

Sulautettujen Java-järjestelmien kehitysnäkymät

Java on vielä lupaus

Sun Microsystemsin kehittämä Java-ohjelmointikieli on ollut merkittävän mielenkiinnon kohteena viime vuodet. Java on kehittynyt täydelliseksi ohjelmointiparadigmaksi ja se on lupaava uusi teknologia erityisesti verkotettuihin sulautettuihin järjestelmiin. Erityisen mielenkiinnon kohteena on myös Javan käyttäminen tietoliikenteen sovelluksissa sulautetuista päätelaitteista verkonhallinta- ja palveluohjelmistoihin.

Sulautetuille järjestelmille ominaisia piirteitä ovat tiukat vasteaikavaatimukset, käytössä olevan muistin pienuus, rinnakkaisuuspiirteet sekä se, että ne sisältyvät laitteeseen, jonka toimintojen ohjauksesta ne vastaavat. Tyypillisiä sulautettuja järjestelmiä sisältäviä laitteita ovat nykyisin prosessinohjauslaitteet, mikroprosessorin sisältävät kodinkoneet, lääketieteelliset instrumentit ja kaikille tutut matkapuhelimet.

Javan synty

Java syntyi osana projektia, jossa pyrittiin kehittämään nimenomaan sulautettuihin järjestelmiin sopivaa ohjelmointitekniologiaa. Ensimmäinen julkinen Java-toimitus (ns. Java Developers Kit ja siihen liittyvä dokumentaatio) tapahtui vuonna 1995. Sen jälkeen Javan saama huomio ja sen hyväksyntä erilaisissa yhteisöissä on ollut yllättävän suurta. Vuonna 1997 Sunin anomus toimittaa Java teknologia kansainväliselle standardointijärjestölle ISO:lle hyväksyttiin äänestyksessä (Suomi äänesti tämän ns. PAS-toimituksen puolesta).

Sulautettujen järjestelmien tulevaisuudessa on nähtävissä muutamia selvää kehityssuuntia. Ne monimutkaistuvat, tunkeutuvat mitä erilaisimpiin tuotteisiin (älykkäät tuotteet) ja verkottuvat kommunikoimaan toistensa kanssa. Mikroprosessoria sisältävien tuotteiden määrä lisääntyy räjähdysmäisesti. Esimerkiksi Motorola arvioi, että vuonna 2000 uudet autot sisältävät noin 35 mikroprosessoria.

Kovin kaukana ei ole aika, jolloin jääkaappisi kykenee il-

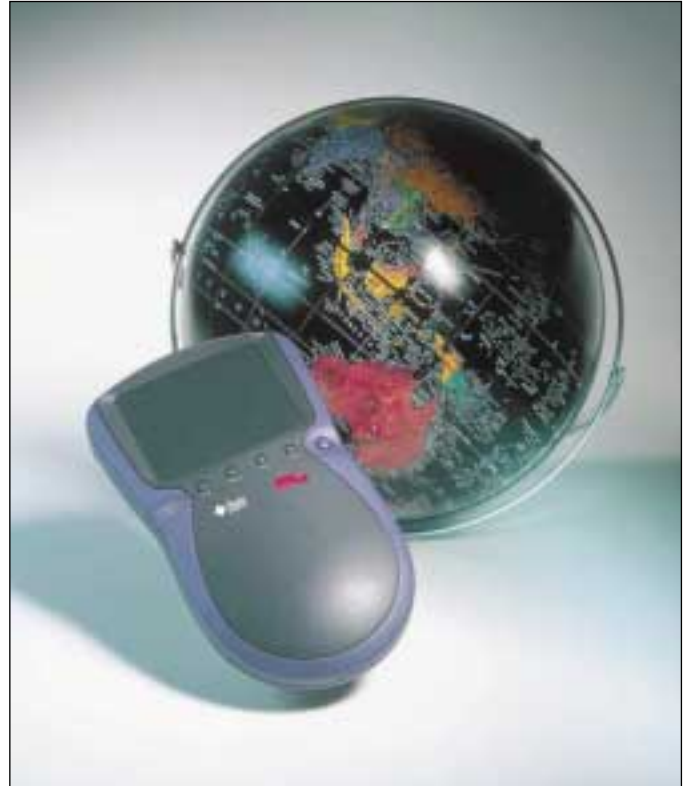
moittamaan kompressoririkostaan autosi Internet-päätteeseen. Sulautettujen järjestelmien verkottumiselle on erityinen tarve etenkin automaatio- ja prosessinohjausjärjestelmissä, joissa on jo pitkään pyritty tehostamaan ohjausta kytkemällä yksittäiset mikrokontrollerit kommunikoimaan keskenään.

