

Embedded Java im Internet der Dinge

Immer mehr Objekte, die kleine und kleinste Computersysteme beinhalten, werden direkt oder indirekt mit dem Internet verbunden. Das «Internet der Dinge» (Internet of Things, IoT) ist im Entstehen. Die Java-Technologie liefert einen wichtigen Beitrag, um nutzbringende eingebettete Systeme für das IoT zu erstellen, die einfach zu betreiben und zu entwickeln sind und die gleichzeitig hohen Sicherheitsansprüchen genügen.



Das Energy Valve ist ein intelligentes Ventil, welches das Raumklima selbstständig optimieren und Daten erheben kann.

Bei der Entwicklung eingebetteter Systeme kommen in den letzten Jahren verstärkt «managed» Laufzeitumgebungen und zugehörige Programmiersprachen wie Java zum Einsatz. Gerade für die Erstellung von vernetzten Komponenten, die Bestandteile des Internet of Things sind, bringt dies viele Vorteile.

Viele Objekte im privaten, kommerziellen und industriellen Umfeld gewinnen seit Längerem enorm an lokaler «Intelligenz». Dieser Trend wird durch die fortschreitende Miniaturisierung digitaler Komponenten sowie deren stetige Leistungssteigerung bei gleichzeitigen Kostenreduktionen gefördert. Daneben sind immer mehr Geräte und Objekte in der Lage, mit anderen Systemen zu kommunizieren, anstatt nur auf sich selbst gestellt ihre Aufgabe zu erfüllen. Durch diese Vernetzung entstehen Gesamtsysteme,

deren Wert weit mehr ist als die Summe seiner Einzelkomponenten. Mit bestehenden Geräten sind so ganz neue Anwendungen möglich. Da für die Vernetzung die allgegenwärtigen und standardisierten Protokolle und Infrastrukturen des Internet benutzt werden, ist auf einfache und kostengünstige Weise der entfernte Zugriff auf die angebandenen Komponenten realisierbar. Auch ohne physische Präsenz kann mit den Bausteinen des resultierenden Internet of Things interagiert werden, sei es durch Menschen, sei es durch andere Computersysteme.

Bewährte Technologie

Für die technische Realisierung solcher eingebetteter und vernetzter Systeme hat sich aus vielen Gründen die Java-Technologie bestens bewährt. Der resultierende Nutzen besteht für Hersteller, Betreiber und

Endanwender sowie aus Sicht der Informatiksicherheit. Beim Zürcher Softwareunternehmen Ergon Informatik wurden auf dieser technologischen Basis im Kundenauftrag bereits diverse eingebettete und mobile Systeme realisiert. Exemplarisch soll hier ein Produkt der Firma Belimo genannt werden, das sogenannte «Energy Valve». Energy Valve ist eine intelligente und vernetzte Komponente aus dem Bereich der Heizungs-Lüftungs-Klima-Automation (HLK). Es dient als Ersatz eines bisher «dummen» Ventils in der Versorgung von Gebäuden mit Kühl- oder Heizwasser. Das Energy Valve kann von einer zentralen Steuerung Vorgaben erhalten, aber dennoch durch eigene Messungen und Algorithmen lokale Optimierungsentscheidungen treffen. Gleichzeitig kann es die erhobenen Daten, die der Optimierung zugrunde liegen, über

Monate hinweg protokollieren und für Auswertungen über das Netz bereitstellen. So ist eine langfristige Analyse des Klimasystems im Gebäude möglich. Durch die Analyse kann erkannt werden, wenn der Wirkungsgrad des Wärmetauschers nachlässt und entsprechende Reinigungs- oder Ersatzmassnahmen ergriffen werden. Bei grossen Immobilien wie am Massachusetts Institute of Technology (MIT) in den USA, wo Energy Valve im Einsatz ist, können so Millionenbeträge an Energiekosten eingespart werden.

