

## Distribuované aplikace a technologie pro jejich vývoj

# Programová stavebnice

**Přinášíme dokončení příspěvku o softwarových technologiích , jež v současnosti pomáhají vytvářet distribuované aplikace schopné provozu v heterogenním prostředí.**

Za řadu produktů nekomerčních jmenujme kupříkladu ORB Orbacus firmy OOC, se kterou jsou rovněž pozitivní zkušenosti (avšak implementace některých významných rysů CORBA 2.3 je podobně jako u většiny nekomerčních ORB doposud ve stavu beta verze). Implementace ORB jednotlivých výrobců se liší zejména verzí standardu CORBA, jejíž požadavky splňují, rozsahem dodávaných podpůrných služeb z kategorie CORBA Services, sadou podporovaných programovacích jazyků a platform, na nichž může být produkt provozován. Implementace všech rysů požadovaných specifikací CORBA 2.3, popř. jednotlivých CORBA Services však rozhodně není snadnou záležitostí, proto si je dobré uvědomit, že na první pohled vysoké ceny této kategorie produktů nejsou s ohledem na náročnost jejich vývoje nikterak nadhodnocené.

Současná verze standardu CORBA 2.3 (březen 2000), podporující např. velmi důležité řízení životního cyklu objektů, interceptory požadavků a jiné rysy staví CORBA do pozice efektivního a prakticky velmi dobře použitelného prostředku tvorby distribuovaných aplikací. Vytvářené systémy již dnes běžně zahrnují nejen počítače, ale kupříkladu i spotřební elektroniku nebo měřicí přístroje. CORBA se tak stává univerzálním prostředkem spolupráce softwarů různých elektronických systémů a její implementace se již pomalu začínají objevovat např. v prvcích síťové infrastruktury (směrovačů), v zařízeních průmyslové automatizace nebo v domácích spotřebičích.

## Technologie Java

Dalším výrazným trendem, který můžeme v dnešních distribuovaných softwarových systémech pozorovat, je stále výraznější přechod na jazyk Java. Jazyk Java je interpretovaný, tedy přenositelný, bezpečný, s přirozenou podporou práce v síti a dobře se hodí na tvorbu distribuovaných aplikací. Je založen na technologii objektově orientovaného programování a syntakticky vychází z jazyka C++, což usnadňuje přechod na něj programátorům, kteří povětšinou ještě donedávna používali jazyk C++.

Je důležité poznamenat, že Java není pouhý programovací jazyk, ale celá nová technologie, umožňující vytvářet aplikace spustitelné na jakémkoli počítači a operačním systému, pro který máme k dispozici virtuální stroj (interpreter) Javy. Běžnou součástí těchto aplikací je možnost dynamického zavádění kódu ze sítě a možnost práce s kódem, o jehož existenci nebylo v době vzniku aplikace nic známo. Brzy po vzniku Javy se rozšířila i technologie transparentního používání distribuovaných objektů RMI (Remote Method Invocation). V současné době Java běžně podporuje i vývoj aplikací složených z komponent (s využitím technologie Enterprise JavaBeans) a samozřejmě je i integrace se standardem CORBA.

S jazykem Java se dnes běžně setkáváme v prostředí WWW, kde technologie appletů dovoluje dynamicky rozšiřovat možnosti WWW prohlížečů formou speciálních programů v Javě, běžících v prostředí samotného WWW prohlížeče. Applety dnes již dávno nejsou jen pomůckou pro zatraktivnění vzhledu WWW stránek animacemi a přehráváním zvuku, ale samozřejmě součástí vícevrstvé architektury aplikací, která navíc minimalizuje náklady na instalaci a správu klientského softwaru dynamickým zaváděním programů přímo ze sítě. Takovýto způsob navíc umožňuje prodejčům softwaru jednoduché poskytování upgradu, snadnou správu licencí a prodej jednotlivých softwarových modulů v reálném čase za použití prostředků elektronického obchodování. Dá se uvažovat i o pronájmu příslušných softwarových modulů na určitou dobu.

Na straně WWW serverů se s Javou setkáváme ve formě technologií servletů (komponenty rozšiřující funkčnost WWW serverů podobně jako CGI) a JSP (Java Server Pages), umožňující dynamicky integrovat informace získávané z distribuovaných komponent do WWW stránek.

Často používaný argument odpůrců Javy o její pomalosti čím dál více ustupuje do pozadí s nástupem JIT (Just-in-Time) překladačů, které za běhu dynamicky překládají instrukce programu v Javě do strojového kódu příslušné platformy. Někteří výrobci jdou i dál a provedou překlad programu v Javě do strojového kódu ještě před jeho prvním spuštěním (obvykle s použitím jazyka C jako mezikódu a optimalizujícího překladače C, který je pro danou platformu obvykle k dispozici u samotného uživatele). Takovýto způsob se používá zejména u programů běžících na straně serveru, kde mírné pozdržení při startu programu není na závadu a obsluha jednotlivých požadavků klientů již probíhá rychlostí odpovídající běžným kompilovaným programům. Žádné z těchto řešení

však není na úkor přenositelnosti programů, jelikož instrukce virtuálního stroje a formát souboru obsahující program v Javě jsou od první verze Javy stále stejné, dobře zdokumentované, a tudíž přenositelné. V brzké době můžeme rovněž očekávat nástup speciálních procesorů (picoJava, MAJC), které budou instrukce Javy vykonávat přímo na úrovni hardwaru.

Nikoli bez zajímavosti je skutečnost, že základní (řádkové verze) vývojové prostředky pro technologii Java poskytuje její tvůrce firma Sun zdarma. Pro profesionální práci je možné zakoupit řadu vývojových prostředí, např. JBuilder firmy Inprise nebo VisualAge for Java od IBM.

### Konkurence, nebo spolupráce ?

Plejáda technologií prosazujících se v současné době při tvorbě distribuovaných aplikací může na první pohled budit dojem zmatku a vyvolávat úvahy o nutnosti strategického rozhodnutí, na kterou z nich se orientovat. Ve skutečnosti však otázka takto postavena vůbec není – většina zmíněných technologií nemá za cíl si konkurovat, ale navzájem se doplňovat. Orientace na binární kompatibilitu a úzká návaznost na jedinou platformu, které mnozí dnešní vývojáři často podléhají, se tedy v kontextu očekávání masového nasazování technologií pro heterogenní prostředí může jevit jako krátkozraká.

*Petr Grygárek, [www.cs.vsb.cz/grygarek](http://www.cs.vsb.cz/grygarek)*