

Perfil:

Emerson Antonio Klisiewicz.

Possui graduação em Ciências da Computação, pela Pontifícia Universidade Católica do Paraná (1994), Especialização em Sistemas de Informações Gerenciais, pela Pontifícia Universidade Católica do Paraná (1995), Especialização em Redes e Sistemas Distribuídos pela Pontifícia Universidade Católica do Paraná (1997), com experiência de mais de 20 anos em áreas de tecnologia de empresas estatais e financeiras. Currículo Lattes:

<http://buscatextual.cnpq.br/buscatextual/visualizacv.do?id=K4282839T3>

AULA 02 – Introdução a Análise de Sistemas

Introdução:

Na segunda aula, vamos continuar a abordar a evolução da Análise de Sistemas, dando ênfase as novas formas de análise. Iniciaremos com a na análise estruturada, passando pela análise essencial e chegando a na análise orientada a objetos.

Contextualizando:**Análise Estruturada**

No final da década de 70, a análise estruturada possibilitou especificar os requisitos lógicos do sistema em um modelo gráfico de alto nível, capaz de ser compreendido pelos usuários e de ser mapeado para a arquitetura do projeto. O modelo gráfico introduzido pela análise estruturada representa os dados utilizados por um sistema, os fluxos que transportam e os processos que os transformam.

A análise estruturada é uma abordagem sistemática para fazer a análise de um sistema de modo a produzir a uma especificação funcional. A especificação funcional define as funções e estruturas de dados que constituem o sistema. A análise estruturada usa técnicas gráficas simples, modulares e complementares.

Componentes do Modelo Estruturado

- Diagrama de Fluxo de Dados (DFD): representação gráfica da rede de processos interligados.
- Dicionário de Dados: descrição das interfaces.
- Especificação dos Processos: descrição do que cada processo faz.

1) DFD:

Temos as seguintes características sobre o DFD:

- Simplicidade (intuitivo) - acessível ao utilizador / cliente.
- Abordagem “Top-Down”, dá uma visão do sistema do geral para o particular apresentando os detalhes através de níveis hierárquicos.

- O Desenvolvimento é iterativo - aperfeiçoamento sucessivo de uma versão inicial.
- O DFD não é procedimental
 - ✓ Não tenta representar processamento condicional ou ciclos com esta forma diagramática. Simplesmente mostra o fluxo dos dados.
 - ✓ Não há indicação explícita da sequência de processamento ou condição lógica é fornecida pelo diagrama.
 - ✓ Os DFDs representam fluxo de informação e não fluxo de controle - O DFD não é um fluxograma.
- O DFD não contém informação temporal.

Regras para a construção de um DFD:

A-) Depósitos de dados

- Dados não podem mover-se diretamente de um depósito de dados para outro depósito de dados:
 - ✓ Os dados têm de ser movidos por um processo.
- Os dados não podem mover-se diretamente de uma Entidade Externa para um depósito de dados:
 - ✓ Os dados têm de ser movidos por um processo que recebe os dados da Entidade Externa e coloca-os no depósito de dados.
- Os dados não podem mover-se diretamente de um depósito de dados para uma Entidade Externa:
 - ✓ Os dados têm de ser movidos por um processo.
- Um depósito de dados é identificado por um substantivo/nome que deve sugerir o respectivo conteúdo:
 - ✓ Dados do Mercado; Clientes; Pagamentos em atraso.
 - ✓ Encomendas pendentes.
- Não é obrigatória a atribuição de nomes aos fluxos de e para depósitos de dados.

B-) Entidades Externas

- Os dados não podem mover-se diretamente entre duas Entidades Externas:
 - ✓ Têm de ser movidos por um processo se os dados são de algum interesse para o sistema.
 - ✓ Caso contrário, o fluxo de dados não é mostrado no DFD.

- Uma Entidade externa é identificada por um substantivo/nome:
 - ✓ Fornecedores.
 - ✓ Armazém.
 - ✓ Administração.
 - ✓ Utilizadores.