Verkkoon kytketyille sulautetuille järjestelmille on toteutettavissa yksi erittäin mielenkiintoinen ominaisuus. Perinteisissä sulautetuissa järjestelmissä ohjelmisto on poltettu ROM-muistiin ja sen päivittäminen on edellyttänyt muistipiirin vaihtoa. Verkotetuissa järjestelmissä voidaan ohjelmisto tai sen osia päivittää verkon kautta. Täten tulevaisuuden kodissa pyykkipesukone voi esimerkiksi ladata verkosta uusia ohjelmia ja ominaisuuksia.

Julkistamisestaan lähtien Java on saanut ohjelmointikieliksi poikkeuksellisen suurta huomiota osakseen. Tämä on näkynyt vaikkapa Javaa käsittelevien kirjojen valtavassa tarjonnassa tai Java-ohjelmistokehittäjien määrän ennätyksellisen nopeassa kasvussa. Javaa on kaupattu



Java soveltuu hyvin langattomiin päätelaitteisiin



Kuva: Sun Microsystems

yleiskäyttöisenä ohjelmointikielenä www-sisältötuotantoon, työasemasovelluksiin ja yritysten tietojenhallintajärjestelmien toteutukseen.

Java suunniteltiin alunperin kieleksi, jolla voitiin tehdä sovelluksia hyvin pienimuistisiin sulautettuihin ympäristöihin kuten leivänpaahtimiin. Nyt kun Javan kehittäjä Sun on voimakkaasti puskemassa Javaa sulautettuihin järjestelmiin, voidaan sanoa, että Java on palaamassa viimeinkin juurilleen. Kuinka Java sitten soveltuu käytettäväksi sulautetuissa järjestelmissä? Millaisia ongelmia suunnittelijat joutuvat toistaiseksi kohtaamaan, ja toisaalta, mitä hyötyä on Javan ominaispiirteistä, kuten sisäänrakennetusta verkotuksesta, sulautettujen järjestelmien sovellusalueella?

Java-teknologia

Javan edustamassa teknologiassa ei ole oikeastaan mitään sellaista yksittäistä piirrettä, jota ei olisi ollut jo olemassa jossain

toisessa ohjelmointikielissä -- vanhimmat Javan piirteet periytyvät jo LISP-kielystä, joka kehitettiin 1960-luvulla. Javan kiehtovuus piilee siinä, että siinä yhdistyy joukko ohjelmistokehitystä tehostavia ominaisuuksia: oliokeskeisyys, syntaksin yksinkertaisuus, virtuaalikoneeseen perustuva suoritustapa, rinnakkaisuuden tuki, automaattinen muistinhallinta, turvaominaisuudet sekä standardoidut luokkakirjastot.

Javan suunnittelijoiden neronleimaus oli ottaa käyttöön tämän hetken suosituimman oliokeskeisen kielen C++:n syntaksi tietyin parannuksin, jolloin nykyisten C/C++ ohjelmistokehittäjien kynnys omaksua Java on varsin matala. Java-kieltä voisikin luonnehtia parannetuksi C++ -kieleksi, josta on poistettu joukko sekaannusta ja ohjelmointivirheitä helposti aiheuttavia piirteitä.

Javan ympäristöriippumattomuus perustuu suoritinriippumattomaan tavukoodiin (byte

code), jota ajetaan annetun spesifikaation mukaisessa virtuaalikoneessa. Kukin Java-luokka kirjoitetaan omaan tiedostoonsa. Tavukoodi tuotetaan kääntämällä nämä lähdekooditiedostot Java-kääntäjällä.

Sivun 91 kuvassa on esitetty Javan suoritukseen liittyvät eri vaiheet sekä eri tavat suorittaa Javaa. Java-virtuaalikoneen voidaan ajatella olevan eräänlainen ohjelmistolla toteutettu virtuaalinen prosessori. Javalla kirjoitettuja sovelluksia voidaan suorittaa kaikissa niissä käyttöjärjestelmäympäristöissä, jotka tarjoavat spesifikaation mukaisen Java-suoritusympäristön (Java Application Environment, JAE). Java-suoritusympäristö koostuu virtuaalikoneesta ja standardista sovellusrajapinnasta (Java Application Programming Interface, Java API).

