MODELAMENTO DE SISTEMAS PROPULSIVOS ESPACIAIS

Marcelo Furlan Salles[1] (UBC, Bolsista PIBIC/CNPq)
Dr. José Nivaldo Hinckel[2] (ETE/DMC/INPE)

RESUMO

Este trabalho, iniciado em agosto de 2001, ao projeto de Iniciação Científica, tem como objetivo o desenvolvimento de um programa geral para modelamento de sistemas propulsivos de uso espacial envolvendo pesquisadores do INPE, IAE e um pesquisador visitante do MAI. Na execução do projeto, nos primeiros 6 meses, foram realizadas apenas atividades introdutórias sobre a linguagem de programação em C ++. Outro programa inserido também, foi um programa de simulação utilizando os métodos e ferramentas de modelamento, (UML – Unified Modelling Language). O método e as ferramentas específicas permitem a criação de diferentes diagramas contendo a descrição dos diferentes componentes do sistema e as interações entre estes componentes. Os diagramas gerados desta forma funcionam como "plantas" do programa em desenvolvimento e como tal facilitam a comunicação entre os diferentes agentes envolvidos no projeto e a documentação decisões tomadas sobre propriedades e funcionalidades de cada subsistema e interações entre eles. Os resultados parciais obtidos são: o aperfeiçoamento da linguagem de programação C ++, uma linguagem mundialmente utilizada em diversas áreas de programação em geral. Início do aprendizado do software de modelamento UML, na simulação de sistemas propulsivos espaciais, através de métodos e ferramentas de modelamento. Iniciou-se também, a elaboração de um banco de dados relativos a propriedades de propelentes líquidos usuais em propulsores de uso espacial e dados moleculares para obtenção de propriedades termodinâmicas de espécies gasosas presentes em produtos de combustão de propulsores a propelentes líquidos. Para dar continuidade a este projeto de Iniciação Científica estão programadas as atividades: Obtenção dos dados dos propelentes e espécies gasosas na literatura especializada, aperfeiçoamento do conhecimento da linguagem de programação C ++ e do software UML, elaboração dos esboços de interfaces para configuração dos componentes do sistema propulsivo e implementação das interfaces em um sistema gráfico.

[1] Aluno do Curso de Bacharelado em Ciências da Aeronáutica, UBC. **E-mail: naca4444@yahoo.com.br**
[2] Tecnologista Sênior da ETE/DMC. **E-mail: hinckel@dem.inpe.br**

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

MODELAMENTO DE SISTEMAS PROPULSIVOS ESPACIAIS

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)

Marcelo Furlan Salles (UBC, Bolsista PIBIC/CNPq)
E-mail: naca4444@yahoo.com.br

Dr. José Nivaldo Hinckel (ETE/DMC/INPE, Orientador)
E-mail: hinckel@dem.inpe.br

COLABORADOR

Dr. José Nivaldo Hinckel (ETE/DMC/INPE)

Junho de 2002

AGRADEÇO

Ao Dr. José Nivaldo Hinckel, pela excelência da qualidade de orientador e amigo, pelo seu apoio e orientação durante a elaboração deste trabalho.

Ao CNPq pela grande oportunidade que tive de mostrar meus conhecimentos, e principalmente, pela experiência adquirida durante a vigência desta bolsa de Iniciação Científica.

ÍNDICE

1. INTRODUÇÃO

Este trabalho é focado em um estudo sobre os métodos e ferramentas de modelamento, na simulação de sistemas propulsivos espaciais.

Esses métodos e as ferramentas específicas permitem a criação de diferentes diagramas contendo a descrição dos diferentes componentes do sistema e as interações entre estes componentes. Os diagramas gerados desta forma funcionam como "plantas" do programa em desenvolvimento e como tal facilitam a comunicação entre os diferentes agentes envolvidos no projeto e a documentação decisões tomadas sobre propriedades e funcionabilidades de cada subsistema e interações entre eles.

Quando falamos em Modelagem de sistemas em geral, existem diversos programas de simulação utilizando os métodos e ferramentas de modelamento. O UML (Unified Modelling Language) é utilizado para gerenciar a complexibilidade associada ao grande número de entidades (componentes e subsistemas) e relações entre estes num modelo de simulação e análise de sistemas propulsivos espaciais.

A utilização de diagramas permite a visualização do sistema completo por diferentes ângulos e com diversos graus de abstração, facilitando a discussão, a compressão e a documentação da natureza e comportamento dos componentes e das interações entre eles.

Para uma determinação particular foi incluído nesse trabalho um banco de dados relativos a propriedades de propelentes líquidos usuais de uso espacial e dados moleculares para obtenção de propriedades termodinâmicas de espécies gasosas presentes em produtos de combustão de propulsores a propelentes líquidos.

O banco de dados será utilizado na elaboração de uma biblioteca especializada ao fornecimento de propriedades termodinâmicas de espécies gasosas e misturas de espécies gasosas.

Os processos termodinâmicos envolvem reações químicas com liberação de energia e escoamentos com grandes gradientes de pressão e temperatura.

2. A UML – UNIFIED MODELLING LANGUAGE

A UML é usada no desenvolvimento dos mais diversos tipos de sistemas. Ela abrange sempre qualquer característica de um sistema em um de seus diagramas e é também aplicada em diferentes fases do desenvolvimento de um sistema, desde a especificação da análise de requisitos até a finalização com a fase de testes.

A definição e documentação de uma arquitetura do sistema, facilita também a manutenção e atualização do código gerado.

A UML é uma tentativa de padronizar a modelagem orientada a objetos. Existem várias metodologias de modelagem orientada a objetos. A UML unificou as melhores idéias de cada uma dessas metodologias orientada a objetos:

2.1. Algumas das Principais Metodologias Orientadas a Objeto

BOOCH - Uma versão de orientação a objeto que defini a noção de que um sistema é analisado a partir de um número de visões, onde cada visão é descrita por um número de modelos e diagramas.

OMT - Técnica de Modelagem de Objetos - Método especialmente voltado para o teste dos modelos, baseado nas especificações da análise de requisitos do sistema, criado por James Rumbaugh. Este método é composto pela junção dos modelos de objetos, funcional e use-case.

