



CENTRO UNIVERSITÁRIO IBMEC/RJ
CURSO DE ENGENHARIA DA COMPUTAÇÃO

CLAYTON JONES ALVES DA SILVA

Notas de aula - Desafios de Engenharia
Capítulo: Modelagem de sistemas de engenharia

Rio de Janeiro

2021.1

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação de classe com atributos	11
Figura 2 – Representação de instância de objeto	11
Figura 3 - Representação de classe com atributos e operações.....	12
Figura 4 – Associação de classes, com nome de associação, multiplicidade e papéis.....	12
Figura 5 – Ligações de instâncias de objetos.....	13
Figura 6 – Diagrama de classes com agregação	13
Figura 7 - Diagrama de classes com generalização.....	14
Figura 8 – ICOM e elementos	16
Figura 9 – Decomposição das funções de um sistema de engenharia	17
Figura 10 – Exemplo: Numeração, relação entre funções e ramificação/unificação de setas no IDEF-0	18
Figura 11 – Nó A0 – Coletar e tratar esgotos e águas pluviais	19
Figura 12 – Decomposição hierárquica dos diagramas IDEF-0 na construção do modelo IDEF-0	20
Figura 13 – Diagrama de estados, na notação UML	23

SUMÁRIO

1. INTRODUÇÃO	4
2. Modelagem	4
2.1 Abstração	4
2.2 Tipos de modelos	5
3. Modelo de objetos	7
3.1 Objetos e Classes	7
3.2 Atributos e operações	8
3.3 Ligações e associações	9
3.4 Agregação	9
3.5 Generalização	10
3.6 Agrupamento	10
3.7 Notação	11
4. Modelagem funcional IDEF-0	14
4.1 Visão geral	14
4.2 ICOM	16
4.3 Decomposição hierárquica	17
5. Modelagem dinâmica: diagramas de estados segundo a UML	21
5.1 Diagramas de interações	21
5.2 Diagramas de estados	22
REFERÊNCIAS	27

1. INTRODUÇÃO

A presente nota de aula objetiva apresentar ao aluno do Ibmec do curso de Engenharia os conceitos principais relativos ao conteúdo da disciplina Desafios de Engenharia.

O texto não tem a pretensão de esgotar os tópicos, o que requer a consulta aos livros e textos recomendados no programa da disciplina.

2. Modelagem

2.1 Abstração

Os engenheiros utilizam muitos tipos de modelos para vários propósitos, antes de construir um sistema ou produto de engenharia. Podemos incluir modelos arquiteturais para mostrar aos clientes; modelos de aviões em escala para testes em túneis de ventos; desenhos técnicos de peças e máquinas...

Os modelos servem para diversos objetivos:

- Testar uma entidade física antes de lhe dar forma;
- Comunicar com o cliente;
- Visualizar o produto.

O principal motivo da modelagem, que possivelmente incorpora todos os motivos apontados anteriormente, é reduzir a complexidade dos sistemas para conseguir tratá-los. Os modelos reduzem a complexidade dividindo-a em um número pequeno de aspectos, segmentados, que podem ser tratados por vez.

Para poder simplificar o sistema a ser tratado, seja na concepção seja na análise de um sistema de engenharia, utiliza-se o conceito de abstração. Abstração é o exame seletivo de determinados aspectos de um problema. O objetivo da abstração é isolar os aspectos que sejam importantes para algum propósito e suprimir os que não o forem. A abstração deve sempre visar a um propósito, porque este determina o que é e o que não é importante.

Todas as abstrações são incompletas e inexatas. Consequentemente, na construção do modelo não se deve procurar a verdade absoluta, mas a adequação a algum propósito. Não há um único modelo 'correto' de uma situação, apenas modelos adequados e inadequados.

2.2 Tipos de modelos

A base do que discutiremos aqui está na Técnica de Modelagem de Objetos (TMO), que combina três visões da modelagem de sistemas: a visão dos aspectos estáticos e estruturais de dados de um sistema; a visão dinâmica, que representa os aspectos temporais e comportamentais de um sistema; e a visão funcional, que representa os aspectos relativos às transformações de funções de um sistema.

Os três tipos de modelos separam um sistema em visões ortogonais que podem ser representadas e manipuladas, com uma notação uniforme. Esses modelos não são completamente independentes, mas cada modelo pode ser em boa medida examinado e entendido pelo que representa. As interconexões entre os diferentes modelos são limitadas e explícitas. Um projeto bem feito consegue isolar os diferentes aspectos de um sistema, limitando o acoplamento entre as suas diferentes visões.

Modelo de objetos

O modelo de objetos descreve a estrutura de objetos de um sistema – sua identidade, seus relacionamentos com outros objetos, seus atributos e suas operações. O modelo de objetos proporciona a estrutura necessária na qual podem ser colocados os modelos dinâmico e funcional. Modificações e transformações não fazem sentido a menos que haja alguma coisa a ser modificada ou transformada.

Os objetos dos sistemas são os seus elementos construtivos, ou seja, as unidades ou componentes em que dividimos o sistema, desde o mais elevado nível até o mais elementar. Saliente-se que são objetos dos sistemas os

componentes utilizados para assegurar a integração entre eles, ou seja, as interfaces utilizadas.

O modelo de objetos deve incorporar os termos próprios ao contexto em que será utilizado. O modelo de engenharia deve conter termos compatíveis com o problema de engenharia que se deseja solucionar.

O modelo de objetos é representado graficamente por diagramas de objetos contendo classes de objetos. As classes são organizadas em níveis hierárquicos compartilhando estruturas e comportamentos comuns e são associadas a outras classes. As classes definem os valores de atributos relativos a cada instância de objetos e as operações que cada objeto executa ou a que se submete.

O modelo dinâmico descreve os aspectos de um sistema relacionados ao tempo e à sequência de operações – eventos que assinalam modificações, sequências de eventos, estados que definem o contexto para os eventos e a organização de eventos e estados. O modelo dinâmico incorpora o controle, que é um aspecto de um sistema que descreve as sequências de operações que ocorrem, independentemente do que as operações fazem, sobre o que elas atuam ou como são implementadas.

