Szene-Trends nachgefragt – in dieser Ausgabe zur Frage, wo die IT in der Flugsicherung helfen oder aber die Sicherheit gefährden kann

Interviews mit Experten für „Java und Sicherheit im Flugzeug“

Für diese und die folgende Ausgabe fragte Szene-Redakteurin Annegret Handel-Kempf (AHK) bei Experten nach zum Thema „Java und Sicherheit im Flugzeug“.

Im März 2015 versetzte vermutlich der Co-Pilot des Fluges U4952 den Airbus 320 in Sinkflug, der deshalb in den Alpen abstürzte – ohne dass von einer Bodenkontrolle Gegenmaßnahmen ergriffen werden konnten. Eine derartige Fernsteuerung, wie bei Roboter-Drohnen, ist unter anderem wegen potenzieller Hackerangriffe umstritten.

Im Mai warf das FBI dem Sicherheitsexperten Chris Roberts vor, während eines Fluges mit der Boeing 737 nicht nur Informationen über seinen Laptop ausgelesen und Hacks im Simulator durchgeführt zu haben. Vielmehr soll er einen kurzzeitigen Seitwärtsflug verursacht haben, indem er die Schubkontrolle manipuliert haben soll.

Nicht nur Roberts, sondern sogar das FBI selbst hatten seit Jahren vor Cyber-Attacken, die den Kurs eines Flugzeuges verändern könnten, gewarnt. Entscheidend für eine funktionierende Flugsicherung sei, dass die Flugsteuerung in modernen Flugzeugen physikalisch getrennt ist. Das bedeutet, dass das WLAN für die Passagiere und die Bordnetzwerke des Flugzeugs nicht miteinander verbunden sind.

Die Luftfahrtbranche steht seit 9/11 im Spannungsfeld zwischen wachsenden Sicherheitsanforderungen in Zeiten von Terror und Kriegen einerseits und wirtschaftlichen „Zwängen“ durch massenhafte Billigflüge und hohe Treibstoffkosten andererseits.

Unterschieden wird zwischen „Security“, der „Luftsicherheit“, und „Safety“, der „Flugsicherung“. Die „Security“, der Schutz vor äußeren Gefahren, bewirkte bei Flug U4952, dass der Pilot wegen der mechanischen Blockade der Cockpit-Tür durch den Co-Piloten nicht rettend eingreifen konnte. Die „Safety“ soll Personen- und Sachschäden durch Unfälle und technische Defekte oder auch durch den „Faktor Mensch“ vermeiden.

JavaSPEKTRUM-Szene-Redakteurin Annegret Handel-Kempf (AHK) fragte nach, wo die IT in der Flugsicherung helfen oder gar die Sicherheit gefährden könnte. Viele wollten sich nicht äußern. Interessante Aussagen zur Rolle von Echtzeit-Java für die Flugsicherheit lesen Sie in der nächsten Ausgabe von JavaSPEKTRUM.

Klare Worte zur Bedeutung der IT in der Flugsicherung recherchierte AHK letztlich auch für diese Ausgabe. Beispielsweise von *Professor Dr.-Ing. Alexander Knoll*, Experte für Flugführung und Flugzeugsysteme an der Hochschule München.

Professor Dr.-Ing. Alexander Knoll ist Experte für Flugführung und Flugzeugsysteme an der Hochschule München. Foto: Hochschule München

AHK: Herr Professor Knoll, sehen Sie generell einen Weg der Flugsicherung darin, den Computer eingreifen zu lassen, wenn der Mensch Fehler macht und umgekehrt? Ist eine derartige, wechselseitige Kontrolle von Mensch und Maschine möglich und wünschenswert?

Prof. Dr.-Ing. Alexander Knoll: Dies wird momentan häufiger diskutiert, insbesondere nach den diesbezüglichen Äußerungen des „neuen“ Chefs der DFS, dem ehemaligen Staatssekretär Scheurle.

