

Modelação para Requisitos de Software

A **Modelação** pode ser definida como o processo de identificar os conceitos adequados e selecionar as abstrações necessárias para construir um modelo que reflita apropriadamente um determinado universo de discurso.

A modelação está, por isso, intrinsecamente ligada aos conceitos de **abstração**, **simplificação** e **formalização**. A **simplificação** envolve a omissão de detalhes da realidade para tornar o modelo mais fácil de compreender, enquanto a **formalização** implica a utilização de uma linguagem precisa e não ambígua para evitar interpretações erróneas.

- **Demasiado realista:** No extremo esquerdo, um coração anatómico, com todos os seus detalhes biológicos, **representa um nível de detalhe excessivo** para a maioria dos contextos. É uma representação fiel, mas a sua complexidade pode obscurecer a mensagem principal.
- **Ideal:** No centro, um coração estilizado com uma flecha, como o ícone universal do amor, atinge o equilíbrio perfeito. Abstrai os detalhes biológicos para comunicar uma ideia clara e universalmente compreendida.
- **Demasiado abstrato:** No extremo direito, uma forma geométrica (um retângulo vermelho atravessado por uma barra preta) é tão abstrata que perde a sua ligação com o conceito original, tornando-se ininteligível sem um contexto adicional.

O Propósito e as Características de um Modelo Eficaz

1. **Abstração:** Um modelo é uma descrição reduzida do sistema. Foca-se nos elementos essenciais, ignorando o que é supérfluo para o objetivo em questão.
2. **Compreensibilidade:** Ao remover detalhes irrelevantes de um determinado ponto de vista, os modelos permitem uma compreensão mais fácil das propriedades do sistema, desde que sejam especificados de forma intuitiva.
3. **Precisão:** Para as propriedades de interesse, o modelo deve fornecer uma representação fiel do sistema. Embora seja uma simplificação, não pode ser incorreto naquilo que se propõe a representar.
4. **Raciocínio:** Um modelo deve ajudar na análise e no raciocínio sobre as propriedades interessantes, mas não óbvias, do sistema. Isto pode ser alcançado através de experimentação (simulação) ou de análise formal.
5. **Economicidade:** Um modelo deve ser drasticamente mais barato de construir e analisar do que o sistema real que representa. Se a construção do modelo for tão ou mais cara que a do sistema, perde o seu propósito fundamental.

Forma: Modelo Físico vs Modelo Simbólico

- **Modelo Físico (ou Icónico):** É uma reprodução material e em escala de um objeto ou processo. A palavra "físico" indica que é concreto e pode ser tocado. Este tipo de modelo imita o sistema original, sendo visualmente semelhante em algumas das suas propriedades, como o modelo arquitetónico (protótipo/maquete) de uma casa ou um modelo molecular, que ajudam a visualizar estruturas tridimensionais complexas.

- **Modelo Simbólico (ou Matemático):** Utiliza relações lógicas e quantitativas para representar um sistema. A sua forma é tipicamente arbitrária ou convencional, sem uma semelhança visual direta com o que representa. Exemplos clássicos incluem:
 - *Fórmulas matemáticas:* A lei de Ohm ($V = R \times I$) representa a relação entre tensão, resistência e corrente através de símbolos, sem se assemelhar a um circuito elétrico.
 - *Linguagem:* A palavra "morango" não tem qualquer relação física com a fruta; é uma convenção simbólica.
 - *Notações de engenharia:* Em UML, uma classe é representada por um retângulo, uma pura convenção da linguagem de modelação.
-

Representatividade: Modelo Descritivo vs. Modelo Prescritivo

- **Modelo Descritivo:** É usado para descrever ou imitar um sistema do mundo real já existente. O seu objetivo é permitir raciocinar sobre as propriedades ou o comportamento desse sistema. Um modelo meteorológico que prevê o tempo é um exemplo clássico. Este tipo de modelo é comum nas ciências naturais e na engenharia inversa, onde se pretende analisar um sistema existente sem o afetar diretamente.
 - **Modelo Prescritivo:** É usado para definir como um sistema, que ainda não foi construído, se deve comportar. É a base da chamada forward engineering. Em engenharia de software, os modelos criados durante a fase de análise descrevem o problema a ser resolvido e servem como um plano para a construção do sistema.
-