O modelo dinâmico é representado graficamente por diagramas de estados. Cada um desses diagramas mostra a sequência de estados e de eventos permitidos em um sistema para uma classe de objetos. Os diagramas de estados também se relacionam com os outros modelos. As ações nos diagramas de estados correspondem a funções do modelo funcional; os eventos de um diagrama de estados tornam-se operações em objetos no modelo de objetos.

O modelo funcional descreve os aspectos de um sistema relacionados a transformações de valores: funções, mapeamentos, restrições e dependências funcionais. O modelo funcional abrange o que um sistema faz, independentemente de como ou quando é feito.

O modelo funcional é representado por meio de diagramas de fluxos de dados. Esses diagramas mostram as dependências entre valores e o processamento de valores de saída a partir dos valores de entrada e das funções, independente de quando e se as funções são executadas.

Cada modelo descreve um aspecto do sistema, mas contém referências a outros modelos. O modelo de objetos descreve a estrutura de dados sobre a qual atuam os modelos dinâmico e funcional. O modelo dinâmico descreve a estrutura de controle dos objetos. Mostra também as decisões que dependem dos valores dos objetos e que provocam ações que os modificam, bem como que chamam as funções. O modelo funcional descreve as funções chamadas pelas operações do modelo de objetos e pelas ações do modelo dinâmico. As funções operam sobre os valores de dados especificados pelo modelo de objetos. Mostra também as restrições relativas aos valores dos objetos.

Existem algumas ambiguidades. É natural, pois a abstração é apenas a abstração que reproduz a captura de uma perspectiva do sistema. Nesse sentido, é imperfeita e possui uma medida de imprecisão.

3. Modelo de objetos

3.1 Objetos e Classes

Um objeto é um conceito, uma abstração, algo com limites nítidos e significativo em relação a um problema em pauta. Todos os objetos possuem identidades e são distinguíveis. Para assegurar precisão quando se trata de objeto utiliza-se a expressão instância do objeto. Usa-se o termo classe de objetos para se referir a grupos de coisas semelhantes.

Uma classe de objetos descreve um grupo de objetos:

- com propriedades semelhantes (atributos);
- o mesmo comportamento (operações);
- os mesmos relacionamentos com outros objetos;
- a mesma semântica.

Os objetos de uma classe possuem os mesmos atributos e padrões comportamentais. A maioria dos objetos deriva sua individualidade das diferenças entre os valores de seus atributos e relacionamentos com outros objetos. Entretanto, é possível a existência de objetos com idênticos valores de atributos e relacionamentos.

Os objetos de uma classe compartilham um objetivo semântico comum, além dos requisitos de atributos e comportamentos em comum. A interpretação de semântica depende do propósito de cada aplicação, sendo uma questão de critério.

O benefício de uso de classes está na abstração do problema. As operações podem ser desenhadas uma vez somente para uma classe, de tal forma que os objetos podem se beneficiar e reutilizá-las.

Diagramas de objetos

Os diagramas de objetos oferecem uma notação gráfica formal para a modelagem de objetos e seu inter-relacionamentos. Existem dois tipos de diagramas de objetos:

- Diagramas de classes – esquema, padrão ou modelo que descreve muitas instâncias possíveis de objetos. Representam o caso mais geral de modelagem de um sistema.
- Diagrama de instâncias de objetos – esquema, padrão ou modelo que descreve as instâncias de objetos e os seus relacionamentos.

3.2 Atributos e operações

Um atributo é um valor de um dado do objeto de uma classe. Cada atributo possui um valor para cada instância de objeto. Os atributos são valores puros, isto é, são dados que não possuem identidade. Nos diagramas cada atributo é definido por um nome de atributo e pode incluir detalhes específicos.

Uma operação é uma função de um objeto ou uma função que lhe é aplicada. Cada operação possui um objeto-alvo. Nos modelos de engenharia, uma operação normalmente possui argumentos, que servem como parâmetros.

3.3 Ligações e associações

Ligações e associações são os meios para estabelecimento de relacionamentos entre objetos e classes.

Uma ligação é uma conexão física ou conceitual entre instâncias de objetos. As ligações mostram o relacionamento entre duas ou mais instâncias de objetos.

Uma associação descreve um grupo de ligações com estrutura e semântica comuns. Tanto as ligações quanto as associações são definidas por um nome. As associações são intrinsecamente bidirecionais. Todas as conexões entre classes devem ser modeladas como associações. Nas associações são estabelecidos papéis entre as classes de objetos. Um papel representa qual é o comportamento da classe em uma associação.

No diagrama várias instâncias de uma classe podem relacionar-se com cada instância de outra classe. A essa característica dá-se o nome de multiplicidade. A multiplicidade especifica quantas instâncias de objetos e classes relacionam-se entre si. O diagrama de instâncias mostra um possível conjunto de linhas. Podem ser “exatamente uma”, “mais de uma”, “n” – n um número especificado ou, genericamente, “muitos”.

Em alguns sistemas, três classes podem relacionar-se por uma associação. Nesse caso, configura-se uma associação ternária. Similarmente as instâncias podem se relacionar em uma ligação ternária.

3.4 Agregação

Do ponto de vista de sistemas de engenharia, a agregação é um relacionamento fundamental. A agregação é o relacionamento “parte-todo” ou “uma-parte-de” no qual os objetos que representam os componentes de alguma coisa são associados a um objeto que representa a estrutura inteira.

A agregação é uma forma estreitamente acoplada de associação com algumas questões semânticas a mais. Algumas propriedades de agregação:

- Transitividade: Se A faz parte de B e B faz parte de C, então A faz parte de C;
- Antissimétrica: Se A faz parte de B, então B é todo de A;
- Propagação de propriedade: Se A faz parte de B, então algumas propriedades de B propagam-se para A.

