

ESPECIFICAÇÃO DE UM GERENCIADOR ORIENTADO A OBJETOS

Marcus Aurélio de Carvalho Macêdo, Décio Fonseca

DEPARTAMENTO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DE PERNAMBUCO
50.739-Recife, PE, Brasil

RESUMO

Este trabalho apresenta as idéias e os conceitos necessários a concepção e a implantação de um gerenciador orientado a objetos, denominado de GOLGO, para a concepção de aplicações em Banco de Dados.

ABSTRACT

This paper presents some aspects of the design and implementation of a software object-oriented (GOLGO) for the development of applications in database. The basic concepts and mechanisms for the objects management are also presented.

1. INTRODUÇÃO

Com a crescente evolução das ferramentas voltadas para o mais variados fins, a Informática vem permitindo o uso do computador de forma mais acessível àqueles usuários que buscam resolver seus problemas. O uso destas ferramentas que auxiliam a captação de informações do mundo real direcionando a representação dos conceitos possibilitam a solução dos mais variados problemas. Novos "ambientes de desenvolvimento" estão sendo criados todos com um único objetivo: transformar o computador numa ferramenta inteligente, que facilite a concepção de aplicações. Novas interfaces permitem um diálogo mais "amigável" entre o homem e a máquina tornando a tarefa de desenvolvimento de aplicações mais fácil, tendo em vista que os conceitos de modelagem de dados são utilizados pelo usuário de forma interativa.

Este trabalho está inserido no contexto do projeto APOIO (em desenvolvimento no DI/UFPE) que pretende ser uma ferramenta inteligente e eficiente para uso do computador, e cujo objetivo final é a criação de um ambiente interativo de especificação e desenvolvimento de aplicações em banco de dados, incorporando os conceitos de modelagem com o enfoque de orientação a objetos (Figura 1). Entretanto, um ambiente com tantos recursos necessita de uma estrutura de

armazenamento e controle de informações mais flexível, permitindo o acesso controlado de seus diversos objetos. Surge então, a proposta de um gerenciador que permita a manipulação dos objetos do sistema, através de uma interface orientada a objetos. O gerenciador que será aqui apresentado, foi criado a partir da análise das facilidades da *filosofia de Hipertexto* e dos conceitos de *programação orientada a objetos* que permitem a modularidade das partes constituintes do Sistema. O GOLGO ou Gerenciador de Objetos Ligados por um Grafo Orientado incorpora estas duas características conforme apresentado em [MACE89]. Sua arquitetura funcional, como será apresentada, é composta por objetos de gerenciamento com funções específicas, que se comunicam através de mensagens, cada um controlando uma parte do sistema, o que ilustra a grande modularidade obtida com o enfoque de orientação a objetos.

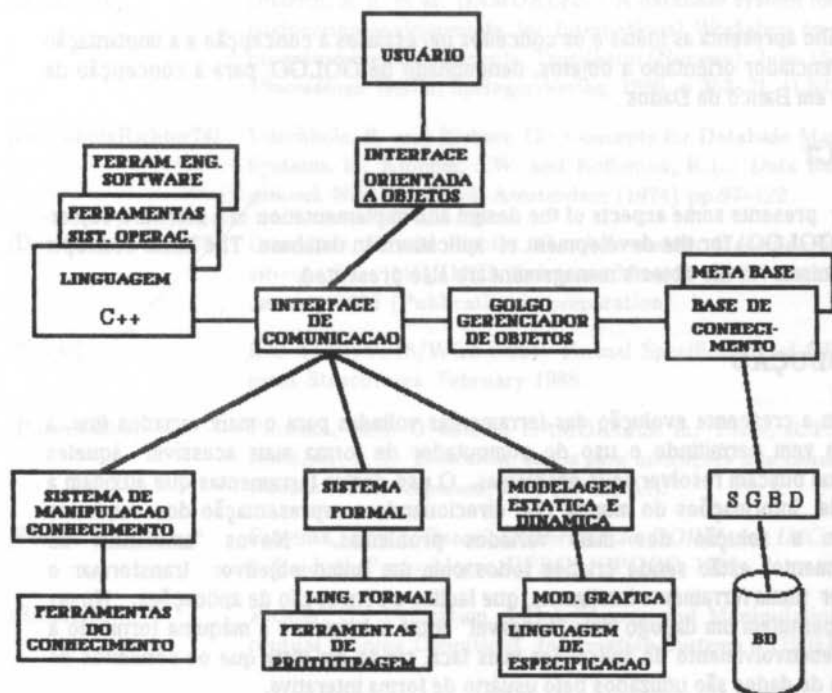


FIGURA 1. Diagrama do Projeto APOIO.