Java tarjoaa uuden lähestymistavan sovellusten siirrettävyyteen. Perinteisesti sovellusten siirrettävyys on pyrittä saavuttamaan lähdekoodin muodossa. Käytännössä tämä ei ole onnistunut kielistandardeista huolimatta. Käännetty koodi joudutaan linkkaamaan kirjastoihin, jotka ovat järjestelmäriippuvaisia ja rajapinnoiltaan

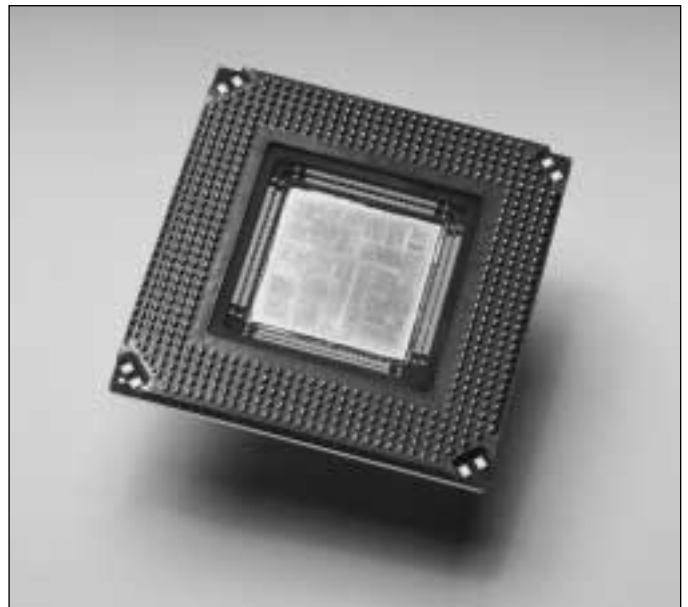
erilaisia. Käännöksen ja linkkauksen jälkeen sovellus voidaan suorittaa ainoastaan siinä käyttöjärjestelmäympäristössä, jossa se tuotettiin.

Javassa siirrettävyys perustuu suoritettavaan koodiin eli tavukoodiin. Rinnakkaisuuden toteutus on Javassa onnistuttu standardoimaan kielen sisäänrakennetuilla mekanismeilla ja standardoituun Java API:in kuuluvilla luokilla toisin kuin esimerkiksi C++ kielessä.

Java sulautetuissa järjestelmissä

Sun Microsystemsin tavaramerkki "Write Once, Run Everywhere" ei ole tarkoitettu merkitsemään pelkästään sitä, että tehtyä Java-sovellusta voidaan suorittaa käyttöjärjestelmästä riippumatta jokaisessa työasemassa, joka tarjoaa standardin Java-sovellusympäristön. Sovellusten tulee toimia myös eritasoisissa ympäristöissä, joiden rajoituinta ääripäätä edustaa älykortti ja tehokkainta supertietokone.

Java-sovelluksille on määriteltä neljä eritasoista ympäristöä: "standard" Java, PersonalJava, EmbeddedJava ja JavaCard, joista ensin mainittu on



Sun Microelectronicsilta on tulossa useita eritasoisia Java-suorittimia.

laajin ja edustaa lähinnä työasemaympäristöjä. Kukin sovellusympäristö tarjoaa virtuaalikoneen ja määritellyn osan standardia Java API:a. Sovellusympäristöt ovat ylöspäin yhteensopivia, mikä tarkoittaa esimerkiksi sitä, että JavaCard-ympäristöön tehtyä sovellusta voidaan suorittaa sellaisenaan PersonalJavaa käyttävässä matkapuhelimessa.

PersonalJava ja EmbeddedJava ovat sulautettuja järjestelmiä varten suunniteltuja suppeampia sovellusympäristöjä, jotka mahdollistavat Javan suorittamisen myös rajoitetuissa ympäristöissä. PersonalJava on tarkoitettu laitteisiin, joissa on kohtuullisen tarkka grafiikkänäyttö, kuten älypuhelimissa, kämmentietokoneissa tai TV-kotipäätteissä.

EmbeddedJava on puolestaan tarkoitettu käytettäväksi laitteissa, joissa ei ole näyttöä lainkaan tai se on hyvin pieni, yleensä merkkipohjainen näyttö. Tällaisia tuotteita ovat matkapuhelimet, kodinkoneet ja hakulaitteet. JavaCard on tarkoitettu laitteille, joille on tyypillistä hyvin rajallinen muisti, esimerkiksi älykortteille.

PersonalJavan ensimmäinen versio 1.0 on valmis ja lisensioitavissa Sunilta. EmbeddedJavan spesifiointi on vielä työn alla ja se tullaan julkaisemaan tämän vuoden kuluessa.