Um relacionamento de agregação é definido como o relacionamento de uma classe estrutural a uma classe de um componente. A existência de um objeto componente pode depender da existência de um objeto agregado do qual faz parte.

Outro aspecto importante a salientar é que a agregação pode ter um número arbitrário de níveis. Assim, a árvore de agregação pode simplificar a utilização de muitas linhas interligando componentes de uma estrutura.

3.5 Generalização

A generalização é o relacionamento entre uma classe em uma ou mais versões detalhadas dela. A classe que é detalhada é chamada de superclasse e a versão que detalha, subclasse. As subclasses representam particularidades diferentes da superclasse. As subclasses diferenciam-se entre si pelos detalhes próprios que estabelecem.

As subclasses herdam as características das suas respectivas superclasses.

3.6 Agrupamento

Duas formas de agrupar classes e associações podem ser utilizadas: módulo ou folha.

Um módulo é uma construção lógica para agrupar classes e associações que incorpora uma perspectiva de uma situação. Por exemplo, os módulos de eletricidade, de encanamento e de ventilação são diferentes visões de um edifício.

Um modelo de objetos pode consistir em um ou mais módulos. Permite que se particione um modelo de objetos em partes possíveis de manipulação. Proporcionam uma unidade intermediária de empacotamento entre um modelo de objetos completo e os blocos básicos de construção de classes e associações. Nomes de classes e de associações devem ser únicos dentro de um módulo. Sempre que possível deve-se usar nomes de classes consistentes de classes e associações através dos módulos.

A folha não é uma construção lógica. É simplesmente uma facilidade de notação.

3.7 Notação

A Figura 1 apresenta uma notação utilizada para representar classe e seus atributos. A classe é representada por um bloco retangular. No cabeçalho é apresentado o nome da classe *Equipamento*. No corpo do bloco são apresentados os atributos *nome* e *peso*, os quais qualificam a classe. Os atributos podem contemplar ainda detalhes opcionais, que são apresentados após o nome do atributo separados por ‘:’. No caso, o nome atende a um padrão ISO e o peso deve ser apresentado como um número real.

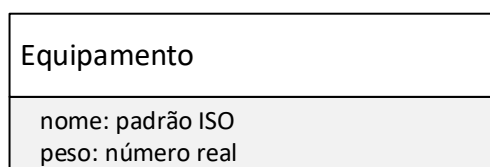


Figura 1 – Representação de classe com atributos

A Figura 2 apresenta uma notação utilizada para representar uma instância de objeto. No cabeçalho é apresentado o nome da instância do objeto *Equipamento*.

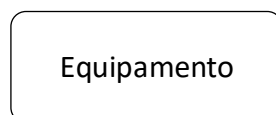


Figura 2 – Representação de instância de objeto

A Figura 3 apresenta a representação da classe Equipamento, incluindo, além dos seus atributos, na parte inferior do corpo do bloco, separada por linha tracejada, uma função desempenhada pela classe de *Processar produto químico*.

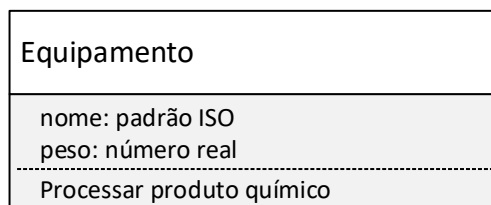


Figura 3 - Representação de classe com atributos e operações

A Figura 4 apresenta o diagrama de classes, contendo as classes *Equipamento* e *Caldeira*, com seus respectivos atributos e funções. Evidencia-se uma associação entre as classes. Está caracterizada uma associação porque as classes podem ser instanciadas em diferentes objetos, portanto a associação se refere a um grupo de ligações. A associação indica que a classe *Equipamento* *Alimenta* a classe *Caldeira*. Observe que o diagrama apresenta a multiplicidade das classes, indicando que existem *M1* instâncias da classe *Equipamento* que se relacionam com *M2* instâncias da classe *Caldeira*. A multiplicidade é apresentada nas extremidades, na parte superior linha da associação. O diagrama evidencia também o papel de cada classe. No caso da classe *Equipamento*, tem o papel de *fonte*; no caso da *Caldeira*, tem o papel de *destino*. Os papéis também são apresentados nas extremidades da associação, na parte inferior da linha.

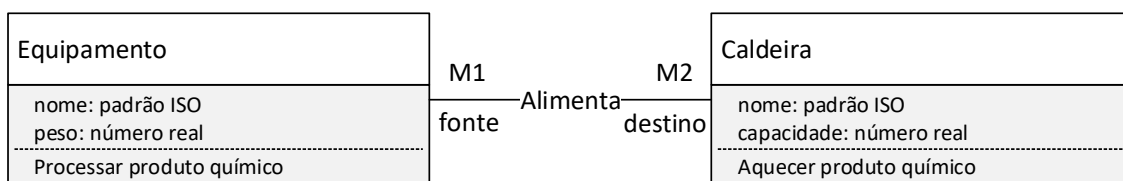


Figura 4 – Associação de classes, com nome de associação, multiplicidade e papéis

A Figura 5 apresenta o diagrama de instâncias, evidenciando as ligações entre as instâncias de *Equipamento* e *Caldeira*. Como no diagrama de classes,

apresenta a instância do objeto *Equipamento* que *Alimenta* a instância do objeto *Caldeira*.

