Szene-Trends nachgefragt – diese Ausgabe greift das Thema „Java und Flugsicherheit" der letzten Ausgabe des vergangenen Jahres noch einmal auf

„Real-Time Specification for Java" (RTSJ) kann zur Safety und Security beitragen"

Die Meldungen zu Zwischenfällen im Luftverkehr sind Mahnung, die technischen Möglichkeiten der Prävention und Überwachung auszubauen. Java kann helfen, wie Dr. James J. Hunt, CEO der aicas GmbH, in einem Interview zur RTSJ bestätigt. Aber auch „Watson", mit Java „im Bauch", an der Schnittstelle von Cognitive Computing und Internet der Dinge.

Dr. James J. Hunt ist Mitbegründer und CEO der aicas GmbH. Er hat einen Bachelor of Science in Computer Science und Physik von der Universität Yale, einen Master in Computer Science von der Universität Boston und einen Doktor-Titel in Informatik von der Universität Karlsruhe.

Dr. Hunt verbrachte viele Jahre als Forscher im Lincoln Laboratory des Massachusetts Institute of Technology (MIT), wo er CAD-Software für Restructurable Wafer Scale Integrated Circuits (RVLSI) und parallele Lisp-Systeme für Signalverarbeitung entwickelt hat.

Dr. Hunt war Leiter des IST-Projekts HIDOORS und technischer Leiter des Artemis-CHARTER-Projekts, welche Werkzeuge und Technologien für Java-basierte sicherheitskritische Systeme entwickelt haben. Er ist ein aktiv mitwirkendes Mitglied des Java Community Process (JCP), insbesondere für Safety-Critical Java (JSR-302) und als Specification Lead für die Real-Time Specification for Java (JSR-282). Dr. Hunt war auch der europäische Co-Vorsitzende der Untergruppe für objektorientierte Technologien des SC-205-Plenums zur Aktualisierung von Software-Sicherheitsstandards in der Luftfahrt. Das Resultat sind die mittlerweile veröffentlichten Standards DO-332 und ED-217.

AHK: Kann die Mensch-Maschine-Schnittstelle in der Flugsteuerung durch Java insgesamt sicherer werden?

Dr. James J. Hunt: Keine Sprache kann alleine die Flugsicherung gewährleisten. Die Erstellung von Software für Flugzeuge ist ein komplexes Verfahren, weil man die Sicherheit des Flugverkehrs gewährleisten will. Je wichtiger die Funktionalität für das Flugzeug ist, desto mehr Aufwand wird bei der Erstellung der Software verlangt.

Allerdings sind die Standards eher auf Sicherheit im Sinne von „Safety" als im Sinne von „Security" ausgerichtet.

AHK: Welche Rolle kann Java für die Flugsicherung spielen?

Dr. James J. Hunt: Java bietet einiges, was zur Sicherheit sowohl im Sinne von „Safety" als auch von „Security" beisteuern kann. Aber hier möchte ich auch vorsichtig sein. Weil Flugsysteme meistens mit Reagieren innerhalb einer kurzen Zeit zu tun haben, kann man nicht mit konventionellem Java arbeiten. Was man gut verwenden kann, ist eine echtzeitfähige Java-Implementierung, die durch die „Real-Time Specification for Java" (RTSJ) mit deterministischer Garbage Collection definiert ist.

Im Wesentlichen definiert die RTSJ eine Möglichkeit für Java-Threads, Echtzeit-Prioritäten zu bekommen und wie normale Threads und Echtzeit-Threads zu interagieren. Mit einem deterministischen Garbage Collector kann man die Vorteile von Garbage Collection in Echtzeit-Systeme bekommen.

Garbage Collection an sich macht die Gewährleistung von Safety und Security einfacher, weil es eine Kategorie von Fehlern ausschließt. Speicher kann nur freigegeben werden, wenn er nicht mehr im Einsatz ist, und auf Speicher kann nur zugegriffen werden, wenn man einen gültigen Zeiger darauf hat. Das bedeutet, dass man nicht aus Versehen in den falschen Daten landet: Probleme, die einen großen Teil der Fehler in der Software verursachen. Zum Beispiel entstehen viele Security-Löcher in Betriebssystemen immer noch durch Pufferüberläufe, was mit Echtzeit-Java nicht vorkommen kann.

