Menedzsmenttudatos alkalmazások: Java Management Extensions

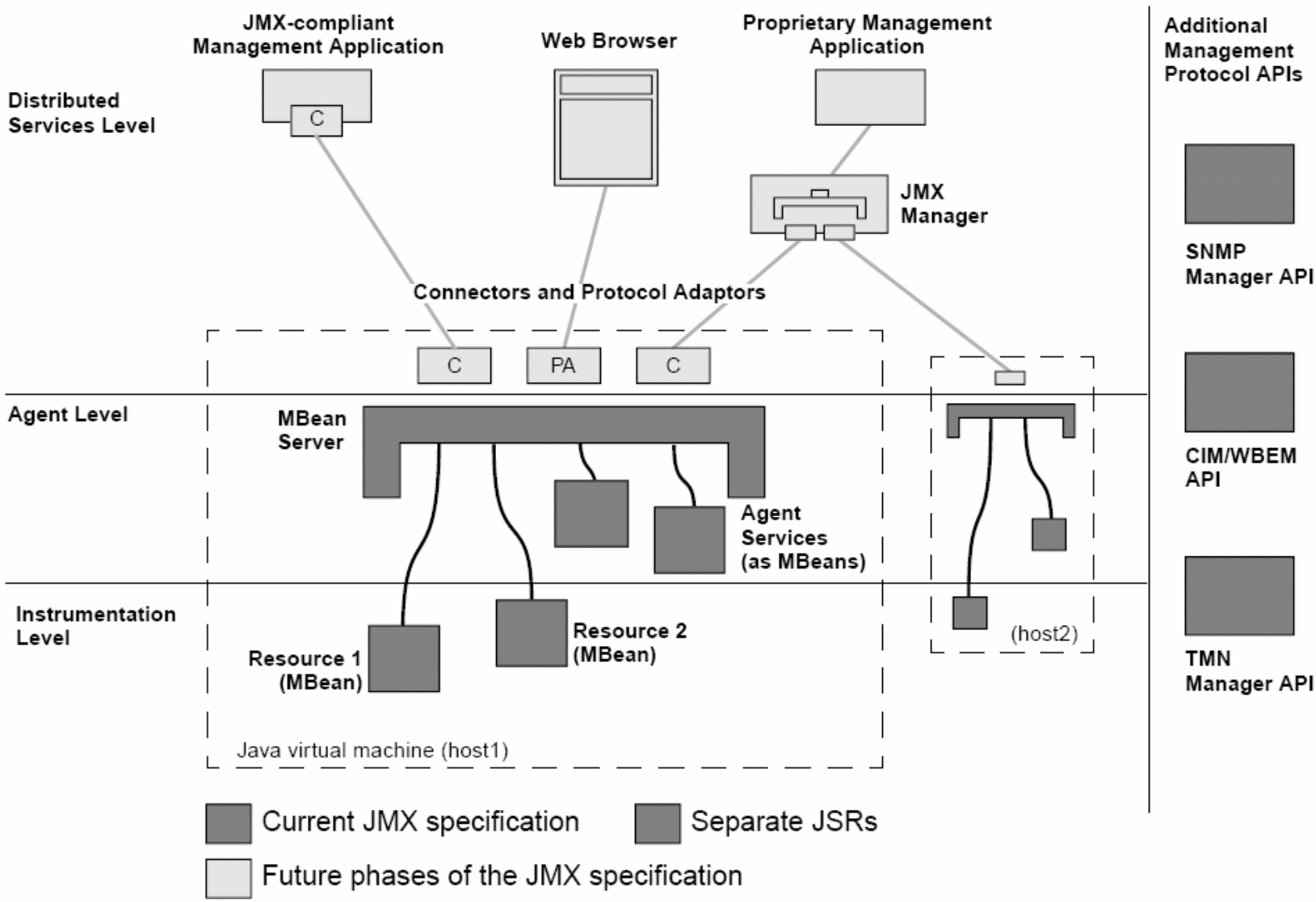
Paller Gábor

2005. nov. 18

Menedzsmenttudatos alkalmazások/JMX

- Alkalmazások csak akkor menedzselhetők hatékonyan, ha fel vannak készítve menedzsmentre
- Pl. az alkalmazás hozzáférhetővé teszi fontos belső indikátorait a menedzsmentrendszer számára, lehetővé teszi saját maga átkonfigurálását vagy tud üzeneteket küldeni hiba esetén a menedzsmentrendszer részére.
- Ez szükségessé teszi, hogy az alkalmazások és a menedzsmentrendszer között alkalmazásszintű interfész (API) legyen.
- A JMX (Java Management Extensions) a legelterjedtebb ilyen szabvány Jávára
- A J2EE-nek mindig része volt, a J2SE 5.0-tól a sztenderd (desktop) Jávának is része.
- A JMX azt mondja meg, hogyan beszélget a Jáva program a menedzsment ágenssel (rajta keresztül a menedzsment rendszerrel). Nem foglalkozik a menedzsmentrendszer és az ágens kapcsolatával.