C-) Fluxos de dados

- Um fluxo de dados tem uma única direção de fluxo entre símbolos.
- Pode fluir nas duas direções entre um processo e um depósito de dados para mostrar uma leitura antes de uma atualização. O último é usualmente indicado por duas setas separadas já que isto acontece em tempos diferentes.
- Todos os detalhes dos fluxos de e para depósitos de dados são definidos ao nível da Especificação dos Processos.
- Os fluxos de dados representam o deslocamento de informações entre:
 - ✓ Um Processo e uma Entidade Externa.
 - ✓ Dois processos.
- Um Processo e um Depósito de Dados.
- Não são permitidos os fluxos de dados entre:
 - ✓ Duas Entidades Externas
 - ✓ Dois depósitos de Dados
 - ✓ Uma Entidade Externa e um Depósito de Dados

2-) Especificação de Processos:

Descreve a forma como os dados de entrada são transformados nos dados de saída. Independe da forma como a função é executada (manualmente ou automatizada) e normalmente usa-se o português estruturado ou o pseudocódigo.

A-) Português Estruturado

É uma versão/adaptação do português porque:

- ✓ Verbos no modo imperativo.
- ✓ Termos definidos no dicionário de dados.
- ✓ Palavras “reservadas” para descrever a lógica da função.
- ✓ Pouco uso de adjetivos e advérbios.

- ✓ Usa-se construções típicas da programação estruturada:
 - Seqüências
 - Decisões
 - Repetições.

B-) Pseudocódigo

O pseudocódigo é orientado para a implementação. Já lembra uma linguagem de programação e o objetivo é facilitar a programação.

Exemplo:

```

ABRIR avaliações-de-alunos, disciplinas, alunos e ender-alunos
DEFINIR quant_reprovações = 0
LER alunos
REPETIR-ENQUANTO existam alunos:
    PESQUISAR primeira ocorrência da chave matr-aluno (de alunos) em avaliações-de-alunos
    REPETIR-ENQUANTO existam em avaliações-de-alunos registro para a chave matr-aluno (de alunos) pesquisada:
        SE média-final for MENOR QUE 5
            INCREMENTAR quant_reprovações
            PESQUISAR cod-disciplina em disciplinas
            LER próximo registro em avaliações-de-alunos
        FIM REPETIR
    IMPRIMIR aviso_de-situação-do aluno
    LER próximo registro em alunos
FIM-REPETIR
  
```

3-) Dicionário de Dados:

Proposto como gramática quase formal para descrever o conteúdo de objetos definidos durante a análise estruturada. Geralmente implementado como parte de uma "ferramenta de projeto e análise estruturada" CASE.

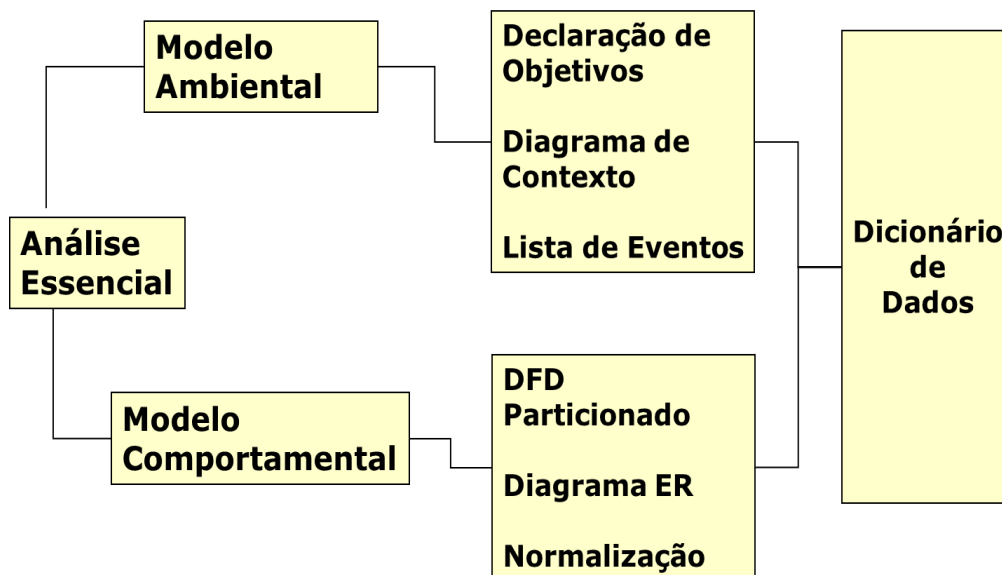
A maioria dos DD contém as seguintes informações:

- ✓ Nome: o nome principal do item de dados, do depósito de dados ou de uma entidade externa *Alias*: outros nomes usados para a primeira entrada.
- ✓ Onde é usado / Como é usado: listagem dos processos que usam o item de dados e como ele é usado. Ex: entrada ao processo, saída do processo, como um depósito de dados, como uma entidade externa.
- ✓ *Descrição de Conteúdo*: notação para representar o conteúdo.
- ✓ Informação Complementar: outras informações sobre tipos de dados, valores previamente estabelecidos (se conhecidos), restrições ou limitações.

Análise Essencial

É o conjunto de técnicas e ferramentas cujo objetivo é auxiliar na análise e definição de sistemas. Na **Análise Essencial** recomenda-se que a especificação do sistema seja apresentada em três perspectivas que se complementam: modelo de processos ou funcional, modelo de dados e modelo de controle. Na Análise Essencial, deve-se considerar **perfeito** o ambiente tecnológico onde será implementado o software a ser projetado (princípio da neutralidade tecnológica). Isto significa considerar que a memória do computador é infinita, seu tempo de resposta é instantâneo, ele não para (não trava), não tem custo, ou seja, é infalível. Este aspecto propicia a análise pensar em uma solução ideal, no desenho do software, fazendo com que não sejam considerados certos requisitos impostos pelas restrições tecnológicas.

Componentes da Análise Essencial:

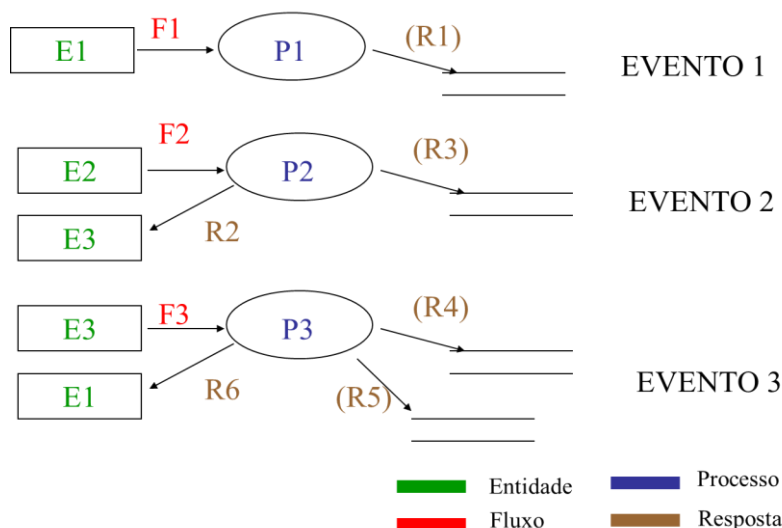


Modelo Ambiental => Voltado para fora do sistema. Mostra a interação do sistema com os elementos externos a ele.

Modelo Comportamental => Voltado para dentro do sistema. Mostra como o sistema deve reagir aos estímulos oriundos do ambiente externo.

A construção do Diagrama de Contexto deve seguir as seguintes etapas:

- Desenhar a bolha que representa o sistema, com o nome do sistema no interior.
- Para cada evento, representar as entidades externas envolvidas, e os fluxos que entram e saem do sistema.
- As entidades externas repetidas devem representadas apenas uma vez.



