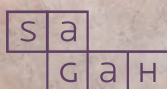


ANÁLISE E PROJETO DE SISTEMAS

Cleverson Lopes Ledur



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Elaborar diagrama de implantação

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar conceitos básicos sobre o diagrama.
- Caracterizar os nós e suas associações.
- Elaborar o diagrama de implantação adequadamente.

Introdução

Os diagramas de implantação são compostos por várias formas UML. As caixas tridimensionais, conhecidas como “nós”, representam o software básico ou elementos de hardware, ou nós, no sistema. As linhas entre nós indicam relacionamentos, e formas menores, contidas nas caixas, representam os artefatos de software que são implantados. Todos esses elementos permitem que a arquitetura do sistema seja representada. Mas criá-los, embora pareça fácil, requer conhecimentos sobre arquitetura, servidores, componentes e uma série de outros artefatos que fazem parte do sistema.

Neste texto, você vai estudar os conceitos básicos sobre o diagrama de implantação. Vai entender os diferentes tipos de nós e suas associações, bem como conhecer a forma de criação de diagramas de implantação.

Conceitos básicos

Um diagrama de implantação da UML descreve uma visão estática da configuração de tempo de execução dos nós de processamento e dos componentes que são executados nesses nós. Em outras palavras, os diagramas de implantação mostram o hardware do sistema, o software que está instalado nesse hardware e o *middleware* usado para conectar as máquinas diferentes entre si. Você pode criar um diagrama de implantação para aplicativos que são usados em

várias máquinas, por exemplo, um aplicativo de ponto de venda executado em um computador de rede *thin client*, que interage com vários servidores internos por trás do firewall corporativo ou de um sistema de serviço ao cliente implantado. Você pode descrever uma arquitetura de serviços da web, como o .NET da Microsoft. Os diagramas de implantação também podem ser criados para explorar a arquitetura de sistemas incorporados, que mostram como os componentes de hardware e software funcionam juntos (BRUEGGE; DUTOIT, 2004).

Os diagramas de implantação são compostos por várias formas UML. As caixas tridimensionais, conhecidas como nós, representam o software básico ou elementos de hardware, ou ainda nós no sistema. As linhas entre os nós indicam relacionamentos e as formas menores contidas nas caixas representam os artefatos de software que são implantados. Veja na Figura 1 a seguir:

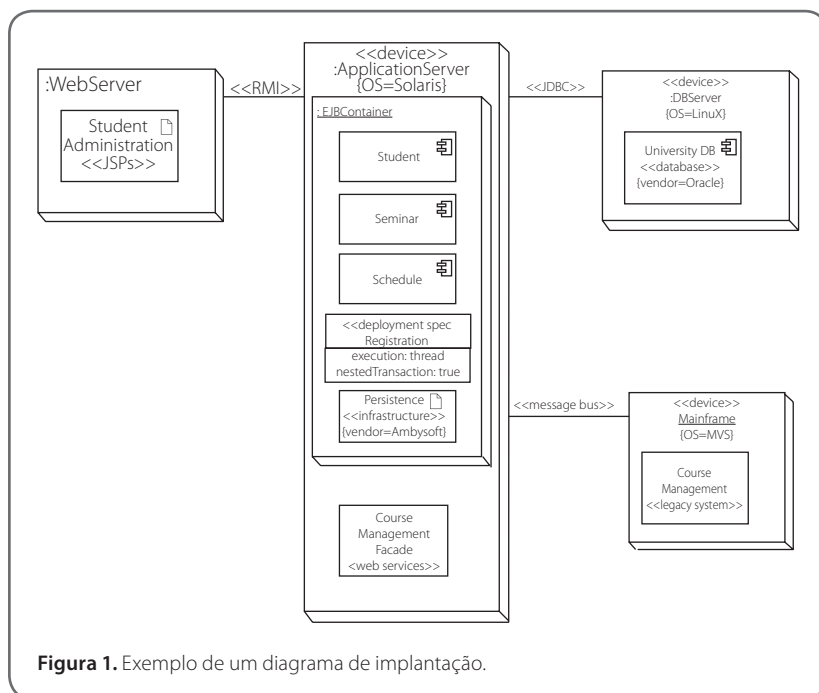


Figura 1. Exemplo de um diagrama de implantação.

Nós e associações

O diagrama de implantação é composto por, basicamente, dois tipos de elementos. Os nós e as associações. Será analisado nesta seção cada um destes elementos e as suas variações, bem como, as suas representações.

Nós

Os nós aparecem como caixas e os artefatos atribuídos a cada nó aparecem como retângulos dentro das caixas. Os nós podem ter subnodos que aparecem como caixas aninhadas. Um único nó, em um diagrama de implantação, pode representar conceitualmente vários nós físicos, como um conjunto de servidores de banco de dados (BRUEGGE; DUTOIT, 2004).