OOSE/Objectory - Os métodos OOSE e o Objectory são baseados na utilização de use-cases, que definem os requisitos iniciais do sistema, visto por um ator externo. Foi desenvolvido pelo ponto de visto de Ivar Jacobson.

Diante desta diversidade de conceitos, "os três amigos", Grady Booch, Rumbaugh e Ivar Jacobson decidiram criar uma Linguagem de Modelamento Unificada - UML.

2.2. Objetivo da UML

O objetivo da UML é descrever qualquer tipo de sistema, em termos de diagramas orientado a objetos. Num padrão de linguagem para visualização, construção e documentação de artefatos de um sistema software, ao longo de todo ciclo de desenvolvimento e diferentes tecnologias de implementação.

Naturalmente, o uso mais comum é para criar modelos de softwares, mas a UML também é usada para representar sistemas mecânicos sem nenhum software.

- A modelagem de sistemas (em geral) usando os conceitos da orientação a objetos;

- Estabelecer uma união fazendo com que métodos conceituais sejam também executáveis;

- Criar uma linguagem de modelagem usável tanto pelo homem quanto pela máquina.

A UML pode ser usada para:

- Mostrar as fronteiras de um sistema e suas funções principais utilizando atores e casos de usos;
- Ilustrar a realização de casos de uso com diagramas de interação;
- Representar uma estrutura estática de um sistema utilizando diagramas de classe;
- Modelar o comportamento de objetos com diagramas de transição de estado;
- Revelar a arquitetura de implementação física com diagramas de componentes e de implementação;
- Estender sue funcionabilidade através de estereótipos.

2.3. Fases de Desenvolvimento de um Sistema em UML

Existem cinco fases no desenvolvimento de sistemas em geral: análise de requisitos, análise, design (projeto), programação e testes. Abaixo vamos ver um por um.

Análise de Requisitos

Esta fase captura as intenções e necessidade dos usuários do sistema a ser desenvolvido através do uso de funções chamadas "use-cases". Através do desenvolvimento de "use-case", as entidades externas ao sistema (em UML chamados de "atores externos") que interagem no sistema que será modelado. Os atores externos e os use-cases, são modelados com relacionamentos que possuem comunicação associativa entre eles ou são desmembrados em hierarquia. O diagrama de use-cases mostrará que os atores externos, ou seja, os usuários do futuro sistema deverão esperar do aplicativo, conhecendo toda sua funcionabilidade.

Análise

A fase de análise está preocupada com as primeiras abstrações (classes e objetos) e mecanismos que estarão presentes no domínio do problema. As classes são modeladas e ligadas através de relacionamentos com outras classes, e são descritas no Diagrama de Classe. As colaborações entre classes também são mostradas neste diagrama para desenvolver os use-cases, modelados anteriormente, estas colaborações são criadas através de modelos dinâmicos em UML.

Design (Projeto)

Na fase de design, o resultado da análise é expandido em soluções técnicas. Novas classes serão adicionadas para prover uma infra-estrutura técnica: a interface do usuário e de periféricos, gerenciamento do banco de dados, comunicação com outros sistemas, dentre outros. O design resulta no detalhamento das especificações para a fase de programação do sistema.

Programação

Na fase de programação, as classes provenientes do design são convertidas para o código da linguagem orientada a objetos escolhida. No momento da criação de modelos de análise e design em UML, é melhor evitar traduzi-los mentalmente em código. A programação é uma fase separada e distinta, onde os modelos criados são convertidos em código.

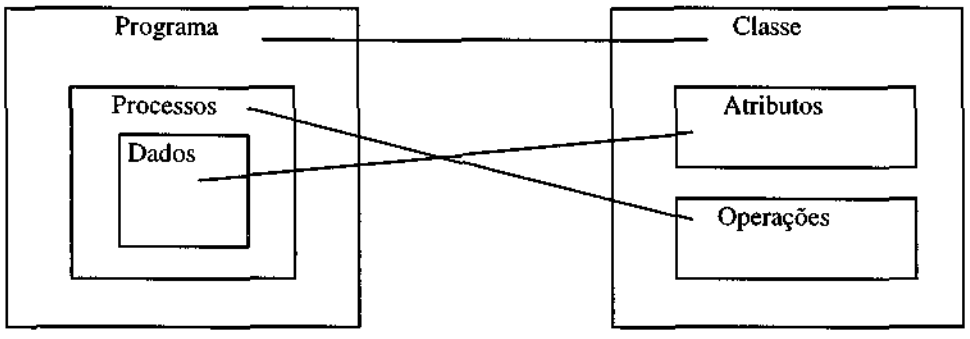
Testes

Um sistema normalmente é rodado em testes de unidade, integração, e aceitação. Os testes de unidade são para classes individuais ou grupos de classes e são geralmente testados pelo programador. O sistema será testado pelo usuário final e verificará se os resultados mostrados estão realmente de acordo com as intenções do usuário final.

2.4. A Mudança do Enfoque

O enfoque de modelagem por objetos, vê o mundo como uma coletânea de objetos que interagem entre si, apresentam características próprias que são representadas pelos seus atributos (dados) e operações (processos).

Enfoque baseado em **sistemas** *versus* enfoque baseado em **objeto**



2.5. Orientação a Objeto

A orientação a objeto não é só teoria, mas uma tecnologia de eficiência e qualidade comprovada usada em inúmeros projetos e para construção de diferentes tipo de sistemas. A orientação a objetos requer um método que integre o processo de desenvolvimento e a linguagem de modelagem com a construção de técnicos e ferramentas adequadas.

Definição: Um objeto é uma instância de uma classe. Ex: Um veículo, uma região e etc.

Objetos se comunicam através de mensagem, isto é, um sinal de um objeto a outro requisitando um serviço através da execução de uma operação. Trata-se de um ciclo completo onde uma mensagem é enviada a um objeto, operações são executadas dentro desse com base nos dados de seu alcance na hierarquia de classes, e uma mensagem contendo o resultado da operação é enviada ao objeto solicitante.