DECLARAÇÃO DOS OBJETIVOS:

A Declaração dos Objetivos do Sistema deve ser elaborada em poucas frases, simples e precisas, em linguagem destituída de jargão técnico. Deve concentrar-se em o que o sistema deve fazer, sem se preocupar em como será deve ser feito. Sua construção é facilitada se forem considerados a Lista de Eventos e o Diagrama de Contexto do Sistema.

Análise Orientada a Objetos

O mundo real é composto por objetos. Cada objeto tem propriedades e comportamentos. Então por que não desenvolver programas que simulem no computador os objetos do mundo real com suas propriedades e comportamentos?

Segundo Yourdon, “Um sistema construído usando um método Orientado a Objetos é aquele cujos componentes são partes encapsuladas de dados e funções, que podem herdar atributos e comportamentos de outros componentes da mesma natureza, e cujos componentes comunicam-se entre si por meio de mensagens” e o objetivo é encontrar os objetos, organizá-los, descrever como interage através de mensagens, definir operações de seus comportamentos.

Conceitos:

1-) Objeto: Desde a mais tenra idade formamos conceitos. Cada conceito é uma ideia ou um entendimento pessoal que temos de nosso mundo. Os conceitos que adquirimos nos permitem dar um sentido e raciocinar sobre as coisas de nosso mundo. Essas coisas as quais nossos conceitos se aplicam são denominados objetos. Um objeto pode ser real ou abstrato, ex.: Uma fatura, uma organização, um avião...

2-) Classe: é a representação do objetos com seus atributos e métodos.

3-) Atributos : são características que descrevem o objeto.

4-) Métodos ou serviços : as ações que um objeto pode executar.

5-) Abstração: significa que, só deve ser representado aquilo que vai ser usado. Pelo princípio da abstração nos objetos são representadas somente as características que são relevantes para o problema em questão

6-) Encapsulamento: os dados e os processos que tratam esses dados estão “encapsulados” numa única entidade. Os objetos agem como uma “caixa preta”, você utiliza sem precisar saber como ele funciona internamente.

7-) Hierarquia de classes: classe que tem características comuns e que podem fazer parte de uma classe (categoria) maior.

8-) Herança: significa que todos os atributos e métodos programados no ancestral já estarão automaticamente presentes em seus descendentes sem necessidade de reescrevê-los.

9-) Polimorfismo: é o princípio relacionado com as diferentes formas de um objeto.

10-) Agregação: é um mecanismo que permite a construção de uma classe agregada a partir de outras classes componentes. Ex. Casa .

11-) Associação: é usada para agrupar certos objetos que ocorrem em algum ponto no tempo ou sob circunstâncias similares.

Síntese

É necessário que o analista saiba se comunicar com os clientes e garantir clareza de ideias colocadas por eles. Programa de computador, documentação, arquivos de configuração, entre outros e existe por causa das necessidades de clientes. Como transformar necessidades em software? Devem ser consideradas as atividades de como entender as necessidades do cliente, planejar a solução, implementar a solução, validar esta solução, entregar o produto ao cliente. Estas atividades são executadas ordenadas ou não, formalmente ou informalmente. Todo processo de transformação tem início e fim. Essa variável temporal, denominada de ciclo de vida, determina as fases do desenvolvimento de software.

Dentro dessa premissa temos as evoluções da análise desde os anos 50 até os dias de hoje.

Análise estruturada:

- Dimensão exata das necessidades
- Expõe o que é feito por gráficos
- Dirigido para uma ferramenta

- Exige análise de cima para baixo através de refinamentos

Análise Essencial:

- Os eventos são a pedra fundamental dos sistemas.
- Especificação de um sistema deve começar pela identificação dos eventos.

Análise Orientada a Objetos:

- A essência é enfatizar, considerar um domínio de problema e uma solução lógica, segundo a perspectiva de objetos (coisas, conceitos e entidades).
- Atualmente temos ferramentas para sua utilização (integrando especificação e implementação).
- Praticamente todas as ferramentas novas de programação permitem suporte a sua utilização.
- Produtividade em função do reuso .
- Produção de códigos mais fáceis de serem entendidos .
- Adequada a construção de sistemas distribuídos e para aplicações voltadas a Internet.