A segunda seção expõe as características do GOLGO e sua arquitetura básica com seus objetos. A terceira seção apresenta o ambiente do GOLGO com os principais objetos de implantação. A quarta seção permite vislumbrar de forma resumida a

funcionalidade do Sistema e o papel preponderante da Interface Orientada a Objetos. A última seção compreende as considerações finais.

2. CARACTERÍSTICAS DO GOLGO

Este tópico descreverá as características do GOLGO (Gerenciador de objetos Ligados através de um Grafo Orientado). O GOLGO é uma proposta de gerenciador que dará suporte a implementação do ambiente para concepção de aplicações em Banco de Dados. O GOLGO objetiva:

- Criar uma estrutura flexível para representação dos objetos necessários ao Ambiente de Desenvolvimento do projeto APOIO (Figura 1);
- Suportar uma interface orientada a objetos;
- Elaborar e manter um sistema de controle de acesso visando o aspecto de segurança;
- Controlar as versões de aplicações modeladas pelo usuário.

Os conceitos de linguagem de programação orientada a objetos descritos em [MACE89] são utilizados na concepção do GOLGO. O GOLGO é uma estrutura em forma de grafo orientado que permite a manipulação de objetos, classes e instâncias. Sendo um grafo, possui o conceito de nó e arco. Estes conceitos têm uma semântica própria para o GOLGO.

2.1 CONCEITO DE NÓ

Um nó representa no GOLGO um objeto, uma classe ou uma instância de objeto. Existem basicamente três tipos de nós no GOLGO. Um nó para instâncias de objeto que chamaremos de NÓ SIMPLES, um nó que representa um objeto ou uma classe que chamaremos de NÓ COMPLEXO, e um outro nó que possui características semelhantes ao NÓ COMPLEXO, que será utilizado para representar instâncias de objetos componentes, que denominamos de NÓ DE COMPOSIÇÃO. Qualquer nó, Simples, Complexo ou de Composição, possui: ENDEREÇO DE LOCALIZAÇÃO e STATUS, ligado ao controle de acesso. Esta informação adicional irá permitir o controle de acesso ao sistema, pois determinados usuários terão apenas a visão dos objetos que lhes são permitidos.

Um NÓ SIMPLES é usado para armazenar as informações ou instâncias de um dado objeto. Sua característica estrutural e conteúdo dependerá da característica do objeto a que pertence.

Um NÓ COMPLEXO possui várias informações adicionais que permitirão a utilização da estrutura do GOLGO. As principais informações são:

- Um IDENTIFICADOR, ou seja, um nome que irá servir de representação externa do objeto e um DESCRITOR do nó que conterà uma descrição breve do objeto (estas informações constituirão o Dicionário de Objetos do Sistema);

- O conjunto de REFERÊNCIAS VERTICAIS usado para a navegação no sistema, que conterà informações como identificador do objeto componente da classe (que chamaremos de sub-objetos para não confundir com o conceito de *Composição de Objetos*), seu STATUS, que é o atributo de controle de acesso, e um APONTADOR para o endereço real dos sub-objetos;

- O conjunto de OPERAÇÕES INTERNAS usado para ativação das operações internas do objeto. Este conjunto irá conterà informações como IDENTIFICADOR DA OPERAÇÃO, seu STATUS e o APONTADOR para o endereço real do "método";

- Um conjunto de informações para a definição visual do objeto como janelas, ícones, etc.

- Um conjunto de restrições ou regras para a formação correta do objeto. Esta informação está associada a classe de objetos, podendo ser usada para a definição de restrições de instâncias;

- Conjunto de REFERÊNCIAS HORIZONTAIS que representam as operações entre objetos de classes diferentes (operações externas). Cada referência horizontal é constituída de um IDENTIFICADOR, seu STATUS e o endereço onde a operação está armazenada;

- Conjunto de LINKS DE COMPOSIÇÃO que terá informações acerca da composição de objetos (ver Links de Composição na seção seguinte).

Um NÓ DE COMPOSIÇÃO possui também algumas informações adicionais que permitirão a utilização de instâncias "especiais" na estrutura do GOLGO. São consideradas instâncias "especiais", aquelas que são definidas usando o conceito de Composição de Objetos. As principais informações deste tipo de Nó, são:

- um IDENTIFICADOR, ou seja, um nome que irá servir de representação externa da instância do objeto; e um DESCRITOR do nó que conterà uma descrição breve da instância do objeto (estas informações constituirão o Dicionário de Objetos do Sistema);

- Conjunto de LINKS DE COMPOSIÇÃO que terá informações acerca da composição de instância "especiais" dos objetos (ver Links de Composição na seção seguinte).