2.6. Classe

Definição: Uma classe é a descrição de um tipo de objeto. É uma coleção de objetos que podem ser descritos com os mesmos atributos e as mesmas operações. Dentro de uma classe todas as operações e atributos são acessíveis aos outros membros da classe, tudo dentro de uma classe é conhecido por ela mesma e por seus descendentes. Usam-se classes para classificar os objetos que identificamos no mundo real. Ex:

Classe Mamíferos	Classe Aves
- Cavalos	- Beija-flores
- Golfinhos	- Águias

Classe **Objeto**

Cliente	<---Nome da Classe / Nome do Objeto	Pablo Barros:Cliente
Nome: String Idade : Num	<---Atributos	Nome : "Pablo Barros" Idade : 20
Criar () Destruir ()	<---Operações	Criar () Destruir ()

3. Notações Genéricas

Na UML são definidos alguns elementos cujas notações podem ser empregadas ao longo dos diversos diagramas propostos. Os principais são pacotes, estereótipo e nota.

3.1. Pacotes

Pacote é um mecanismo de agrupamento, onde todos os modelos de elementos podem ser agrupados. Em UML, um pacote é definido como: "Um mecanismo de propósito geral para organizar elementos semanticamente relacionados em grupos". Pacotes podem importar modelos de elementos de outros pacotes. Na grande maioria dos casos, os pacotes possuem relacionamentos com outros pacotes. Os relacionamentos permitidos entre pacotes são de dependência, refinamento e generalização (herança). Um pacote pode somar associações para classes referenciadas mas não pode modificar seus conteúdos (atributos e operações).

Um pacote pode ter visibilidade para indicar como outros pacotes terão acesso a seu conteúdo. A UML define quatro tipos de visibilidade para pacotes:

Privada - nenhuma outra classe pode usar diretamente, significa que esse elemento apenas pode ser usado/referenciado por elementos definidos no mesmo pacote (-)

Protegida - só subclasses dessa classe podem usar, significa que esse elemento pode ser usado/referenciado por um elemento definido no mesmo pacote ou num outro pacote que seja uma especialização (através da relação de herança) do primeiro (#)

Pública - qualquer outra classe pode usar, significa que esse elemento pode ser usado/referenciado por qualquer outro elemento independente do local onde é definido (+)

Implementação - declarado no corpo de uma operação

3.2. Estereótipos

Um estereótipo é um metatipo, isto é, um tipo que descreve tipos. Permite definir novos tipos de elementos no metamodelo do UML. O conceito de estereótipo providencia um mecanismo de classificar elementos, de tal forma que eles se comportem, de alguma forma, como se fossem instâncias de novos, mas virtuais, construtores definidos no metamodelo.

3.3. Notas

Nota ou anotações ilustra um comentário e não tem qualquer impacto semântico, no sentido que o seu conteúdo não altera o significado do modelo no qual ela se encontra. Deve ser descrito informalmente: requisitos, restrições, observações ou explicações nas notas.

Conclusão: Uma classe é um modelo utilizado, para criar objetos específicos e relacionados, uma descrição de um conjunto de instâncias que compartilham as mesmas operações, atributos, relacionamentos e semântica. Uma instância de uma classe é um objeto real, possui dados e aguarda por executar serviços.

Uma classe define os atributos compartilhados por todos os objetos criados a partir dela e estabelece relacionamentos com outras classes: duas classes podem compartilhar alguns atributos mas manipulam outros atributos diferentes.

Na programação orientada a objeto, a herança permite expressar relacionamentos entre classes onde as classes bases subjacentes derivam em subclasses mais complexas. Uma classe básica contém todos os atributos que são compartilhados entre as classes derivadas da classe base. Subclasses usam atributos compartilhados existentes, reimplementam atributos compartilhados ou criam novos atributos.

4. Relacionamentos

Os relacionamentos ligam as classes/objetos entre si criando relações lógicas entre estas entidades. Estão citadas abaixo cada tipo de relacionamento e suas perspectivas sub-divisões:

4.1. Associações

Uma associação representa que duas classes possuem uma ligação entre elas, para cada X existe um Y, e suas sub-divisões:

- Associações Normais;
- Associação Recursiva;
- Associação Qualificada;
- Associação Exclusiva;
- Associação Ordenada ;
- Associação de Classe;
- Associação Ternária;
- Agregação: a agregação é um caso particular da associação. A agregação indica que uma das classes do relacionamento é uma parte, ou está contida em outra classe.

4.2. Generalizações (herança)

A generalização é um relacionamento entre um elemento geral e um outro mais específico. O elemento mais específico possui todas as características do elemento geral e contem ainda mais particularidades. Existem alguns tipos de generalizações que variam em sua utilização a partir da utilização. Generalização normal e restrita. As generalizações restritas se dividem em generalização de sobreposição, disjuntiva, completa e incompleta.

4.3. Dependência e Refinamentos

Além das associações e generalizações, existem ainda dois tipos de relacionamentos em UML.

O relacionamento de dependência: Quando uma classe recebe um objeto de outra classe como parâmetro, uma classe acessa o objeto global da outra. Nesse caso existe uma dependência entre estas duas classes, apesar de não ser explícita.

Refinamentos: É um tipo de relacionamento entre duas descrições de uma mesma coisa, mas em níveis de abstração diferentes e podem ser usados para modelar diferentes implementações de uma mesma coisa.

5. Diagramas

Os diagramas utilizados pela UML são compostos de nove tipos. Todos os sistemas possuem uma estrutura estática e um comportamento dinâmico. A UML suporta modelos estáticos (estrutura estática), dinâmicos (comportamento dinâmico) e funcional.

5.1. Diagrama Use-Case

A modelagem de um diagrama use-case é uma técnica usada para descrever e definir os requisitos funcionais de um sistema. Eles são escritos em termos de atores externos, use-cases e o sistema modelado. Este diagrama apenas se resume a determinar que funções deverão ser suportadas pelo sistema modelado.