Ich persönlich sehe dies kritischer, denn:

- Die Verantwortung liegt rein rechtlich beim Flugzeugführer. Die Flugsicherung unterstützt diesen nur.
- Die Flugsicherung hat nur einen minimalen Anteil der Informationen zur Verfügung, die der Pilot vor Ort zur Verfügung hat. Auch beim Einsatz von Drohnen hat sich schon gezeigt, dass das kleine Subset an Informationen, das der Remote-Pilot zur Verfügung hat, oftmals nicht ausreicht: Siehe Absturz der Drohne der Border Patrol an der mexikanischen Grenze, wo der „Pilot“ aufgrund seiner Daten einfach nicht bemerkte, dass der Motor nicht mehr lief.
- Die Flugsicherungsmitarbeiter haben kein flugzeugspezifisches Wissen. Dieses wäre aber nötig, um die Besonderheiten des betroffenen Flugzeugs analysieren zu können. Abgesehen davon kennt die Flugsicherung Randbedingungen wie das Wetter, Verkehr, der nicht auf dem Radar auftaucht, zum Beispiel Segelflugzeuge usw., nicht ausreichend.

Bruce Schneier, vom „Economist“ als „Security-Guru“ bezeichnet und Schneier-on-Security-Blogger, ist ein US-amerikanischer Experte für Kryptographie und Computersicherheit. Durch die Plattformunabhängigkeit erobert Java den Bereich der Echtzeitsysteme, in denen oft ein völlig zuverlässiges, millisekundengenaues Timing erforderlich ist (Computer in Kernkraftwerken, Flugzeugen, Autos …). Die Redakteurin fragte Schneier, was er von der Real-Time Specification for Java (RTSJ) mit Blick auf Flugsicherung hält.

Bruce Schneier: I don't know anything about it.

Prof. Marc Erich Latoschik: Die Mensch-Maschine-Schnittstelle hat immer Potenzial zur Verbesserung“

Professor Marc Erich Latoschik vom Lehrstuhl für Mensch-Computer-Interaktion (Informatik IX) an der Universität Würzburg spricht vom Ziel einer „Sicherstellungswahrscheinlichkeit“ an der Schnittstelle Mensch-Maschine im Flugzeug. Foto: privat

Professor Marc Erich Latoschik vom Lehrstuhl für Mensch-Computer-Interaktion (Informatik IX) an der Universität Würzburg nahm zu folgenden Fragen der Szene-Redakteurin Stellung.

AHK: Herr Professor Latoschik, weder Mensch noch Maschine sind zu 100 Prozent zuverlässig: Kann sichergestellt werden, dass Pilot oder Computer sofort regulierend eingreifen, wenn der andere Steuerungspart Gefahren nicht umgeht beziehungsweise verursacht?

Prof. Marc Erich Latoschik: Die Frage ist, was Sie mit „sichergestellt“ meinen. Es kann nur eine Sicherstellungswahrscheinlichkeit geben. Wenn beide Partner, Mensch und Maschine, nicht zu 100 Prozent zuverlässig sind, dann kann man auch keine Verlässlichkeit des einen als Fehlerkorrektur des anderen erreichen.

Daher gibt es in der IT etwa fehlertolerante Systeme – etwa durch Backupsysteme, welche bei Bedarf anspringen oder welche nebenläufig die gleichen Berechnungen durchführen und diese miteinander vergleichen. Dieses fängt vor allem hardwarenahe Probleme ab. Semantische Probleme im Programmablauf kann man beweisbar im Allgemeinen nicht sicherstellen, nur in Spezialfällen in den Griff bekommen. Dabei wird die Absicherung der Sicherstellungswahrscheinlichkeit mit zunehmender Komplexität deutlich aufwendiger und dann wieder fehleranfälliger.

Für den menschlichen Benutzer gibt es diese Backupstrategien auch schon seit Ewigkeiten. Denken Sie an die Verteilung der Zugriffsschlüssel für Interkontinentalraketen (ICBMs) usw.