2.2 CONCEITO DE ARCO

No GOLGO existem três tipos de arcos: Links Verticais, Links Horizontais e Links de Composição.

Os *Links Verticais* ou *Referências Verticais* são usados para a navegação do grafo. Eles apontam do *objeto-pai* (classe) para os *objetos componentes* ou *sub-classe-objetos*. Basicamente estão representados pelo conjunto de referências verticais contidas em cada objeto do sistema. Os links verticais serão visualizados por uma janela acionada pelo usuário que mostrará os links verticais através de um menu. Cada link vertical terá um STATUS para permitir o controle do acesso aos objetos (figura 2).

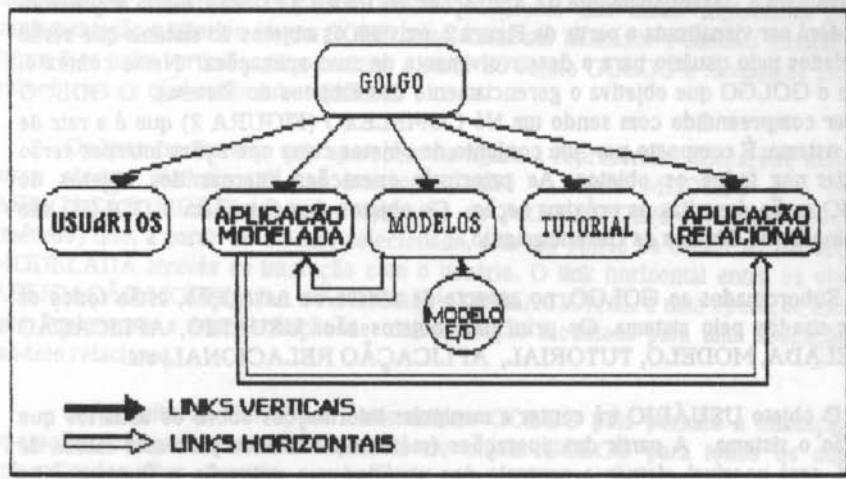


FIGURA 2 - Grafo representativo da arquitetura básica do sistema.

Os *Links Horizontais* ou *Referências Horizontais* definem as operações consideradas externas, pois envolvem a manipulação de objetos de classes diferentes. Estas operações são necessárias pois representam a manipulação do sistema como um todo. Os links horizontais serão representados através do conjunto de referências horizontais no objeto considerado como o *objeto origem* apontando para o "método" ou procedimento para manipulação dos *objetos-destino*. O conjunto de referências horizontais será visualizado sob a forma de uma janela que deverá ser acionada pelo usuário e mostrará os links horizontais pelo seu IDENTIFICADOR dispostos na forma de menu. O usuário irá ver apenas os links horizontais permitidos pelo STATUS de cada link (figura 2).

Para incorporar o conceito de *Composição de Objetos* descrito em [MACE89], foi criado um link (apontador) que parte do *objeto composto* para o *objeto componente*. Este link possui uma semântica diferente dos outros citados acima. Existe um atributo para classificação dos *links de composição*. O atributo indica: Link forte e Link fraco. Um link de composição é considerado como *Link Forte* quando o objeto componente é imprescindível à composição do *objeto composta*. Por outro lado, o *Link Fraco* indica

que o *objeto componente* poderá ser opcional (existir ou não) na composição do *objeto composta*.

2.3 ARQUITETURA BÁSICA DO AMBIENTE

O que será apresentado nesta seção é a Arquitetura Básica do Ambiente Interativo para o Desenvolvimento de Aplicações em Banco de Dados. Nesta arquitetura que poderá ser visualizada a partir da Figura 2, existem os objetos do sistema que serão manipulados pelo usuário para o desenvolvimento de suas aplicações. Neste contexto, situa-se o GOLGO que objetiva o gerenciamento dos Objetos do Sistema. O GOLGO pode ser compreendido com sendo um Nó COMPLEXO (FIGURA 2) que é a raiz de todo o sistema. É composto por um conjunto de objetos cujas operações internas serão herdadas por todos os objetos. As principais operações internas dos objetos do GOLGO serão descritas na próxima seção. Os objetos que compõem o GOLGO são denominados de Objetos de Gerenciamento.

Subordinados ao GOLGO, no aspecto de acesso ou navegação, estão todos os objetos usados pelo sistema. Os principais objetos são: USUÁRIO, APLICAÇÃO MODELADA, MODELO, TUTORIAL, APLICAÇÃO RELACIONAL, etc.