5.2. Diagrama de Classes

O diagrama de classes demonstra a estrutura estática das classes de um sistema onde estas representam as "coisas" que são gerenciadas pela aplicação modelada. Classes podem se relacionar com outras através de diversas maneiras: associação, dependência, especialização ou em pacotes. Todos estes relacionamentos são mostrados no diagrama de classes juntamente com suas estruturas internas, que são os atributos e operações.

5.3. Diagrama de Objetos

O diagrama de objetos é uma variação do diagrama de classes e utiliza quase a mesma notação. A diferença é que o diagrama de objetos mostra os objetos que foram instanciados das classes. O diagrama de objetos é como se fosse o perfil do sistema em um certo momento de execução. A mesma notação do diagrama de classes é utilizada com 2 excessões: os objetos são escritos com seus nome sublinhados e todas as instâncias num relacionamento são mostradas.

5.4. Diagrama de Estado

O diagrama de estado é tipicamente um complemento para a descrição das classes. Este diagrama mostra todos os estados possíveis que objetos de uma certa classe podem se encontrar e mostra também quais são os eventos do sistemas que provocam tais mudanças. Diagramas de estado capturam o ciclo de vida dos objetos, subsistemas e sistemas. Eles mostram os estados que um objeto pode possuir como os eventos (mensagens recebidas, timer, erros e etc.) afetam estes estados ao passar do tempo.

5.5. Diagrama de Sequência

Um diagrama de sequência mostra a colaboração dinâmica entre os vários objetos de um sistema. O mais importante aspecto deste diagrama é que a partir dele percebe-se a sequência de mensagens enviadas entre os objetos. Ele mostra a interação entre os objetos, alguma coisa que acontecerá em um específico da execução do sistema. Em particular, os objetos são representados pelas suas "linhas de vida" e interagem por troca de mensagens ao longo de um determinado período de tempo.

5.6. Diagrama de Colaboração

Um diagrama de colaboração mostra de maneira semelhante ao diagrama de sequência, a colaboração dinâmica entre objetos. No diagrama de colaboração, além de mostrar a troca de mensagens entre os objetos, percebe-se também os objetos com os seus relacionamentos.

Se a ênfase do diagrama for o decorrer do tempo, é melhor escolher o diagrama de sequência, mas se a ênfase for o contexto do sistema, é melhor dar a prioridade ao diagrama de colaboração.

5.7. Diagrama de Atividade

Diagramas de atividade capturam ações e seus resultados. Eles focam o trabalho executado na implementação de uma operação (método), e suas atividades numa instância de um objeto. O diagrama de atividade é uma variação do diagrama de estado e possui um propósito um pouco diferente do diagrama de estado, que é o de capturar ações.

5.8. Diagrama de Componente e de Execução

O diagrama de componente e o de execução são diagramas que mostram o sistema por um lado funcional, expondo as relações entre seus componentes e a organização de seus módulos durante a execução. Apresenta a estrutura dos códigos gerados. Normalmente usado em sistema hardware e software de computadores.

6. Uma Primeira Análise de Sistemas Propulsivos

Sistemas propulsivos espaciais são sistemas complexos, com grande número de componentes interagindo entre si.

Este diagrama serve para comunicação entre os diferentes agentes envolvidos e documentação de decisões tomadas e acordadas.

Os diagramas criam um vocabulário para a descrição dos componentes e uma sintaxe para a descrição das relações entre eles.

Esse diagrama foi uma primeira análise dos componentes do sistema e as interações entre estes componentes. Este diagrama gerado desta forma, contém os diversos caminhos para melhor visualização do programa em desenvolvimento, sendo que a informação está principalmente na topologia e não no tamanho ou na colocação dos símbolos.

Na Fig 1. observa-se que foi usado diversos tipos de notações genéricas da UML. Foram usados alguns elementos como pacotes, que podem ter visibilidade para indicar como outros pacotes terão acesso a seu conteúdo, como exemplo: visibilidade privada, visibilidade pública e etc. Foi também usado associações, que representam que duas classes possuem uma ligação entre elas. Outra ferramenta da UML que foi inserida no diagrama foi o relacionamento de agregação, que indica que uma das classes do relacionamento é uma parte, ou está contida em outra classe.

A generalização que é um relacionamento entre um elemento geral e um outro mais específico. O elemento mais específico possui todas as características do elemento geral e contem ainda particularidades, e o *relacionamento de dependência* que mostra quando uma classe recebe um objeto de outra classe como parâmetro, uma classe acessa o objeto global da outra, foram usados neste diagrama.

Podem ser construídos vários tipos de diagramas que sumarizam a informação derivada de diagramas e modelos mais fundamentais.

A Figura 1 mostra um diagrama das entidades relacionadas aos processos termodinâmicos no interior da câmara de combustão.