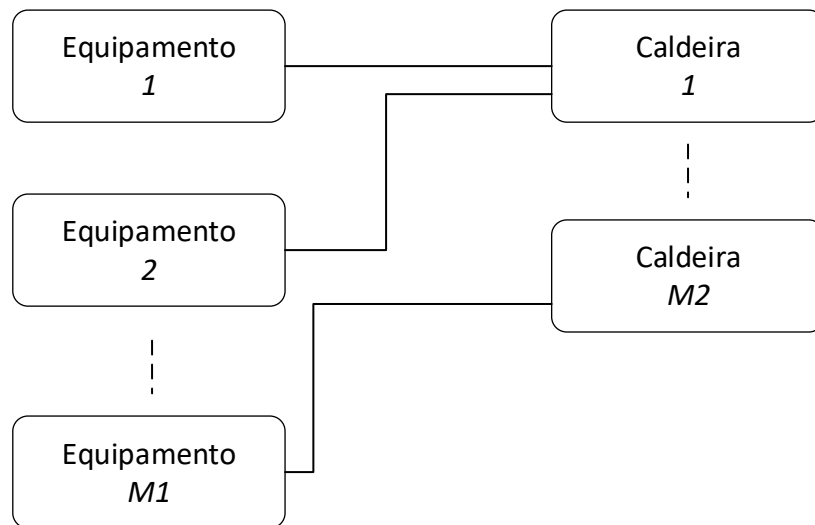


Figura 5 – Ligações de instâncias de objetos

A Figura 6 apresenta o diagrama de classes evidenciando a agregação. Por uma questão de simplificação, foram subtraídas as informações já comentadas. Observe que a classe *Computador* constitui das classes componentes *Monitor*, *Teclado*, *Torre* e *Mouse*. A classe *Torre* se decompõe nas classes *Placa-mãe*, *HD* e *Memória*. A figura apresenta tipicamente uma árvore de agregação.

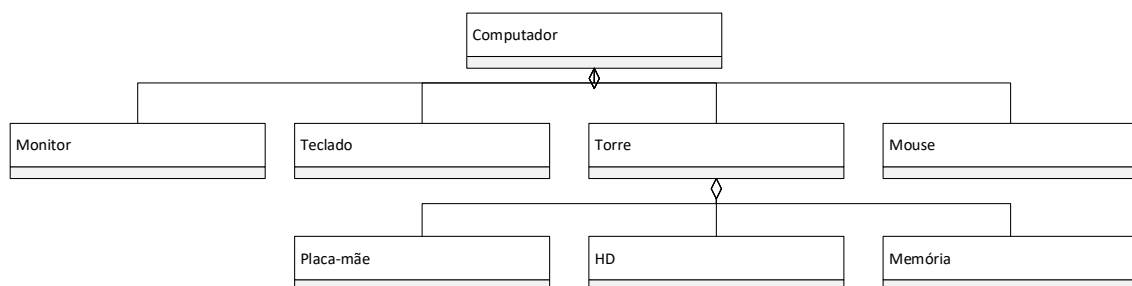


Figura 6 – Diagrama de classes com agregação

A Figura 7 apresenta o diagrama de classes evidenciando a generalização. Por uma questão de simplificação, foram subtraídas as informações já comentadas. Observe que a superclasse *Equipamento* pode ser detalhada nas subclasses *Bomba*, *Aparelho de trocar calor* e *Tanque*. Como comentado anteriormente, as subclasses herdam atributos da superclasse. Por

exemplo, admitindo que a classe *Equipamento* tem os atributos de *nome*, *fabricante*, *peso* e *preço*. A classe *Tanque* também deverá possuir os mesmos atributos, incluindo os seus próprios atributos, como *volume* e *pressão*.

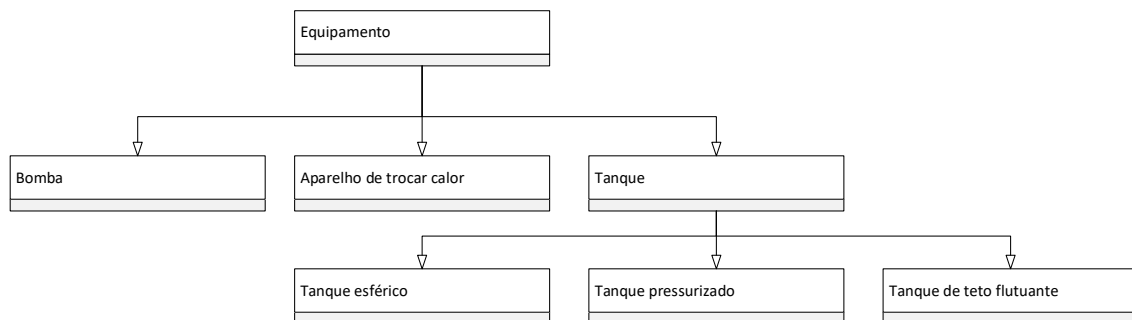


Figura 7 - Diagrama de classes com generalização

4. Modelagem funcional IDEF-0

4.1 Visão geral

Uma família de linguagens de modelagem de sistema muito utilizada pela engenharia é intitulada IDEF. Foi criada pelo programa de fabricação integrada assistida por computador (do inglês, *Integrated Computer Aided Manufacturing* – *ICAM*) da Força Aérea Americana, cuja abreviatura decorre de *ICAM DEFINition* (MENZEL C., 1998) e que, atualmente, considerando as evoluções por que passou a família, passou a significar *Integrated DEFINition*.

A família IDEF contempla, entre outras, atualmente, as seguintes linguagens:

- **IDEF-0 – Contempla a modelagem funcional dos sistemas;**
- IDEF-1 – Contempla a modelagem conceitual dos sistemas;
- IDEF-2 – Contempla a modelagem de simulação;
- IDEF-3 – Contempla a modelagem dos processos;
- IDEF-4 – Contempla a modelagem orientada a objetos;
- IDEF-5 – Contempla a modelagem de ontologias, ou seja, o modelo de dados que representa os conceitos e os seus relacionamentos, dentro de um domínio.

Outras linguagens pertencem à família IDEF, cada uma proposta para a captura de um conceito ou propriedade dos sistemas de engenharia.

O NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY – NIST (1993) recomenda o uso da modelagem IDEF-0 para projetos que exigem uma técnica de modelagem para análise, desenvolvimento, reengenharia, integração ou aquisição de Sistemas de Informação; para incorporar técnicas de modelagem a sistemas ou empreendimentos na análise de processos de negócios ou metodologia de Engenharia de Software. Não obstante a recomendação do padrão do NIST para aplicação em Sistemas de Informação, a modelagem IDEF-0 é aplicável a análise e projetos de Sistemas de Engenharia indistintamente.