Jetzt soll aber der eine Partner Backup für den anderen sein, also eine heterogene Absicherung. Dabei werden sich nach meiner Auffassung die Schwierigkeiten verschieben, aber nicht beheben lassen. Diese sind prinzipieller Natur. Es kann nur eine Sicherstellungswahrscheinlichkeit angestrebt werden.

AHK: Wie wichtig ist eine Absicherung der Kommunikationskanäle zwischen Bodenstation und Flugzeug, gegebenenfalls mithilfe von Java?

Prof. Marc Erich Latoschik: Eine Absicherung ist absolut notwendig. Die zugrunde liegende Technik ist eigentlich nicht so wichtig, wenn die Sicherheit gewährleistet ist. Hier gilt der alte IT-Grundsatz: Je mehr Software involviert ist, umso höher die Fehleranfälligkeit. Java mit der VM ist hier eher anfälliger als ein schlankes spezifisches System, dafür aber hoffentlich vielfach getestet.

AHK: Sollte ein korrigierender, möglicherweise rettender Eingriff aus der Ferne – wie bei Drohnen – möglich sein? Könnten gefährliche Manipulationen der Flugsteuerung durch Piloten, physikalische Angreifer oder Hacker überhaupt aus der Ferne neutralisiert werden?

Prof. Marc Erich Latoschik: Dies ist am Ende eine Nutzen-Kosten-Abwägung. Es wird keinen hundertprozentigen Schutz vor Hackerangriffen geben können. Es wird auch keine hundertprozentige Schutzmaßnahme durch einen Remotezugriff geben können. Beides ist mit einem Risiko versehen. Leider ist dieses Risiko schwer zu kalkulieren.

Dazu kommt, ob obiges Risiko größer oder kleiner als das Risiko ist, welches vom Piloten ausgeht, kann ich Ihnen mangels Daten nicht sagen. Hier wüssten die Versicherer bestimmt besser Bescheid. Zu letzterem könnte man also gegebenenfalls eine Berechnung durchführen, zu ersterem kaum.

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

Prof. Marc Erich Latoschik: Die Mensch-Maschine-Schnittstelle hat immer Potenzial zur Verbesserung. Meistens hört der iterative Verbesserungsprozess (user-centered design) nach einer Anzahl von Iterationen aus Ressourcenmangel auf. Man erreicht also immer eine nur teiloptimale Lösung.

Ob dies durch Java besser wird? Nun, Java hilft beim Prototyping, also dem schnellen Erstellen von Teilprototypen, dafür bekommt man mehr Softwareteile, welche im Sinne einer Fehleranfälligkeit ein höheres Risiko darstellen. Sicher gibt es auch hier irgendwann einen break-even. Hierfür verlässliche Zahlen zu erreichen, ist höchst aufwendig. Machbar zwar, aber unter den Rahmenbedingungen heutiger Softwareerstellung kaum vorstellbar.

Eher würde man gegebenenfalls das Prototyping mit Java oder anderen High-Level-Tools machen und dann eine optimierte Schnittstelle auf ein proprietäres zugeschnittenes System übertragen.

Marc Latoschik, Jahrgang 1968, stammt aus Herford in Nordrhein-Westfalen. Er studierte Informatik und Mathematik in Bielefeld, Paderborn und am New York Institute of Technology. Bis 1997 war er als Softwareentwickler selbstständig, 1996 übernahm er die Leitung des Labors für Künstliche Intelligenz und Virtuelle Realität der Universität Bielefeld. Dort promovierte er 2001 über das Thema „Multimodale Schnittstellen in der Virtuellen Realität“.

Von 2007 an lehrte und forschte Latoschik an der Hochschule für Technik und Wirtschaft in Berlin, 2009 folgte eine Professur für Intelligent Graphics an der Universität Bayreuth. Seit Mai 2011 hat Marc Latoschik den Lehrstuhl Mensch-Computer-Interaktion (Informatik IX) in Würzburg inne.

Interviews:
Annegret Handel-Kempf (AHK)