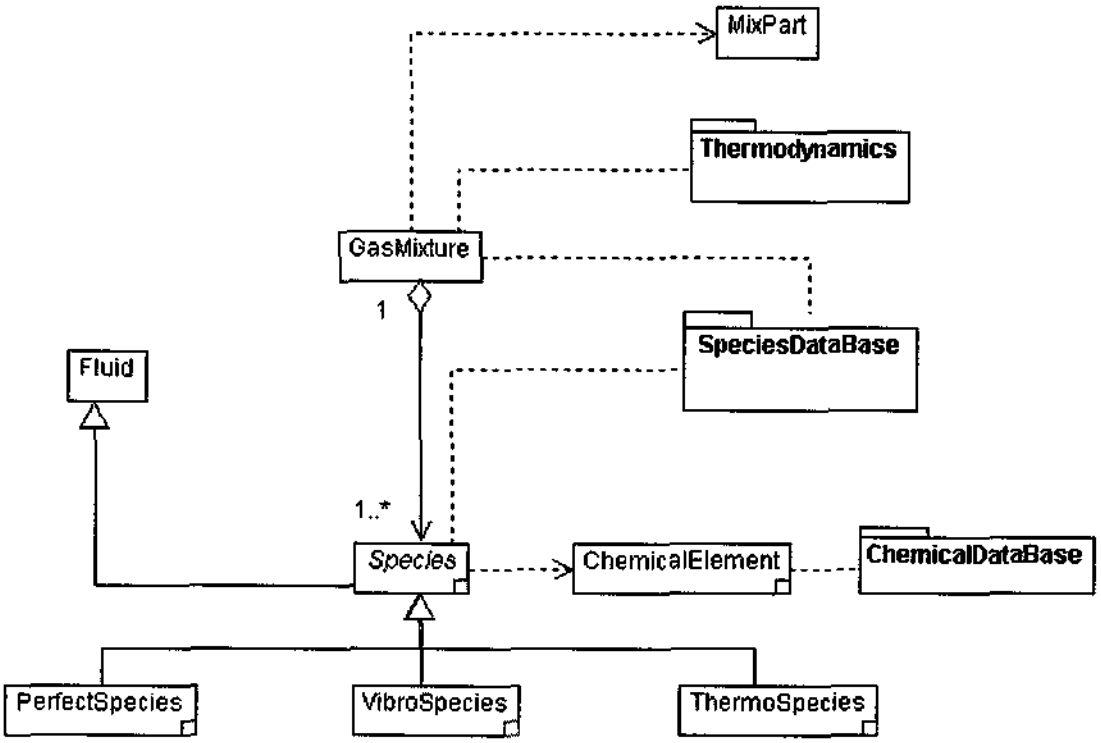


Figura 1. Diagrama de classes envolvidas nos processos termodinâmicos de uma câmara de empuxo.

As setas e caixas com diferentes geometrias, têm uma semântica bem definida representando as relações entres as entidades. A relação entre *GasMixture* e *Species* é uma relação de composição; uma mistura gasosa é composta de 1 ou mais espécies gasosas.

Uma espécie gasosa (*Species*) por sua vez pode apresentar um comportamento de gás perfeito (*PerfectSpecies*) ou de um gás com uma estrutura interna ativa, rotação, vibração e energia eletrônica (*VibroSpecies*, *ThermoSpecies*). Em algumas situações, o modelo de descrição de um gás perfeito é suficiente. Em condições de elevadas temperaturas e reações químicas é necessário levar em conta as trocas de energia associadas à estrutura interna das partículas que compõem a espécie química.

As entidades *ChemicalDataBase* e *ChemicalSpecies* contém dados para cada um dos elementos químicos e espécies químicas de interesse ao processo.

A Figura 3 mostra a classe representando a entidade *Species* dos diagramas acima.

Species (public Class/Interface)

Bases: public **Fluid**.
Submodel: Species
Comment: *This class describes the attributes and funcionalities of a gas species. It is an abstract class to be used as base class for the derivation of the specialized class for different gas models. All operations inside de class are made in the SI unit system. Variables are stored as doubles in the SI. Methods that return dimensional units do so using specialized classesl.*

Methods:
public **Name () : String**

public **Symbol () : String**

protected **Sigma () : double**

public abstract **Energy (mTemp: in double) : SpecificEnergy**

public abstract **Enthalpy (mTemp: in Temperature) : SpecificEnergy**

public abstract **Cv (mTemp: in Temperature) : SpecificHeat**

public abstract **Cp (mTemp: in Temperature) : SpecificHeat**

public **Entropy (mT: in Temperature, mP: in Pressure) : SpecificHeat**

public **Gibbs (mT: in Temperature, mP: in Pressure) : SpecificEnergy**

public abstract **Helmholtz (mT: in Temperature, mP: in Pressure) : SpecificEnergy**

public abstract **Mu (mT: in Temperature, mP: in Pressure) : MolSpecificEnergy**
Chemical Potential

public abstract **Gamma (mT: in Temperature) : double**

public abstract **SoundSpeed (mT: in Temperature) : Velocity**

public abstract **Prandtl (mT: in Temperature) : double**

public abstract **Viscosity (mT: in Temperature) : DynamicViscosity**

public abstract **Thermoconductivity (mT: in Temperature) : Thermoconductivity**

Get/Set:

Attributes:
protected **mR: double**
Gas Constant

protected **msUr: double = 8314**
Universal Gas Constant

protected **msNa: double = 6.23e24**
Avogadro´s Number

private **msK: double**
Boltzmann Constant

private **nG: geom**

Figura 3. Documentação da Classe *Species*

7. Conclusões e Perspectivas

A UML é a linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema e pode ser utilizada com todos os processos ao longo do ciclo de desenvolvimento e através de diferentes tecnologias de implementação.

A UML vai além de uma simples padronização em busca de uma notação unificada, uma vez que contém conceitos novos que não são encontrados em outros métodos orientados a objeto.

A utilização de um processo de desenvolvimento torna mais eficiente calcular o progresso do projeto, controlar e melhorar o trabalho. Um processo de desenvolvimento descreve "o que fazer", "como fazer", "quando fazer", e "porque deve ser feito". Quando são definidas e relacionadas as atividades de um processo, um objetivo específico é alcançado.

Foi desenvolvido também um banco de dados relativos a propriedades de propelentes líquidos usuais em propulsores de uso espacial e dados moleculares para obtenção de propriedades termodinâmicas de espécies gasosas presentes em produtos de combustão de propulsores a propelentes líquidos

Este banco de dados fornece as propriedades microscópicas das partículas (massa, momentos de inércia, energias de vibração, eletrônica, etc) utilizadas em métodos de termodinâmica estatística para a obtenção das propriedades termodinâmicas.

Estas propriedades termodinâmicas das espécies individuais são utilizadas na descrição de processos termodinâmicos no interior de câmaras de combustão.

A UML pode ser usada para mostrar as fronteiras de um sistema e suas funções principais utilizando atores e casos de uso, ilustrar a realização de casos de uso com diagramas de interação, representar uma estrutura estática de um sistema utilizando diagramas de classe, modelar o comportamento de objetos com diagramas de transição de estado, revelar a arquitetura de implementação física com diagramas de componente e de aplicação e estender sua funcionabilidade através de estereótipos.

Será esperado na próxima fase deste projeto, a elaboração dos esboços de interfaces, para configuração dos componentes do sistema propulsivo e implementação das interfaces em um sistema gráfico.

Essas interfaces gráficas, servirão para entradas de dados de componentes de sistemas propulsivos para um programa de modelamento e simulação.