Convém relembrar a definição funcional dos sistemas de engenharia como um passo na aplicação do método da Engenharia de Sistemas. A função representa a tarefa a executar por um sistema de engenharia e por seus componentes, genericamente tratados como mecanismo. Para executar uma função, o mecanismo requer um conjunto de entradas que são transformadas para gerar saídas. Tanto entradas quanto saídas podem ser informações, bens ou serviços. A transformação ocorre em um domínio que envolve, normas, leis e outras questões regulatórias, padrões e outras questões que orientam ou restringem a atividade do sistema ou componente.

A modelagem IDEF-0 apresenta as seguintes características (NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY - NIST, 1993):

- É capaz de representar graficamente ampla variedade de negócios, sistemas e quaisquer outros tipos de empreendimento, em qualquer nível de detalhe;
- É uma linguagem simples e coerente, assegurando expressão precisa e rigorosa e promovendo consistência para uso e implementação;
- Ele amplia a comunicação entre analistas de sistemas, desenvolvedores e usuários, por meio da facilidade de aprendizagem e ênfase na apresentação hierárquica de detalhes do sistema;

- Já foi testada e aprovada pela indústria e Forças Armadas;
- Pode ser gerada por várias ferramentas computacionais comerciais.

4.2 ICOM

Os diagramas IDEF-0 utilizam esses conceitos inerentes às atividades ou funções desempenhadas pelos sistemas e seus componentes. Utilizam como entidade básica o elemento referido pelo acrônimo ICOM (do inglês, *Input-Control-Output-Mechanism*). Graficamente, os ICOMs são representados por caixa (boxes) e setas. Cada caixa representa uma função, descrita sinteticamente como rótulo (*label*) no interior. Cada seta possui, evidentemente, um sentido (origem-destino) e representa o encaminhamento de uma informação, serviço ou bem, seja de entrada, saída, controle ou mecanismo. Cada seta possui um rótulo que descreve sinteticamente o tipo de informação, serviço ou bem. As setas de entrada, controle e mecanismo sempre possuem como destino a caixa do ICOM. As setas de saída sempre possuem como origem a caixa do ICOM. O mecanismo é o sistema ou componente que executa a função. A Figura 8 apresenta a representação de ICOMs.

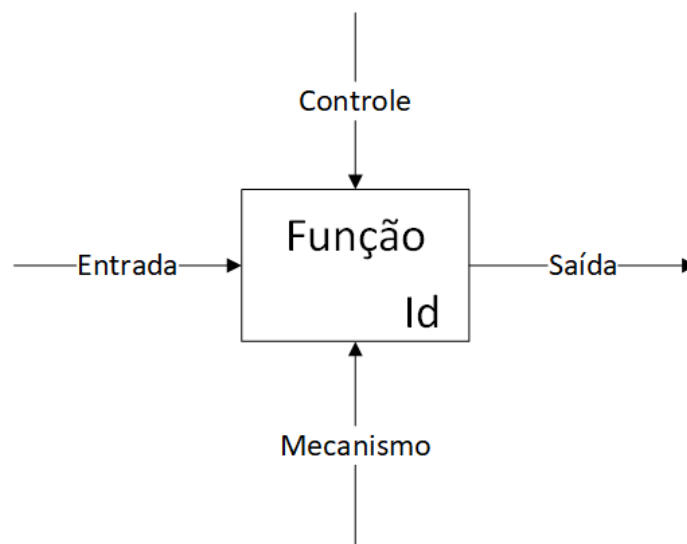


Figura 8 – ICOM e elementos

Fonte: o Autor

4.3 Decomposição hierárquica

Além da utilização de ICOMs, os diagramas IDEF-0 admitem que cada função pode ser decomposta sucessivamente em outras funções, menos complexas, do mais alto nível até as funções mais elementares, de mais baixo nível. A Figura 9 ilustra a decomposição hierárquica de mecanismos.

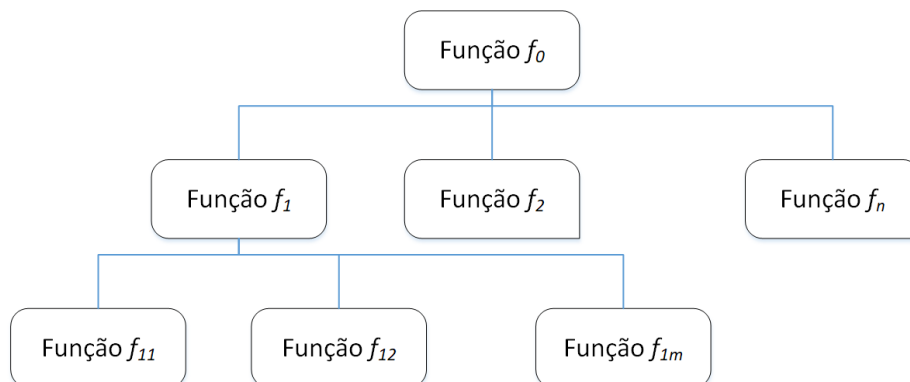


Figura 9 – Decomposição das funções de um sistema de engenharia

Fonte: o Autor

A decomposição das funções de mais alto nível em funções de nível inferior é representada por um conjunto de diagramas IDEF-0. O identificador da função (Id) é indicado no canto inferior direito do respectivo ICOM por um número que varia de 1 a n . Os diagramas IDEF-0 capturam os relacionamentos entre as funções de mesmo nível pelo uso de setas que interligam os respectivos ICOMs. Admitem que as setas sejam ramificadas ou unificadas com um mesmo rótulo, despoluindo visualmente o gráfico.

A Figura 10 ilustra os identificadores das funções (1 – Gravar, 2 – Liberar e 3 – Pagar). As funções 1, 2 e 3 são a decomposição da função de um nível acima. A saída da função 1 (Autorização) se ramifica como controle das funções 2 e 3. De outra forma, as saídas das funções 2 e 3 (Documentos contábeis) se unificam.

