

Na prvom riadku sme použili direktívu `taglib` na importovanie našej knižnice tagov, ktorej tagy budeme používať s prefixom `tw`. To, aby tá bola k dispozícii práve pod URI `infoware` (ktorú sme použili), musíme zabezpečiť neskôr v nastavení aplikácie v súbore `web.xml` (o chvíľu si ukážeme, ako na to). V tele stránky potom jednoducho používame nami implementovaný vlastný tag `date` s určeným prefixom `tw`.

Teraz už máme pripravené všetky komponenty na vytvorenie prvej webovej aplikácie používajúcej tagové rozšírenia. Najprv musíme štandardným spôsobom vytvoriť enterprise aplikáciu a do nej pridať webovú aplikáciu, ako sme to už robili v predchádzajúcich častiach veľakrát. Následne tejto novej webovej aplikácii nastavíme koreňový kontext (napr. `tagext`) a vložíme pripravené komponenty. Po správnom postupe by obsah takejto webovej aplikácie mal vyzerať podobne ako na obrázku 1. Posledným krokom k skompletizovaniu našej webovej aplikácie je nastavenie URI knižnice tagov, pod ktorou bude prístupná (teda v podstate URI, pod ktorým bude prístupný nami definovaný deskriptor knižnice tagov). Toto nastavenie treba urobiť v spomínanom súbore `web.xml` našej webovej aplikácie, ale nie je nevyhnutné zasahovať doň priamo, pretože túto možnosť už aplikácia `Deploytool` poskytuje. Stačí vybrať záložku `File Ref's` a na nej kliknúť na tlačidlo `Add Tag Library`, čím pridáme jeden riadok do tabuľky `JSP Tag Libraries`. V tomto riadku stačí nastaviť URI na „`infoware`“ a `Location` na cestu k súboru deskriptora, teda na „`./WEB-INF/tds/infoware.tld`“, ako možno vidieť aj na obrázku 2. Teraz je už aplikácia kompletná, takže ju stačí uložiť a umiestniť na server (funkcia `Deploy`), čo je už opäť nám známy a v tomto seriáli mnohokrát opakovaný krok. Po zadaní adresy `http://localhost:8080/tagext/hello.jsp` (pokiaľ máte nastavený aplikačný server tak, aby akceptoval požiadavky klientov na porte 8080) by ste mali vidieť výsledok našej snahy, podobný tomu na obrázku 3. Na záver ešte podotknem, že všetky uvedené zdrojové kódy by ste mali nájsť na CD `Revue`, prípadne na našej webovej stránke.

**Záver**

V tejto časti ste sa mohli prvýkrát oboznámiť s tagovými rozšíreniami. Na jednoduchom príklade ste si mohli overiť, že pridanie a implementovanie vlast-



Obr. 3

ného tagu do vašej webovej aplikácie je veľmi rýchle a jednoduché a že tagové rozšírenia významne odľahčujú vrstvu `view` architektúry `MVC`, teda stránky `JSP`. Umožňujú presunúť všetkého `Java` kódu z nich do tried, čím umožňujú tvorbu stránok `JSP` aj takým dizajnérom, ktorí nemajú o implementácii jednotlivých tagov ani potuchy a poznajú len ich funkciu. Takýto dizajner sa potom môže sústrediť na dizajn stránky bez toho, aby sa zatažoval nejakou aplikačnou logikou, dokonca nemusí ani vedieť programovať v `Java`. Ďalšou potenciálnou výhodou tagov, ktorú ste si možno všimli, je opätovná použiteľnosť – pokiaľ je aplikačná logika zapuzdrená v obslužných triedach tagov, možno ju využiť aj v iných stránkach, resp. aplikáciách bez nutnosti akéhokoľvek zásahu do kódu. Keby bola táto aplikačná logika vložená do stránok `JSP`, jej opätovné použitie by bolo oveľa komplikovanejšie.

Tagové rozšírenia (alebo ak chcete vlastné tagy) prinášajú teda nesporné výhody do sveta technológie `JSP` a bolo by veľkou chybou ich nevyužívať. Preto sa im budeme naďalej podrobne venovať aj v najbližších častiach tohto seriálu. Zatiaľ vám prajem príjemné prežitie letných dní.

Andrej Chu

**Technológie InterSystems (IV.)**

Databázové aplikácie bývajú základom väčších celkov. Typickým príkladom sú podnikové aplikácie. V takom prípade však treba zabezpečiť vzájomnú kompatibilitu jednotlivých stavebných prvkov celej podnikovej informačnej architektúry. Pomôcť nám v tom môže integračná platforma `Ensemble`, uvedená spoločnosťou `InterSystems` na trh v novembri minulého roku.

Typickým problémom ľubovoľného integračného procesu je predovšetkým technologická a časová náročnosť. Hlavnou výhodou platformy `Ensemble` je okrem iného to, že umožňuje tento problém minimalizovať. Nielen vďaka jednotlivým vlastnostiam, ale aj vďaka komplexnému pohľadu na podnik a celú problematiku integrácie.

Pre platformu `Ensemble` sú typické predovšetkým štyri nasledujúce vlastnosti. Každá z nich tvorí základ úspešnej integrácie, no samostatne by v drvivej väčšine prípadov jednotlivé vlastnosti na úspešnú integráciu nemohli stačiť.

Základom celého prostredia je podpora *komplexnej integrácie a vývoja*. `Ensemble` ponúka jednotné grafické prostredie na vývoj na báze technológie `XML`. Vždy možno plnohodnotne využiť už existujúce aplikácie a dáta.

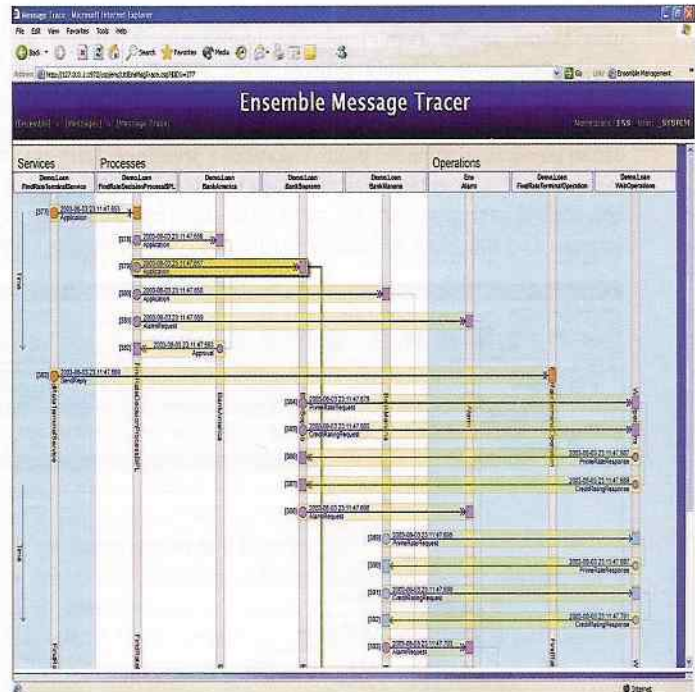
Druhým stavebným kameňom `Ensemble` je *univerzálna architektúra služieb*, ktorá ponúka kon-

zistentnú a efektívnu objektovú reprezentáciu nesúrodých programových modelov a dátových formátov. Vďaka tomu možno pristupovať k už existujúcim prvkom napríklad ako k opakovane použiteľným komponentom `.NET` či `J2EE`.

Úspech `Ensemble` stojí takisto na *perzistentnom objektovom stroji*, teda nám už dobre známej databázovej platforme `Caché`. V rámci `Ensemble` slúži `Caché` na ukladanie a správu všetkých metadát, správ a stavových informácií o jednotlivých procesoch. Vďaka `Caché` získavajú organizácie všetky výhody objektového prístupu k živým aj archívnym správam.

Štvrticu vlastností `Ensemble` uzatvára *prispôsobiteľná správa celého systému*, ktorá okrem iného ponúka rýchlu diagnostiku vrátane odstránenia problému, automatické sledovanie kritických zdrojov a vyhľadávanie úzkych hrdiel alebo prispôbovanie chybových hlásení, čím sa uľahčuje ich ďalšie spracovanie v nástrojoch už využívaných spoločnosťou.

Platforma `Ensemble` je k dispozícii pre operačné platformy `HP Alpha OpenVMS`, `HP Alpha Tru64`, `Unix`, `HP-UX`, `IBM AIX`, `Linux`, `Sun Solaris` a `Micro-`



soft `Windows`. Cena licencie systému `Ensemble Enterprise` sa začína na 125 000 USD za jeden procesor.

Nabudúce sa pozrieme na podporu technológie `XML` v `Caché`.

Marek Kocan