Perspetiva: Modelo Estrutural vs. Modelo Comportamental

- **Modelo Estrutural:** Foca-se nos **aspectos estáticos** de um sistema. Descreve os componentes ou módulos que fazem parte do sistema e a sua arquitetura. Em UML, os diagramas de classes, componentes e de implementação (deployment) são exemplos de modelos estruturais, mostrando as "peças" do sistema e como se encaixam.
 - **Modelo Comportamental (Behavioural):** Enfatiza os **aspectos dinâmicos, funcionais e temporais** do sistema. Aborda o comportamento do sistema, sendo especialmente relevante na fase de análise para entender como o sistema reage a eventos e interações. Exemplos incluem máquinas de estados finitos, redes de Petri e diagramas de fluxo de dados (DFDs).
-

Modelos UML

Modelo de Domínio

O propósito de um modelo de domínio é capturar os elementos comuns (propriedades e variáveis) de um domínio específico que é relevante para o sistema. Ele expressa verdades duradouras sobre o universo em que o sistema irá operar.

Modelo de Casos de Uso

Os modelos de casos de uso servem dois propósitos essenciais:

1. Definir a fronteira do sistema, clarificando o que está dentro e fora do seu âmbito.
2. Especificar as funcionalidades que o sistema disponibiliza aos seus utilizadores (denominados atores).

Modelo de Classes

Em métodos de desenvolvimento orientados a objetos, os modelos de classes são indispensáveis para indicar as classes existentes e as suas relações. A estrutura básica de uma classe em UML inclui o seu Nome, os seus Atributos (dados) e as suas Operações (métodos). As relações entre classes definem a estrutura estática do sistema e o seu significado arquitetónico é crucial. As principais relações são:

- **Generalização (Δ):** Representada por um triângulo vazio, estabelece uma relação "é-um-tipo-de" (herança). Permite que uma subclasse herde atributos e operações de uma superclasse, sendo o mecanismo fundamental para a reutilização de código e a criação de hierarquias de tipos.
- **Associação (—):** Uma linha simples que denota uma ligação estrutural entre classes, indicando que os seus objetos estão conectados. A Navegação (seta na ponta da linha) pode ser usada para indicar a direção em que a associação pode ser percorrida.
- **Agregação (\Diamond):** Representada por um losango vazio, é um tipo de associação que modela uma relação "tem-um" ou "parte-de". Nesta relação, o ciclo de vida da parte é independente do todo (ex: um professor pertence a um departamento, mas se o departamento for extinto, o professor continua a existir).
- **Composição (\blacklozenge):** Representada por um losango preenchido, é uma forma forte de agregação que denota uma relação de 'pertença' onde o ciclo de vida da parte está intrinsecamente ligado ao do todo. Se o todo for destruído, a parte também o é (ex: os quartos de um hotel não podem existir sem o hotel).
- **Multiplicidade (*, 1, 0..1):** Anotações numéricas nas pontas de uma associação que indicam quantos objetos de uma classe podem estar relacionados com um objeto da outra classe.

Modelo de sequência

Os modelos de sequência são um tipo de diagrama de interação que modela os aspetos dinâmicos relacionados com a troca de mensagens entre objetos ao longo do tempo. São particularmente úteis para representar uma instância de um caso de uso, mostrando como um grupo de objetos colabora para realizar uma tarefa.

Modelo de Estados

Enquanto os modelos de classes são estáticos, os modelos de estado definem o comportamento dinâmico e temporal de uma instância de uma classe. Um estado é uma condição que persiste por um período de tempo significativo. Num diagrama de estados convencional, apenas um estado está ativo de cada vez. Estes diagramas são excelentes para modelar o ciclo de vida de um objeto, mostrando como ele transita entre diferentes estados em resposta a eventos.

Modelo de Atividades

Os modelos de atividades são úteis para representar o fluxo de controlo entre as atividades de um processo de negócio. São adequados quando o comportamento de um sistema é impulsionado principalmente pela conclusão de ações. Os seus elementos básicos incluem um ponto de início, um de fim, atividades (ações) e decisões (ramos condicionais).