In der Vergangenheit durfte man dynamische Allokation gar nicht in kritischen Flugsystemen anwenden. Das geht gut mit einfachen Kontrollsystemen, die man als Zustandsmaschine abbilden kann. Aber komplexere Kontrollsysteme, die den Piloten teilweise oder gänzlich ersetzen können, sind ohne dynamische Speicherverwaltung zunehmend schwierig zu programmieren. Deswegen habe ich mit meinen Kollegen von der Subgruppe für objektorientierte Technologie (OOT) im Plenum des „RTCA Special Committee 205/EUROCAE WG 71" (Software Considerations in Aeronautical Systems) an der Leitlinie für dynamische Speicherverwaltung gearbeitet. Diese Leitlinie ist jetzt ein Teil der aktuellen Standards für die Entwicklung von Software in Flugzeugen. Das heißt, dass Echtzeit-Java im Flugzeug verwendbar ist.

AHK: Könnten gefährliche Manipulationen der Flugsteuerung durch Piloten mit Hilfe von Java aus der Ferne neutralisiert werden? - Wenn ja, wie?

Dr. James J. Hunt: Natürlich muss man die Flugsysteme so entwerfen, dass Manipulationen der Flugsteuerung durch Piloten ausgeschlossen sind und Fernsteuerung möglich ist. Java bietet die Schnittstellen, um sichere Kommunikation außerhalb der Flugzeuge zu programmieren. Ein Beispiel dafür ist GateLink in der Boeing 787: Das gewährleistet einen sicheren Datenaustausch zwischen Flugzeug und Flughafensystemen und ist mit JamaicaVM programmiert.

Aber Echzeit-Java bietet nicht nur sichere Kommunikation und Datenverwaltung mit einem starken Thread-Modell.

Hintergrund

Immer komplexer werdende Echtzeit-Systeme in Luft- und Raumfahrt, aber auch in Automobilindustrie, Medizintechnik und Industrie-Automatisierung, verlangen von Hard- und Software beispielsweise, dass eine Maschine ohne Verzögerung stoppen kann, wenn Probleme auftreten – obwohl vielerlei unterschiedliche Betriebssysteme und Prozessorarchitekturen im Einsatz sind. Die aicas GmbH, die 2001 von James J. Hunt, Andy Walter und Fridtjof Siebert gegründet wurde, beschäftigt sich deshalb vor allem damit, Echtzeit und Java unter einen Hut zu bringen. Also dafür zu sorgen, dass Systeme auf Ereignisse in festgelegten Zeiträumen garantiert und vorhersehbar reagieren (Echtzeit), während die Vorteile von Java wie Objektorientierung und Plattformunabhängigkeit zum Tragen kommen.

Dafür nutzt das Unternehmen seine JamaicaVM, eine hart echtzeitfähige Java Virtual Machine. Die Realtime Specification for Java (RTSJ) ermöglicht die Entwicklung von Echtzeit-Anwendungen sowie sicheren Zugriff auf bestimmte Speicher-Bereiche und somit beispielsweise auch die Entwicklung von Treibern und Firmware.

Echtzeitfähigkeit ist für Java eine besondere Herausforderung. Implementierungen, die in diesen Bereich vordringen wollen, müssen das Problem der automatischen Speicherbereinigung (Garbage Collection), die den Programmfluss unterbricht, lösen.

Die Sprache ist so definiert, dass man formale Methoden einsetzen kann, um Fehler zu vermeiden. Durch formale Spezifikationen kann ein Tool wie KeY, ursprünglich vom Karlsruhe Institute of Technology (KIT), die Korrektheit des Codes beweisen. Die RTSJ und die Java-Spezifikation geben dort die Grundlage für diese Werkzeuge, wo RTSJ 2.0 von JSR-282 und Java 8 den neuesten Stand der Sprache definiert.

Interview: Annegret Handel-Kempf (AHK)

▼ Watson wacht über Wetter und Widrigkeiten

Wie können Big Data, Internet of Things (IoT), Java und das Web für Sicherheit bei Flügen genutzt werden?

Auf eine „digitale Reise" hat sich *Laurent Martinez*, Leiter des neuen Bereiches „Services by Airbus", begeben. Unterstützt vom IBM Watson IoT. Das heißt, beim „gigantischen Technologiesprung" des in Toulouse beheimateten Luftfahrtunternehmens bedient sich der 47-Jährige kognitiver Computer. Dabei geht es nicht einfach nur um Wartung, Workshops und reibungslose Beförderung. „Wir sprechen über das Leben der Passagiere", betonte der Airbus-Manager bei der Eröffnung der weltweiten Zentrale des Watson-Geschäftsbereichs IoT in den Münchner Highlight Towers im Dezember 2015 (ausführlicher Bericht in Ausgabe 2/2016).

In der Parkstadt Schwabing ist auch der Standort des ersten europäischen Watson Innovation Centers angesiedelt. Zudem ein Watson IoT Client Experience Center – eines von acht neuen Centern in Asien, Europa und Lateinamerika. In diesen Centern bekommen Kunden und Partner direkten Zugriff auf Technologien, Werkzeuge und Know-how von IBM.