O objeto USUÁRIO irá conter e manipular informações sobre os usuários que utilizarão o sistema. A partir das operações (métodos) definidos para esta classe de objetos, será possível efetuar o controle dos usuários que utilizarão o Sistema. As principais operações internas do objeto USUÁRIO são: Criação, Remoção, Alteração, Consulta, Pesquisa, etc. Cada usuário no Sistema possuirá uma SENHA, um CÓDIGO interno, e outras informações necessárias ao Sistema.

O objeto APLICAÇÃO MODELADA irá conter todas as aplicações geradas a partir do modelo selecionado. A esta classe de objeto estará associada o Controle de Versões efetuado por um Objeto de Gerenciamento descrito no tópico seguinte. Este objeto permitirá ao usuário do Sistema, representar a sua Aplicação definida a partir do método de Modelagem associado a cada Modelo utilizado. Serão utilizados os Nós de Composição para a representação de instâncias que herdam a Composição definida no Modelo utilizado.

O objeto MODELO irá conter todos os modelos implementados no sistema e que serão usados para a geração dos objetos APLICAÇÃO MODELADA. Cada objeto do Modelo será representado com sua sintaxe e semântica que permitirão a definição conceitual de qualquer aplicação. Este objeto utilizará o conceito de Composição de Objetos para efetuar a Modelagem. Um determinado objeto do Modelo poderá ser construído a partir da composição de outros.

O objeto TUTORIAL irá controlar informações como textos e questões visando o aprendizado dos usuários. Cada aplicação modelada poderá ter um tutorial correspondente funcionando como documentação e aprendizado ao mesmo tempo, sendo criada pelo usuário especialista da aplicação.

O objeto APLICAÇÃO RELACIONAL irá conter todas as aplicações modeladas e transformadas para o modelo relacional que serão *exportadas* para a implementação posterior em um SGBD relacional a ser definido. Podemos visualizar na figura 2 os links verticais que permitem a partir do objeto GOLGO a navegação para os sub-objetos ao qual estão subordinados.

Os links horizontais que aparecem na figura 2 representam operações externas entre objetos distintos. O link horizontal entre o objeto MODELO e o objeto APLICAÇÃO MODELADA é a operação de *MODELAGEM* que é um processo (método) que, a partir do modelo selecionado, cria um objeto da classe APLICAÇÃO MODELADA através da interação com o usuário. O link horizontal entre os objetos APLICAÇÃO MODELADA e APLICAÇÃO RELACIONAL é uma operação externa que implementa a transformação de uma aplicação modelada para uma aplicação no modelo relacional.

O conceito de *herança* é muito útil no GOLGO pois permite a utilização de operações básicas do sistema contidas no objeto GOLGO para todos os objetos subordinados. Por exemplo, a criação de uma APLICAÇÃO MODELADA deverá utilizar as operações básicas CRIAÇÃO DE NÓ e CRIAÇÃO DE LINKS VERTICAIS herdadas dos objetos de gerenciamento do GOLGO.

3. AMBIENTE DE IMPLEMENTAÇÃO DO GOLGO

Como vimos na seção anterior o GOLGO é um grafo genérico, ou seja, seus nós podem ser classes, objetos e instâncias dos mais diversos tipos. O usuário pode navegar por este grafo e acessar os objetos que lhes são permitidos pelo STATUS, e assim modelar sua aplicação com base nos objetos do Modelo Estático/Dinâmico (em desenvolvimento no projeto Apoio do Departamento de Informática da UFPE) de forma interativa.

No GOLGO, cada objeto é uma entidade que além de possuir os conceitos básicos do paradigma da orientação a objetos, possuem outras características associadas tais como: Definição Visual, o próprio objeto possui uma ou mais formas de se apresentar ao usuário, através de janelas, ícones, etc.; possui um mecanismo de controle de acesso visando garantir a segurança; uma forma de associação que permite explicitar e controlar a composição de objetos; um Dicionário de Objetos contendo informações acerca da semântica dos objetos e suas operações. Nesta seção serão abordados: o ambiente de implementação do GOLGO e sua arquitetura funcional.

3.1 ARQUITETURA DO GOLGO

A implementação de um grafo genérico, cujos nós são de classes distintas, se torna possível quando uma linguagem de programação fornece construtores de tipos de dados permitindo assim um vasto conjunto de aplicações, com estruturas de dados coerentes. Aliando a estas características as propriedades de orientação a objetos, uma linguagem torna-se uma poderosa ferramenta no desenvolvimento de sistemas complexos. A linguagem C++ foi escolhida como ferramenta para implementar o Ambiente de Desenvolvimento do Projeto APOIO por permitir a modularidade do sistema, pois os módulos são os próprios objetos com suas operações correspondentes, o que motivou a opção por esta linguagem.