PersonalJava asettaa seuraavat vähimmäisvaatimukset käytettävälle laitteelle: 32-bittinen suoritin, ROM-muistia kaksi megatavua, RAM-muistia 0,5-1 megatavua, verkkoyhteys ja reaaliaikakäyttöjärjestelmä.

Kuten vaatimuksista havaitaan PersonalJava ei sellaise-

naan sovellu useisiin sulautettuihin laitteisiin, esimerkiksi perinteisessä GSM-matkapuhelimessa suoritin- ja muistivaatimukset ovat liian suuria.

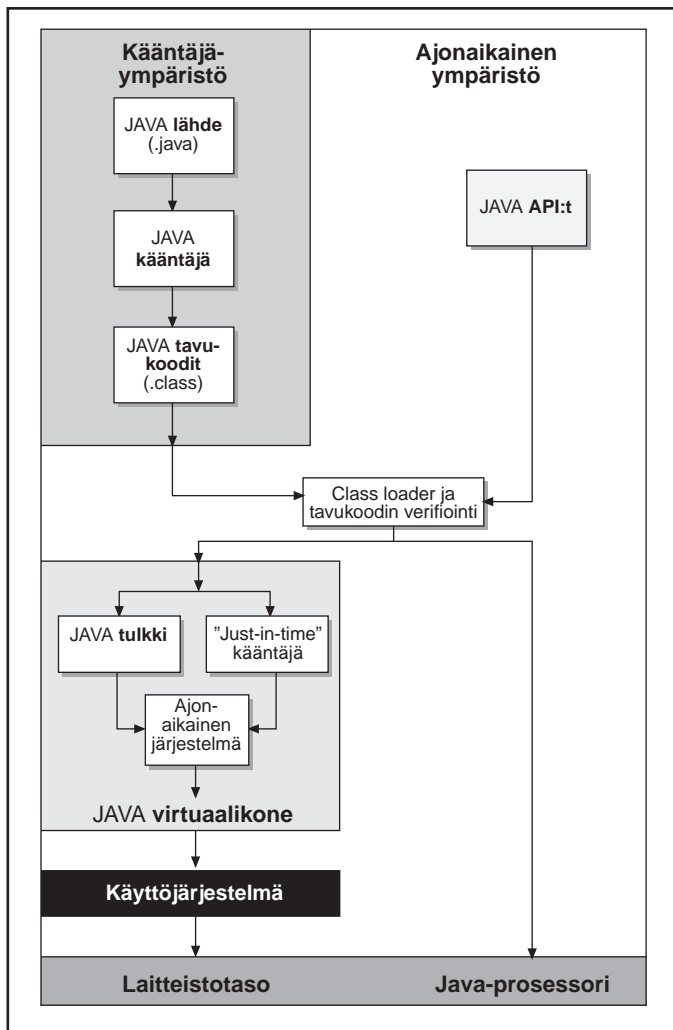
Joukko johtavia reaaliaikakäyttöjärjestelmien toimittajia, kuten Chorus, Lucent Technologies, QNX ja WindRiver Systems ovatkin lisensioineet PersonalJava- ja EmbeddedJava-ympäristöt ja tulevat toimittamaan niitä omiin käyttöjärjestelmiinsä liitettynä.

Eri reaaliaikakäyttöjärjestelmiin integroitu PersonalJava toteuttaa näiden välille standardin Java-sovellusympäristön, mikä synnyttää Javalla kirjoitetuille sulautetuille sovelluksille laajat markkinat.

Älykorteissa vahvoilla JavaCard-standardi määrittelee sekä älykorttiin toteutettavissa olevan Java-virtuaalikoneen että Java API:n, joiden avulla älykortissa voidaan suorittaa pieniä Java-sovelluksia. Standardin uusin versio 2.0 määrittelee eri muistityyppien vähimmäismääriksi: 16 kilotavua ROM-, 8 kilotavua EEPROM- ja 256 tavua RAM-muistia.

Älykortit edustavat sovellus- aluetta, jolla Java on yllättävän vahvasti lyönyt läpi: noin 95 prosenttia valmistajista mukaan lukien huomattavimmat nimet, kuten Gemplus, Motorola ja Schlumberger ovat sitoutuneet JavaCard-standardin käyttöön tuotteissaan.