Martinez freut vor allem das Zusammenspiel von Cognitive Computing und Internet der Dinge in Watsons rasend schneller Strukturierung, Analyse und Interpretation von historischen und aktuellen Daten. Für „Security", „Safety", aber auch für Effizienz und Nachhaltigkeit benötige er möglichst gute und schnelle Lösungsvorschläge fürs Risikomanagement.

„250 Gigabyte Daten entstehen während eines einzigen Fluges", berichtet der smarte Franzose. Auch sonst hat er imposante Stückzahlen wie die 300 Millionen Teile pro Flugzeug und Zehntausende Parameter im Flugservice-Umfeld zu bieten, die teils über Sensoren digital vernetzt sind. „Auswertung" im „valuable"-Sinn liefert hier nur ein einziger. Kein Mensch, trotz aller Cleverness. Sondern Watson, der IBM-Supercomputer, der erwachsen geworden ist. Er spielt nicht mehr nur Jeopardy-Quiz, sondern arbeitet für Wirtschaft und Forschung. Der Code für Watson ist übrigens größtenteils Java mit einigen wesentlichen Brocken in C++ und Prolog.

Füllte er einst einen Raum von der Größe eines Schlafzimmers, ist der Rechner jetzt nur noch so groß, wie drei aufeinander gestapelte Pizzaschachteln. Auf Smartphone-Größe werde er vorerst nicht schrumpfen, erklärte IBM-Vizepräsident *John Kelly*. Schließlich brauche Watsons „Hirn" Platz für all seine schnell abzuwägenden Entscheidungsvorschläge.

Watson ist die Verkörperung von Cognitive Computing. Dieses beschreibt eine neue Klasse von Systemen, die lernen, argumentieren und in natürlicher Sprache mit den Menschen interagieren können. Dabei werden ihnen bestimmte Fähigkeiten nicht explizit anprogrammiert, sondern sie lernen und bilden ihr Verständnis aus Interaktionen und Erfahrungen, die sie mit ihrer Umgebung machen. Dadurch halten sie gleichzeitig Schritt mit den wachsenden Datenvolumina, der steigenden Komplexität sowie unvorhersehbaren Informationen, die im Internet der Dinge entstehen. Sie helfen damit bei der Erschließung der rund 80 Prozent an unstrukturierten Daten, um das Unsichtbare sichtbar zu machen und die Welt besser zu verstehen.

Beim „Watson"-Willkommen in der bayerischen Landeshauptstadt beschrieb Martinez, wie ein Sicherheitsszenario in einer Flug-Notfallsituation aussehen könnte: Der Pilot fragt in natürlicher Sprache den kognitiven Computer, was er tun soll. Eine Turbine brenne. Watson antwortet: „Setze zur Notlandung in Istanbul an". Der Pilot sagt: „Mir ist heute nicht so nach Istanbul. Wie wäre es mit Athen?". Watson antwortet: „Idiot". Sagt nichts mehr und unternimmt aber auch nichts, um die Maschine eigenständig zu landen.

Als Scherz war diese Schockszenario-Schilderung des Vortragenden gedacht. Verkehrt daran: Watson sagt nicht: „Idiot", weil er keine Emotionen hat. Gefühle und Kreativität aus Inspiration fehlen ihm. Gegebenenfalls auch ein paar letzte Informationen zur Situationseinschätzung. Deshalb wird der Mensch als Entscheider immer noch gebraucht.

Richtig ist, dass er nicht als Autopilot mit einer Notlandung einspringen kann. Dafür ist er nicht gebaut.

Doch Watson kann vom Wetter lernen und aus seiner Erfahrung unendlich vieler, von ihm in Relation gebrachter Komponenten, die jedes menschliche Hirn sprengen würden, heraus warnen. *David Kenny*, CEO von The Weather Company, die im Oktober 2015 von IBM übernommen wurde, erzählte in München von der Herausforderung, die Atmosphäre auf dem ganzen Planeten Erde zu „mappen" und ins kognitive System zu bringen.

Von Wetterballons, 50.000 Flügen täglich, von 40 Millionen Smartphones und insgesamt drei Milliarden Referenzpunkten kämen die Daten für Vorhersagemodelle, Big-Data-Analytics- und Cognitive-Computing-Szenarien, insbesondere im Internet der Dinge. IoT, Wetter, Watson beziehungsweise Cognitive Computing und Klima seien über Sensoren und Datenanalyse eng miteinander verbunden. Dem Piloten könnten Entscheidungshilfen bei Wetterturbulenzen gegeben werden. Er hätte die Chance, Tornados auszuweichen und Treibstoff sparende Flugrouten zu wählen. Hier hilft Watson wirklich.

Text und Fotos: Annegret Handel-Kempf