Java-Plattform für ein intelligentes Ventil

Ein wesentlicher Bestandteil für die Realisation des Energy Valve ist die für Belimo auf Basis der Java-Technologie entwickelte vernetzte HLK-Plattform «Shared Logic». Diese umfasst einerseits die lokale Datenverarbeitungs-Engine auf den HLK-Komponenten, wie dem oben vorgestellten Ventil. Andererseits gibt es innerhalb dieser Plattform eine leistungsfähige Unterstützung für die verteilte Verarbeitung, eine erweiterbare Bibliothek von Logikkomponenten sowie einfach zu bedienende grafische Design-Tools zur Erstellung von HLK-Gesamtsystemen auf Basis der «Shared Logic»-Komponenten. All diese Funktionalität wurde dank der spezifischen Vorteile der Java-Technologie nutzbringend realisiert. Woher kommt nun der Nutzen der Java-Technologie für die Entwicklung eingebetteter Systeme? Das Memory-Management ist gerade im Vergleich zu hardwarenäheren Sprachen wie C oder C++ deutlich ausgereifter und robuster.

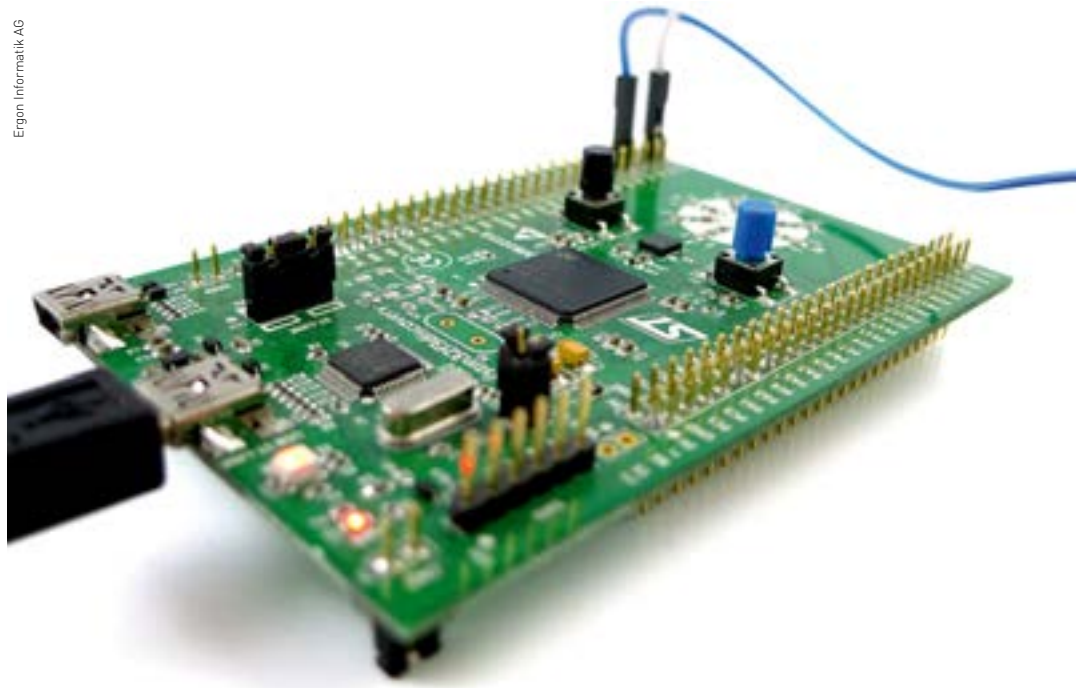
Die Präsenz des Garbage-Coll-ectors macht die Implementierung fehlerärmer und ermöglicht erst den effizienten Einsatz von Software-Patterns, die die Verständlichkeit und Wartbarkeit des Programmcodes erhöhen. Gerade in komplexeren Systemen erhöht dies die Softwarequalität deutlich. Auch für die Sicherheit bringt das fortgeschrittene Memory-Management grosse Vorteile: Direkte Zugriffe auf Speicherstellen sind nicht mehr möglich, unbenutzte Stack-Overflows – eine häufige Quelle von Angriffsvektoren – gehören der Vergangenheit an, und eine Schwachstelle wie die vor Kurzem öffentlich bekannt gewordene Heartbleed-Sicherheitslücke wäre aufgrund der beim Arrayzugriff immer automatisch vorgenommenen Indexprüfung nicht möglich gewesen.

Grosse Entwickler-Community

Da Java gerade auch im Bereich der Netzwerkprogrammierung sehr stark ist, ist es im professionellen Enterprise-Umfeld die gängige Programmiersprache und Laufzeitumgebung für Server-Anwendungen und Internet-Services. Dies hat zur Folge, dass es eine riesige Entwickler-Gemeinschaft gibt, die zudem eine enorme Menge an ausgereiften Bibliotheken und leistungsfähigen Frameworks hervorgebracht hat. Diese wiederum lassen sich oft nutzbringend im Kontext eingebetteter Systeme und IoT-Komponenten auf Java-Basis einsetzen.