Na implementação do GOLGO foi considerada esta modularidade, e cada classe de objetos que compõe o sistema é responsável por um controle específico de determinada estrutura. O sistema como um todo, pode ser visto como objetos que se comunicam entre si como se fossem partes de uma engrenagem que se inter-relacionam.

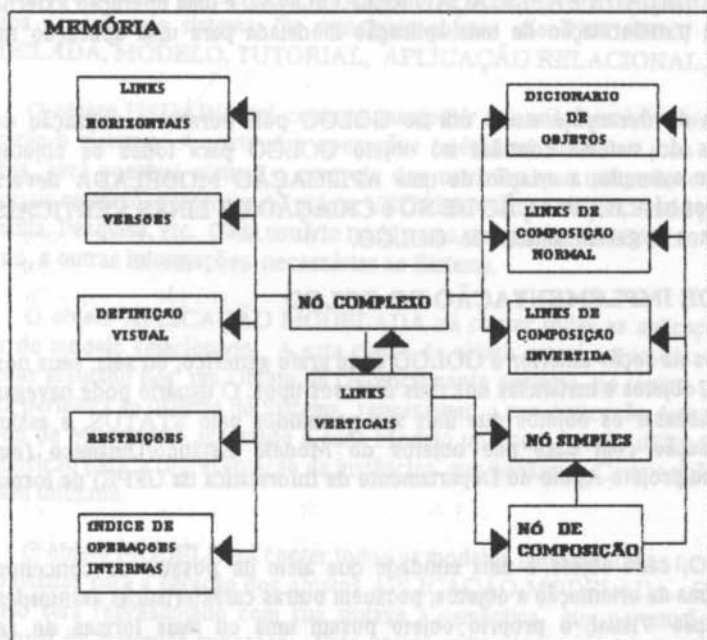


FIGURA 3 - Diagrama mostrando os objetos de gerenciamento do GOLGO e seus relacionamentos. As setas indicam os apontadores existentes em cada objeto que permitem a associação das instâncias.

O GOLGO tem em sua arquitetura funcional as seguintes classes de objetos (figura 3):

- Nó Complexo;
- Nó Simples;
- Nó de Composição;
- Links Verticais;
- Links Horizontais;
- Índice de Operações Internas;
- Links de Composição Normal;
- Links de Composição Invertida;
- Definição Visual;
- Dicionário de Objetos;
- Restrições;
- Memória;
- Versões;

Os objetos citados acima, que doravante chamaremos de *Objetos de Gerenciamento*, são o suporte básico do GOLGO que permitem a manipulação e controle dos demais objetos (ditos *Objetos do Sistema*) do Ambiente de Concepção de Aplicações em Banco de Dados (figura 1). Em seguida serão descritos sucintamente os objetos de gerenciamento. Os conceitos referentes aos objetos que serão descritos a seguir estão especificados na seção 2 deste trabalho.

3.1.1 OBJETO: NÓ COMPLEXO

Sua estrutura interna é constituída por um conjunto de apontadores para o endereço onde o objeto ou classe é referenciado nos outros objetos de gerenciamento. A partir de um Nó Complexo, o usuário, através da Interface, pode acessar todos os objetos de gerenciamento e assim efetuar o processamento desejado. Algumas operações internas deste objeto são: Inserção, Remoção, Atualização de Apontador, Recuperação de Apontadores, etc.

3.1.2 OBJETO: NÓ SIMPLES

Este objeto contém a informação propriamente dita. Sendo um objeto genérico, pois o tipo de informação armazenada não tem uma estrutura definida, este objeto será definido como uma SuperClasse e todos os objetos do sistema são definidos como *Classes Derivadas* [WIEN88] desta SuperClasse. Isto permite que as propriedades do objeto Nó Simples sejam herdadas pelas Classes Derivadas e que quando for criada uma instância de um objeto qualquer do sistema, ela seja uma instância da Classe Nó Simples sendo manipulada pelo gerenciador como um Nó Simples. As operações internas do objeto Nó Simples são todas polimórficas porque o objeto tem instâncias genéricas, sendo consideradas *Funções Virtuais* [WIEN88]. O objeto Nó Simples terá operações tais como: inserção, remoção, alteração, imprime, etc.