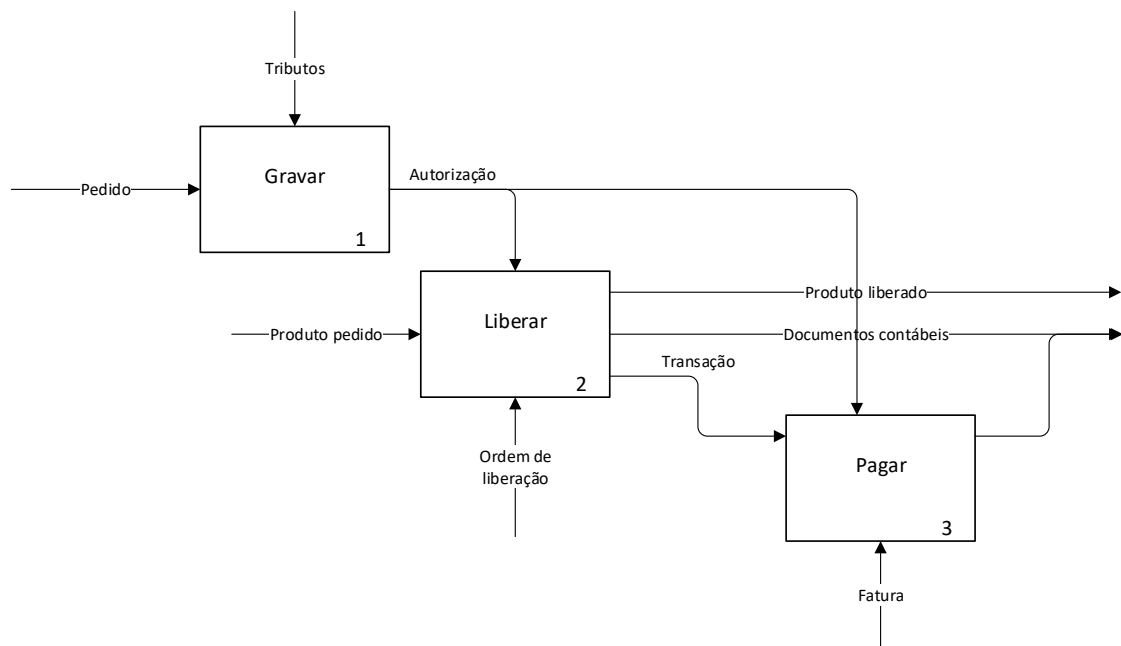


Figura 10 – Exemplo: Numeração, relação entre funções e ramificação/unificação de setas no IDEF-0

Fonte: o Autor

Uma vez compreendidas as regras de um diagrama IDEF-0, pode-se entender a modelagem IDEF-0 para sistemas de engenharia. O modelo compreende um conjunto de diagramas IDEF-0 organizados hierarquicamente do mais alto nível até o nível mais elementar. Cada diagrama IDEF-0 de um nível se decompõe em diagramas filhos, a partir de uma de suas funções. Portanto, exceto o diagrama de mais alto nível, todos os demais possuem um 'pai'. Os ICOMs são inscritos em uma folha de título que contém: à esquerda, a identificação do diagrama; no centro, o título do nó; à direita, a numeração da folha de título.

O diagrama de mais alto nível é representado somente por um ICOM relacionado à função 0 do sistema. A sua identificação é A-0. O diagrama A-0 é um diagrama de contexto, ou seja, contém a função de mais alto nível e suas interações, da qual derivam todos os detalhes do sistema, sejam entradas, saídas, controles e mecanismos.

A identificação do diagrama IDEF-0 inferior ao diagrama A-0 é A0. O diagrama A0 contém todas as funções 'filhas' da função raiz, do diagrama de

contexto A-0, representadas pelos elementos dos respectivos ICOMs. Como várias funções são representadas na mesma folha, normalmente ficam evidenciadas as interfaces funcionais. As setas de um ICOM podem ser ramificadas ou unificadas.

A Figura 11 ilustra a folha de título A0, que contém as funções 1 – Coletar, 2 – Bombear, 3 – Escoar e 4 – Tratar. Como ilustra a Figura 12, o nó raiz A-0, que contém a função 0, se decompõe na filha A0, que contém as funções 1, 2, 3 e 4.