Na fase de programação, as classes provenientes do design são convertidas para o código da linguagem orientada a objetos escolhida. No momento da criação de modelos de análise e design em UML, é melhor evitar traduzi-los mentalmente em código. A programação é uma fase separada e distinta, onde os modelos criados são convertidos em código.

Apêndice A

Espécies Atômicas

- Hidrogênio: H

PA = 1.00794		peso atômico
m = 0.167372.E-26	Kg	massa
Hf = 217999	J/g mol	entalpia de formação definida a 298.15 K
Hf = 216035	J/g mol	entalpia de formação definida a 0 K
Ne = 4		nº de níveis de energia

Energia (K)		Degenerescência	
e (0) = 0		g (0) = 2	
e (1) = 118351.5752		g (1) = 2	
e (2) = 118351.6256		g (2) = 2	
e (3) = 118352.1004		g (3) = 4	
e (4) = 140268.7639		g (4) = 2	

- Nitrogênio: H

PA = 14.0067			peso atômico
m = 2.325862.E-26	Kg		massa
Hf = 472.68	J/g mol		entalpia de formação definida a 298.15 K
Hf = 470.82	J/g mol		entalpia de formação definida a 0 K
Ne =			nº de níveis de energia

Energia (K)		Degenerescência	
e (0) = 0		g (0) = 4	
e (1) = 27683.228		g (1) = 6	
e (2) = 27695.774		g (2) = 4	
e (3) = 41528.044		g (3) = 2	
e (4) = 41528.600		g (4) = 4	

- Carbono: C

PA =12.011			peso atômico
m = 1.994469.E-26	Kg		massa
Hf = 716.67	J/g mol		entalpia de formação definida a 298.15 K
Hf = 711.19	J/g mol		entalpia de formação definida a 0 K
Ne =			nº de níveis de energia

Energia (K)		Degenerescência	
e (0) = 0		g (0) = 1	
e (1) = 23.616		g (1) = 3	
e (2) = 62.496		g (2) = 5	
e (3) = 14677.387		g (3) = 5	
e (4) = 31173.134		g (4) = 1	
e (5) = 48578.688		g (5) = 5	

- Argônio: Ar

PA = 39.948			peso atômico
m = 0.166053.E-26	Kg		massa
Hf = 0	J/g mol		entalpia de formação definida a 298.15 K
Hf = 0	J/g mol		entalpia de formação definida a 0 K
Ne =			nº de níveis de energia

- Oxigênio: O

PA = 15.9994			Peso atômico
m = 2.656764.E-26	Kg		Massa
Hf = 249.17	J/g mol		entalpia de formação definida a 298.15 K
Hf = 246.79	J/g mol		entalpia de formação definida a 0 K
Ne = 4			nº de níveis de energia

Energia (K)		Degenerescência	
e (0) = 0		g (0) = 5	
e (1) = 227.7067948		g (1) = 3	
e (2) = 326.6574986		g (2) = 1	
e (3) = 22830.18985		g (3) = 5	
e (4) = 48619.72491		g (4) = 1	

- Helio: He

PA = 4.00260			peso atômico
m = 0.162741.E-26	Kg		massa
Hf = 0	J/g mol		entalpia de formação definida a 298.15 K
Hf = 0	J/g mol		entalpia de formação definida a 0 K
Ne =			nº de níveis de energia

- Oxigênio: 0_2

PA = 31.9988			peso atômico
m = 5.313529.E-26	Kg		massa
Hf = 0	J/g mol		Entalpia de formação definida a 298.15 K
Hf = 0	J/g mol		Entalpia de formação definida a 0 K
f = 1/2			Fator de simetria c/ relação ao núcleo
Ne = 6			n° de níveis de energia

Estado	Energia (K)	Degenerescência (1)
$A^3\Sigma^-g$	e (0) = 0	g (0) = 3
$a^1\Delta g$	e (1) = 11340.93932	g (1) = 2
$b^1\Delta^+g$	e (2) = 18877.95798	g (2) = 1
$c^1\Sigma^-u$	e (3) = 46996.09841	g (3) = 1
$A'^3\Delta u$	e (4) = 49910.90077	g (4) = 6
$A^3\Sigma^+u$	e (5) = 50367.20640	g (5) = 3
$B^3\Sigma^-u$	e (6) = 71014.98204	g (6) = 3

ω_e (0) = 2273.53318	$\omega_e \chi_e$ (0) = 17.23761083
ω_e (1) = 2171.534233	$\omega_e \chi_e$ (1) = 18.56012165
ω_e (2) = 2061.275744	$\omega_e \chi_e$ (2) = 20.04723341
ω_e (3) = 1142.799924	$\omega_e \chi_e$ (3) = 18.32416351
ω_e (4) = 1222.953752	$\omega_e \chi_e$ (4) = 28.77538240
ω_e (5) = 1149.691628	$\omega_e \chi_e$ (5) = 17.49543250
ω_e (6) = 1020.170323	$\omega_e \chi_e$ (6) = 15.27121054

$\omega_e \rho_e$(0) = 6.80305184.E-2	$\omega_e Z_e$ (0) = - 1.83119052.E-2
$\omega_e \rho_e$(1) = 0	$\omega_e Z_e$ (1) = 0
$\omega_e \rho_e$(2) = - 2.0574398.E-2	$\omega_e Z_e$ (2) = 0
$\omega_e \rho_e$(3) = - 0.351653172	$\omega_e Z_e$ (3) = 7.91323.E-4
$\omega_e \rho_e$(4) = 0	$\omega_e Z_e$ (4) = 0
$\omega_e \rho_e$(5) = - 0.791323016	$\omega_e Z_e$ (5) = 0
$\omega_e \rho_e$(6) = - 8.5193023.E-2	$\omega_e Z_e$ (6) = - 3.4494481.E-2