Syyt suosioon ovat selvät. Standardi kehitettiin yhdessä johtavien valmistajien kanssa. Standardoitu suoritusympäristö mahdollistaa sen, että yhden valmistajan kortille kirjoitettu Java-sovellus toimii myös mui-



Kuva 1. Javan eri suoritustavat.

den valmistajien korteissa. Java-sovellusten pieni koko mahdollistaa useampien sovellusten suorittamisen yksittäisellä kortilla. Lisäksi älykorttisovellusten kehitystyö tehostuu, kun konekielen sijasta voidaan käyttää korkeamman tason Javaa.

Älykorttituotteille on ennustettu valtaisia markkinoita -- erään ennusteen mukaan maailmassa olisi vuoteen 2000 mennessä käytössä yli kolme miljardia älykorttia. Esimerkiksi maailman suurin kuluttajia palveleva luottokorttiyhtiö Visa International aikoo ottaa käyttöön älykortteissaan JavaCard-standardiin perustuvan suoritekniologian. Yhtiön tavoitteena on löytää älykorkeille uusia sovellusalueita esimerkiksi matkapuhelinverkoissa. JavaCard-tuotteet soveltuvat myös hyvin erilaisten sulautettujen laitteiden konfigurointiominaisuuksien toteuttamiseen. Tämä on eräs tulevaisuuden tuotekehitysalue, johon VTT Elektroniikka on panostamassa.

Ongelmia ...

Javan käyttö sulautetuissa jär-

jestelmissä ei ole vielä täysin ongelmatonta, keskeisimmät ongelmat liittyvät varsin suureen muistin tarpeeseen, suorituskykyyn, reaaliaikapiirteisiin, laiteistorajapintaan ja puutteellisiin kehitysyökaluihin.

Java-virtuaalikone suunniteltiin ja se soveltuu parhaiten 32-bittiseen suoritinympäristöön, joiden käyttöön sulautetuissa järjestelmissä ollaan vasta siirtymässä (4 ja 8 bittiset mikrokontrollerit ovat yhä varsin yleisiä sulautetuissa laitteissa). Virtuaalikone ja Java API vaativat muistia liikaa, jotta useimpia sulautettuja järjestelmiä voitaisiin toteuttaa taloudellisesti.

Toisaalta Javaan liittyy piirteitä, joiden avulla voidaan tehokkaammin hyödyntää käytössä olevaa muistia. Ensinnäkin tavukoodi on huomattavasti kompaktimpaa kuin konekielikoodi. Toiseksi, oliokeskeisyys mahdollistaa modulaariset ja yleiskäyttöiset ohjelmistokomponentit, joita voidaan tavukoodin muodossa dynaamisesti ladata muistiin ja poistaa muistista tarpeen mukaan.

Suorituskyvyssä Java on vielä

hieman jäljessä C/C++ kieltä. Pelkästään tulkittavana kielellä se ei pysty kilpailemaan hyvin optimoitujen C-kääntäjien kanssa. Tosin suorituskyvyn parantamiseen löytyy useita keinoja, kuten native code - tai Just-In-Time (JIT) -kääntäjät. Edelliset kääntävät Java-koodia käytettävän prosessorin konekielelle ja jälkimmäisten kohdalla tekniikkana on kääntää useimmin suoritettavat Java-metodit suoritus-aikaisesti konekieleksi, mikä nostaa useimmissa tapauksissa suoritusnopeuden hyvin lähelle C++-kielen tasoa.

JIT-kääntäjien haittapuolia sulautetuissa järjestelmissä on ikävä kyllä se, että ne lisäävät muistinkulutusta, ja että kääntäjien käyttämien optimointitekniikoiden aiheuttamat viiveet vaikeuttavat reaaliaikavaatimusten varmentamista. Mikään ei myöskään estä kääntämästä Javaa suoraan konekieleksi ja jättämästä virtuaalikonetta pois -- tällöin tosin ympäristöriippumattomuus ja Javan virtuaalikoneen tarjoamat edut menetetään. Lisäksi konekielikäännös ei sovellu tapauksiin, jossa Java-sovelluksia ladataan dynaamisesti suoritettavaksi.

Myös virtuaalikoneet kehittyvät jatkuvasti. Keväällä 1998 Sun julkaisee koodinimeä HotSpot kantavan virtuaalikoneen, jonka väitetään tietyissä tapauksissa jopa ylittävän C++-koodin suoritusnopeuden. Viime kädessä kuitenkin Java-suorittimet, jotka sisältävät sekä Java-virtuaalikoneen että valikoituja sovellusrajapintoja piillä toteutettuina ratkaisevat suorituskykyongelmat lopullisesti. Lisäksi on muistettava, että monissa tapauksissa jo nyt saavutettava suoritusnopeus on riittävä.