3.1.3 OBJETO: NÓ DE COMPOSIÇÃO

Sua estrutura interna é constituída por um conjunto de apontadores para o endereço onde o objeto ou classe é referenciado nos outros objetos de gerenciamento: Links de Composição Normal, Links de Composição Invertida, Nó Simples e Dicionário de Objetos. A partir de um Nó de Composição, o Sistema, através da Interface, pode acessar outras instâncias de Nó de Composição e assim efetuar a Composição eficiente de objetos. Algumas operações internas deste objeto são: Inserção, Remoção, Atualização de Apontador, Recuperação de Apontadores, etc.

3.1.4 OBJETO: LINKS VERTICAIS

A estrutura interna deste objeto, permitirá que sejam criadas dois tipos de listas encadeadas: uma lista contendo informações sobre todos os Nós Complexos e uma outra lista associada a cada elemento da primeira lista, contendo todos os sub-objetos subordinados àquele Nó Complexo. O objeto Links Verticais será usado pela Interface para a navegação no Sistema e acesso aos objetos e suas instâncias. Algumas operações internas: Inserção, Remoção, Atualização de Apontadores, Alteração de Status, etc.

3.1.5 OBJETO: LINKS HORIZONTAIS

Sua estrutura interna conterá apontadores, identificador e Status destas operações que estão associadas a objetos Nós Complexos. Algumas operações internas deste objeto são: Inserção, Remoção, Ativação, etc.

3.1.6 OBJETO: ÍNDICE DE OPERAÇÕES INTERNAS

Este objeto possui uma estrutura interna que permitirá que as operações sejam agrupadas por objeto do sistema representados pelos Nós Complexos mantendo informações tais como, Identificador, Status e Apontador para as operações internas. Com isto, torna-se possível em tempo de execução, a utilização de qualquer método definido para os objetos do Sistema, pelo usuário autorizado. Algumas operações internas deste objeto são: Inserção, Remoção, Alteração de Status, Ativação de Operação, etc.

3.1.7 OBJETO: LINKS DE COMPOSIÇÃO NORMAL

A estrutura interna deste objeto permite manter e controlar a composição de objetos através de estruturas semelhantes a do objeto Links Verticais, contendo uma lista de objetos compostos e outras listas associadas a cada objeto composto, contendo informações dos objetos componentes. Algumas operações internas: Inserção, Remoção, Alteração, Pesquisa, etc.

3.1.8 OBJETO: LINKS DE COMPOSIÇÃO INVERTIDA

A função do objeto Links de Composição Invertida é manter e controlar os Links que apontam para um objeto componente, ou seja, este objeto fornecerá para um objeto componente, todos os objetos compostos que o utilizam em suas composições. Sua estrutura e operações internas são semelhantes ao objeto Links de Composição Normal.

3.1.9 OBJETO: DEFINIÇÃO VISUAL

No sistema é possível associar um ou mais formas gráficas de apresentação de um objeto. Sua estrutura interna permitirá, para cada objeto do sistema, o acesso e manutenção das diversas formas de definição visual além de controlar todo os processos de visualização dos objetos do sistema, tais como: mecanismos de janela, exibição de gráficos, etc. . Algumas operações internas são: Inserção, Remoção, Montador de Definição Visual, Gerenciador do mecanismo de janela, etc.

3.1.10 OBJETO: DICIONÁRIO DE OBJETOS

Este objeto além de permitir a documentação dos objetos e operações, possibilita o acesso aos objetos do sistema através do nome de identificação. Sua estrutura interna em forma de árvore de pesquisa binária permite a rápida recuperação de informações sobre os objetos do sistema. Algumas operações internas deste objeto são: Inserção, Remoção, Consulta, Alteração, Pesquisa, etc.

3.1.11 OBJETO: RESTRIÇÕES

Este objeto permite a associar a um determinado objeto do sistema, representado por um Nó Complexo, um conjunto de restrições previamente definidas pelo usuário, com o objetivo de orientar a formação de suas instâncias. Suas operações internas principais são: Inserção, Remoção, Alteração, etc.

3.1.12 OBJETO: MEMÓRIA

O objeto do gerenciador GOLGO responsável pelo controle e armazenamento das instâncias de todos os objetos de gerenciamento é denominado MEMÓRIA. Sua estrutura é dividida em duas sub-estruturas: Memória Real e Memória Virtual. A sub-estrutura Memória Real é responsável pela alocação e recuperação de qualquer informação alocada na memória RAM e pode ser decomposta em: Tabela de Alocação, Áreas de Alocação e Área de Overflow. A Tabela de Alocação é usada para controlar os endereços bases das diversas áreas nas quais a memória RAM está particionada; as Áreas de Alocação são porções de memória reservadas para cada objeto de gerenciamento do Sistema; a Área de Overflow é usada para alocação de informações, estendendo as Áreas de Alocação.

