

Material de Apoio 14

Framework

- **Etimologia:** *frame* (quadro, estrutura), *work* (trabalho) = Estrutura de trabalho
- **Motivação:** demanda crescente de software, necessidade de aumentar o desempenho de produção de software, melhora de qualidade, rentabilizar o dispêndio de esforços, aumentar lucros.
- **Alavanca do sucesso:** o paradigma de orientação a objetos.

Introdução

Framework é uma técnica que é aplicada tanto no projeto quanto no desenvolvimento de um software orientado a objetos. Ela implica, basicamente, em explorar o potencial de reutilização de partes de software já desenvolvidas e/ou desenvolver novos componentes de software prevendo sua reutilização no futuro. Embora o conceito de framework possa ser aplicado nos diferentes paradigmas de programação, é na orientação a objetos que ele encontra seu substrato ideal para implementação, uma vez que existe um mapeamento bastante interessante entre os conceitos associados a estes dois assuntos.

A idéia básica para construção de um framework é não desenvolver uma solução para uma aplicação específica, mas sim capturar o comportamento geral de um domínio de aplicação e montar uma estrutura de controle capaz de representa-lo. A realização de um software para uma implementação específica consiste em instanciar o referido framework, através da especialização de seus componentes. Esta especialização pode ser entendida como customização e extensão da estrutura do framework. A Figura 1 compara o processo de desenvolvimento de software utilizando frameworks e utilizando OO pura (são consideradas apenas macro-etapas do processo de desenvolvimento).

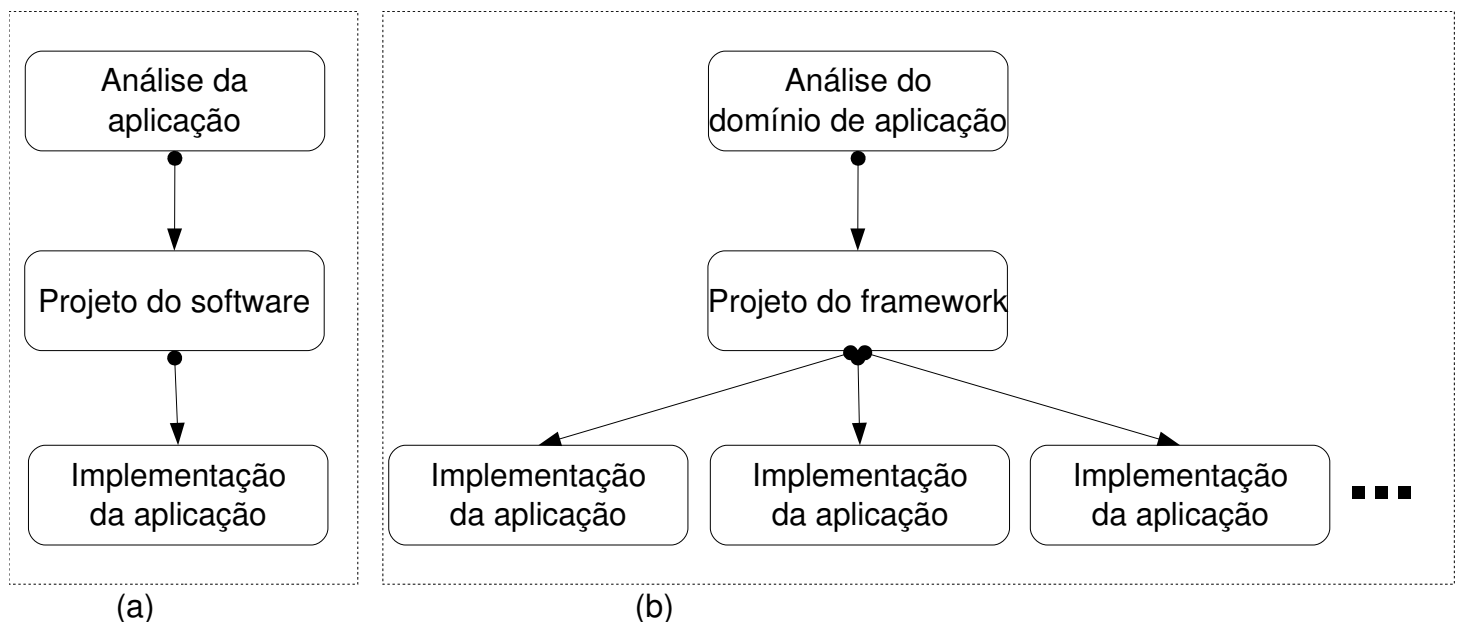


Figura 1. Comparação dos processos de desenvolvimento utilizando puramente o paradigma de orientação a objetos (a) e frameworks (b). Visão macroscópica.

A bibliografia é rica em definições para framework. Algumas são:

- “A framework is a set of classes that embodies an abstract design for solutions to a family of related problems, and supports reuse at a larger granularity than classes.” [Johnson and Foot, 1988]
- “A framework is a set of objects that collaborate to carry out a set of responsibilities for an application subsystem domain. [Damian, 1991]
- “A framework is an architecture developed with the objective to obtain the maximum reuse, represented as a group of abstract and concrete classes, with a great potential specialization.” [Mattsson, 1996].

Ainda apresentando definições, alguns termos são comumente aplicados a framework. O núcleo básico do framework é normalmente referenciado como *kernel*. Este kernel é o responsável pelo motor de execução do framework e compreende a parte genérica de uma aplicação para todos os problemas de um determinado domínio. As partes adaptáveis do framework são denominadas de *hot spots*. Hot spots são os pontos de flexibilização do framework, onde ocorre a customização do software para uma determinada aplicação. Por analogia, o kernel também é referenciado, algumas vezes, como frozen spots.

