Transição da atividade de Análise para Projeto

Estudo de caso – Locadora de Veículos

Refinamento referente aos aspectos estáticos e estruturais da atividade de projeto, tomando-se por base a modelagem da atividade de análise. Persistência de objetos, considerando o mapeamento das classes para tabelas de banco de dados relacional.

Refinamento dos aspectos estáticos e estruturais

Como refinamento dos aspectos estáticos e estruturais das técnicas de modelagem da UML, para a atividade de projeto o foco concentra-se na principal técnica de modelagem estrutural, o Diagrama de Classes. É recomendado:

- Refinar as classes, definindo as classes de projeto ou novas classes, ou seja, uma classe de análise pode resultar em mais de uma classe de projeto.
- Definir o estereótipo das classes, que são classes de fronteira (<<boundary>>), de controle (<<control>>) ou de entidade (<<entity>>).
- Estabelecer o tipo de dados de cada atributo, correspondente à linguagem de programação que será adotada na implementação.
- Detalhar as operações, listando todas as operações identificadas nos diagramas comportamentais e de interação.
- Revisar a visibilidade das classes e operações, definindo o nível de acessibilidade de um atributo ou operação por outros objetos, sendo a visibilidade do tipo privada, pública, protegida ou de pacote.
- Rever os tipos de relacionamentos estabelecidos entre as classes, que são do tipo associação (podendo ser associação do tipo reflexiva, binária, ternária, classe associativa e agregação), generalização, dependência ou realização; além de indicar a navegabilidade de cada associação.
- Definir as classes abstratas, as interfaces, padrões de projeto (design patterns), componentes de softwares reutilizáveis, frameworks e demais detalhes pertinentes às tecnologias de desenvolvimento a serem utilizadas durante a implementação, definindo, assim, a arquitetura de um sistema orientado a objetos.

