Verlässliche Echtzeitsysteme

Fallstudie: Reaktorschutzsystem

Peter Ulbrich

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)
www4.informatik.uni-erlangen.de

06. Juli 2015

Fragestellungen

- Wie sind kommerzielle verlässliche Systeme aufgebaut?
  - Welche Fehler gilt es zur Laufzeit zu tolerieren?
  - Welche Mechanismen werden für die Fehlertoleranz eingesetzt?
  - Welche Maßnahmen stellen die Korrektheit der Implementierung sicher?
- Schwerpunkt:
  - Grundverständnis der Funktion
  - Struktureller Aufbau hinsichtlich Fehlertoleranz
  - Verifikation der eingesetzten Software

☞ Fallstudie: Primäres Reaktorschutzsystem Sizewell B

Gliederung

Sizewell B

Der derzeit einzige Druckwasserreaktor in Großbritannien

(Quelle: John Brodrick)

- Standort: Suffolk, UK
- Betreiber: EDF Energy
- Erbauer (u.a.):
  - Westinghouse
  - Framatome (Areva)
  - Babcock Enegery
  - GEC-Alsthom
- Entwurf: 1980-82
- Bau: 1988-95
- Laufzeit: 2035
- Leistungsdaten:
  - Elektrisch: 1195 MW
  - Thermisch: 3479 MW

Entstehungsgeschichte

1969 Erste Ankündigung als Advanced Gas-cooled Reactor, (AGR)

1974 Steam Generating Heavy Water Reactor, (SGHWR)

- Mit schwerem Wasser moderierter Siedewasserreaktor
  - (engl. *Boiling water reator, BWR*)

1980 Ankündigung als Druckwasserreaktor

- (engl. *Pressurized water reactor, PWR*)

1982 - 1985 Begutachtung des Sicherheitskonzepts

1987 Erteilung der Baugenehmigung

1988 Baubeginn am 18.07.1988

1995 Netzsynchronisation am 14.02.1995

- Kommerzieller Betrieb seit 22.09.1995

2005 Erhöhung der thermischen Leistung auf 3479 MW

- Die Nettoleistung erhöht sich von 1188 MW auf 1195 MW
- Leistungserhöhung hängt aber von der Temperatur des Meeres ab

Das Reaktorschutzsystem

Einzige Aufgabe: Fehlertoleranz

☞ Das Reaktorschutzsystem: Der kritische Kern der Leittechnik

- **Zweck:** Durchführung einer Reaktorschnellabschaltung (RESA)
  - Auch SCRAM, reactor emergency shutdown, reactor trip
  - Falls ein unsicherer Reaktorzustand festgestellt wird
- **Funktionsweise** der Schnellabschaltung
  - Einfangen freier Neutronen, Stoppen der Kettenreaktion
    - Reaktorleistung reduziert sich auf die Nachzerfallswärme (engl. *decay heat*)
    - Diese beträgt ca. 5% der thermischen Leistung ⇝ ca. 174 MW (Sizewell B)
  - → Einschießen der Steuerstäbe (engl. *control rod*) in den Reaktorkern
    - In Druckwasserreaktoren werden diese von oben eingeschossen
    - Normalbetrieb: Magnete/Motoren pressen sie gegen vorgespannt Federn
  - → Zusätzlich: Einleiten von Neutronengiften, z. B. Borsäure

⚠ **Sicherheitsanforderung**: fail-**operational**

→ Den sicheren Zustand (engl. *fail-safe*) nimmt der Reaktor ein

Sizewell B Reaktorschutzsystem

⚠ Ausschluss: Anticipated Transient without Scram (ATWS)

- Verursacht durch Fehler im Entwurf oder der Implementierung
- Äußere Störeinflüsse

→ Gleichtaktfehler sind in jedem Fall zu vermeiden!

☞ Diversitärer Aufbau des Schutzsystems

- Primäres Schutzsystem (engl. *primary protection sys., PPS*)
  - Basierend auf digitaler Sicherheitsleittechnik
  - Überwachung von Reaktorparametern
    - Neutronenfluss im Reaktordruckbehälter
    - ^{16}N-Gehalt im Primärkühlkreislauf
  - Überwachung der Steuerstäbe
  - Reaktorinstrumentierung (engl. *reactor instrumentation*)
  - Stromkreisunterbrecher (engl. *circuit breakers*) ⇝ SCRAM
- Sekundäres Schutzsystem (engl. *secondary protection sys., SPS*)
  - Basierend auf diskret aufgebauten, analogen Schaltungen

Primäres Reaktorschutzsystem

- Zuverlässigkeitsanforderung: Toleranz eines ausgefallenen Kanals
  - Auch wenn ein Kanal aktuell gewartet wird
    - Wartungen/Tests während des Betriebs sind unumgänglich
    - Der Reaktor wird nur zur Revision und zur Wiederbefüllung heruntergefahren
    - → Diese Revisionsintervalle betragen typischerweise 18 Monate
- Zulässige Ausfallwahrscheinlichkeiten des PPS
  - Failure upon demand ⇝ f/d
  - Ausfall eines einzelnen Kanals: $10^{-3} f/d$
  - Insgesamt (das redundante System aus vier Kanälen): $10^{-4} f/d$

→ Ausfallwahrscheinlichkeit: $10^{-5} f/a$ ($\equiv$100 000a)

☞ Vierkanaliger, redundanter Aufbau des PPS

- Außerdem wird sichergestellt, dass maximal ein Kanal gewartet wird

⚠ Darüber hinaus: Jeder unsichere Zustand führt zur RESA

- Auch wenn das PPS nicht mehr aktiv in der Lage ist, dafür zu sorgen

→ Passivität der Systeme hat Auslösung der Sicherheitsfunktionen zur Folge

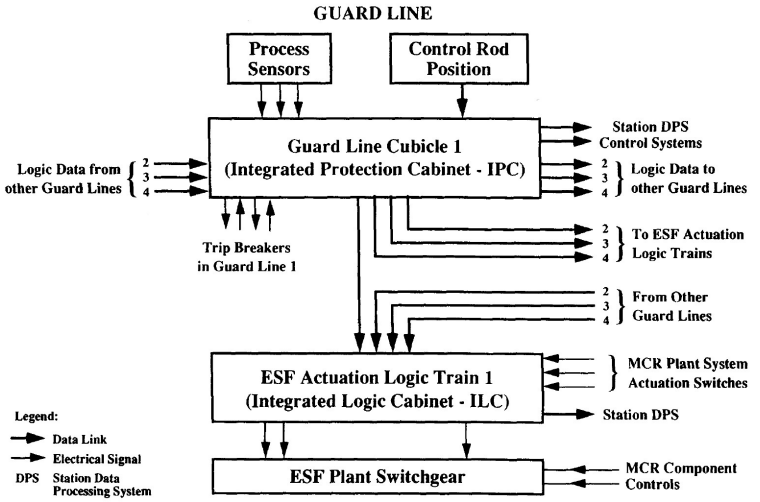
Aufbau des primären Schutzsystems

- 4-fach redundante Sicherheitsleittechnik
  - Redundanz umfasst jeweils Sensorik, Berechnung und Aktuatoren
  - → Die Replikation umfasst den kompletten Kontrollpfad (engl. *guardlines*)
  - Einzelne Redundanzen sind räumlich separiert
  - Aufstellorte der Kontrollrechner, Kabelkanäle, Stromversorgung, ...
  - → Vermeidung von Gleichtaktfehlern durch Umwelteinflüsse
- Unabhängige Arbeitsweise der einzelne Replikate
  - Sie bestimmen eigenständig ob eine RESA vonnöten ist
  - → Durchführung der RESA wird durch Mehrheitsentscheid ermittelt
  - Jedes Replikat führt den Mehrheitsentscheid selbst durch
  - Die Logik des Mehrheitsentscheids bezieht sich auf einen Wahrheitswert
  - → Implementierung durch einen dedizierten Schaltkreis

⚠ Notwendige Kommunikation erfolgt über optische Medien
- Keine gegenseitige elektrische Beeinflussung
- Keine Störungen durch elektromagnetische Interferenz

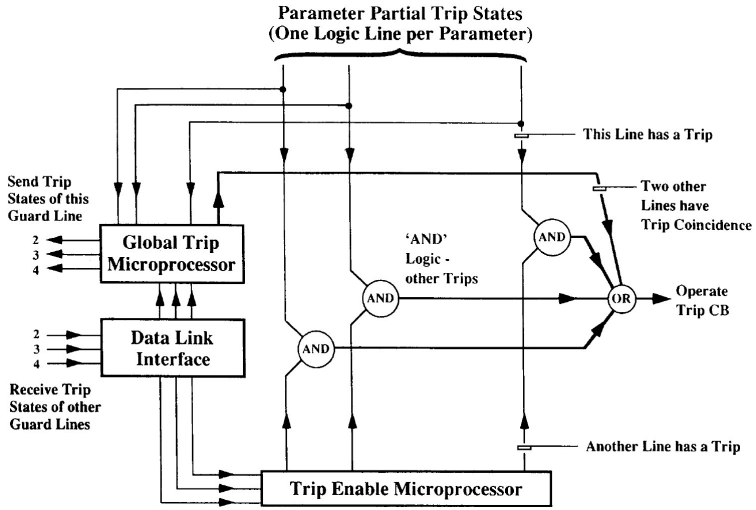
Eine Guardline des primären Schutzsystems

Quelle Grafik: [2]



Implementierung des Mehrheitsentscheids

Quelle Grafik: [2]



Softwareverifikation

- Verifikation und Validierung bestand aus verschiedenen Aktivitäten:

Engineering Confirmatory Analysis — NNC Ltd.
- Begutachtung (engl. *review*) relevanter Entwicklungsdokumente
- Anforderungen/Spezifikationen für System/Code, Quellcode und -daten

Independent Design Assesment — Nuclear Electric
- Überprüfung der Systemanforderungen in Systementwurf/-spezifikation
- Einbeziehung von Software-Entwurf und -Spezifikation

MALPAS Analysis — TA Consultancy Services Ltd.
- Formale Verifikation der Softwareimplementierung mit MALPAS

Object/Sourcre Code Comparison — Nuclear Electric
- Nachweis der Äquivalenz zwischen Binär- und Quellcode mit MALPAS

Dynamic Testing — Rolls Royce and Associates Ltd.
- Durchführung von ca. 55 000 zufällig erzeugten Testfällen

⚠ Geschätzter Aufwand: 250 Mannjahre
- In etwa derselbe Aufwand wurde bereits von Westinghouse investiert

MALPAS

MALvern **P**rogramming **A**nalysis **S**uite

- Entwicklung durch Royal Signals and Radar Establishment
  - Forschungseinheit des britischen Verteidigungsministeriums
  - Stationierung in Malvern (Worcestershire) $\rightsquigarrow$ Namensgebung
- besteht aus folgenden Analysewerkzeugen

Kontrollflussanalyse $\mapsto$ Kontrollflussgraph ...
  - Schleifen, Ein-/Ausstiegspunkte, Reduzierbarkeit, ...

Datenflussanalyse $\mapsto$ erreichende Definitionen, ...
  - Verwendung nicht initialisierter Daten, nie geschriebene Ausgaben

Informationsflussanalyse (engl. *program dependency graph*)
  - Daten- und Kontrollflussabhängigkeiten von Ausgabevariablen

Semantische Analyse $\mapsto$ symbolische Ausführung
  - Funktionale Zusammenhänge zwischen Ein- und Ausgaben

Einhaltung von Vor- und Nachbedingungen
  - (engl. *Compliance analysis*)

Softwareverifikation mit MALPAS [4]

- Zu prüfen: Softwareimplementierung des PPS
  - Implementierung in PL/M-86 und ASM86 bzw. PL/M-51 und ASM51
  - Umfasst ingesamt ca. 100 000 *Lines of Code*
    - Ca. 40 000 Zeilen für einen Hauptprozessor, ca. 10 000 bei Hilfsprozessoren
    - Anwendung, Betriebssystem, Kommunikation, Selbsttest, ...
- Referenz: Anforderungs- und Entwurfsdokumente
  - Software Design Requirements (SDR)
    - Abstrakte Beschreibung der von der Software zu erbringenden Funktionalität
  - Software Design Specification (SDS)
    - Architekturelle Umsetzung der funktionalen Anforderungen
    - Enthält detaillierte Information zur Funktion einzelne Softwarekomponenten
    - Beschreibt bereits alle Programmvariablen, sowie Ein- und Ausgaben

☞ Ablauf: Verifikation erfolgt Prozedur für Prozedur (engl. *unit proof*)
  - Aufgerufene Prozeduren werden durch geeignete Platzhalter ersetzt
  - Beginnend bei Blattprozeduren

MALPAS-Analysis [4] (Forts.)

⚠ MALPAS verwendet eine eigene Zwischensprache: MALPAS IL
  - Für den PL/M-86-Code wurde ein eigener Übersetzer entwickelt
  - Problem: MALPAS IL unterstützt anders als PL/M-86 keine Zeiger

→ Lösung: Dereferenzierung per Zeiger angesprochener Objekte
  - Kodierrichtlinien $\rightsquigarrow$ eingeschränkte Verwendung von Zeigern
  - → Dereferenzierung erfolgt größtenteils automatisiert, teilweise manuell

- Semantische Analyse $\rightsquigarrow$ Extraktion funktionaler Zusammenhänge
  - Ergebnis ist der mathematische Zusammenhang: Eingabe $\mapsto$ Ausgabe

→ Manueller Abgleich mit den Anforderungen/der Spezifikation

- Formulierung von Vor- und Nachbedingungen in MALPAS IL
  - Ansatz: primäre Quelle SDR, Verfeinerung mithilfe von SDS
    - → Schwierig wegen unterschiedlich detaillierter SDR/SDS
  - Analyse war sehr mühsam $\rightsquigarrow$ alternative Formulierungen waren oft nötig
    - Ungünstiger, schwer zu vereinfachender Ausdruck ließ Analyse scheitern
    - → Neuformulierung wies der algebraischen Vereinfachung den Weg

Qualitätssicherung und Ergebnisse

- Problem: korrekte Formulierung von Vor-/Nachbedingungen

1. Standardisierter Analyseprozess (ISO 9001)
2. Detaillierte Vorgehensbeschreibung für die Durchführung (ca. 200 Seiten)
3. Detaillierte Protokollierung der Analyse
   - Eingabe für die MALPAS-Analyse und ihre Ergebnisse
   - Für jede Analyse wurden vorgefertigte Formulare ausgefüllt
   - Ableitung der math. Spezifikation, Interpretation der Ergebnisse, ...
4. Umfangreiche gegenseitige Begutachtung (engl. *peer-review*)
   - Einhaltung des Prozesses, Verständnis des PPS erweitern
   - Überprüfung von Terminierungsbeweisen, Termersetzungsregeln, ...

☞ Ergebnisse: Abweichungen von der Spezifikation
  - Diese wurden kommentiert und kategorisiert

→ Lieferung von insgesamt ca. 2000 Kommentaren an Nuclear Electric

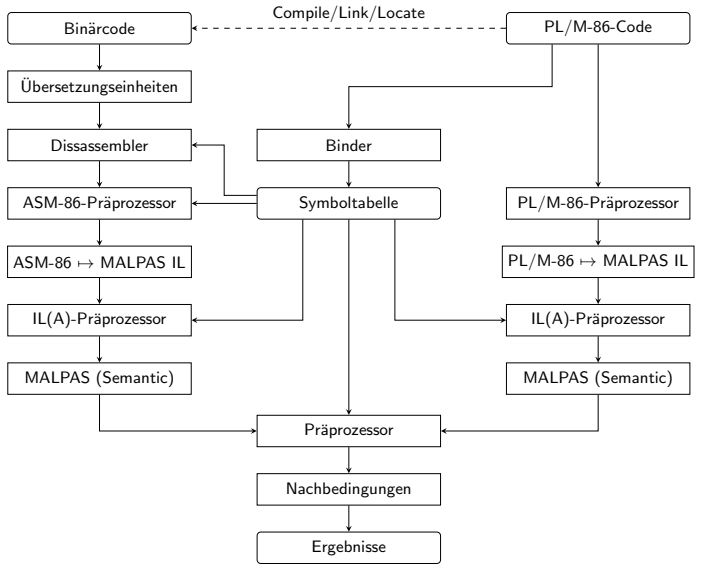
Kategorie 1	mögliche Fehlfunktion im PPS	$\rightsquigarrow$ keine
Kategorie 2	Änderungen in Anforderungen/Spezifikation	$\rightsquigarrow$ ca. 40%
Kategorie 3	nicht-kritische Änderungen am Quelltext	$\rightsquigarrow$ ca. 8%
Kategorie 4	keinerlei Änderung erforderlich	$\rightsquigarrow$ ca. 52%

Äquivalenz von Quell- und Binärcode [3]

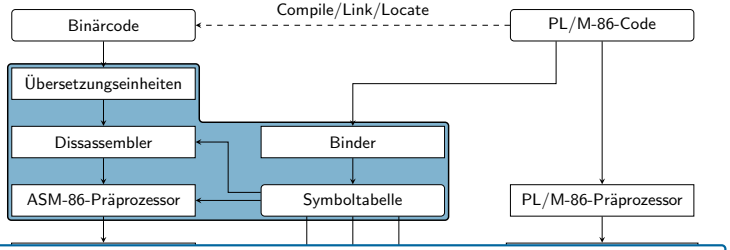
Traue Nichts und Niemandem, . . . auch nicht dem Übersetzer!

- Problem: Passt der Binärcode auch zum Quellcode?
  - Was hilft der korrekteste Quellcode, wenn der Übersetzer fehlerhaft ist?
  - Bewiesenermaßen korrekte Übersetzer existierten damals nicht
    - Nimmt man Assemblierer und Binder dazu, ist das auch heute noch so
  - Rekonstruktion des Quellcodes aus dem Binärcode ist nicht möglich
    - → Kein Vergleich originärer vs. rekonstruierter Quellcodes
- ☞ Idee: Man trifft sich in der Mitte ⇝ MALPAS IL
  - Übersetzer PL/M-86 ⇝ MALPAS IL existiert bereits
  - Übersetzer Binärcode ⇝ MALPAS IL entwickelt man noch
    - Rekonstruktion der Übersetzungseinheiten, Disassemblierung, . . .
  - Vergleich ↦ Verifikation der Nachbedingungen mit MALPAS
    - Quellcode ⇝ Extraktion von Nachbedingungen
    - Binärcode ⇝ Extraktion der Implementierung
- → Zu zeigen: die Implementierung erfüllt die Nachbedingung
  - Quell- und Binärcode sind identisch

Ablauf des Vergleichs: Quell- vs. Binärcode

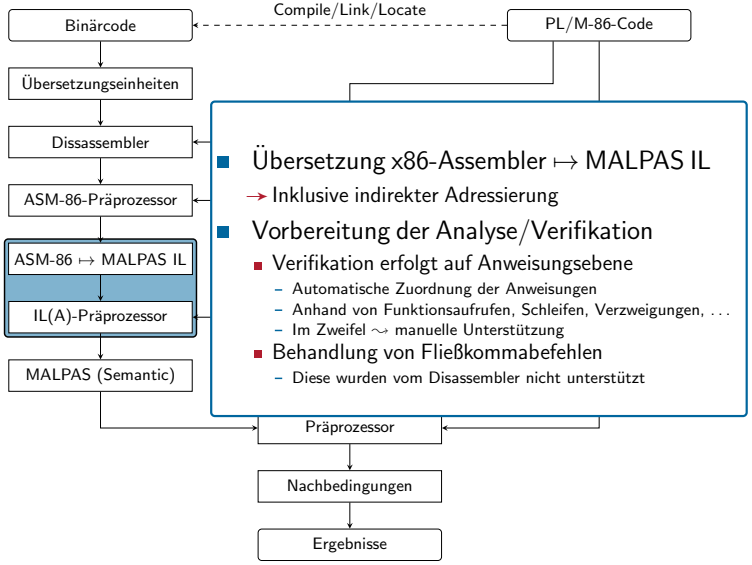


Ablauf des Vergleichs: Quell- vs. Binärcode



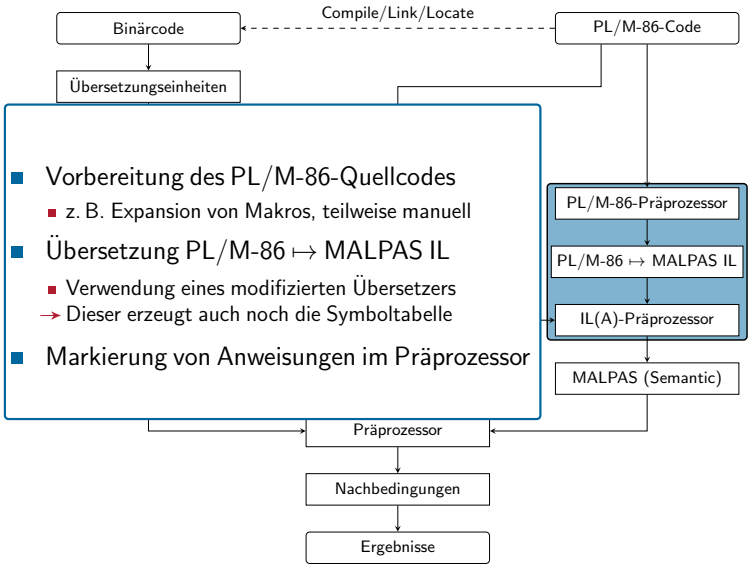
- Rekonstruktion von Text- und Datensektionen
  - → Ursprüngliche Übersetzungseinheiten
- Dissassemblierung des Binärcodes
- Aufbereitung des Assemblercodes
  - Unterstützt durch eine aus dem Quelltext bestimmte Symboltabelle
  - Bestimmung von globalen Variablen aus Adressen
  - Bestimmung von Funktionsparametern
  - Einfügen von Variablendeklarationen

Ablauf des Vergleichs: Quell- vs. Binärcode



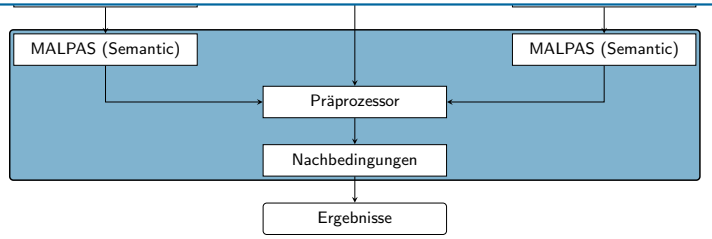
- Übersetzung x86-Assembler ↦ MALPAS IL
  - → Inklusive indirekter Adressierung
- Vorbereitung der Analyse/Verifikation
  - Verifikation erfolgt auf Anweisungsebene
    - Automatische Zuordnung der Anweisungen
    - Anhand von Funktionsaufrufen, Schleifen, Verzweigungen, . . .
    - Im Zweifel ⇝ manuelle Unterstützung
  - Behandlung von Fließkommabefehlen
    - Diese wurden vom Disassembler nicht unterstützt

Ablauf des Vergleichs: Quell- vs. Binärcode



Ablauf des Vergleichs: Quell- vs. Binärcode

- Funktionalen Zusammenhänge zwischen Ein- und Ausgabe
  - → Eingabe für die Prüfung der Nachbedingungen
    - MALPAS vergleicht nicht direkt den erzeugten MALPAS IL-Code
    - → Es stellt die extrahierten math. Zusammenhänge gegenüber
- Formulierung des Verifikationsproblems in MALPAS IL
  - Eliminierung verbliebener, problematischer Konstrukte
    - Speicherreferenzen durch indirekte Adressierung, Registerzuweisungen, temporäre Variablen
  - Zuordnung der Anweisungen durchführen: ASM-86 ↔ PL/M-86
    - ASM-86-Anweisungen werden zu Prozedurimplementierungen in MALPAS IL
    - PL/M-86-Anweisungen werden zu Nachbedingungen in MALPAS IL
- Überprüfung der Nachbedingungen durch MALPAS
  - Wurden keine Bedrohungen (engl. *threats*) gefunden, waren Binär- und Quellcode identisch



Ergebnisse und Bewertung des Ansatzes

⚠ 11 Abweichungen zwischen Binär- und Quellcode [1]
- Eine davon stellte sich als ernsthafter Defekt des Übersetzers heraus
- Ergebnisse wurde nicht offiziell veröffentlicht, sickerten jedoch durch

- Bewertung des Ansatzes

Generalisierbarkeit ⇝ Portierung für andere Programmiersprachen
- Ansatz ⇝ allgemein gehalten, Implementierung ⇝ sprachabhängig
- PL/M ist eine sehr einfache Sprache und erleichtert die Verifikation
  - Komplexere Sprachen könnten dieses Vorhaben erschweren
  - Optimierungen wie das Ausrollen von Schleifen etc. gar unmöglich machen

Automatisierbarkeit war in weiten Teilen gegeben
- Andere Teile erforderten aber signifikante manuelle Eingriffe
  - Insbesondere die Markierung von Anweisungen war problematisch

Formalität konnte nicht vollständig durchgehalten werden
- Insbesondere war die Abbildung von Ganzzahlen nicht 100%-ig korrekt
  - Alle Ganzzahlen wurden auf denselben MALPAS IL Ganzzahltyp abgebildet
  - Unabhängig von der Bitbreite (8-,16- oder 32-Bit) der Ganzzahl
  - Falls nötig, wurde diese Unterscheidung manuell eingebracht

Gliederung

Zusammenfassung

Sizewell B $\rightsquigarrow$ primäres Reaktorschutzsystem

- Einziger Zweck: sichere Abschaltung des Reaktors

Redundanz $\rightsquigarrow$ Absicherung gegen Systemausfälle

- 4-fach redundante Systeme

Diversität $\rightsquigarrow$ Abfedern von Software-Defekten

- Unterschiedliche Hardware und Software

Isolation $\rightsquigarrow$ Abschottung der einzelnen Replikate

- Technisch $\mapsto$ optische Kommunikationsmedien
- Zeitlich $\mapsto$ nicht-gekoppelte, eigenständige Rechner
- Räumlich $\mapsto$ verschiedene Aufstellorte und Kabelrouten

Verifikation $\rightsquigarrow$ umfangreiche statische Prüfung von Software

- Vielschichtiger Prozess, Betrachtung von Quell- und Binärcode

Literaturverzeichnis

[1] Buttle, D. L.:
Verification of Compiled Code.
Eindhoven, The Netherlands, University of York, Diss., Jan. 2001. – 262 S.

[2] Moutrey, G. ; Remley, G. :
Sizewell B power station primary protection system design application overview.
In: *International Conference on Electrical and Control Aspects of the Sizewell B PWR*, 1992. –
ISBN 0–85295–550–8, S. 221–231

[3] Pavey, D. J. ; Winsborrow, L. A.:
Demonstrating Equivalence of Source Code and PROM Contents.
In: *The Computer Journal* 36 (1993), Apr., Nr. 7, S. 654–667.
http://dx.doi.org/10.1093/comjnl/36.7.654. –
DOI 10.1093/comjnl/36.7.654

Literaturverzeichnis (Forts.)

[4] Ward, N. J.:
The Rigorous Retrospective Static Analysis of the Sizewell 'B' Primary Protection System Software.
In: Górski, J. (Hrsg.): *Proceedings of the 12th International Conference on Computer Safety, Reliability, and Security (SAFECOMP '93).*
Heidelberg, Germany : Springer-Verlag, Okt. 1993. –
ISBN 3–540–19838–5, S. 171–181