Questões práticas para uso de framework

A utilização de frameworks, como dito anteriormente, é em muito, justificada pelo aumento da rentabilidade de empresas de desenvolvimento de software. De forma pontual, os benefícios a serem atingidos são:

- **Modularidade:** alcançada através do encapsulamento apoiado em uma interface de serviços estável. Este encapsulamento promove a qualidade de software através do aumento da localidade de código, reduzindo o impacto de eventuais mudanças no design e reduzindo o esforço necessário para compreensão e manutenção de software.
- **Reuso:** as interfaces estáveis propiciam reuso pois definem componentes genéricos que podem ser reaplicáveis para criar novas aplicações.
- **Extensibilidade:** o framework consegue extensibilidade pois provê métodos explícitos que possibilitam às aplicações estenderem suas interfaces (variações requeridas pelas instanciações de uma aplicação).

No entanto, ao se fazer uso de frameworks deve-se também ter em conta os riscos desta técnica. O primeiro, e mais evidente, é em relação ao desempenho. Em geral, a construção de software (ou camadas de software) genéricas e de facilmente reutilizáveis implica em baixo desempenho de execução. Outro aspecto a ser considerado é o fluxo de desenvolvimento de um framework – este ciclo é bastante complexo e, não raro, necessita várias iterações. Deve-se igualmente notar a necessidade do envolvimento de pessoal habilitado no domínio da aplicação para projeto do framework em si.

Classificação de frameworks

A literatura aponta duas possíveis classificações de frameworks, uma por escopo e outra considerando a técnica de implementação de aplicações com o framework. Considerando o escopo, são identificados três tipos básicos:

- System infrastructure frameworks. Este tipo de framework é considerado de uso restrito. Consistem em camadas infra-estrutura de software, sendo sendo utilizados como base para construção de outras aplicações. Como exemplo frameworks para suporte à comunicação de dados e apoio à utilização de banco de dados. Normalmente este tipo de framework não é vendido diretamente aos clientes, e sim incorporado ao software desenvolvido.
- Middleware integration frameworks. Este tipo de framework prevê a integração de diferentes componentes freqüentemente usados em aplicações distribuídas, de forma a obter um ambiente de execução integrado.
- Enterprise application frameworks. São os frameworks voltados para domínio de aplicação.

No caso desta disciplina, os frameworks de aplicação são o tema principal. Considerando a técnica de implementação de aplicações com o framework, encontramos:

- Whitebox. A utilização de frameworks whitebox explora fortemente os conceitos da programação orientada a objetos, em particular a herança e a ligação dinâmica. Seu reuso e a extensão de suas funcionalidades se dá pela herança de classes abstratas e sua especialização pela implementação de métodos. Características operacionais de frameworks deste tipo são expostas ao programador de aplicação.
- Blackbox. Neste tipo de framework o reuso se dá pela associação de componentes para construção de uma aplicação. Basicamente o framework apresenta um conjunto de componentes oferecendo um conjunto de funcionalidades que deve ser associado aos componentes da aplicação propriamente dita. Neste caso, características operacionais do framework ficam ocultas do programador de aplicação.
- Graybox. Agrega as características dos dois anteriores.

Documentação

A medida do sucesso de um framework é dada pelo número de instâncias de aplicações dele realizadas. Para viabilizar este sucesso, é vital que o framework seja acompanhado de uma farta e útil documentação para o usuário – note, usuário, neste caso, refere-se ao programador responsável pelo desenvolvimento de aplicações. Nesta documentação não devem faltar os seguintes itens:

- Identificação do domínio de aplicação. Deve ser caracterizado o domínio de problemas para qual o framework foi desenvolvido, habilitando potenciais usuários à facilmente encontrar a solução proposta.
- Fundamentos de utilização. Devem ser fornecidas as informações necessárias para reutilização dos componentes do framework, caracterizando as funcionalidades oferecidas e aquelas que o usuário deve estender. Especial atenção deve ser dada ao fluxo de informações entre o framework e a extensão proposta.

- Caracterização de aplicações. Exemplos de aplicações para as quais o framework é voltado. Estes exemplos devem caracterizar o problema bem como apresentar o processo de instanciação da aplicação com o framework.
- Design do framework. Descrição detalhada das classes do framework e os seus relacionamentos e colaborações, identificando os componentes a serem especializados pelo usuário.

Exercícios

1. Em uma aplicação voltada a simulação orientada a eventos se justifica a construção de um framework? Caso positivo, identifique esta estrutura.
2. Construa um framework para uma “agenda eletrônica”. Esta agenda deve possuir as seguintes características:
 - O sistema de agenda pode ter vários usuários;
 - Os usuários selecionam o tipo de evento para o qual ele deseja ser avisado;
 - Ao selecionar um evento o usuário escolhe o alarme(data e hora) que deseja e como ele deve ser avisado;
 - O usuário é avisado através dos canais de comunicação do sistema com o usuário
 - Ex. de evento: tarefa, aniversário
 - Ex. de canal: celular, email

(Exemplo retirado da internet, trabalho acadêmico na disciplina do prof. Tavares, na UCP)

Referências

- A. Damian. *Software Frameworks*. Notas de aulas - não publicado. Acesso: <http://sem.ucalgary.ca/courses/SEN/609.03/W98/adi2/framework.html>.
- R. Johnson, B. Foot. Designing Reusable Classes. *Journal of Object-Oriented Programming*. V 1(12):22-35. 1988.
- M. Mattsson. *Object-Oriented Frameworks: a survey of methodological issues*. PhD thesis. Dept. of Computer Science, UCK. 1996.