B_e (0) = 2.079916293	α_x (0) = 2.2923447.E-2
B_e (1) = 2.052116396	α_x (1) = 2.4602951.E-2
B_e (2) = 2.014966802	α_x (2) = 2.6141432.E-2
B_e (3) = 1.317193129	α_x (3) = 2.0013278.E-2
B_e (4) = 1.381218355	α_x (4) = 3.7695750.E-2
B_e (5) = 1.310042447	α_x (5) = 2.0372970.E-2
B_e (6) = 1.178315940	α_x (6) = 1.7153724.E-2

α (0) = 9.217411061.E-5	α_z (0) = - 4.094964241.E-6
α (1) = 0	α_z (1) = 0
α (2) = - 6.178350845.E-5	α_z (2) = 0
α (3) = 0	α_z (3) = 0
α (4) = - 1.395606.E-3	α_z (4) = 0
α (5) = - 1.395606.E-3	α_z (5) = 0
α (6) = - 9.071036446.E-4	α_z (6) = 0

D_e (0) = 6.962203191.E-6		β_e (0) = - 1.1028250542.E-10	(7)
D_e (1) = 7.15068193.E-6		β_e (1) = - 1.4967604183.E-8	(7)
D_e (2) = 7.706046764.E-6		β_e (2) = 0.110785222.E-6	(7)
D_e (3) = 15.1070745.E-6		β_e (3) = - 6.6411840229.E-7	(7)
D_e (4) = 7.047367945.E-6		β_e (4) = - 1.712386462.E-7	(7)
D_e (5) = 6.89170351.E-6		β_e (5) = - 0.431630736.E-6	(7)
D_e (6) = 6.54639895.E-6		β_e (6) = -2.5740294223.E-7	(7)

nº Quântico Vibracional Máximo Antes da Dissociação	
ν (0) = 36	(3)
ν (1) = 30	(3)
ν (2) = 27	(3)
ν (3) = 15	(5)
ν (4) = 15	(5)
ν (5) = 11	(3)
ν (6) = 21	(3)

- Nitrogênio: N_2

PA = 28.0134			peso atômico
m = 4.651737.E-26	Kg		massa
Hf = 0	J/g mol		entalpia de formação definida a 298.15 K
Hf = 0	J/g mol		entalpia de formação definida a 0 K
f = 1/2			fator de simetria c/ relação ao núcleo
Ne = 3			nº de níveis de energia

Estado	Energia (K)	Degenerescência
$X^1\Sigma^+g$	e (0) = 0	g (0) = 1
$A^3\Sigma^+u$	e (1) = 71585.64104	g (1) = 3
$B^3\Pi g$	e (2) = 85328.80683	g (2) = 6
$W^3\Delta g$	e (3) = 85434.11035	g (3) = 6

- Cianogen: CN

PA = 26.0177			peso atômico
m = 4.320331.E-26	Kg		massa
Hf = 435.1	J/g mol		entalpia de formação definida a 298.15 K
Hf = 436.8	J/g mol		entalpia de formação definida a 0 K
f = 1			Fator de simetria c/ relação ao núcleo
Ne =			nº de níveis de energia

Energia (K)	Degenerescência
e (0) = 0	g (0) = 2
e (1) = 13129.92	g (1) = 4
e (2) = 37149.12	g (2) = 4
e (3) = 77695.2	g (3) = 4

ω_e (0) = 2978.7984	$\omega_e \chi_e$ (0) = 18.88416

B_e (0) = 2.734416	α_e (0) = 0.024768

- Monóxido de Carbono: CO

PA = 28.0104			peso atômico
m = 4.651239.E-26	Kg		massa
Hf = -11053	J/g mol		entalpia de formação definida a 298.15 K
Hf = -113.81	J/g mol		entalpia de formação definida a 0 K
f = 1			fator de simetria c/ relação ao núcleo
Ne = 4			nº de níveis de energia

Estado	Energia (K)	Degenerescência
$X^1\Sigma^+$	e (0) = 0	g (0) = 2
$A^3\Pi r$	e (1) = 70048.92051	g (1) = 6
$a'^3\Sigma^+$	e (2) = 80319.99112	g (2) = 3
$d^3\Delta i$	e (3) = 87937.71249	g (3) = 6
$e^3\Sigma^-$	e (4) = 92412.48588	g (4) = 3

ω_e (0) = 3121.860775	$\omega_e \chi_e$ (0) = 19.11881008
ω_e (1) = 2508.364471	$\omega_e \chi_e$ (1) = 20.66072456
ω_e (2) = 1767.671741	$\omega_e \chi_e$ (2) = 15.06103515
ω_e (3) = 1686.151082	$\omega_e \chi_e$ (3) = 15.30130959
ω_e (4) = 1608.141021	$\omega_e \chi_e$ (4) = 15.37468682

$\omega_e \rho_e(0) = 0$	$\omega_e Z_e(0) = 0$
$\omega_e \rho_e(1) = 0$	$\omega_e Z_e(1) = 0$
$\omega_e \rho_e(2) = 0$	$\omega_e Z_e(2) = 0$
$\omega_e \rho_e(3) = 0$	$\omega_e Z_e(3) = 0$
$\omega_e \rho_e(4) = 0$	$\omega_e Z_e(4) = 0$

$B_e(0) = 2.778667278$	$\alpha_x(0) = 0.025184804$
$B_e(1) = 2.433303887$	$\alpha_x(1) = 0.027394164$
$B_e(2) = 1.934568959$	$\alpha_x(2) = 0.027221511$
$B_e(3) = 1.885938562$	$\alpha_x(3) = 0.025638865$
$B_e(4) = 1.846804042$	$\alpha_x(4) = 0.025221622$

$\alpha(0) = 0$	$\alpha_z(0) = 0$
$\alpha(1) = 0$	$\alpha_z(1) = 0$
$\alpha(2) = 0$	$\alpha_z(2) = 0$
$\alpha(3) = 0$	$\alpha_z(3) = 0$
$\alpha(4) = 0$	$\alpha(4) = 0$