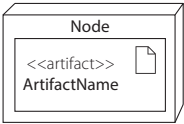
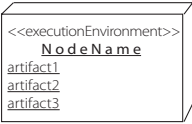

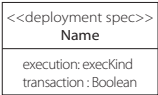
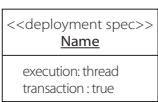

Existem dois tipos de nós (OBJECT MANAGEMENT GROUP, 2011):

- Nó do dispositivo.
- Nó de ambiente de execução.

Os **nós de dispositivos** são recursos de computação física com memória de processamento e serviços para executar o software, como computadores típicos ou telefones celulares. Um **nó de ambiente de execução** (EEN) é um recurso de computação de software que é executado dentro de um nó externo e que ele próprio fornece um serviço para hospedar e executar outros elementos de software executáveis.

Os nós podem aparecer de diferentes formas. Veja como está especificado na UML, conforme o Quadro 1 (OBJECT MANAGEMENT GROUP, 2011):

Quadro 1.

Nome	Forma
Artefato	
Nó	
Artefato implantado em um nó.	
Nó com artefatos implantados (1).	
Nó com artefatos implantados (2).	
Especificação de implantação.	
Especificação de implantação com propriedades.	
Especificação de implantação com valores de propriedades.	
Artefato com propriedades de implantação em forma de anotação.	

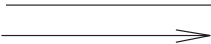
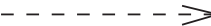



Relações

As relações são geralmente formas de linhas ou setas que representam alguma informação entre dois nós de um diagrama de implantação. Leia abaixo as relações que podem ocorrer entre nós.

- **Associação:** uma associação implica que dois elementos modelo tenham um relacionamento, geralmente, implementado como uma variável de instância em um ou ambos os elementos.
- **Dependência:** um objeto de origem depende do seu objeto de destino.
- **Generalização:** uma generalização é usada para indicar herança.
- **Implantação:** uma implantação é um tipo de relacionamento de dependência que indica a fixação de um artefato em um nó ou destino executável, normalmente em um diagrama de implantação.
- **Manifesto:** um relacionamento manifesto indica que a fonte do artefato incorpora o elemento do modelo de destino, geralmente nos diagramas de componente e implantação.

No Quadro 2, você pode conhecer a forma utilizada no diagrama para representar cada uma das relações (OBJECT MANAGEMENT GROUP, 2011).

Quadro 2.

Nome	Representação
Associação	
Dependência	
Generalização	
Implantação	
Manifesto	

A elaboração de um diagrama de implantação é bastante simples e não requer um grande esforço como nos outros diagramas da UML. Esse tipo de diagrama é associado à etapa de análise e projeto, principalmente, na modelagem dos comportamentos durante a implementação do sistema.

Para a criação, você pode fazer alguns questionamentos e apontamentos gerais de forma que facilite a modelagem e projeto do sistema:

- Você identificou o escopo do seu sistema? Deve-se saber se está diagramando um único aplicativo ou a implantação em toda uma rede de computadores.
- Com quais sistemas legado você precisa interagir? Certifique-se de que conhece o software e os protocolos de operação com os quais estará trabalhando e o monitoramento que será implementado. Também confirme de ter considerado as limitações de seu hardware físico.
- Qual arquitetura de distribuição está sendo usada? Agora, você deve ser capaz de responder a mais estas perguntas:
 - Quantos níveis o seu aplicativo terá?
 - Qual é o aplicativo que será implantado?
 - Você tem todos os nós que precisa? Sabe como eles estão conectados?
 - Você sabe quais componentes serão em quais nós?

Antes mesmo de iniciar a criação do seu diagrama de implantação, você deve entender a arquitetura geral do sistema. Dessa forma, é necessário identificar quais serão os nós e artefatos que serão utilizados no diagrama. Para guiar a criação de um diagrama, considere os seguintes passos:

1. Identifique o escopo do modelo.
 - O diagrama aborda como implantar uma versão de um único aplicativo ou descreve a implantação de todos os sistemas dentro de sua organização?
2. Considere questões técnicas fundamentais.
 - Com quais sistemas existentes o sistema precisa interagir/integrar?
 - Quão robusto o sistema precisa ser (haverá hardware redundante para failover)?
 - O que/quem precisará se conectar e/ou interagir com sistema e como eles o farão (via internet, trocando arquivos de dados etc.)?

