

Java EE の “ 抽象化 ” は SOA へとつなげるために

- Java SE
 - Java SE 6 となるコードネーム Mustung については、約 1 年間毎週のスナップショットを公開し続けておりさまざまなスペックが盛り込まれている
 - JSR-199 や Web サービス など世代を反映するものばかり
 - すべてのリリースで課題となっているのは “ 互換性 ”
- Java EE
 - 高度化することで開発者が生産性を上げることが難しくなってきた
 - EoD (Ease of Development) に関わる POJO ベースプログラミング手法の採用、アノテーション、リソース・インジェクション、新たな API として取り上げられることの多い Persistence API が大きなポイント
 - JSP (JSR-52)、StAX (JSR-173)、Web サービスメタデータ (JSR-181)、Persistence API (JSR-220)、JAXB (JSR-222)、JAX-WS (JSR-224)、アノテーション (JSR-250)、JSF (JSR-252) などさまざまなスペックが追加
 - 複雑化すると同時に抽象化を行うべく扱いの容易さを追加
 - 今後、SOA 実現のために相互運用性が問われてくれば、デベロッパーは EJB だけの知識では収束できなくなってくることが予想
 - SOA に関わる開発者は、いくつかの技術を組み合わせる必要性が必須になると言及し、Java (EJB など) テクノロジーだけではなく XSLT や SQL、BPEL、Rule、XQuery など]
 - これらの連携した統合開発環境 (IDE) が求められるはず