Architektúra



Architektúra (magyarázat)

- MBean server: Jáva menedzsment szempontjából ő az ágens (további, nem Jávás rétegei lehetnek, pl. egy teljes CIMOM …)
- MBean-ek: az alkalmazás által menedzsmentre kiajánlott erőforrásokat jelképezik
- Connector-ok: protokoll adapterek (pl. SNMP)
- A JMX (JSR3) csak az MBean-ek és az MBean server kapcsolatával foglalkozik
- A JSR 160 (JMX Remoting) írja le a Connector-ok és az MBean server kapcsolatát

MBean-ek

- Jáva osztályok kötött interfésszel
- 4 fajtájuk van
  - Standard MBean: könnyű írni, képességei kötöttek
  - Dynamic MBean: viselkedését változtathatja a körülményeknek megfelelően
  - Open MBean: Dynamic MBean+metainformációk, amelyek alapján a menedzsmentrendszer előzetes ismeret nélkül is felderítheti és használhatja
  - Model MBean: Dynamic MBean, amihez szükség szerint lehet implementációs modulokat illeszteni

Standard MBean

- Kötött műveletek, kötött attribútumok
- A menedzsmentműveleteket a konténer Jáva hívásokká képezi le
- Az MBean által implementálandó metódusfejléceket tartalmazza a specifikáció
- Meg van szabva:
  - A konstruktorok formátuma
  - Hogy az attribútumokat getter/setter metódusokon keresztül kell elérni
  - Hogy a nem getter/setter metódusok hívható menedzsmentoperációkként látszanak a menedzsmentinterfészen
  - A mód, ahogy az MBean értesítéseket küldhet

Standard MBean példa

- Kell egy interfész:

```
public interface MyClassMBean {
  public Integer getState();
  public void setState(Integer s);
  public void reset();
}
```

- És kell egy implementáció:

```
public class MyClass implements MyClassMBean {
  private Integer state = null;
  private String hidden = null;

  public Integer getState() {
    return(state);
  }
  public void setState(Integer s) {
   state = s;
  }
  public String getHidden() {
    return(hidden);
  }
  public void setHidden(String h) {
    hidden = h;
  }
  public void reset() {
    state = null;
    hidden = null;
  }
}
```

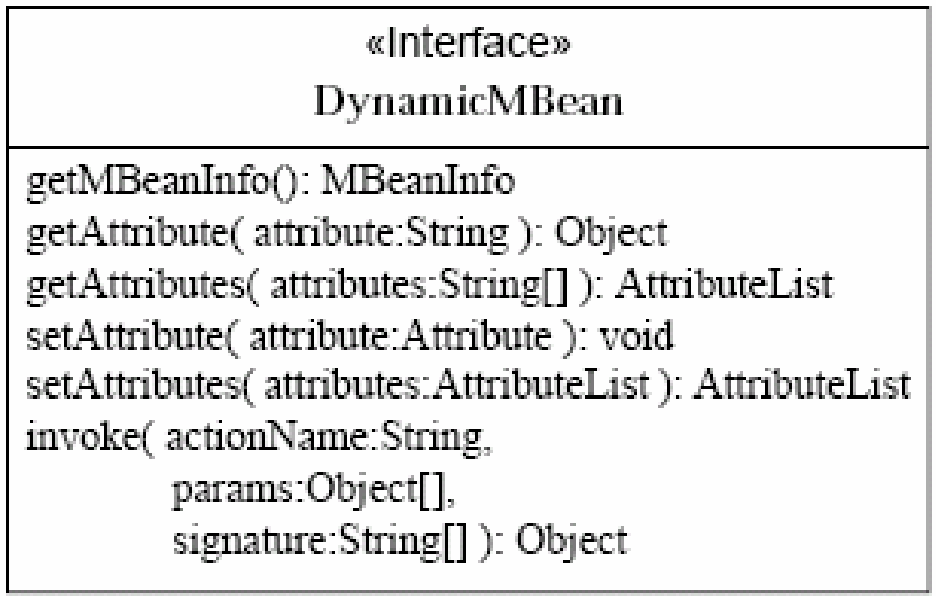
State nevű állapotváltozó getter és setter metódusai

Ezek a menedzsmentinterfészen hívható metódusok, mert nincsenek feltűntetve az MyClassMBean interfészen

Menedzsmentinterfészen hívható metódus

Dinamikus MBean

- Általános, minden metódus és attribútum elérésére alkalmas interfész, minden dinamikus MBean-nél ugyanaz
- Leírja önmagát egy táblával.



Notification

- Ha az MBean implementálja a getNotificationInfo() metódust, az leírja, milyen eseményeket tud küldeni a bean.
- Implementálja az NotificationBroadcaster vagy NotificationEmitter interfészeket. Ezek a következő szolgáltatásokat nyújtják:
  - addNotificationListener: közli a bean-nel, hogy egy újabb érdeklődő van a notification-jeire
  - removeNotificationListener: elvesz egy érdeklődőt
- Szokásos integrálás az ágenssel:
  - Az ágens meghívja a getNotificationInfo-t és publikálja az eseményeket (pl. trap-ek) a menedzsmenttechnológiának megfelelő módon (pl. mint CIM osztályokat)
  - Regisztrálja önmagát a bean-nél addNotificationListener hívással
  - Ha a bean eseményt akar küldeni, meghívja az addNotificationListener-rel regisztrált objektumot – ez a hívás az ágensbe jut
  - Az ágens elküldi az eseményt (pl. CIM event vagy SNMP trap)