$D_e(0) = 8.807382005.E\text{-}6$	(2)	$\beta_e(0) = 1.5009154534.E\text{-}9$	(4)
$D_e(1) = 9.150571603.E\text{-}6$	(2)	$\beta_e(1) = -3.8065816568.E\text{-}8$	(4)
$D_e(2) = 9.222510059.E\text{-}6$	(2)	$\beta_e(2) = 8.9748519542.E\text{-}8$	(4)
$D_e(3) = 9.481488501.E\text{-}6$	(2)	$\beta_e(3) = 2.143880386.E\text{-}8$	(4)
$D_e(4) = 9.740466942.E\text{-}6$	(2)	$\beta_e(4) = -1.3955718041.E\text{-}8$	(4)

nº Quântico Vibracional Máximo Antes da Dissociação	
$\nu(0) = 8$	(5)
$\nu(1) = 8$	(5)
$\nu(2) = 7$	(5)
$\nu(3) = 7$	(5)
$\nu(4) = 6$	(5)

- Óxido de Nitrogênio: NO

PA = 30.0061			peso atômico
m = 4.982633.E-26	Kg		massa
Hf = 9.0291	J/g mol		entalpia de formação definida a 298.15 K
Hf = 8.9788	J/g mol		entalpia de formação definida a 0 K
f = 1			fator de simetria c/ relação ao núcleo
Ne = 4			nº de níveis de energia

Estado	Energia (K)	Degenerescência
$X^2\Pi$ r	e (0) = 0	g (0) = 4
$A^2\Sigma^+$	e (1) = 63256.4915	g (1) = 2
$B^2\Pi$ r	e (2) = 66100.79417	g (2) = 4
$C^2\Pi$ r	e (3) = 74997.27912	g (3) = 4
$D^2\Sigma^+$	e (4) = 76376.6271	g (4) = 2

ω_e (0) = 2739.709913	$\omega_e \chi_e$ (0) = 20.25067536
ω_e (1) = 3416.083909	$\omega_e \chi_e$ (1) = 23.17281545
ω_e (2) = 1496.032131	$\omega_e \chi_e$ (2) = 11.97055908
ω_e (3) = 3445.852042	$\omega_e \chi_e$ (3) = 21.5815368
ω_e (4) = 3343.555558	$\omega_e \chi_e$ (4) = 32.92623131

$\omega_e \rho_e$(0) = 0	$\omega_e Z_e$ (0) = 0
$\omega_e \rho_e$(1) = 0	$\omega_e Z_e$ (1) = 0
$\omega_e \rho_e$(2) = 0	$\omega_e Z_e$ (2) = 0
$\omega_e \rho_e$(3) = 0	$\omega_e Z_e$ (3) = 0
$\omega_e \rho_e$(4) = 0	$\omega_e Z_e$ (4) = 0

B_e (0) = 2.40555003	α_x (0) = 0.024602951
B_e (1) = 2.872502548	α_x (1) = 0.027552428
B_e (2) = 1.657462026	α_x (2) = 0.017265229
B_e (3) = 2.87753824	α_x (3) = 0.043163073
B_e (4) = 2.88127904	α_x (4) = 0.031293228

α (0) = 0	α_z (0) = 0
α (1) = 0	α_z (1) = 0
α (2) = 0	α_z (2) = 0
α (3) = 0	α_z (3) = 0
α (4) = 0	α (4) = 0

D_e (0) = 0.776935324.E-6	(2)	β_e (0) = - 2.3545113338..E-9	(4)
D_e (1) = 7.769353248.E-6	(2)	β_e (1) = - 1.3594610066.E-8	(4)
D_e (2) = 7.049968688.E-6	(2)	β_e (2) = - 5.5329522564.E-8	(4)
D_e (3) = 8.0265709879.E-6	(2)	β_e (3) = 2.8993721796.E-7	(4)
D_e (4) = 8.34460896.E-6	(2)	β_e (4) = - 1.5666180427.E-7	(4)

nº Quântico Vibracional Máximo Antes da Dissociação	
ν (0) = 23	(2)
ν (1) = 23	(2)
ν (2) = 3	(3)
ν (3) = 29	(3)
ν (4) = 27	(5)

Bolsa de Iniciação Científica - PIBIC

Bolsista: Marcelo Furlan Salles

Coordenador: José Nivaldo Hinkel

Título: **Modelamento do Sistema Propulsivo do Foguete.**

04 de Fevereiro de 2002 - Relatório Parcial

No primeiro mês, fui colocado numa sala com um computador a minha inteira disposição. Em seguida, foi feita a instalação do programa a ser usado no desenvolver do projeto. Foi instalado um software de programação do tipo C ++ Builder. Em decorrer da bolsa, foi estudado esse programa através de livros e apostilas pego na Biblioteca do INPE, Internet e através do meu coordenador Nivaldo. Durante meses nesses estudos, fui aperfeiçoando meus conhecimentos nessa linguagem e desenvolvendo pequenos programas testes.

Adquiri uma apostila via internet da Universidade de Caxias do Sul de programação em C ++, contendo conceitos de programação orientada a objetos, classes, abstrações, herança, operadores e etc. Com exemplos e exercícios para treinamento.

Uma apostila da Universidade Federal de Minas Gerais – UFMG, contendo os conceitos básicos da linguagem de programação C. Introduzindo as funções, entrada e saída, comandos de retorno (if, for), declarações de variáveis, operadores aritméticos e de atribuição.

Foi também requerido no plano do projeto, um software de modelamento. Foi adotado então um software chamado UML – Linguagem Unificada de Modelamento.

Nesse período, fui à procura de bibliografias sobre esse software, apostilas, informações complementares. Foi meio demorado a encontrar um livro

específico sobre esse assunto, mas adquiri o livro para conhecimento e estudo do software em questão.

Livro: Modelamento de Objetos através da UML, análise e desenho orientados a objeto, editora Makron books, autor José Davi Furlan. Adquirido pela biblioteca do INPE – CPTEC. Contendo nesse livro diversas informações sobre diagramas de classe (objeto, atributo, associação, herança, dependência), diagrama de "use case", diagrama de estado e etc.

Estou no aguardo da instalação de um novo software em meu PC, chamado Linux, começa agora uma nova fase de estudos sobre essa nova linguagem, tomando o mesmo procedimento até agora adotado.

Marcelo F. Salles
Bolsista

Nivaldo Hinkel
Coordenador