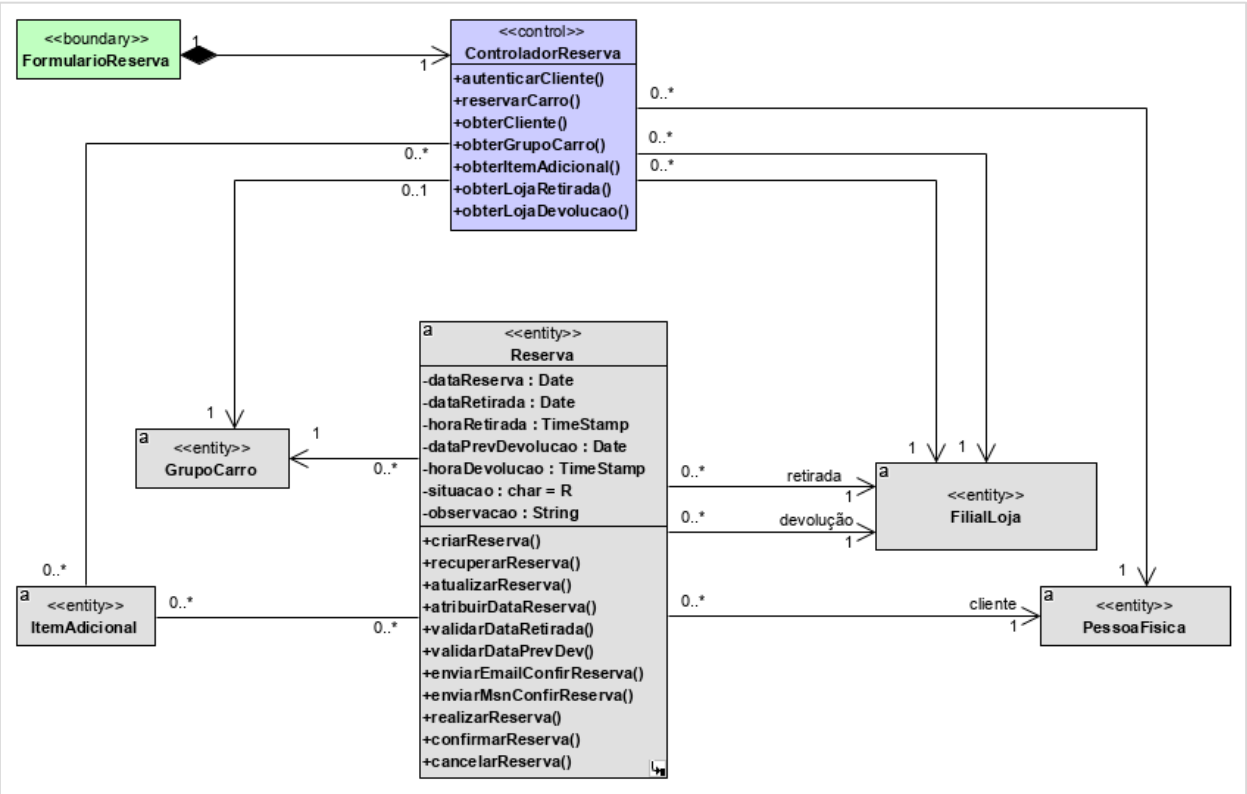
Modelagem de Classe de Projeto

Na representação do Diagrama de Classes da atividade de projeto, é importante revisar o relacionamento estabelecido entre as classes de objetos. Os relacionamentos usuais estabelecidos no Diagrama de Classes de análise são do tipo associação e generalização, no entanto, na versão do diagrama de projeto é comum inserir componentes de softwares, bem como aplicar padrões de projeto (design patterns). Assim, podem surgir os relacionamentos do tipo realização e dependência.

Ainda faz parte da definição das classes de projeto definir o estereótipo das classes. Uma classe indicada com um estereótipo representa uma classificação do elemento.

- **Estereótipo de fronteira *(<<boundary>>)*:** identifica uma classe que serve de comunicação entre os atores externos e o sistema. Muitas vezes uma classe de fronteira é associada à própria interface do sistema. A utilização de classes de fronteira é importante quando é preciso definir a existência de uma interface para o sistema, sendo desnecessária em sistemas muito simples cujas interfaces não apresentam nenhuma característica especial.
- **Estereótipo de controle *(<<control>>)*:** identifica classes que servem de intermédio entre as classes de fronteira e as outras classes do sistema. Os objetos de controle são responsáveis por interpretar os eventos ocorridos sobre os objetos de fronteira, como os movimentos do mouse ou o pressionamento de um botão, e retransmiti-lo para os objetos das classes de entidade que compõem o sistema.
- **Estereótipo de entidade *(<<entity>>)*:** classes de entidade também são chamadas de classes do negócio, e são aquelas que representam os conceitos do domínio do sistema, ou seja, a classe contém informações recebidas ou geradas por meio do sistema. É com base nas classes de entidade que se define quais delas geram objetos que devem ser persistentes, no qual o mecanismo de armazenamento geralmente é um sistema de gerenciamento de banco de dados.

- **Diagrama de Classes (Projeto) - md_Locacao_dc**: recorte do Diagrama de Classes de projeto do sistema "Locação de Veículos" referente à classe "Reserva", correspondente à visão de classes participantes do caso de uso "Reservar Carro".



Fonte: elaborada pela autora.

Persistência de objeto para o Modelo Relacional

Para especificar o mapeamento de classes para tabelas do modelo de dados relacional, é usual adotar técnicas de modelagem de dados e/ou definir o uso de frameworks de mapeamento objeto relacional, como estratégia de armazenamento persistente. Assim, como parte da documentação que envolve o projeto de banco de dados, deve-se apresentar no mínimo a construção do esquema do banco de dados.

Considerando que foi definido o uso de SGBDR como mecanismo de armazenamento dos objetos, é necessário fazer o mapeamento dos valores de atributos de objetos das classes persistentes para as tabelas de banco de dados relacional, com base no Modelo de Classes.

- **Identificação dos objetos das classes:** primeiramente, deve-se identificar se os objetos das classes são objetos transientes ou objetos persistentes. Normalmente os objetos de entidade são os objetos persistentes, os quais devem ser armazenados em meio físico durante a execução do sistema para serem manipulados. Os objetos transientes existem somente durante uma sessão de uso do sistema e geralmente são os objetos de fronteira e de controle.
- **Análise das classes e seus relacionamentos (Esquema do Banco de Dados Relacional)**: na sequência, são analisadas as classes persistentes e seus relacionamentos e aplicadas as alternativas de mapeamento apresentadas a seguir, considerando a notação indicada para manter uma padronização da representação das tabelas e, assim, constituir o esquema do Banco de Dados Relacional (BDR):

  Nome da Tabela (coluna 1, coluna 2, coluna 3, coluna 4,… coluna n)
  - Cada coluna representa um atributo da classe mapeada, no entanto, atenção aos atributos derivados, pois eles não são mapeados para uma coluna.
  - Destaca-se a coluna que representa a chave primária com sublinhado simples e as colunas que representam chaves estrangeiras com sublinhado tracejado.
  - Representa-se em cada tabela derivada de classe, no geral, uma coluna que indica o identificador (Id) para a chave primária. Essa estratégia de notação dos “Ids” define a identidade independente dos objetos, conforme os princípios da orientação a objetos.
- **Mapeamento de Classes para Tabelas:** Segundo Rumbaugh (1997), as principais alternativas de mapeamento de classes para tabelas são:
  - **Mapeamento de associação binária**: para as classes relacionadas com associação binária, com multiplicidade um-para-muitos, mapeia-se cada classe em uma tabela. Para associação binária com multiplicidade um-para-um, pode-se mapear as classes cada uma em uma tabela ou unir os atributos das duas classes em uma única tabela.