Laitteiston ohjaaminen hankalaa

Javan muistinhallinta perustuu automaattiseen, taustalla pyörivään roskienkeruuseen. Roskienkeruun aiheuttama epädeterministisyys koituu ongelmaksi järjestelmissä, joissa on kovat reaaliaikavaatimukset. Lisäksi roskien keruu kuluttaa prosessoriaikaa. Ongelman ratkaisemiseksi on kehitteillä useita tekniikoita, jotka tarjoavat ohjelmoijille paremman kontrollin roskienkeruun ohjaukseen. Esimerkiksi NewMonics, Inc. on kehittänyt pERC (Portable Executive for Reliable Control) nimisen Javan kanssa yhteensopivan virtuaalikoneen, johon on lisätty tuki parille epästandardille piirteelle, joilla roskienkeruu on saatu inkrementaaliseksi ja preemptiiviseksi.

Neljäs keskeinen ongelma Java-kielen käytössä on Java-sovelluksen ja laitteiden välinen rajapinta. Koska Java ei tarjoa muistiosoitimia on laitteiston ohjaaminen Java-koodista käsin hankalaa tai jopa mahdotonta. Tämän ongelman ratkaisemiseksi Javalle on määriteltäviä niin kutsuttu Java Native Interface (JNI) -rajapinta, joka mahdollistaa C/C++-koodin kutsumisen Java-koodista. Kyseistä rajapintaa ei voida kuitenkaan käyttää silloin kun halutaan säilyttää täydellinen käyttöjärjestelmäriippumattomuus.

Sopivien ja riittävän kehittyneiden kehitysyökalujen puute on myös ongelma, joka toistaiseksi hidastaa Javan käyttöönottoa sulautetuissa järjestelmissä. Ongelman poistamiseksi on muun muassa Sun itse kehittämissä koko joukkoa työkaluja, joilla voidaan tukea sulautetun Java-suoritusympäristön sisältäviä tuotteita ja niissä suoritettavia sovelluksia. Työkaluilla tuetaan sulautetun koodin tuottamista, ja testaamista, esimerkiksi muistin käytön suhteen sekä testatun koodin ja tarvittavien resurssien viemistä tuotteeseen.

Joissakin tapauksissa voi olla tarpeellista käyttää C/C++-kieltä Javan rinnalla varsinkin, jos yritys on siirtymässä sulautettujen ohjelmistojen kehitystyössä Java-kielen käyttöön ja huomattava osa olemassa olevasta ohjelmistoista C/C++-koodia. Tätä siirtymävaihetta tukeakseen Sun sisällytti ensimmäiseen kaupalliseen Java-prosessoriinsa, microJava 701, tuen myös C/C++-koodin tehokkaaseen suoritukseen.

Java-koodin ja C/C++-koodin yhdistäminen ei ole täysin ongelmatonta: esimerkiksi käytettävän reaaliaikakäyttöjärjestelmän tulee ottaa kantaa siihen, kuinka jaetaan rinnakkaisuuden hallinta C/C++ ja Java-koodin kesken. Myöskään nykyiset debuggerit eivät kovin hyvin tue yhdistetyn koodin debuggausta.

... ja mahdollisuuksia

Sulautettujen ohjelmistojen kehittäjien näkökulmasta Javan tekevät mielenkiintoiseksi sen oliokeskeiset menetelmät ja komponenttiarkkitehtuurit. Ne sopivat työasemasovelluksia paremmin juuri sulautettuihin sovelluksiin, koska jälkimmäisillä on hoidettavana yleensä varsin rajattu joukko tarkkaan määriteltyjä tehtäviä.

Sulautettujen ohjelmistojen tulee yleensä myös toimia ehdottoman oikein, mikä edellyttää huolellista testausta. Kattavasti testattujen komponenttien



Java-päätteet ovat tulossa markkinoille.

käyttö vähentää ohjelmistovirheiden määrää ja Javan testattavuutta puolestaan parantaa sen tiukka oliokeskeisyys, sillä toisin kuin C/C++:lla Javalla ei ole mahdollista kirjoittaa ei-oliokeskeistä koodia.

Javaan hyvin istuva komponenttijaottelu sekä mahdollisuus komponenttien dynaamisen lataamisen verkon kautta, tarjoavat aivan uudenlaisen lähestymistavan sulautettujen järjestelmien konfigurointiin ja ylläpitoon. Vanhoja komponentteja voidaan korvata uusilla tai kokonaan uusia komponentteja voidaan ladata tarpeen mukaan.

