

Applications réparties à composants et JavaEE

Jonathan Lejeune

Sorbonne Université/LIP6-INRIA

SRCS – Master 1 SAR 2019/2020

sources :

Développons en Java, Jean-Michel Doudoux

Cours précédent de Lionel Seinturier

Wikipédia

Cours de Jérôme Hugues et Ada Diaconescu (Telecom ParisTech)

Un cycle de vie complexe

- Définition des interfaces, de la topologies de l'application
- Implantation des interfaces
- Tests
- Configuration
 - ⇒ Paramétrage des entités applicatives et de l'env. d'exécution
- Déploiement sur plusieurs nœuds
 - ⇒ Positionner les entités applicatives et les préparer à s'exécuter

Et des évolutions

- Correction de bugs
- Évolutions/extensions des interfaces
- Déploiement sur une nouvelle architecture (topologie, OS, ...)

Construire une application répartie

Les outils à notre disposition

- Remote Procedure Call
- Objets répartis
- Message Oriented Middleware

Leurs limites

Ces outils butent sur :

- le passage à l'échelle de l'application (nombre d'entités)
- l'évolution de l'application

Pas de standard de déploiement et de configuration

- Instanciation manuelle des objets sur le système
 - Politiques de configuration à plusieurs niveaux :
 - bus à objets, serveur d'activation, objet, connexion
- ⇒ Configuration et déploiement noyés dans le code applicatif

Solution : séparer le code d'une entité des mécanismes de configuration et de leur mise en place

Difficile d'étendre les fonctionnalités des objets

- Impossibilité de connaître les clients d'une interface
- ⇒ difficile d'assurer le suivi des modifications simplement

Solution : étendre le modèle de programmation pour ajouter la notion "d'interface requise" à celle "d'interface publiée"

Définition selon [Szyperski 97]

***Unité de composition** qui a, par **contrat**, spécifié uniquement ses **interfaces** et ses **dépendances** explicites de contextes. Un composant logiciel peut être déployé indépendamment et est sujet à composition par des entités tierces.*

Une notion commune à plusieurs systèmes sous différentes formes

- Logicielle : EJB, Java Beans, OSGi
- Modélisation : composants UML
- Matérielle : composants électronique

En résumé

Outil facilitant la **réutilisation**, la **composition** et le **déploiement**

Propriétés des composants

- unité de déploiement indépendante
 - ⇒ Séparation forte par rapport à l'environnement
 - ⇒ Déploiement partiel impossible
- unité de composition par une entité tierce
- unité qui n'a pas d'état persistant
 - ⇒ notion de copie non définie
 - ⇒ un composant est soit disponible soit indisponible

Propriétés des objets

- Unité d'instanciation ; il a un identifiant unique
 - ⇒ instanciation partielle impossible
 - ⇒ plan de construction : la classe
- Encapsule son état et son comportement
 - ⇒ État change, pas son identifiant
- L'état peut être persistant

Apports du composant sur l'objet

- plus haut niveau abstraction
- meilleure encapsulation, protection, autonomie
- communications plus explicites
port, interface, connecteur
- connectables/composables
- séparation de la partie métier de la partie technique
- meilleure couverture du cycle de vie
conception, implémentation, packaging, déploiement, exécution

Un nouveau modèle de programmation

Structuration d'une application répartie en définissant :

- Des composants implantant une fonctionnalité
- Un container définissant les frontières des composants
- Un framework à composants permettant l'automatisation de :
 - la connexion et l'assemblage
 - la configuration
 - l'instanciation et du déploiement

Java Enterprise Edition



Qu'est ce que c'est ?

Un ensemble de concepts et de spécifications définies par Oracle pour le développement d'applications Java réparties pour les entreprises.

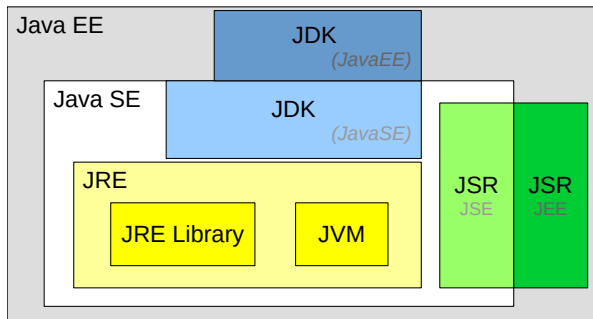
En pratique

- Une Extension de l'API de de Java Standard Edition (Java SE) :
 - correspondance objet Java et données relationnelles (SGBD)
 - services Web
 - architectures distribuées multi-tier
- Des composants modulaires s'exécutant dans un conteneur JavaEE
- Un **serveur d'applications** automatisant interactions, dépendance et déploiement des conteneurs et des composants

Domaines applicatifs

E-commerce, Systèmes d'informations, Sites web, services,

Positionnement de Java Enterprise Edition



Entités associées à chaque version de Java EE

- **Java Specification Requests (JSR)** : spécif. de la version considérée
- **Java Development Kit (JDK)** : les bibliothèques logicielles
- **Java Runtime Environment (JRE)** : le seul env. d'exécution

API de communications

- **Java RMI** : requête/réponse entre objets Java
- **Java IDL** : requête/réponse entre objets Java et non-Java (via CORBA)
- **JAX-RPC/JAX-WS** : requête/réponse Web Service (HTTP+SOAP+XML)
- **JMS** : messages + boîte à lettres

API de connexion

- **JNDI** : annuaires (LDAP, DNS, ...), et espace de noms d'objet (ENC)
- **JDBC** : bases de données
- **JavaMail** : serveur mail

API de développement de composants d'application JavaEE

- **EJB** : Composants distribués transactionnels
- **Servlet** : Composants Web

Java Enterprise Edition : bref historique

- Déc. 1999 : J2EE 1.2
EJB 1.1, JDBC 2.0, JNDI 1.2, RMI-IIOP 1.1, Servlet 2.2, JMS 1.0, JavaMail API 1.2
- Nov. 2003 : J2EE 1.4
ajout de supports pour Web Services (JAXP, JAX-RPC, JAXR)
- Mai 2006 : Java EE 5
EJB 3.0
- Dec. 2009 : Java EE 6
EJB 3.1
- Juin 2013 : Java EE 7
EJB 3.2
- Depuis fin 2017 : Java EE 8
support pour HTML5, HTTP 2.0 et Cloud computing

Implémentations existantes

- Une implémentation de référence : Oracle GlassFish 5.0
- Commerciales : WebSphere (IBM), JEUS 8 (TmaxSoft), ...
- Open-source : Wildfly 12/JBoss (RedHat), Geronimo (Apache),

Processus de certification

- payant sauf pour plates-formes open-source
- assez-lourd à mettre en œuvre (plusieurs dizaines de milliers de tests)
- une certification par version de JavaEE

Caractéristiques

- Composée de :
 - conteneur(s) de composants de type EJB
 - conteneur(s) de composants Web (Servlet)
 - ...
- Des connecteurs ou/et des dépendances entre composants
- L'ensemble archivé dans une **Enterprise Application Archive (EAR)**
- Un serveur d'applications déployant et gérant le tout

