aramis

AUTOMOTIVE · RAILWAY · AVIONICS

MULTICORE SYSTEMS

ARAMiS Schlussberichte

Version	1.1
Laufzeit des Vorhabens	01.12.2011 - 31.03.2015
Förderkennzeichen	01IS11035
Verbreitung	Öffentlich
Fälligkeitsdatum	16.11.2015
Datum	16.11.2015

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Versionen

Version	Datum	Beschreibung
1.0	16.11.2015	Erste Version
1.1	16.11.2015	Öffentliche Version

Verbreitung

Der vorliegende Bericht ist öffentlich.

Projektkoordination

Prof. Dr.-Ing. Jürgen Becker / Dr.-Ing. Oliver Sander
Karlsruher Institut für Technologie (KIT)
Institut für Technik der Informationsverarbeitung (ITIV)
Engesserstr. 5
76131 Karlsruhe
Telefon: +49 721 - 608 - 42502 / - 42512
Telefax: +49 721 - 608 - 42511
Email: becker@kit.edu / sander@kit.edu

Vorbemerkungen

Das vorliegende Dokument beinhaltet neben einer partnerübergreifenden Darstellung in Part I, die Schlussberichte der folgenden Verbundpartner von ARAMiS in Part II:

1. Airbus Group Innovations
2. Audi Electronics Venture GmbH
3. Robert Bosch GmbH
4. Daimler AG
5. Fortiss GmbH
6. Fraunhofer AISEC und Fraunhofer IESE
7. Intel Deutschland GmbH
8. Karlsruher Institut für Technologie
9. Liebherr-Aerospace Lindenberg GmbH
10. OFFIS e.V.
11. Symtavision GmbH
12. SYSGO AG
13. Technische Universität Braunschweig
14. Technische Universität Kaiserslautern
15. Technische Universität München (EISEC, LIS, INSEC)
16. Technische Universität München (SSE)
17. Universität Stuttgart (IPVS)
18. Universität Stuttgart (ISTE)
19. Universität Paderborn

Zusammenfassung

Multicore-Technologie ist heute in Rechenzentren, PCs, Laptops, Tablets und Smartphones, die hauptsächlich Durchsatz orientierten arbeiten, weit verbreitet und etabliert. Im Kontext von sicherheitskritschen oder gemischt-kritischen Systemen finden Multicore-Architekturen bisher kaum Verwendung, obwohl auch bei aktuellen und zukünftigen Anwendungen in den Mobilitätsdomänen eine hohe Nachfrage an zusätzlicher Rechenleistung besteht. Der Grund hierfür liegt in den besonderen Herausforderungen beim Übergang von sequentieller zu paralleler Ausführung, welche den Einsatz von Multicore-Technologie bisher verhindern. Insbesondere die große Anzahl gemeinsam genutzter Ressourcen in eingebetteten Multicore-Architekturen führt zu zeitlichen und räumlichen Interferenzen und damit zu einem nicht deterministischen Systemverhalten. Zum Beispiel das Deployment von Software auf Multicore-Prozessoren stellt eine enorme Herausforderung bzgl. Effektivität und Effizienz dar.

Sowohl der Automobilbereich, die Bahn als auch die Luftfahrt sind Domänen, welche genau diesen Herausforderungen bei der Entwicklung ihrer nächsten Produktgenerationen entgegensehen und bei denen die benötigte Rechenleistung einzig durch Multicore-basierte Systeme bereitgestellt werden kann. Im vom BMBF geförderten Projekt **A**utomotive **R**ailway **A**vionics **M**ult**i**core **S**ystems haben führende Industrieunternehmen und Forschungseinrichtungen mit vereinten Kräften domänenübergreifende Lösungen entwickelt, welche die Verwendung von Multicore-Architekturen in sicherheitskritischen Anwendungen ermöglichen. Das Konsortium hat die Herausforderungen der Multicore-Technologie umfassend analysiert und zahlreiche Lösungen für den Bereich der genannten Mobilitätsdomänen entwickelt. Die erarbeiteten Ansätze adressieren u.a. Safety, Security, Migration, Segregierung, Deployment, Legacy-Code und Virtualisierung. Die Anwendbarkeit der Lösungen konnte durch verschiedene industrielle Demonstratoren gezeigt werden. Die Betrachtung von Industrie-relevanten Anforderungen inkl. der Unterstützung von Legacy-Systemen ist eine wesentliche Stärke des ARAMiS Projekts und stellt die praktische Anwendbarkeit der bereitgestellten Lösungen sicher.

Inhaltsverzeichnis

I. Partnerübergreifende Darstellung

1 Aufgabenstellung

ARAMiS hatte zum Ziel, durch den Einsatz von Multicore-Technologie in den Mobilitätsdomänen Automobil, Avionik und Bahn die technologische Basis zur weiteren Erhöhung von Sicherheit, Verkehrseffizienz und Komfort zu schaffen. Die gewonnenen Erkenntnisse bilden das unabdingbare Fundament für die erfolgreiche Vernetzung von Embedded Systems zu Cyber Physical Systems (CPS). Das Projekt leistete einen wichtigen Beitrag zum Erhalt und zur Stärkung der weltweiten Wettbewerbsfähigkeit deutscher Unternehmen der Domänen Automobil, Avionik und Bahn.

Die Arbeiten in ARAMiS folgten einem typischen, im industriellen Umfeld verbreiteten Forschungs- und Entwicklungsansatz: Startpunkt war die Untersuchung von Anwendungsszenarien. Hieraus wurden Funktionsanforderungen abgeleitet und ein Systementwurf erarbeitet. Es folgte die detaillierte Bearbeitung von Hardware- und Software-Aspekten mit der abschließenden Darstellung und Evaluierung der Projektergebnisse durch die Demonstratoren. Parallel hierzu wurde an durchgängigen Entwicklungsmethoden und einer Reference Technology Plattform (RTP) gearbeitet.

1.1 Domänenübergreifende Ansätze

Jede der betrachteten Domänen (Automotive, Avionik und Bahn) hat eigene Anforderungen an Multicore-Architekturen. Viele der Anforderungen überlappten sich jedoch auch und ermöglichten so das gemeinsame Erarbeiten von domänenübergreifend einsetzbaren Lösungen. Die Identifikation von domänenübergreifenden Anforderungen und die Erarbeitung von entsprechenden gemeinsamen Lösungen waren in ARAMiS von größter Wichtigkeit.

In Zusammenarbeit mit allen Domänen sowie den Partnern aus der Wissenschaft konnten wesentliche Grundsatzfragen untersucht werden, die das Vorgehen für zukünftige Entwicklungen verbessern und absichern.

1.2 Domäne Automotive

Ausgangspunkt der Anforderungsentwicklung für die Domäne Automotive waren Thesen und Szenarien. Nach Konsolidierung

und Bewertung wurden aus den wichtigsten Szenarien notwendige Funktionen und hieraus funktionale bzw. nichtfunktionale Anforderungen abgeleitet, die von einer technischen Architektur basierend auf Multicore und Virtualisierung zu erfüllen sind.

Unter Berücksichtigung von CPS-Vernetzungsszenarien und einer technischen Fahrzeugarchitektur, die die funktionale Hochintegration auf wenige Steuergeräte anstrebt, ergaben sich im Projekt die folgenden Themenschwerpunkte:

- Konzeption Automotive-tauglicher Integrationsplattformen zur Funktionspartitionierung unter Berücksichtigung der aktuellen und zukünftigen Fahrzeugarchitekturen.
- Analyse von Safety / Qualifizierungsaspekten unter Nutzung von Multicore im Kontext funktionaler Hochintegration für verschiedene Fahrzeugdomänen.
- Analyse der Systemarchitektur unter dem speziellen Aspekt der Safety und Qualifizierbarkeit für Fail Operational-Systeme. Qualifizierbarkeit der Ausfallsicherheit / Hochverfügbarkeit der Funktion des Gesamtsystems durch Kombination von Hardware- und reiner Software-Redundanz. Erarbeitung eines Konzeptes unter Gewährleistung der Einhaltung der Echtzeitanforderungen der Funktion bei Verlagerung.
- Bewertung von Chip/System-Design, Absicherung und Produktion (Prozesse, on-Chip Safety Features für ASIL-x Unterstützung) im Kontext ISO 26262.
- Bewertung von Low Level Software bei sicherheitskritschen Applikationen (ASIL-C/D) und deren Integration in AUTOSAR.
- Darstellung von Abhängigkeiten zur optimierten Auslastung einer Multicore Architektur (Visualisierung, Optimierungsvorschläge). Klärung der Auswirkung der Multicore Architektur auf den Software-Entwicklungsprozess. Methoden der Migration existierender Software-Assets auf Multicore Architekturen.
- Entwicklung von Security-Methoden zum Schutz gegen äußere Angriffe unter Berücksichtigung von Vernetzungsszenarien im Sinne von Cyber Physical Systems.
- Bewertung bestehender Virtualisierungstechnologien unter den Aspekten Safety und Security zur Unterstützung der funktionalen Hochintegration in verschiedenen Fahrzeugdomänen. Erarbeitung von Konzepten zur Erfüllung der automotiven Anforderungen.

1.3 Domäne Avionik

Ausgehend von den Anwendungsszenarien "Free Flight" sowie „Kabinenmanagement“ wurden die Anforderungen für die Avionik-Anwendungen definiert. Darauf basierend wurden dann unterschiedliche Architekturkonzepte entwickelt und auf ihre Zertifizierbarkeit hin analysiert.

Die untersuchten Ansätze waren dabei:

- Verwendung von "Standard" Multicore-Prozessoren in Kombination mit sicherem Monitor.
- Verwendung einer "einfachen Ablaufsteuerung" für den Multicore-Prozessor, die den sicheren Betrieb gewährleisten kann.
- Entwicklung einer neuartigen Sicherheits-Architektur für Multicore-Prozessoren, die durch ihre Struktur die funktionale Sicherheit sowie das deterministische Verhalten sicherstellt und die Bestimmung der WCET (Worst-Case Execution Time) und Daten-Latenz ermöglicht.

Weiterhin wurden die folgenden Architekturaspekte berücksichtigt:

- Parallelisierung und Verteilung sicherheitskritischer echtzeitfähiger Anwendungen auf heterogenen verteilten Architekturen.
- Referenzarchitekturen, die einerseits Garantien für diese Vielfalt von Qualitätsmerkmalen von Diensten geben, andererseits eine Virtualisierung von konkreten Zielarchitekturen unterstützen.
- Ko-Allokation von sicherheitsrelevanten und von unkritischen Teilfunktionen auf dem Multicore-Prozessor.
- Kosteneffizientes Beherrschen der Komplexität.
- Produktlinienübergreifende Harmonisierung von skalierbaren Architekturansätzen.
- Anbindung an bestehende und zukünftige Reference Technology Platform.
- Erweiterbarkeit/Skalierbarkeit auf Applikationsebene.
- Zertifizierbare Lösungsansätze für sicherheitskritische Software-Regel- / Steuerfunktionen.
- Performanz, Kosten, Sicherheit und Zuverlässigkeit.

Die Architekturkonzepte wurden in den Arbeiten von der Systemebene bis zu Hardware- und Softwareelementen verfeinert. Die wesentlichen Ergebnisse konnten im Rahmen des Projektes mittels eines Avionik-Demonstrators, der im Kern ein Avionik-Prozessor (Multicore) auf einer Leiterkarte ist, validiert werden. Der Demonstrator konnte essentielle ausgewählte Aspekte für eine Zertifizierbarkeit durch die relevanten Behörden aufzeigen.

Ein weiteres Ziel dabei war, die derzeitige Performanz eines Hochsicherheitsprozessors um etwa den Faktor 3 zu steigern - dies bei vergleichbaren Bauvolumen und vergleichbarer Leistungsaufnahme, d.h. bei vorgegebenem physikalischem Umfeld. Die Ansätze für die Verwendung von "Standard" Multicore-Prozessoren wurden schwerpunktmäßig im Demonstrator zum experimentellen Kabinenmanagement-Server untersucht.

1.4 Domäne Bahn

Die generelle Vorgehensweise der Domäne Bahn folgte der in den Domänen Automotive und Avionik: Die Anforderungen an eine zertifizierbare Multicore-Plattform für eine Konsolidierung von Bahnanwendungen wurden ausgehend von den Szenarien definiert. Basierend auf den Anforderungen wurden entsprechende Architekturen entwickelt. Die zentralen Elemente der Architektur waren dabei eine Multicore-Prozessorplattform sowie eine Virtualisierungslösung. Dabei ergaben insbesondere sich die folgenden Fragestellungen:

- Ko-Allokation von sicherheitsrelevanten und von unkritischen Teilfunktionen auf dem Multicore-Prozessor.
- Kosteneffizientes Beherrschen der Komplexität.
- Erweiterbarkeit und Skalierbarkeit auf Applikationsebene unter effizienter Ausnutzung von Virtualisierungstechniken.
- Zertifizierbare Lösungsansätze für sicherheitskritische und echtzeitfähige Software-Regel-/Steuerfunktionen.
- Performanz, Kosten, Sicherheit und Zuverlässigkeit.

2 Voraussetzungen

In der Fahrzeugindustrie, in der Avionik und bei der Bahn werden neue Funktionen zur Realisierung von mehr Sicherheit, Komfort, zur Erhöhung der Verkehrseffizienz und Energieeinsparung über elektronische Steuergeräte realisiert. Zentraler Bestandteil der Steuergeräte sind Mikrocontroller, in denen ein einzelner Prozessorkern (Singlecore-Prozessor) die Softwarealgorithmen ausführt. Der in der Vergangenheit verfolgte Ansatz zur Integration neuer Funktionen war ein zusätzliches Steuergerät vorzusehen. Dieser Ansatz ist bedingt durch Bauraum und Kosten an seine Grenzen der Skalierbarkeit gestoßen. Dies bedeutet, dass zukünftige Steuergeräte mehr Funktionen gleichzeitig ausführen müssen. Darüber hinaus benötigen fortschrittliche Funktionen durch zumeist aufwendige Signalverarbeitung oder zunehmende

Vernetzung mit anderen Funktionen und Steuergeräten eine hohe Rechenperformanz.

Die bisher in Steuergeräten verwendete Singlecore-Technologie stößt hier an ihre Leistungsgrenze: Eine weitere Erhöhung der Performanz durch steigende Taktraten ist insbesondere aufgrund der Verlustleistungsdichte und der damit verbundenen Problematik der Wärmeabfuhr nicht möglich. Die durch die Singlecore-Technologie vorgegebenen Grenzen stehen somit letztendlich der Integration fortschrittlicher Funktionen in Fahrzeugen, Flugzeugen und bei der Bahn und damit der weiteren Erhöhung von Sicherheit, Komfort und Verkehrseffizienz entgegen.

In klassischen IT Bereichen werden wegen der bekannten Leistungsgrenze von Singlecore-Prozessoren seit einiger Zeit Multicore-Prozessoren verwendet. Dort steigt die zur Verfügung stehende Rechenleistung eines Prozessors (CPU) mit der Anzahl der Rechenkerne (Cores). Eine vergleichbare Entwicklung ist vermehrt auch bei Embedded Systems zu beobachten, die einen Anteil von über 90% aller weltweit eingesetzten Prozessoren ausmachen. Der relative Anteil der in den Mobilitätsdomänen Automobil, Avionik und Bahn eingesetzten Bausteine beträgt lediglich unter 10%. Daher orientieren sich Entwicklung- und Herstellung von Multicore-Hardwarebausteinen eher an den volumenstarken Märkten Consumer, Entertainment und Communications. In diesen Domänen ist der Paradigmenwechsel von Single- zu Multicores in Teilen bereits erfolgreich vollzogen.

Multicore-Systemen müssen für den Einsatz in den Mobilitätsdomänen Automobil, Avionik und Bahn weitreichende und spezifische funktionale und nicht funktionale Anforderungen erfüllen, die über diejenigen des General Purpose Computings weit hinausgehen. Diese sind insbesondere:

- Echtzeitfähigkeit,
- Leistungsfähigkeit,
- Zuverlässigkeit und Verfügbarkeit,
- zertifizierbare Funktionssicherheit (Safety),
- Sicherheit gegen Angriffe (Security),
- Kompatibilität zu bestehenden Konzepten,
- Energieeffizienz

Die Übertragung von Lösungen aus Domänen, die Multicore-Systeme bereits erfolgreich einsetzen (z.B. aus dem General Purpose Computing), ist nur begrenzt möglich. Für die Mobilitätsdomänen müssen zur Erfüllung der genannten Anforderungen spezialisierte Architekturen und Methoden entwickelt werden. Auch bereits existierende Erfahrungen von Singlecore-Lösungen sind nur sehr eingeschränkt übertragbar, da

Multicores aufgrund Ihrer parallelisierenden Eigenschaften erheblich komplexere Systemzustände erlauben, die damit die Sicherstellung oben genannter Eigenschaften erschweren.

3 Planung und Ablauf des Vorhabens

Abbildung 1 zeigt den Ablauf des ARAMiS Projekts anhand seiner Teilprojekte (TP). Gestartet wurde mit der Erarbeitung von Szenarien und Anforderungen (TP 1). Hierauf aufbauend wurden der Systementwurf (TP 2), die HW- und SW- Entwicklung (TP 3 / TP 4) und schließlich die Demonstratoren (TP 6) angegangen. Fragestellungen zu Architekturen, Safety, Zertifizierbarkeit, Security und Virtualisierung sind auf verschiedenen Systemebenen relevant, aber nicht einzelnen Modulen direkt zuordenbar. Sie wurden daher auf System-, Hardware- und Softwareebene betrachtet. Die entsprechende Durchgängigkeit wurde über das zugehörige Teilprojekt (TP 5) sichergestellt, das auch Fragestellungen zu Werkzeugen und Methoden übergreifend untersuchte.

Die Arbeiten verliefen im Wesentlichen wie geplant.

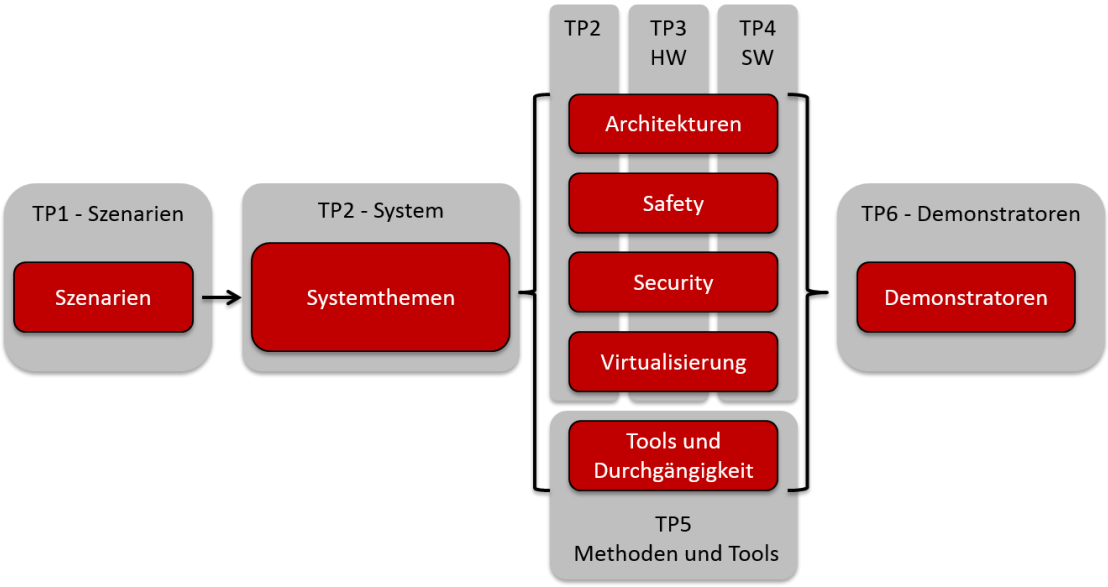

Abbildung 1 ARAMiS Projektstruktur

Die Arbeitsinhalte der einzelnen TPs und ihre technischen und wissenschaftlichen Ansätze umfassen:

- **TP 1 Szenarien und Anforderungen:** In TP 1 wurden die relevante Szenarien für den zukünftigen Einsatz von Multicore-Technologie in den Mobilitätsdomänen Auto-

motive, Avionik und Bahn definiert und verfeinert. Hierauf aufbauend wurden die funktionalen und nicht-funktionalen Anforderungen an zukünftige Multicore-Systeme und Virtualisierungslösungen abgeleitet.

- **TP 2 Systementwurf:** In TP 2 wurde auf der Basis der in TP 1 erhobenen Anforderungen die Einbettung zukünftiger Multicore-Systeme in Netzwerke analysiert. Ziel war Systemarchitekturen für Multicore-Systeme zu erarbeiten. Besonders berücksichtigt wurden hierbei die Anforderungen aus den Bereichen Safety / Zertifizierbarkeit und Security. Die Ergebnisse aus TP 2 wurden in TP 3 (Hardware), TP 4 (Software) und TP 5 (Methoden und Tools) aufgegriffen.
- **TP 3 Hardware:** TP 3 entwickelte die Hardware basierend auf den in TP 1 ermittelten Anforderungen und der in TP 2 spezifizierten Systemarchitekturen. Besondere Schwerpunkte der Arbeiten bildeten heterogene Hardware-Architekturen, Security, Safety / Zertifizierbarkeit sowie Virtualisierungsunterstützung.
- **TP 4 Software:** TP 4 entwickelte Software-Architekturen für den Einsatz von Multicore-Technologie und Virtualisierung im Rahmen der Anforderungen und des Systementwurfs aus TP 1 bzw. TP 2 und in enger Abstimmung mit TP 3. Im Fokus standen dabei anwendungsspezifische Fragestellungen wie beispielsweise die Weiterverwendung von Legacy Code, der Umgang mit Multicore-Hardware auf Middleware- und Betriebssystemebene, Security und Safety / Zertifizierbarkeit, sowie Realisierungskonzepte für Virtualisierung unter den gegebenen Randbedingungen.
- **TP 5 Durchgängige Entwicklungsmethodik und Werkzeuge:** TP 5 widmete sich dem Entwurf durchgängiger Methoden und der Betrachtung von Werkzeugen, die beim Design von Multicore-Systemen unterstützen. Es wurde dementsprechend in enger Abstimmung mit TP 2, TP 3 und TP 4 durchgeführt.
- **TP 6 Demonstratoren:** TP 6 hatte die Validierung der Architekturvorlagen für System, Hard- und Software aus TP 2, TP 3 und TP 4 und der entsprechenden Methoden und Tools aus TP 5 zum Ziel. Durch Demonstratoren konnte die Funktionsfähigkeit sowie Safety und Security der Konzepte nachgewiesen werden.

Die oben genannten TPs untergliedern sich weiter in Arbeitspakete (APs) und Tasks, eine vollständige Liste dieser ist in der ARAMiS Vorhabensbeschreibung des Projekts zu finden.

4 Wissenschaftliche- und technische Ausgangslage

4.1 Stand der Technik in der Domäne Automotive

Die Elektrik/Elektronik-Architektur aktueller Fahrzeuge besteht, je nach Fahrzeugklasse und Ausstattung aus einer unterschiedlichen Anzahl von Sensoren, Aktoren und bei Vollausstattung aus über 70 funktionsspezifischen, Singlecore-Steuergeräten, die funktional in Domänen gekapselt und über unterschiedliche Systembusse gekoppelt sind.

Die schwer beherrschbare Vernetzungskomplexität des evolutionär gewachsenen Bordnetzes, der Kostendruck, die weiter anhaltende Funktionsmehrung und die Tatsache, dass neue Funktionen z.B. der Fahrdynamik oder Fahrerassistenz sich gegenseitig beeinflussen, hat zu einer partiellen und fahrzeug-domänenspezifischen Zentralisierung der Funktionen auf weniger Steuergeräte geführt. Beispielhaft seien hierfür die Fahrzeugdomänen Infotainment, Fahrdynamik und Chassis genannt:

- Infotainment: Das zentrale Steuergerät der Infotainment-Domäne ist die sogenannte Headunit. Sie ist innerhalb des Fahrzeugs aktuell das Steuergerät mit der größten Rechenleistung und in der Regel als Mehrprozessorsystem ausgelegt. Die Headunit ist die zentrale Komponente für die Steuerung verschiedener Audio- und Videoquellen. Sie beherbergt sowohl fahr- bzw. zeitkritische Anwendungen wie z. B. die Ausgabe der PDC-Signale (Park-Distance Control), der Warnsignale (Gongs) oder der Rückfahrkamera als auch vielfältige Entertainment- und Telematikfunktionen. Zukünftige Konzepte sehen eine weitere Zentralisierung von Funktionen auf der Headunit vor, um die Komplexität des E/E-Gesamtsystems zu reduzieren und Kosten einzusparen. Als Beispiel sei hier die Anzeigensteuerung genannt: Eine zentrale Anzeigensteuerung, die kundengerecht, der aktuellen Fahrsituation angepasst und flexibel Informationsausgabe über Kombi, Headup-Display und zentrales Display steuern würde bringt ASIL-relevante Funktionalität wie die Steuerung der Check-Control-Meldungen auf die Headunit. Denkt man parallel an eine umfangreichere, offene Fahrzeugvernetzung erhält man einen komplexen Mix aus nicht-funktionalen Anforderungen bezüglich Security, Safety, Zeitkritikalität als auch Zertifizierbarkeit der mit den aktuellen Lösungsansätzen in einem machbaren Kostenrahmen nicht mehr ohne weiteres zu realisieren ist.

- Fahrdynamik: Die Weiterentwicklung der Fahrdynamik-Regelsysteme (Hoch-, Quer-, Längsdynamik), die sich durch eine komplexe Vernetzung von Sensorik, Steuergerät und Aktorik auszeichnet und deren Funktionsziele sich gegenseitig beeinflussen, erfordern eine neue Systemarchitektur um ungewünschte Redundanzen in Sensorik bzw. unerwünschte Funktions-Interferenzen zu vermeiden. Dies führt zu einer Zentralisierung von Regelfunktionen in Steuergeräte. Die Tendenz wird weiter gefördert durch die Vernetzung bisher unabhängiger Systeme zum Beispiel aus dem Bereich aktive und passive Sicherheit, um neue Generationen von Sicherheitssystemen zu ermöglichen.
- Chassis: Zur Reduktion von Kabellängen und Gewicht und damit zur Ausschöpfung von Kosten- und Energiepotentialen ist auch hier eine bauraumorientierte Zentralisierung von Funktionen zu beobachten (kurze Wege Sensorik / Aktorik).

Zur Erreichung der notwendigen Rechenleistung der zentralisierten Komponenten wurden bisher mehrere bzw. leistungsfähige Prozessoren in einem Steuergerät verbaut, wobei die Leistungsfähigkeit durch die Erhöhung der Taktrate gesteigert wurde. Diesem Vorgehen sind aber durch die Power-Wall (physikalische Grenze der möglichen Energieentsorgung für Transistoren) Grenzen gesetzt. Dies macht den Einsatz von Multicore-Systemen unabdingbar. Vor dem Hintergrund einer umfangreicheren, offenen Fahrzeugvernetzung müssen Multicore-Systeme in Fahrzeugen einen komplexen Mix an nicht-funktionalen Anforderungen bezüglich Security, Safety, Echtzeitfähigkeit und Zertifizierbarkeit erfüllen. Hierzu existierte zu Beginn des Projekts in der Fahrzeugdomäne kein Lösungsansatz.

4.2 Stand der Technik in der Domäne Avionik

Die Rolle der Elektronik ist bereits heute essentiell für alle relevanten Avionik Systeme: Ohne Embedded Systems kann heute kein Verkehrs- oder Militär-Flugzeug mehr fliegen. So sind bereits heute über 1700 einzelne Prozessoren in einem Langstreckenflugzeug verbaut. Ein wichtiges Ziel ist, die Anzahl der eingebetteten Rechner in einem Flugzeug zu reduzieren und gleichzeitig für zukünftige Funktionen die verfügbare Rechenleistung und Energieeffizienz zu steigern. Für zukünftige Vernetzung spielt Security eine wichtige Rolle. Weniger Rechner in einem Flugzeug bedeutet unter anderem Gewichts- und Energieersparnis und Reduktion der Netzwerkskomplexität. Dieses Ziel lässt sich nur durch die Verwendung von hochintegrierten Mehrprozessor-Elementen, den sogenannten Multicore

Prozessoren, erreichen. Schon heute ist der Einsatz der Integrated-Modular-Avionics-Architektur (IMA) Stand der Technik, um die Anzahl der Rechner in einem Flugzeug zu senken. In einer IMA-Architektur teilen sich softwarebasierte Flugzeugfunktionen (zum Beispiel Flight Management System, Fuel Management System, Built-In Test und Cabin Environment Control) mit unterschiedlicher Kritikalität die Rechenkapazität eines Prozessors in Form von Partitionen. Mittels Partitionen werden Programme in Zeit und Speicher separiert, d.h. jedes Programm läuft zu einer fest vorgegebenen Zeit und hat seinen eigenen Speicherbereich. Für zukünftige Funktionen wird zur Erreichung eines ökologischeren, ökonomischeren und sicheren Flugverkehrs, wie zum Beispiel in den Szenarien Single European Sky Research (SESAR), eine Erhöhung der Rechenleistung notwendig.

Alle wesentlichen sicherheitskritischen Funktionen werden in Flugzeugen durch Embedded Systems bestimmt und sind damit von entscheidender Bedeutung für Sicherheit, Leben und Gesundheit der Passagiere sowie zum Schutz der Umwelt. Dementsprechend aufwendig sind die Nachweise für die funktionale Sicherheit der in der Luftfahrt eingesetzten elektronischen Systeme (siehe SAE ARP 4754, SAE ARP 4761, RTCA/DO-160, RTCA/DO-254 und RTCA/DO-178B).

Den Systemen und Systemelementen (zum Beispiel Geräten und Software-Konfigurationseinheiten) werden Design Assurance Levels (DAL) zugeordnet. Diese richten sich nach der Einstufung des Fehlereffekts in Bezug auf die zu implementierende Flugzeugfunktion und reichen von Catastrophic über Hazardous, Major, Minor bis zu No Safety Effect. Für IMA-Architekturen bedeutet dies, dass Sicherheit und Zuverlässigkeit von Hardware-Komponenten (Rechner-Hardware) und Software-Komponenten (Anwendungs-Software) getrennt zertifiziert werden können. Für einfachere Systeme wird aus Kostengründen typischerweise der Ansatz verfolgt, Hardware- und Software-Komponenten als gemeinsames System zu betrachten und entsprechend auch gemeinsam, d.h. als Gesamtheit auf seine Sicherheit und Zuverlässigkeit hin zu prüfen und zu zertifizieren.

Ein wesentliches Element des Sicherheitsnachweises ist der Nachweis des deterministischen Verhaltes. Das bedeutet zum einen, es dürfen nur vorgesehene Funktionen implementiert sein und nur diese, und die implementierten Funktionen müssen zu festgelegten Zeitpunkten aktiviert werden. Keine Funktion darf eine andere Funktion (negativ) beeinflussen. Das "sichere Verhalten" muss aktiv und positiv nachgewiesen werden. Entsprechend aufwendig sind die Nachweise und benötigten Tests. Im Fall der Hardware wird bei der Zertifizierung ein Sicherheitsnachweis letztendlich bis auf "Einzelelement Ebene", d.h. bis auf das einzelne logische Element hin, gefordert. Je

komplexer ein integrierter Schaltkreis ist desto aufwendiger sind die geforderten Nachweise.

Für bisherige einfachere Singlecore-Prozessoren akzeptieren die Zulassungsbehörden, dass dieser Baustein in seiner Funktionalität fest vorgegeben ist und durch die Software bestimmt wird. Entsprechend werden Singlecore-Prozessoren als zertifizierbar betrachtet, wobei der Sicherheitsnachweis sich wesentlich auf die Software konzentriert.

Zukünftige Avionik-Rechner werden zunehmend in Multicore-Technologie ausgeführt werden müssen, bis hin zum völligen Ersatz von Singlecore-Prozessoren durch Multicore-Prozessoren. Nach heutigen Maßstäben und nach zu Beginn des Projekts bekannten Verfahren können Multicore- Prozessoren nicht für sicherheitskritische Funktionen in der Luftfahrt eingesetzt werden. Das wesentliche Problem ist dabei, dass das Verhalten des Multicore-Prozessors als nicht deterministisch angesehen wird. Dies liegt ist im Wesentlichen in der Architektur der derzeitigen Multicore-Prozessoren begründet. Besonderes Augenmerk erfordern daher die folgenden kritischen Aspekte:

- Gemeinsame Ressourcen-Nutzung Worst-Case Execution Time (WCET) und Daten-Latenz sind nicht bestimmbar
- Enge Kopplung der Einzelprozessoren Separierung der verschiedenen Software-Funktionen ist erschwert
- Hohe Integrationsdichte mit bis zu zweitausend Registern hohe Komplexität, die fehlerträchtig ist
- Nicht verfügbarer Dokumentation, notwendige Einblicke in Eigenschaften und Abläufe fehlen

Falls Multicore-Prozessoren in der Avionik nicht eingesetzt werden könnten, würde die notwendige weitere Steigerung der Leistungsfähigkeit der Rechnerleistung in der Avionik nicht implementiert und genutzt werden können. Im Extremfall, d.h. zukünftiger Nichtverfügbarkeit von heutigen Singlecore-Prozessoren, entstünde ein riesiges Problem für zukünftige Avionik-Systeme, da zertifizierbare Prozessoren nicht mehr verfügbar und Multicore-Prozessoren für sicherheitskritische Anwendungen in der Luftfahrt nicht einsetzbar wären.

4.3 Stand der Technik in der Domäne Bahn

Innerhalb von Personenverkehrszügen kommen Fahrzeug-steuerungssysteme zum Einsatz. Diese Systeme steuern die wesentlichen Subsysteme eines Zuges, wie den Antrieb, die Bremse, die Energieversorgung, das Bordnetz, die Türen oder die Klimatisierung. Ein derartiges Fahrzeugsteuerungssystem besteht nach dem Stand der Technik zu Beginn des Projekts aus

mehreren verschiedenen Rechnerknoten. Die Rechnerknoten sind über heterogene Bussysteme wie Profinet, Ethernet, MVB oder WTB verbunden. Die Vielzahl der Rechnerknoten und Bussysteme führt zu hohem Gewicht, hohem Verbrauch an Bauraum sowie einer komplexen Verkabelung. Indem mittels Multicore- und Virtualisierungstechnologie die Funktionalität mehrerer heutiger Rechnerknoten auf einem einzelnen Knoten vereint wird, könnte die Anzahl der Rechnerknoten und damit Gewicht, Bauraum, Komplexität der Verkabelung und letztlich Kosten verringert werden. Im Desktop- und v. a. im Server-Bereich sind ausgereifte und verbreitete Lösungen verfügbar, um heterogene Funktionen auf einem einzelnen Rechner zu konsolidieren. Aufgrund der besonderen Anforderungen hinsichtlich funktionaler Sicherheit, Echtzeit und der Einsetzbarkeit unter rauen Umweltbedingungen sind diese Lösungen für den Bahnbereich nicht geeignet. Eine Plattform bestehend aus Multicore-Prozessorplattform, Betriebssystem, Virtualisierungslösung und Middleware, welche die Komponenten eines Fahrzeugsteuerungssystems im Bahnbereich konsolidieren kann, muss die folgenden wesentlichen Eigenschaften aufweisen:

- Heterogene Systeme sind auf einfache Art und Weise und mit geringem Anpassungsaufwand auf die Plattform integrierbar. Dabei kann möglichst viel bestehende Software wiederverwendet werden.
- Die Plattform eignet sich für Echtzeit-Anwendungen mit typischen Reaktionszeitanforderungen im zwei- bis einstelligen Millisekunden Bereich.
- Konzepte zur Hochverfügbarkeit werden unterstützt, z.B. Live Migration.
- Es bestehen so wenige Abhängigkeiten zur Hardware wie möglich, um diese wenn nötig ohne großen Aufwand flexibel wechseln zu können.
- Die Plattform schafft keinen Vendor-Lock-In, also keine Abhängigkeit zu einem einzigen Hersteller.
- Die Plattform ist zertifizierbar bis SIL (Safety Integrity Level) 2. Die maßgebliche Norm ist dabei CENELEC EN 50128.

4.4 Stand der Wissenschaft

Realzeitanwendungen in Embedded Systems stellt Industrie und Forschung vor große Herausforderungen. Offene Fragestellungen zu Beginn des Projekts waren u.a. die Ermittlung der WCET aufgrund von Wettstreitigkeiten bei Zugriffen auf geteilte Ressourcen in solch einem System, die sichere

Abstraktion/Abschottung der Plattform gegenüber nicht autorisierten Zugriffe und generell die effiziente Ausschöpfung der Potentiale von Nebenläufigkeit. Die in der Literatur genannten Konzepte und Methoden nehmen meist Vereinfachungen bei der Systemarchitektur an (keine geteilten Ressourcen, keine Rekonfiguration der Tasks während der Laufzeit) oder schalten für die Performance wichtige Features aus (Caches, Hardware-Beschleuniger).

Virtualisierungskonzepte in Embedded Systems gingen bisher von Para-Virtualisierung (Anpassung der Betriebssysteme / Anwendungen an zwischengeschaltete Software-Schicht und damit verbundenen Performance-Einbußen) aus. Die Erforschung der Verwendung von Hardware-Unterstützung bei der Virtualisierung auf den Cores, Hardware-Beschleunigern und I/O-Devices, und insbesondere welche Implikationen dies bzgl. Security, Safety und Performance hat, war ein bisher in der Literatur nicht adressierter Forschungsgegenstand.

Sicherheitskritische Anwendungen erfordern eine Maskierung von statischen und transienten Hardwarefehlern, zumal neuere Technologien mit wachsenden Fehlerraten einhergehen. Die Erhöhung von Zuverlässigkeit und Verfügbarkeit durch Multicore war zu Beginn des Projekts auf Replikation ganzer Cores und ihrer Software beschränkt. Replikation auf Task-Ebene, die besonders kosten- und energieeffizient wäre, erfordert komplexere Ablaufsteuerungen (Roll-Back & Recovery). In diesem Kontext werden jedoch oft die Aufwände für Fehlererkennung / Zustandsspeicherung unterschätzt, komplexere Blockierungseffekte und Fehlerabhängigkeiten durch Nutzung gemeinsamer Ressourcen ignoriert und eine quantitative Bewertung der Effekte ausgelassen. Performanz- und Zuverlässigkeitsgarantien sind damit nicht möglich.

Um die hohe parallele Rechenleistung von Multicore-Plattformen in Anwendungen nutzbar zu machen, müssen bestehende Methoden des Software-Engineering für die Programmierung paralleler Systeme erweitert werden. Dies beginnt bei der Auswahl geeigneter Modellierungs- und Programmierparadigmen für parallele Echtzeitsysteme und setzt sich über den Softwareentwicklungsprozess bis in die Qualitätskontrolle und die Analyse geforderter Eigenschaften wie Safety, Performance, Energie-Effizienz und Produktkosten fort. Für die effiziente Unterstützung einer derart erweiterten Softwareentwicklungsmethodik für Multicore-Plattformen sind darauf abgestimmte Entwicklungswerkzeuge erforderlich. Viele der in den Industriedomänen eingesetzten Werkzeuge boten zu Beginn des Projekts noch keine hinreichende Unterstützung im Bereich der für CPS und Multicore typischen Themen wie die hochgradige Vernetzung und Parallelität auf mehreren Ebenen des Gesamtsystems.

5 Zusammenarbeit mit anderen Stellen

ARAMiS hat Methoden und Werkzeuge des Software-Engineering, die im Projekt SPES 2020 erarbeitet wurden, für Smart Mobility, CPS und Multicore angepasst und erweitert. Die in ARAMiS entwickelten Methoden und Werkzeuge wurden für den Aufbau von Demonstratoren verwendet, auf Praktikabilität evaluiert und werden im Sinne der Nachhaltigkeit der Projektergebnisse als Treiber für zukünftige Entwicklungen bei Werkzeugherstellern dienen.