Als Folge des langjährigen Einsatzes von Java-Technologie im Unternehmensumfeld sind hier auch viele unterstützende Werkzeuge und Entwicklungsplattformen entstanden, die den Implementierungsprozess signifikant verbessern: Gerade was das Build-Management, das automatisierte Testing von Komponenten, Gesamtsystemen und Benutzerschnittstellen, die fortlaufende Integration (Continuous-Integration) oder die permanente automatische Erstellung von

Ergon Informatik AG



Ahead-of-time-Compilation-Ansätze ermöglichen die Ausführung von Java-Code auf sehr ressourcenbeschränkten Plattformen wie ARM Cortex-M Systemen.

Software-Releases angeht, sind im Java-Umfeld sehr viele leistungsfähige Tools vorhanden.

Java in Echtzeit

Gelegentlich hat Java noch den Ruf, einen unnötig hohen Ressourcenbedarf auf der Zielplattform zu haben. Unterdessen ist jedoch eine Vielzahl leistungsfähiger Laufzeitumgebungen (Java Virtual Machine, JVM) optimiert für den Embedded-Einsatz verfügbar, beispielsweise Oracle J2SE Embedded, Jamaica VM, OpenJDK für ARM Prozessoren, MicroEJ, IBM J9, Dalvik/Android und andere mehr. Selbst zur Erfüllung von Real-Time-Anforderungen gibt es unterdessen Lösungen: Einige Java Virtual Machines sind sogar bereits für den Einsatz in der Automobil- und Flugzeugindustrie zertifiziert. Performancenachteile aufgrund der Interpretation gibt es dank Compilertechnologien wie Just-in-time-Compilation (JIT) oder Ahead-of-time-Compilation (AOT) kaum noch oder sie werden sogar ausgeglichen. Die immer günstigere und gleichzeitig immer leistungsfähiger werdende Hardware trägt ihr Übriges zur Kompensation bei.

Auf der Java Virtual Machine können zudem auch Softwarekomponenten ausgeführt werden, die nicht in Java selbst, sondern in anderen modernen Programmiersprachen entwickelt wurden, die auf der JVM basieren wie beispielsweise Scala oder Ruby. Auch das Mischen von Komponenten, die in unterschiedlichen Sprachen erstellt wurden, ist so möglich. Dies erhöht die Flexibilität im Entwicklungsprozess weiter. Neben dem Einsatz einer herkömmlichen JVM gibt es inzwischen auch eine Reihe anderer Ausführungstechniken für Programmiersprachen aus der Java-Familie oder dem C#-Umfeld, die auf dem Ahead-of-time-Compilation-Ansatz beruhen. Beispiele dafür sind die Android ART Runtime, .Net Native, die RoboVM zum Einsatz von Java unter iOS oder das JAhead-Framework von Ergon, das die Ausführung von Java-Code auf sehr ressourcenbeschränkten Plattformen wie ARM Cortex-M-Systemen erlaubt.

Als Schlussfolgerung lässt sich feststellen, dass Java die perfekte Realisierungstechnologie für Embedded-Komponenten im Internet der Dinge ist. Auf der

Basis einer grossen Menge ausgereifter existierender Softwarekomponenten lässt sich unter Einsatz von Rapid Prototyping und den vielfältigen automatisierbaren Testverfahren qualitativ hochstehende eingebettete Software erstellen. Da Java häufig auch auf Mobilgeräten und für serverseitige netzwerk-basierte Services eingesetzt wird und an den Hochschulen gelehrt wird, steht eine riesige Entwicklerbasis zur Verfügung. Java verfügt über ausgezeichneten Netzwerksupport und ein breites Spektrum bewährter Bibliotheken für nahezu jeden Anwendungsfall. In Kombination mit der Laufzeitumgebung und dem strikt geprüften Speichermodell ist für ein hohes Mass an Security gesorgt – ein wichtiger Aspekt, wenn man die in der jüngeren Vergangenheit erfolgten Angriffe auf IoT-Komponenten und eingebettete Systeme zukünftig vermeiden will. ☺

Peter K. Brandt, Lead Engineer Mobile and Embedded, und Gert Brettlecker, Technologieverantwortlicher für den Bereich Industry & Mobile Solutions Ergon Informatik AG www.ergon.ch