Javan ympäristöriippumattomuus tarjoaa mielenkiintoisen edun sulautetuissa järjestelmissä, jotka sisältävät sekä sulautetun ohjelmisto- että laitteisto-osan. Yleensä kyseisten osien kehitystyö tapahtuu rinnakkain. Kehittävää ohjelmistoa suoritetaan ja testataan ensin ohjelmistopohjaisesti emuloidussa kohdeympäristössä, sen jälkeen mahdollisesti laitteistopohjaisesti emuloidussa ympäristössä ja lopulta todellisessa kohdeympäristössä. Hyvin usein ongelmana on se, että emuloidussa ympäristössä suoritettu ja testattu koodi ei toimikaan lopullisessa kohdeympäristössä. Javan tapauksessa kukin edellä mainituista ympäristöistä tarjoaakin täsmälleen samanlaisen Java-suoritusalueen, jolloin emuloiduissa ympäristöissä testatun koodin pitäisi toimia ongelmitta myös lopullisessa kohdeympäristössä. Java siis oleellisesti helpottaa kohdeympäristön simuloimista.

Java-suorittimet

Eräs mielenkiintoisimmista sulautetuista Java-teknologioista ovat Java-suorittimet. Ne ovat prosessoreja, jotka käyttävät konekielenään Java-tavukoodia. Tällä tavoin Javan suorituskykyongelmat voidaan ratkaista elegantisti. Lisäksi suorittimet mahdollistavat Java-teknologian integroimisen sulautettuihin elektroniikkalaitteisiin tehok-

kaasti ja halvalla.

Sun on valinnut Java-suorittimiin kaksi erilaista lähestymistapaa. Ensimmäkin Sun Semiconductors suunnittelee ja valmistaa Java-suorittimia. Toisaalta Java-suorittimen ydin on lisensoitavissa ja ulkopuoliset puoli-johdevalmistajat voivat näin valmistaa kilpailevia tuotteita.

Nykyinen Java-tuotekartta sisältää kolme eri perhettä: picoJava, microJava ja ultraJava -suorittimet. PicoJava on prosessoriydin ja soveltuu siten parhaiten elektroniikkayrityksille, jotka haluavat kehittää omia tuotteitaan esimerkiksi ASIC-suunnittelun kautta. microJava ja ultraJava-suorittimet ovat itsenäisiä prosessoreja ja niiden välinen ero on lähinnä tehossa.

MicroJava-suoritin on tarkoitettu lähinnä sulautettuihin kulutus-elektroniikkatuotteisiin, joissa hinta ja pieni tehonkulutus ovat tehokkuuden optimointia tärkeämpiä suunnitteluohjenuoria. UltraJava-suoritin on suoritusarvo tehoikkain jäsen, mutta siitä syystä myös kalliimpi ja enemmän tehoa kuluttava. UltraJava löytäneekin käyttökohteensa lähinnä kommunikaattori- ja PDA-tyyppisten tuotteiden tai jopa työasemien, markkina-alueelta. Tätä artikkelia kirjoitettaessa picoJava ja microJava (microJava 701-suoritin) on julkistettu, ultraJava on vielä lupausten asteella.

PicoJava-ydin on varsin pieni ja mahdollistaa sen, että samaan piirituotteeseen voitaisiin upottaa esimerkiksi erillinen DSP tai mikrokontrolleriydin. Tällaisen hybridiratkaisun etuna olisi joustavuus, kullakin suoritusnopeudella suoritetaan ne tehtävät, johon kyseinen arkkitehtuuri parhaiten soveltuu. Tällainen toteutus on askel kohti helposti konfiguroituvia tuotteita ja system-on-chip ajattelua. Näiden aspektien tutkiminen on osa VTT Elektroniikan pitkän tähtäimen tutkimusta.

Tällä hetkellä Java-teknologiaa ovat lisensoineet useimmat merkittävät valmistajat muiden mukana Texas Instruments ja Motorola. Näiden suurten valmistajien mukanaolo luo uskottavuutta Java-teknologian tulevalle tunkeutumiselle sulautettuihin järjestelmiin sekä laite-elektroniikan että ohjelmistosuunnittelun kautta.

Java VTT Elektroniikassa

VTT Elektroniikassa, jonka tutkimus- ja tuotekehitystoiminta



on erikoistunut erityisesti sulautettuihin järjestelmiin ja mikroelektroniikkaan, Java-teknologia nähdään strategisesti mielenkiintoisena ja lupaavana teknologiana. VTT onkin panostanut merkittävästi resurssejaan Java-teknologian seuraamiseen aina standardointipyrkimyksistä tuotekehityshankkeisiin saakka. VTT:n tutkijat ovat myös antaneet Java-teknologiansiirtopalvelua.

