (7, 8, 9) para, então, dar início ao efetivo desenvolvimento dos algoritmos de construção das janelas.

4. RESULTADOS

O Motif é composto de *procedures* e funções, que permitem um rápido e fácil acesso à baixos níveis do sistema *versus Window*. Ele possibilita uma escrita rápida de aplicativos, que utilizem poucas linhas de código de programa. No entanto, o Motif requer muito mais memória disponível do que outros aplicativos similares a ele.

O Motif tem uma filosofia de abstração, que exige um raciocínio, no qual todos os componentes da janela a ser construída são objetos (*widgets*), que seguem entre eles, uma determinada hierarquia. Ou seja, todo objeto é dinamicamente alocado e pertence a uma classe. Essa classe, por sua vez, possui uma estrutura que é estaticamente alocada e inicializada, e contendo as operações dessa classe. Cada classe inferior pode herdar todos os recursos oferecidos pela sua classe superior. O fluxograma da Figura 1 mostra as classes básicas de objetos do Motif.

A classe básica é a *Core class*. Ela é usada como um suporte de superclasses. Possui recursos que são herdados pelas outras classes, ou seja, recursos comuns a todos os objetos. São exemplos desses recursos, a localização xy e a espessura da borda das janelas.

A *Composite class* e a classe *XmPrimitive* são classes de uma camada abaixo da *Core class*.

A classe *XmPrimitive* não possui outras classes abaixo dela. Ela possui recursos para o desenho da janela e para a utilização das cores.

A *Composite class* possui duas classes : a *Constrain class* e a *Shell class*. A *Constrain class* fornece aplicativos de *layout*. Esta classe possui uma classe inferior, a *XmManager*. A *XmManager* suporta recursos visuais e contextos gráficos.

A *Shell class* é utilizada para proporcionar a comunicação entre objetos e o gerenciador de janelas. Ela possui várias categorias, como, as caixas de diálogo e os menus de erro.

Pode-se, então, concluir que o Motif é um ninho de classes e objetos. O esquema que apresenta as suas principais classes resulta em novos esquemas e assim por diante.

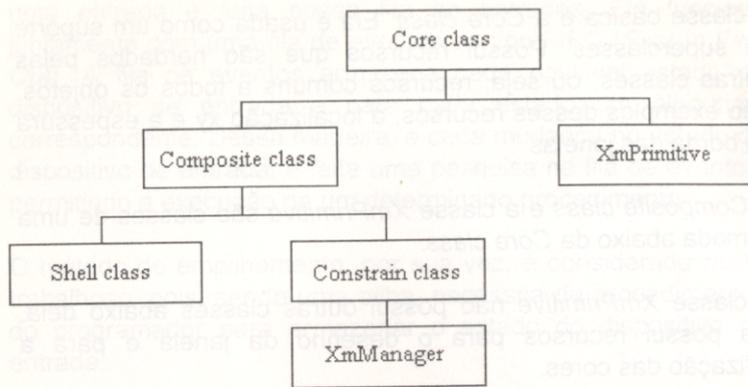


Figura1 - Classes básicas de objetos do Motif

5. CONSIDERAÇÕES GERAIS

Este trabalho ainda se encontra em fase de andamento, não foi desenvolvido nenhum algoritmo até o presente momento. Visto que a massa de conceitos a ser absorvida para o desenvolvimento da presente interface é bastante grande.

Como, em um futuro próximo, o Núcleo de Modelagem Molecular, que possui duas *workstations* e um microcomputador, terá seus equipamentos ligados em rede, a interface deverá funcionar em rede. Para tanto será utilizado um *software* que simula uma janela gráfica, em microcomputador, idêntica a que aparece na *workstation*. Como ele estará ligado em rede com a *workstation*, o microcomputador será o seu próprio terminal.

BIBLIOGRAFIA

1. RIO, P.A.D.R., PINTO, P.S.S., EARP, M.V.S. *Desenvolvimento de interface gráfica para visualização*. In : JORNADA DE INICIAÇÃO CIENTÍFICA DO CETEM, 2, 1994, Rio de Janeiro : CETEM/CNPq, 1994. p.121
2. FOLEY, J.A., VAN DAM, S.F., HUGHES, J. *Computer Graphics - Principles and Practice*, Reading, MA: Addison-Wesley, 1990.
3. GL PROGRAMMING 1, Supplement 1, Fortran Code Supplement - Student's Workbook, Silicon Graphics Computer Systems, 1991.
4. GL PROGRAMMING 2 - Student's Workbook, Silicon Graphics Computer Systems, 1991.
5. POWER VISION GRAPHICS PROGRAMMING - Student's Workbook, Silicon Graphics Computer Systems, 1991.
6. TAYLOR, A., *UNIX Para Quem Usa DOS*. Rio de Janeiro: Lutécia, 1991.
7. OSF/MOTIF - Programmer's Guide - Release 1.1, Rio de Janeiro: Prentice-Hall do Brasil, 1988.
8. NYE, A. *X Toolkit Intrinsic Programming Manual OSF/Motif 1.2 Edition*. O'Reilly & Associates, 1993.
9. YOUNG, D.A. *The XWindows System® Programming and Applications With Xt*. OSF/Motif Edition, 1993.