- O middleware, incluindo o sistema operacional e as abordagens/protocolos de comunicação, o sistema usará?
 - Com que hardware e/ou software seus usuários interagirão diretamente (PCs, computadores de rede, navegadores etc.)?
 - Como você pretende monitorar o sistema assim que ele for implantado?
 - Quão seguro o sistema precisa ser (você precisa de um firewall, de hardware fisicamente seguro etc.)?
3. Identifique a arquitetura de distribuição.
- Você pretende adotar uma abordagem *fat-client* em que a lógica de negócios está contida em uma aplicação de desktop ou uma abordagem de *thin-client* em que a lógica de negócios é implantada em um servidor de aplicativos?
 - Seu aplicativo terá duas camadas, três camadas ou mais?
 - Sua estratégia de arquitetura de distribuição será, muitas vezes, predeterminada para sua aplicação, especialmente, se você estiver implantando o sistema em um ambiente técnico existente.
4. Identifique os nós e suas conexões.
- A estratégia de distribuição definirá os tipos gerais de nós que você terá, mas não os detalhes exatos. É preciso tomar decisões da plataforma, como o hardware e os sistemas operacionais a serem implantados, incluindo a forma como os vários nós serão conectados.
5. Distribua software para os nós.
- Ambas as versões dos diagramas de implantação indicam o software que é implantado em cada nó, informações críticas para qualquer pessoa envolvida no desenvolvimento, instalação ou operação do sistema.

Um exemplo que se pode utilizar para descrever essas ações é o diagrama apresentado na Figura 2, em que há um aplicativo da web que é implantado no servidor de aplicativos wsrsv-01 e vários esquemas de banco de dados - para o servidor de banco de dados dbsrv-14. É possível ver que tem alguns ambientes de execução como <<JSP server>> Tomcat 7 e o <<executionEnvironment>> Catalina Servlet Container. Nesses nodos, se tem alguns artefatos e uma especificação de implantação (um arquivo XML com informações de como deve ocorrer a implantação).

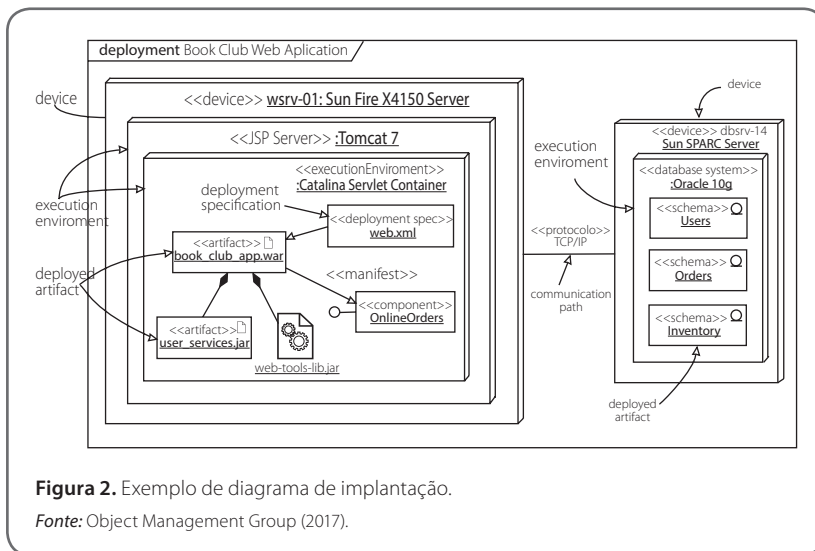


Figura 2. Exemplo de diagrama de implantação.

Fonte: Object Management Group (2017).

Do outro lado, existe um caminho de comunicação para um servidor de banco de dados em que é rodado um ambiente *Oracle* e dentro têm alguns `<<schema>>` para as entidades.

Veja que a especificação desse diagrama requer conhecimentos técnicos da arquitetura da aplicação. Por isso, é interessante que ao criá-la, você tenha todas as informações necessárias e também o acompanhamento de pessoas do time que irão trabalhar diretamente na criação desta arquitetura.



Exemplo

Para você ver mais um exemplo de solução de problemas e melhoria de processos, acesse o artigo "Proposta de melhoria do processo de uma fábrica de polpas por meio da metodologia de análise e solução de problemas" (SOUZA et al., 2015).



Referências

BRUEGGE, B.; DUTOIT, A. H. *Object-Oriented Software Engineering: using UML, Patterns and Java*-(Required). Nova Jersey: Prentice Hall, 2004.

OBJECT MANAGEMENT GROUP. *UML Superstructure Specification 2.4.1*. Needham, 2011. Disponível em: <<http://www.omg.org/spec/UML/2.4.1>>. Acesso em: 8 set. 2017.

OBJECT MANAGEMENT GROUP. *Deployment Diagrams Overview*. Needham: UML, 2017. Disponível em: <<http://www.uml-diagrams.org/deployment-diagrams-overview.html>>. Acesso em: 12 set. 2017.

Leitura recomendada

AMBLER, S. W. *UML 2 Deployment Diagrams: an agile introduction*. Agile Modeling, Toronto, 2014. Disponível em: <<http://www.agilemodeling.com/artifacts/deploymentDiagram.htm>>. Acesso em: 8 set. 2017.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016.

SCHACH, S. R. *Engenharia de Software: os paradigmas clássico e orientado a objetos*. 7. ed. Porto Alegre: McGraw-Hill, 2009.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R.; BETTS, A. *Gerenciamento de operações e de processos: princípios e práticas de impacto estratégico*. 2. ed. Porto Alegre: Bookman, 2013.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS