GESELLSCHAFTLICHE RELEVANZ DER INFORMATIK ALS STRUKTURTECHNOLOGIE

HUBER B. KELLER

Institut für Angewandte Informatik, KIT, Karlsruhe

Beitrag zur Tagung des FORUM Technologie & Gesellschaft am 15. Oktober 2014 in Berlin

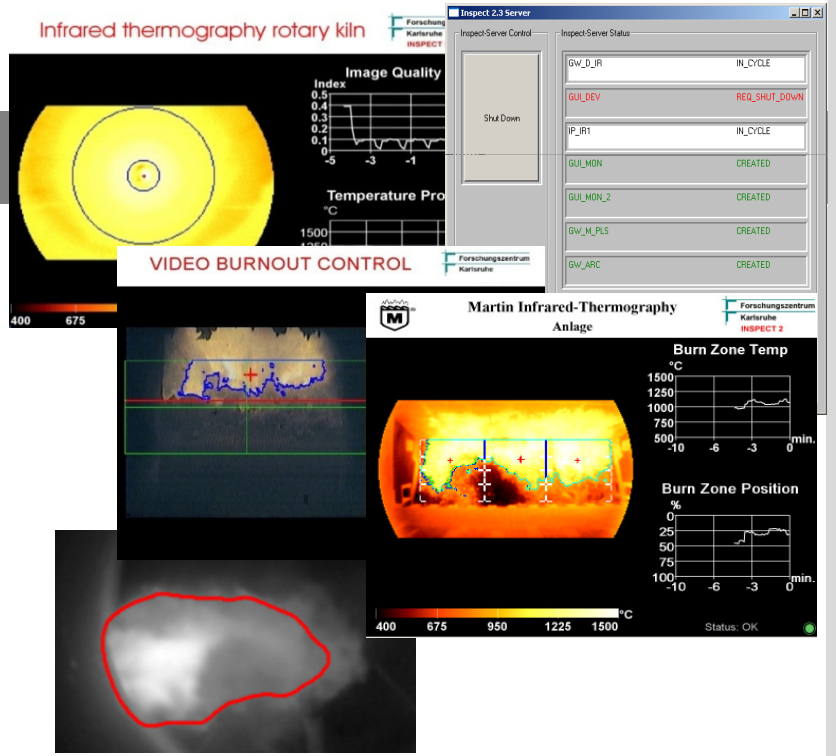
Karlsruher Institut für Technologie

Gesellschaftliche Relevanz der Informatik als Strukturtechnologie - eine Betrachtung der Herausforderungen und Chancen

Karlsruhe Institute für Technology – Institut für Angewandte Informatik (IAI)

Dr. Hubert B. Keller

Tagung „Chancen und Risiken in der Wagnisgesellschaft“
15.10.2014, Bundesanstalt für Materialforschung und –prüfung (BAM),
Unter den Eichen 42-44, 12205 Berlin



Unsichtbare Automation

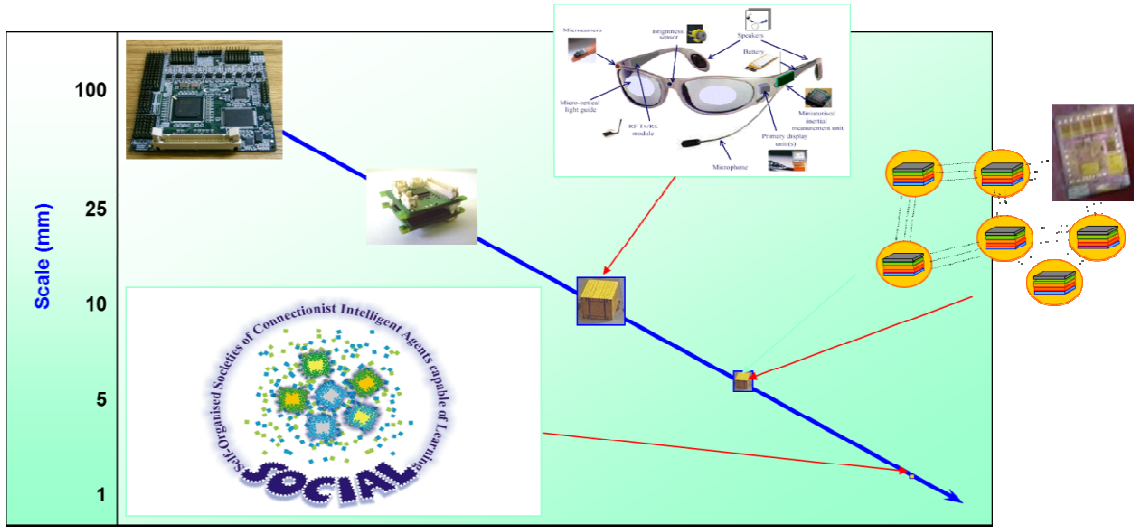
- 90% aller Computer sind nicht im PC, sondern als "eingebettete Systeme" hochintegriert in die zu automatisierenden oder überwachenden Systeme wie ABS, ESP, Herzschrittmacher usw. untrennbar eingebaut
- Informatik spielt die Schlüsselrolle bei solchen Systemen mit höchsten Zuverlässigkeits- und sicherheitskritischen Anforderungen

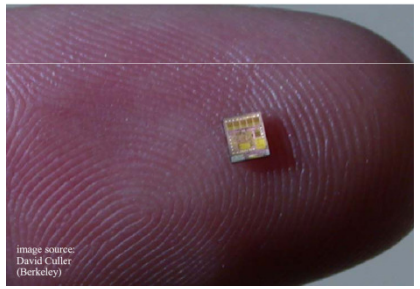
- Diese Systeme sind allerdings meist vernetzt, offen und damit potentiell gefährdet (siehe Stuxnet Angriff)
- Die gleichzeitig ablaufenden, teils sicherheitskritischen Automationsaktivitäten werden quasi unsichtbar abgearbeitet

- Alle Welt spricht dennoch nur vom Internet und den Anwendungen dort und sieht nicht die allumfassende, vernetzte Automatisierung

Unsichtbare Computer

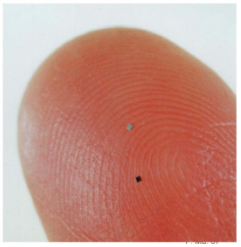
- Computer verschwinden größenmäßig
  → wir verlassen uns auf unsichtbare Computer ohne direkten Zugriff durch einen Menschen
- Unser Leben und unsere Gesundheit hängen ab vom absolut korrekten Funktionieren unter allen Umständen





µ-Chip (Hitachi)

- Size: 0.4 mm²
- Carrier frequency: 2.45 GHz
- Operating distance: 0-25 cm
- Memory capacity: 128bit ROM
- Anti-collision: no
- Response time: 20 ms



Echtzeit und Security

- Der Zentralverband der deutschen elektrotechnischen Industrie hat 2006 die
  - Informationsverarbeitung unter Echtzeitbedingungen sowie
  - Securityanforderungen

  auf solchen integrierten Computern als eine der zukünftigen Herausforderungen definiert
- Solche Systeme, z. B. im Auto, regeln nicht nur, sondern treffen auch höhere Entscheidungen wie automatisches Bremsen aufgrund automatisierter Interpretation von Messgrößen (Bilder)
- Die Vernetzung dieser Systeme führt zu einem Security- und nachfolgend auch zu einem Safety-Problem
- Dies betrifft alle automatisierten Bereiche wie Auto, Zug, Gebäude, verfahrenstechnische Anlagen etc.
- Die Sicherheit der Kommunikation berührt also direkt die Sicherheit des technischen Systems

Komplexität und Implementierung

Zwei Aspekte treten hier in Vordergrund

- Der erste Aspekt betrifft offene Türen durch eine Nichtbeherrschung der Komplexität der Software der Automatisierung
  - Wie viele zu testenden Abläufe in einer Software gibt es?
  - Welche Parameter eines Systems wirken sich unsicher aus?
  - Wie stelle ich fest, ob die Kommunikationsstrukturen sicher sind?
- Der zweite Aspekt betrifft die Güte der Implementierung bzw. der verwendeten Programmiersprache und dessen Laufzeitsystem
  - Wie fehlersicher ist die syntaktische Struktur der Sprache?
  - Wie sicher ist die Sprache/Implementierung bzgl. der Prüfung von algorithmischen Eigenschaften der Sprachmittel?
  - Indexbereiche von Vektoren werden z. B. in C und C++ nicht geprüft und sind damit zur Laufzeit durch gezielte Bereichsüberschreitung potentiell eine Gefahr
  - Sprachen wie Ada bieten Laufzeitprüfungen und reagieren mit Abbruch der Programmteile

Zuverlässigkeit und Beherrschbarkeit

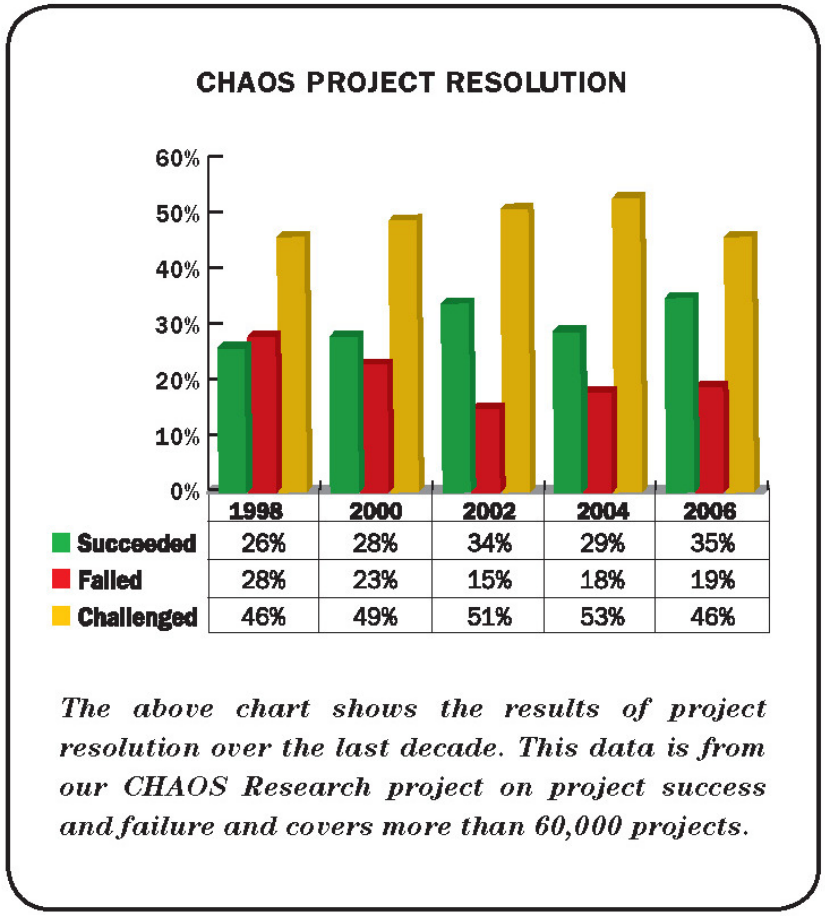
- Die Frage ist daher, wie zuverlässig sind diese Systeme von Ihrer Komplexität und Ihrem Verständnis bzw. Beherrschbarkeit durch den Benutzer

- Zuverlässigkeit betrifft die intendierte Funktion (Zielfunktion) beim bestimmungsgemäßen Gebrauch und bei Störungen über einen definierten Zeitraum (das ist nicht Verfügbarkeit!)

- Klassische Zuverlässigkeit nicht auf Software anwendbar
  - Flugzeugflügel verschwinden nicht 5 Sekunden lang
  - Ampelsystem schaltet alle Ampeln auf grün wegen unerwarteter Situation

- Beherrschbarkeit betrifft das Verständnis über das Verhalten des Systems und dessen Kontrollierbarkeit (fehlerhaftes Verhalten durch falsche Bedienung bei korrekter Funktion!)

Zuverlässigkeit von Software

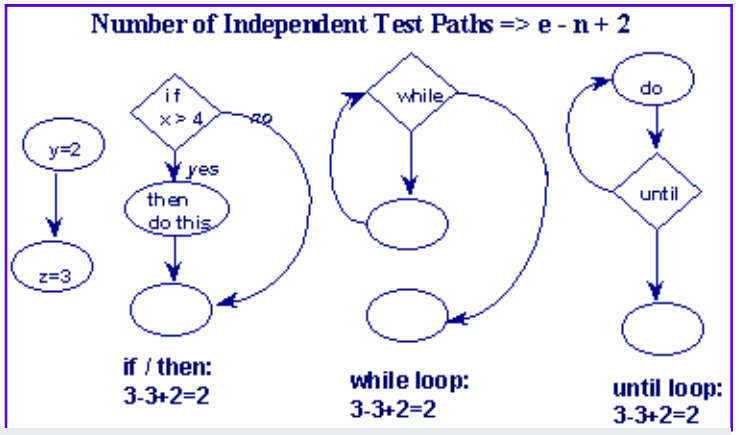


Aus: The Trends in IT Value. 2008, The Standish Group International, Incorporated

plexity

s

s



lass	Actual Problems
Value	29
urn Statement	1
Dereference	28
Condition	4
d Variable	36
e Code	20
ss Assignment	9

Determinismus im Zeitverhalten

Echtzeit

- rechtzeitig (zeitlich definiert, ESP Eingriff oder Airbag nicht zu früh und nicht zu spät),
- gleichzeitig (Worst-Case-Szenario): Airbag, Gurtstraffer, Schiebedach, Seitenscheiben, Bremsbeläge anlegen, Motorleistung drosseln, etc,
- zuverlässig (die Funktion muss über eine vorher festgelegte eindeutig definierte Zeitspanne erbracht werden, Flug über Atlantik, Wartungsintervalle bei den Bremsen),
- deterministisch (bei gleichen Situationen immer gleichartig mit den gleichen zeitlichen Randbedingungen reagieren, was bei interpretativer Verarbeitung nicht immer gewährleistet ist),
- sicher im Sinne Safety sein (bedeutet meist fehlertolerant und damit redundant mit Rückfalloption, sicherer Zustand muss erreichbar sein, tolerierbares Risiko),

...