Reaaliaikajärjestelmien ryhmän tutkimustiimeissä on tutkittu Javan käyttämistä erilaisten langattomien tiedonsiirtojärjestelmien toteutuksessa. Esimerkkinä mainittakoon mukautettavan virheenkorjausrutiinin toteuttaminen reaaliaikaiseen sulautettuun tiedonsiirtojärjestelmään.

Javan toimiminen reaaliaikaisten käyttöjärjestelmien kanssa on myös ollut kiinnostuksen kohteena, samoin kuin Java-älykorttien soveltaminen elektroniikkatuotteissa.

VTT Elektroniikassa on myös käynnissä tutkimusprojekti, jossa kehitetään virtuaalisuunnitteluteknologioita tukemaan elektroniikka- ja tietoliikennetuotteiden tuotekehitystä. Kyseisessä projektissa Javaa sovelletaan sekä työkalukehityksessä että tuotteita ja niiden ominaisuuksia simuloivissa virtuaalimalleissa. Tuote ja sen ominaisuudet, lähinnä toiminnallisuus, mallinnetaan oliokeskeisten ja Javalla toteutettujen virtuaalikomponenttien avulla. Tulevaisuudessa, kun sulautetut Java-suoritusympäristöt otetaan käyttöön, voidaan sulautettujen järjestelmien simulointimalleista suoraan generoida tuotteeseen sulautettavat Java-sovellukset, mikä tulee huomattavasti nopeuttamaan järjestelmäkehitystä.

Tulevaisuuden lupaus

Suhteellisen nuoresta iästään huolimatta Java edustaa riittävän kypsää teknologiaa käytäväksi monilla sovellusalueilla. Sulautetuissa järjestelmien teknologia-alueella on toistaiseksi vielä ongelmia, jotka kaipaavat ratkaisua. Lyhyesti sanottuna, todellisten reaaliaikaisten sulautettujen järjestelmien toteutta-

minen pelkästään Javaa käyttäen ei vielä tällä hetkellä onnistu. Eletään vaihetta, jossa odotetaan ratkaisuja tässä artikkelissa esitettyihin ongelmiin. Näiden ratkaisujen eteen tehdään töitä laajalla rintamalla.

On myös muistettava, että vaikka Java ei yleistyisikään ajonaikaisena järjestelmän sulautetuissa laitteissa, se tulee silti olemaan hyödyllinen suunnittelu ja testaus teknologia. Ongelmista huolimatta Java on erittäin varteen otettava teknologia, jonka kehitystä on syytä seurata hyvin huolellisesti. Yleisten asiantuntija-arvioiden mukaan Java tulee saavuttamaan vahvan aseman sulautettujen järjestelmien alueella seuraavien 2–3 vuoden kuluttua. ●

Marko Salmela toimii VTT Elektroniikassa tutkijana. Hän työskentelee Virtuaaliprotyypointiprojektin (VRP) [3] projektipäällikkönä tehden tutkimusta mm. Java-teknologioihin pohjautuvasta elektroniikka- ja tietoliikennetuotteiden virtuaalisuunnittelusta.

Petri Mähönen toimii VTT Elektroniikassa erikoistutkijana. Hänen radiomodeemeihin liittyvät tutkimuskohteensa keskittyvät erityisesti verkkoteknologiaan, software radio -arkkitehtuurien protokollien suunnitteluun sekä Java-teknologioilla toteutettuihin protokollin ja DSP-osiin. Hän on myös vastuussa standardointiaktiviteeteista.

ENGLISH SUMMARY

Embedded Java

The Java-technology developed by Sun Microsystems has been one of the focal points of IT-technology discussions during last years. Java has evolved and matured to be complete programming-paradigm and it is also promising new technology for embedded and networked system design and implementation. One of the special interest areas in near future will be the use of Java-technology for telecommunications and datacommunications starting from embedded terminals (such as software-radios) up to the levels of applications and network management software.

Aiheesta enemmän

Gosling James et al.: The Java Language Specification, Addison-Wesley
PersonalJava: <http://java.sun.com/products/personaljava/>

Realtime pERC, NewMonics: <http://www.newmonics.com>

VRP-projekti: <http://www.ele.vtt.fi/vrp>

Chew, Frederick F.: The Java/C++ Cross Reference Handbook, Prentice Hall

Java home page: <http://java.sun.com/>