- **Mapeamento de classe associativa**: para as classes relacionadas com associação de classe associativa, mapeia-se cada classe em uma tabela. Para classes relacionadas com multiplicidade muitos-para-muitos, cria-se a terceira tabela com apenas a chave primária composta.
- **Mapeamento de agregação**: para classes relacionadas com associação do tipo agregação, mapeia-se a classe "Todo" e "Parte" para tabelas individuais. O identificador da classe "Todo" é indicado como chave estrangeira na tabela que representa a classe "Parte".
- **Mapeamento de composição**: para classes relacionadas com associação do tipo composição (tipo especial de agregação), mapeia-se a classe "Todo" e "Parte" para tabelas individuais. O identificador da classe "Todo" torna-se parte da chave primária na tabela que representa a classe "Parte".
- **Mapeamento de generalização**: existem três abordagens para o mapeamento do relacionamento do tipo generalização em tabelas. A abordagem normal define que a superclasse e as subclasses são mapeadas cada uma em uma tabela com a utilização de um "Id" compartilhado e a criação de um atributo tipo na tabela que representa a superclasse, para identificar os tipos de objetos representados pelos objetos das subclasses. A segunda e terceira abordagens são consideradas alternativas de mapeamento de generalização. A segunda abordagem define a eliminação da tabela correspondente à superclasse e mapeia-se uma tabela correspondente à cada subclasse, reproduzindo todos os atributos da superclasse em cada tabela da superclasse. A terceira abordagem define a criação de uma única tabela correspondente à superclasse, unindo todos os atributos das subclasses ao nível da superclasse.

SAIBA MAIS

Seguindo as tendências do desenvolvimento ágil de software que enfatiza a documentação de análise e projeto com artefatos mínimos, o desenvolvimento simplificado com entregas incrementais antecipadas, equipes de projeto pequenas e altamente motivadas, além de priorizar como princípios de desenvolvimento a comunicação ativa entre desenvolvedores e clientes e a satisfação do cliente, entre os atuais modelos de processo de desenvolvimento ágil, o Processo Unificado Ágil (AUP – Agile Unified Process) é uma evolução do Processo Unificado que contempla as fases clássicas de Concepção, Elaboração, Construção e Transição, entretanto, em cada atividade, a equipe itera para alcançar a agilidade e entregar incrementos de software significativos para os usuários o mais rápido possível. De acordo com Pressman e Maxim (2016, p. 78), "[…] embora o AUP tenha conexões históricas e técnicas com a linguagem de modelagem unificada, é importante notar que a modelagem UML pode ser usada com qualquer modelo de processo ágil".

PESQUISE MAIS

O artigo “Desenvolvimento e Avaliação de um Perfil UML para Modelagem de Jogos Educacionais Digitais” tem o objetivo de apresentar o desenvolvimento e a avaliação do UP4EG, um perfil UML para modelagem de JEDs por meio de diagramas de classes UML. Para realizar a leitura, acesse a plataforma Biblioteca Virtual da Kroton/EBSCO HOST/Buscar por artigos e busque pelo título do artigo.

RODRIGUES DE OLIVEIRA, L. et al. Desenvolvimento e Avaliação de um Perfil UML para Modelagem de Jogos Educacionais Digitais. **Revista Brasileira de Informática na Educação**, [s. l.], v. 26, n. 2, p. 124–143, 2018.

O artigo “Um Estudo Comparativo entre Banco de Dados Orientado a Objetos, Banco de Dados Relacionais e Framework para Mapeamento Objeto/Relacional, no Contexto de uma Aplicação Web” tem o objetivo de apresentar um estudo comparativo entre o uso de Banco de Dados Relacional (BDR), banco de dados relacional com o uso de frameworks para mapeamento de objeto/relacional e Banco de Dados Orientado a Objeto (BDOO). Para realizar a leitura, acesse a plataforma Biblioteca Virtual da Kroton/EBSCO HOST/Buscar por artigos e busque pelo título do artigo.

OLIVEIRA, M. M. A. et al. Um Estudo Comparativo Entre Banco De Dados Orientado a Objetos, Banco De Dados Relacionais E Framework Para Mapeamento Objeto/Relacional, No Contexto De Uma Aplicação Web. **HOLOS**, [s. l.], v. 31, n. 1, p. 182–198, 2015.