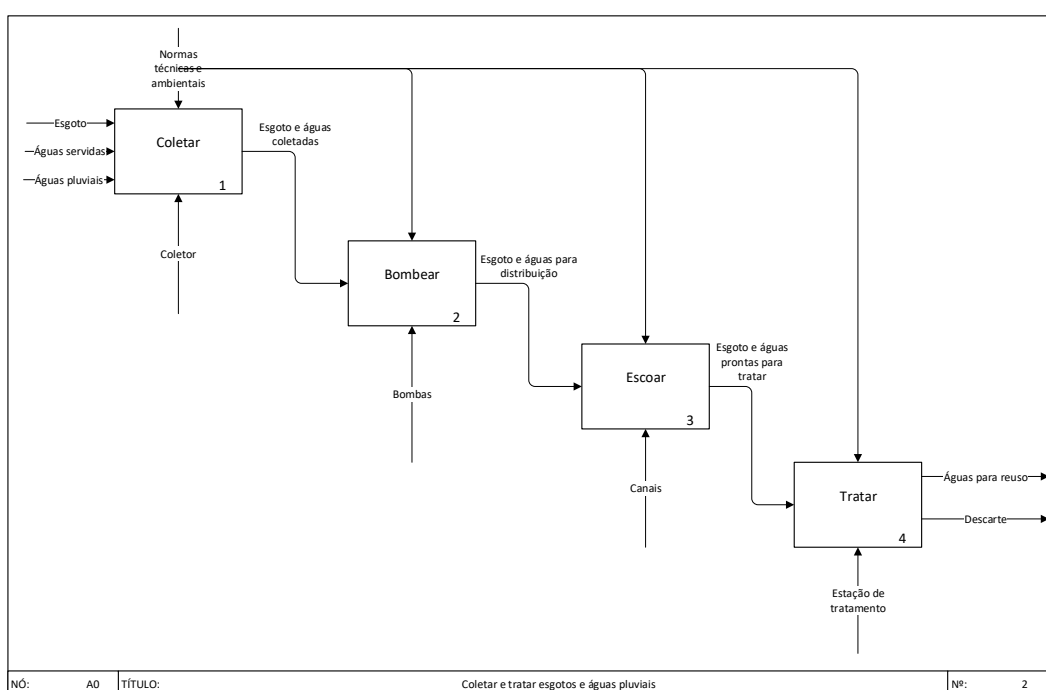


Figura 11 – Nó A0 – Coletar e tratar esgotos e águas pluviais

Fonte: o Autor

Em todas as folhas, as funções (tantas quantas sejam necessárias) serão numeradas cardinalmente a partir do 1. Na folha inferior à A0, a função 1 de A0 se decompõe na folha A1, a função 2 de A0 se decompõe na folha A2 e assim por diante. Se houver decomposição da folha A1, a decomposição da i -ésima função de A1 será representada pela folha A1,i. Por exemplo, A13 é a folha que resulta da decomposição da função 3 de A1.

5. Modelagem dinâmica: diagramas de estados segundo a UML

Em qualquer sistema minimamente interessante, os objetos não estão estáticos, mas interagem entre si. A modelagem do comportamento de um sistema consiste, segundo a abordagem orientada por objetos, em dois tipos distintos de especificações. Por um lado, na modelagem do comportamento inter-objetos, ou seja na identificação dos seus padrões de trocas de mensagens (diagramas de interação). Por outro lado, na modelagem do comportamento intra-objeto, ou seja na identificação dos estados em que um objeto se pode encontrar ao longo do seu ciclo de vida e dos eventos envolvidos (diagramas de estados e de atividades).

5.1 Diagramas de interações

Uma interação é a especificação do comportamento de um conjunto de instâncias, representado pela sua troca de mensagens, num determinado contexto, e com vista à concretização de um dado objetivo.

As interações são utilizadas para modelar o fluxo de controle de uma operação, classe, componente, caso de utilização ou um sistema na sua globalidade. Sempre que existe uma ligação (link) entre instâncias, pode ocorrer uma ou mais interações.

As interações ocorrem provocadas por estímulos. Um estímulo é uma comunicação entre dois objetos que veicula informação com a expectativa que determinada atividade seja realizada. Um estímulo provocará a invocação de uma operação, o envio de um sinal, ou a criação ou destruição de um objeto.

Uma mensagem é a especificação de um estímulo, ou seja, especifica os papéis que os objetos emissor e receptor devem acordar para que uma ação seja realizada. Regra geral, a recepção de uma mensagem resulta na realização de uma ação (que por sua vez pode provocar uma mudança de estado) no objeto destino.

Quando um objeto envia uma mensagem a outro objeto, este por sua vez pode enviar uma outra mensagem a outro objeto, e assim sucessivamente.

Criam-se deste modo sequências de mensagens executadas, em geral, sobre o mesmo processo ou atividade de execução.

Uma interação é representada por um diagrama de interação. Os diagramas de interação são usados para especificar a realização de um caso de utilização bem como a realização de uma operação envolvendo diferentes objetos.

Os diagramas de interação podem ser apresentados nas seguintes formas:

- Diagrama de sequência: é um diagrama de interação com ênfase na ordenação temporal das mensagens trocadas entre os objetos
- Diagrama de colaboração: é um diagrama de interação com ênfase na organização estrutural dos objetos que trocam mensagens entre si.

5.2 Diagramas de estados

Um diagrama de estados, também conhecido por diagrama de transição de estado ou por máquina de estados, permite modelar o comportamento interno de um determinado objeto, subsistema ou sistema global. Estes diagramas representam os possíveis estados de um objeto, as correspondentes transições entre estados, os eventos que fazem desencadear as transições, e as operações (ações e atividades) que são executadas dentro de um estado ou durante uma transição. Os objetos evoluem ao longo do tempo através de um conjunto de estados como resposta a eventos e à passagem de tempo.

5.2.1 Estados

A ilustra um diagrama de estados do modelo de componentes de sistemas de engenharia, representando o comportamento de manutenção, com um estado inicial e um estado final.

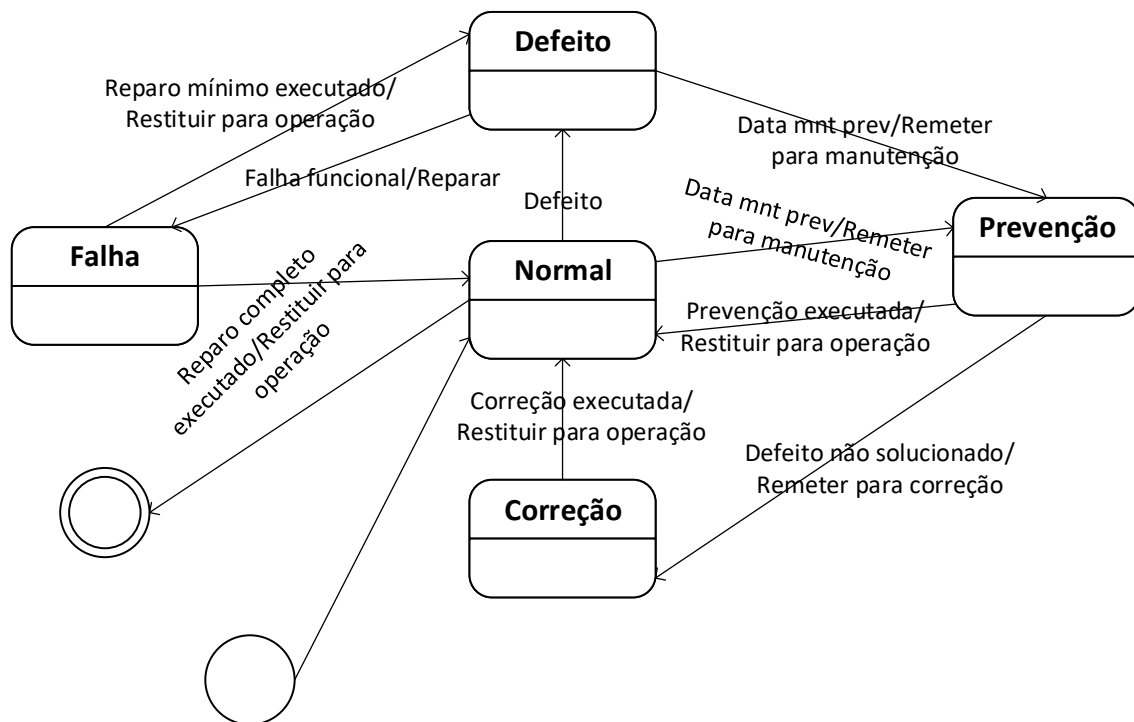


Figura 13 – Diagrama de estados, na notação UML

Na notação UML, os estados são representados por retângulos com os cantos arredondados, exceto o estado inicial e final que têm ícones particulares. As transições são representadas graficamente por uma linha a cheio dirigida. A Figura 13 ilustra a visão detalhada de um estado, que para além do nome, pode ainda incluir um segundo compartimento com as ações e atividades realizadas em cada estado (não apresentadas na figura).

Um estado é uma situação registrada por um objeto durante o seu respectivo ciclo de vida, durante a qual uma condição é verificada, vai executando alguma atividade, ou simplesmente espera que determinado evento ocorra.

Possui diferentes partes, a saber:

- Nome: designação inequívoca;
- Ações de entrada e de saída: Ações executadas, respectivamente, no início (à entrada) ou no fim (à saída) do estado.
- Transições internas: transições que ocorrem mas que não alteram a mudança de estado.

- Sub-estados: uma estrutura aninhada de um estado, envolvendo sub-estados disjuntos (i.e., sequencialmente ativos) ou concorrentes (i.e., concorrentemente ativos).
- Eventos deferidos: uma lista de eventos que não são tratados no estado corrente, mas tratados pelo objeto num seu outro estado.

5.2.2 Transições

Uma transição é uma relação entre dois estados que especifica que um objeto que se encontre no primeiro estado, realizará um conjunto de ações e mudará para o segundo estado quando um determinado evento ocorrer e determinadas condições se verificarem. Um evento é uma ocorrência de um estímulo que pode corresponder a uma transição de estado.

Uma transição tem diferentes partes, a saber:

- O estado de origem e de destino, que a transição interliga;
- Evento de gatilho (*event trigger*): é o evento cuja recepção pelo objeto no estado origem proporciona a realização da transição, caso a condição de guarda seja satisfeita;
- Condição de guarda (não apresentada na Figura 13): expressão lógica que é avaliada quando a transição é lançada pela recepção do evento de gatilho - se a expressão for avaliada verdadeira a transição ocorre, caso contrário a transição não ocorre e o evento é perdido;
- Ação.

É possível existirem situações de transições sem gatilho (*triggerless*), i.e. transições sem eventos de gatilho. São transições que ocorrem apenas porque o estado origem termina a sua atividade normalmente. Uma transição pode ter múltiplos estados-origem (representando a junção de múltiplos estados concorrentes) bem como múltiplos estados-destino (representando a divisão para múltiplos estados concorrentes), não sendo comum este tipo de utilização.

5.2.3 Probabilidade e estados: modelo Markovianos

Os modelos de estados podem ser matematicamente representados por:

$S=\{s_1, s_2, \dots, s_n\}$, o conjunto de estados;

$T=\{t_1, t_2, \dots, t_m\}$, o conjunto de transições;

$I: T \rightarrow S$, Mapeamento da transição para estado de origem;

$O: T \rightarrow S$, Mapeamento da transição para estado de destino;

$L=\{\lambda_1, \lambda_2, \dots, \lambda_m\}$, o conjunto de taxas de transição.

Os modelos de estados em que as transições obedecem a modelos matemáticos analíticos pré-determinados são chamados de modelos determinísticos.

Há casos em que os estados de um modelo de estados podem apresentar uma probabilidade de ocorrência. A probabilidade de cada estado também pode depender da probabilidade de ocorrência de estados anteriores. Nesse caso os modelos reproduzem os chamados processos Markovianos. O modelo de estados Markoviano tem como característica a capacidade de ‘esquecer’ a sequência de estados onde esteve no passado. Apenas o estado atual é relevante para a determinação dos estados futuros.

Considerando essa definição, nos modelos de estados em que se caracteriza o processo Markoviano a definição completa do sistema requer também as seguintes definições

- $P[X(t_0)=s_i]$: probabilidade do objeto ocupar um estado s_i do conjunto de estados S em um instante de tempo t_0 ;
- $P[X(t_0)=s_i|X(t_{-1})=s_j]$: probabilidade do objeto ocupar um estado s_i do conjunto de estados S em um instante de tempo t_0 , dado que ocupava no instante anterior t_{-1} ocupava o estado s_j do conjunto de estados S .

REFERÊNCIAS

MENZEL C., M. R. J. The IDEF Family of Languages. In: BERNUS P., M. K. . S. G. **Handbook on Architectures of Information Systems**. Berlin: [s.n.], 1998. Cap. 10, p. 209-241.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY - NIST. INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0). **FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION - FIPS PUB 183**, Gaithersburg, 21 December 1993.

RUMBAUGH, J. et al. **Modelagem e Projetos baseados em Objetos**. [S.l.]: Campus, 1994.

SILVA, A.; VIDEIRA, C. **UML Metodologias e Ferramentas CASE**. [S.l.]: Centro Atlântico, 2001.