A sub-estrutura Memória Virtual controla o armazenamento em disco rígido e está associada a cada Área de Alocação podendo ser decomposta em: Área de Armazenamento Virtual, Estrutura de Controle de Memória Virtual e Lista de Espaços Vazios. A Área de Armazenamento Virtual é o lugar destinado ao armazenamento propriamente dito. A Estrutura de Controle de Memória Virtual efetua o controle de alocação na Área de Armazenamento Virtual. A Lista de Espaços Vazios destina-se ao controle de blocos livres na Área de Armazenamento visando o reaproveitamento destes.

O objeto MEMÓRIA possui uma estrutura que abrange tanto a Memória Real quanto a Memória Virtual. A FIGURA 4 ilustra a representação das estruturas em memória RAM. As principais operações que manipulam a estrutura do objeto MEMÓRIA que serão usadas pelos demais objetos de gerenciamento são: Inicialização de Memória, Transferência Física Real-Virtual, Transferência Física Virtual-Real, Transferência Lógica, Alocação Real, Desalocação Real, Alocação Virtual, Desalocação Virtual, Compactação, etc.. Outras operações estão especificadas em [MACE89a].

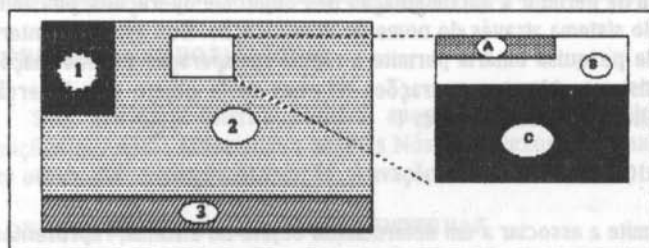


FIGURA 4 - Localização das áreas de memória em RAM, Tabela de Alocação (1), Área Total de Alocação (2), Área de Overflow (3), Sub-área de Controle de Memória Virtual (A), Sub-área da Lista de Espaços Vazios (B), Sub-área de Armazenamento Real (C).

3.1.13 OBJETO: VERSÕES

Este objeto é responsável pelo controle de versões de objetos do Sistema. Sua estrutura interna permitirá que sejam criadas dois tipos de listas encadeadas: uma lista contendo informações sobre todos os Nós Complexos que possuem mais de uma versão ativa e uma outra lista associada a cada elemento da primeira lista, contendo apontadores para as instâncias de Nós Complexos que representam cada versão criada no Sistema. O objeto Versões será usado pela Interface para o acesso a todas as versões de Aplicações Modeladas criadas pelos usuários. Entre as operações internas tem-se: Inserção, Remoção, Atualização de Apontadores, Alteração de Status, etc.

3.2 FUNCIONAMENTO DO GOLGO

O GOLGO, como vimos nas seções anteriores, possui uma estrutura funcional composta por objetos chamados Objetos de Gerenciamento, e sua finalidade é efetuar o controle dos objetos do sistema. Cada Objeto de Gerenciamento tem sua função no Sistema com seus respectivos métodos que manipulam suas instâncias. Para um determinado Objeto do Sistema, existe um conjunto de instâncias em cada Objeto de Gerenciamento. A instância do Nó Complexo possui apontadores para as demais instâncias dos Objetos de Gerenciamento.

Entretanto, para que o sistema funcione é necessário a existência de um método que manipule todos os Objetos de Gerenciamento possibilitando a *NAVEGAÇÃO* no Sistema. A navegação permite que um determinado usuário, desde que tenha acesso, visualize os objetos, manipule-os através do envio das mensagens disponíveis, e crie as instâncias. A visualização de objetos é realizada através do objeto Definição Visual. Um mesmo objeto poderá ter mais de uma forma de apresentação visual (ícones, janelas, gráficos, etc) que poderão ser escolhidas pelo usuário. A navegação somente é efetuada a partir do objeto Links Verticais. Para cada classe de objeto do Sistema, está associada uma instância do objeto Links Verticais que possui um parâmetro que define a permissão de acesso a um determinado objeto referenciado pelos Links. A manipulação de um objeto do Sistema é realizado através de suas operações internas. Para cada objeto do Sistema existe associado um conjunto de apontadores que permitem o envio da mensagem ao objeto, em tempo de execução, para a realização da operação pretendida. Este conjunto de apontadores é controlado pelo objeto de gerenciamento Índice de Operações Internas. Outras operações podem também, ser acionadas pelo usuário. São operações representadas pelo objeto de gerenciamento Links Horizontais. Estas operações envolvem objetos do Sistema de classes distintas. O método que realizará toda a manipulação dos objetos de gerenciamento para o acesso e representação dos objetos do Sistema, é a *INTERFACE*. Além das operações acima, a *INTERFACE* poderá manipular com Restrições, Controle de Versões, Links de Composição e Dicionário de Objetos. Portanto, a *INTERFACE* orientada a objetos permitirá a utilização dos objetos do Sistema usando os objetos de gerenciamento para manipulá-los.