Security

- sicher im Sinne Security (da diese Systeme hoch vernetzt sind, kommunizieren und auch remote erreichbar und damit potentiell einbruchs- und manipulationsgefährdend sind,
  welche Software ist in meinem Auto verbaut bzw. welche wird bei der Wartung eingespielt, Car2X),

- Implementierungstreu, d.h. die mathematischen Eigenschaften der Algorithmen werden programmiersprachlich und vom Laufzeitsystem umgesetzt und gesichert (gezielte Bereichsüberschreitung von Indices können nicht zur Abarbeitung von eigentlich nur als Text eingegebene Information führen)

- Strukturtreu, d.h. ausgeführte Programmaufrufe sind tatsächlich vom echten Initiator angestoßen

...

Komplexität und Beherrschbarkeit

- benutzungsfreundlich (das heißt, dem Benutzer gegenüber ist das System verständlich und im Verhalten nachvollziehbar ausgelegt),
- Windows Prozesse: → rundll32.exe, svchost -3158 ???
- verlässlich im Sinne der Summe obiger Eigenschaften und
- beherrschbar im Sinne der Komplexität
  → testbar?
- beherrschbar im Sinne mentaler Nachvollziehbarkeit durch den Benutzer → ein psychologisch intellektuelles Problem
  siehe dazu insbesondere die Forschungen von Dörner: ich verstehe das System und meine Handlungen bewirken das intendierte Ziel

- Dies betrifft insbesondere das Verständnis über das Verhalten und die Konfiguration von Systemen (→ Windows)

Resümee

- Die von uns Menschen mit Hilfe der Informatik erzeugten Systeme genügen diesen Kriterien nicht und zwar meist ohne Betrachtung der Fähigkeiten und Fertigkeiten des Menschen im Umgang damit

Denn

- Wir vernetzen diese Systeme miteinander auf Basis unzuverlässiger Konzepte und Methoden und
- die resultierenden Verhaltensweisen ergeben sich nicht mehr nur (!) aus dem System alleine, sondern aus den Wechselwirkungen dieser Systeme untereinander und in Verbindung mit dem Eingriff des Menschen

...

- Eine Ameise hat ein sehr einfaches Repertoire an Verhaltensmuster, in Summe ergeben viele Ameisen intelligentes Verhalten, das Ganze ist sehr robust

- Der Mensch hat ein sehr komplexes Repertoire an Verhaltensweisen
- Er bildet sich ein mentales Modell, dessen was er glaubt zu verstehen

- In der Wechselwirkung mit komplexen Systemen ergibt sich durch falsche oder unvollständige mentale Vorstellungen eher eine Divergenz zur intendierten Funktion und Sicherheit

- Kognitionspsychologen wie Dörner zeigen dies durch Experimente!
  → Intelligenz scheint in der Vernetzung zu verschwinden.

...

- Wie können bei der Übertragung und der interpretierten Fusion von Daten deren Nachvollziehbarkeit bzw. schon allein die Gewissheit über deren Korrektheit gesichert werden?
- Wie weiß ich, dass die eintreffenden Informationen (z.B. Car2X) korrekt sind und richtig interpretiert werden?
- Hierzu muss die Informatik grundlegende Vorgaben leisten
  - Systemarchitekturen sind klar und einfach zu strukturieren
  - Analyse- und Designphasen müssen ausgeweitet werden
  - Logische Korrektheit muss Vorrang vor Effizienz haben
  - Programmiersprachen müssen adäquate Konstrukte besitzen (Größen mit mathematischen Eigenschaften etc.)
  - Programmiersprachen brauchen Laufzeitsysteme, welche diese Eigenschaften zur Ausführungszeit prüfen (Bereichsüberschreitung etc.)
  - Funktionen müssen beherrschbar sein – insbesondere in Grenzbereichen

Zukunft der technologischen Entwicklung

Automatisierung für die Gesellschaft wird allumfassend

- Lassen wir uns vom Computer zentral steuern, damit wir als Masse intelligent bezogen auf die Anforderungen wirken?
- Was bleibt als persönliche Handlungskompetenz?
- Ziel muss Kontrolle (Beherrschbarkeit) über die benutzten Systeme sein
- Kann die kognitive Last der einlaufenden Informationsflut bei Vernetzung getragen werden (Verständnis der Systemstrukturen)?

- Unsere Gesellschaft sollte und muss diese Entwicklungen diskutieren und zumindest Leitplanken setzen, sonst sind wir nur noch vom Computer geführte Nutzer unserer eigenen Technologie.
- Möglicherweise muss die intelligente Leistungsfähigkeit moderner Technologien eingeschränkt werden ….

Diskussion