4. CONCLUSÃO

O GOLGO da forma como foi criado, usando toda a capacidade de modularidade da filosofia de programação orientada a objetos, tem uma estrutura hierarquizada de classes de objetos bastante simplificada. A idéia básica da criação deste gerenciador foi, além de permitir a manipulação de objetos conforme o paradigma da orientação a objetos, permitir o desenvolvimento dos objetos de gerenciamento com funções específicas, explicitando a comunicação entre os objetos através das mensagens que eles trocam. O

GOLGO então, é um sistema composto por partes distintas, que juntas fornecem todo o controle necessário à manipulação dos objetos do Sistema.

Foram definidas as estruturas internas (data hiding) de cada objeto de gerenciamento com suas respectivas operações internas, que estão sendo implementadas usando a linguagem de programação orientada a objetos C++ [WIEN88].

5. BIBLIOGRAFIA

- [BROW86] BROWN P. J: "Interactive Documentation, in Software " - Practice and Experience ,Vol 16, Mar 1986, pp 291-299.
- [BROW87] BROWN, P. J.: "Hypertext : The Way Forward", University of Kent, Canterbury, 1987.
- [BUSH45] BUSH, V. : "As We May Think", Atlantic Monthly, nº176 , pp 101-108, julho 1945.
- [CONK87] CONKLIN J. : "A Survey of Hypertext", MCC TR, Nr. STP-356-86 Rev1, Austin, TX, Feb 1987.
- [COX 86] COX, B. , HUNT, B.: "Objects, Icons, and Software IC's", BYTE Magazine, 11(8), August 1986, 161-176.
- [GARR86] GARRETT, N., SMITH, K. E., MEYROWITZ, N.: "Intermedia: Issues, Strategies and Tactics in the Design of a Hypermedia Document System, IRIS, Brown University.
- [HOOK87] HOOK, S. : "Object-Oriented Programming: An Evolutionary Approach", 1987.
- [KAY77] KAY, A. et. al: "Personal dynamic media", Computer, março 1977.
- [MACE89] MACEDO, M.A.C., FONSECA, D., "GOLGO - Gerenciador Orientado a objetos", 4º SBBD - Campinas - SP, Abril de 1989.
- [MACE89a] MACEDO, M.A.C., SOUZA, P.S., MELO, E.R., NETO, M.C.A: " Gerenciamento de Memória no GOLGO", Relatório do Projeto Apoio nº 01/89, DI-UFPe, Abril de 1989.
- [PASC86] PASCOE, G. : "Elements of Object-Oriented Programming", BYTE Magazine, 11(8), August 1986, 136-144.

- [SCHE86] SCHEIFLER R.W.,GETTY J.: "The XWINDOW System". MIT Laboratory for Computer Science - Digital Equipment Corp. MIT Project Athena 1986.
- [SCHM86] SCHMUCKER, K. : "MacApp: An Application Framework", BYTE Magazine, 11(8), August 1986, 189-194.
- [TAKA88] TAKAHASHI, T.: "Introdução a Programação Orientada a Objetos", III EBAI, Curitiba, Janeiro de 1988.
- [VASC88] VASCONCELOS, A. M. L., MELO, A. C. V., ALBUQUERQUE, E.S., MEIRA, S.R., "Hipertexto - Conceitos e Características", D.I.- UFPe- Setembro 1988.
- [WIEN88] WIENER, R.S., PINSON, L.J.: "An Introduction to Object-Oriented Programming and C++", Addison-Wesley Publishing Company, 1988.

PRELIMINARES

Este trabalho tem como objetivo de introduzir ao leitor o conceito de Engenharia de Software, bem como a importância da Engenharia de Software no desenvolvimento de sistemas de software. A engenharia de software é a aplicação dos princípios da engenharia de software para a produção de software. A engenharia de software é a aplicação dos princípios da engenharia de software para a produção de software.

A Engenharia de Software é a aplicação dos princípios da engenharia de software para a produção de software. A engenharia de software é a aplicação dos princípios da engenharia de software para a produção de software.