Die „Multicore for Avionics Arbeitsgruppe“ (MCFA) wurde von Freescale ins Leben gerufen, um Multicore-Prozessoren in zertifizierungspflichtigen Anwendungen in der Luftfahrt einsetzen zu können. Der Austausch mit der MCFA erfolge durch in ARAMiS beitragende Personen, die auch in der MCFA aktiv sind. Konzepte wurden ausgetauscht und in Einklang gebracht.

Die im AUTOSAR Konsortium laufenden Arbeiten, welche sich auch mit Multicore-Technologie befassen, wurden in ARAMiS berücksichtigt. Konzepte, welche für die Domäne Automotive in ARAMiS erarbeitet wurden, wurden über im Projekt beitragende Personen mit den Konzepten und Weiterentwicklungen im AUTOSAR Konsortium abgestimmt. Umgekehrt wurden neue Erkenntnisse aus AUTOSAR in ARAMiS eingebracht.

II. Partnerspezifische Darstellung

Die folgenden Kapitel stellen die erzielten Ergebnisse partnerweise in zusammengefasster Form dar. Eine detaillierte, übergreifende Beschreibung der technischen Inhalte und Ergebnisse des Projekts ist im gemeinsamen Abschlussbericht (Final Report vom 21.10.2015) des Projekts zu finden. Darauf basierend ist eine Veröffentlichung der Projektinhalte und Ergebnisse als Buch geplant.

1 Airbus Group Innovations

1.1 Wissenschaftlich-technische Ergebnisse

Zu Beginn des Projektes wurde ein übergeordnetes Szenario erstellt, welche alle im Projekt behandelten Domänen inkludiert. Ausgehend von diesem High Level Szenario, wurden Domänen spezifische Szenarien abgeleitet und hieraus Requirements extrahiert. Für die Avionik Domäne wurden zwei Domänen spezifische Szenarien definiert. Eines (Unmanned Ariel Vehical) konzentriert sich dabei auf die Zertifizierungs-Aspekte von Multicore-Systemen. Das zweite Szenario (Kabinen Management System) fokussiert auf den Security Aspekt bei der Verwendung von Multicore Prozessoren in Avionik Systemen. Für beide Szenarien wurde auf Grundlage der Requirements eine System-architektur definiert.

Für erstere wurden modernen Multi-Core Architekturen (wie zum Beispiel FreescaleP4080) untersucht. Eine gute Übersicht der technischen Ergebnisse wurden in der Veröffentlichung bei der EDCC European Dependable Computing Conference geliefert.

Des Weiteren konzentrieren sich die Arbeiten auf eine Einbindung von I/O Elementen in eine Multi-Core Hardware. Schwerpunkt-mäßig wurden Limitierungen und mögliche Umgehungsstrategien dieser Limitierungen der Freescale QorIQ PowerPC Plattform untersucht. Untersuchung eines software-basierten Workarounds zur Umgehung der Freescale IOMMU (PAMU) Limitierungen wurden fortgeführt. Außerdem wurde an einem Konzept bzgl. eines hardware-basieren Workarounds zur Umgehung der Freescale IOMMU (PAMU) Limitierungen gearbeitet.

Darüber hinaus wurden Arbeiten an Zertifizierungsarchitekturen für Multicore bearbeitet. Die Aspekte von der zur Zertifizierung von Multicore-Prozessoren wurden untersucht und im Rahmen einer Arbeitsgruppe namens MCFA (Multicore for Avionics) mit anderen Firmen und der Zertifizierungsbehörde (EASA und FAA) in einem speziellen Treffen mit EASA und FAA und Aerospace OEMs und Zulieferer zum Thema Zertifizierung von Multicore Prozessoren geteilt. Die Ergebnisse wurden in einem Dokument der Zertifizierungsbehörde publiziert („generic CRI“ – Certification Review Item) und Projektintern geteilt. Ein direktes Weitergeben des „generic CRIs“ ist aufgrund von Geheimhaltungsverein-barungen ist zu diesem Zeitpunkt nicht möglich. Jedoch wurden die Arbeiten zu einem späteren Zeitpunkt in einem als CAST-32 (Certification Authorities Software Team Position Paper Multi-core Processors) genannten Dokument im Mai 2014 veröffentlicht und freigegeben. Neben den Zertifizierungsaspekten wurden Monitoringansätze auf Basis von Hardwareperformancecountern entwickelt. Diese stehen im engen Zusammenhang zur im Projekt

entwickelten Multicore WCET-Analyse (Worst Case Executions Time) sowie zum Themenbereich Software Virtualisierung.

Im Themenbereich Software Virtualisierung wurde das "Run-Time Fixed-Size Weighted DMA Transaction Based Temporal Separation Concept" zur zeitlichen Separierung (temporal separation) erforscht und dokumentiert. Außerdem wurde daran gearbeitet die bisherigen Konzepte für Multi-Core auf Multi-Core Multiprozessoren zu erweitern (Multi-Prozessor I/O Virtualisierung (MPIOV)). Zu MPIOV wurde eine gemeinsame Publikation mit TUM(LIS) erarbeitet.

Im Projekt wurde ein software-basierter Ansatz zur Lösung der Freescale IOMMU (PAMU) Limitierungen entwickelt und dokumentiert. Außerdem wurde an dem Hardware-basierten Ansatz (I/O Memory Protection Unit (IOMPU)) zur Lösung der IOMMU Limitierungen gearbeitet. Eine Publikation zu IOMPU mit TUM(LIS) ist hieraus resultiert. Daneben wurde an dem Hardware-basierten Ansatz (Safeguarding Interrupts (SgInt)) gearbeitet der bei IOMMU Limitierungen oder Fehlen einer IOMMU ermöglicht, Interrupts abzusichern. Die Arbeiten bzgl. SgInt wurden in Kooperation mit TUM(LIS) publiziert.

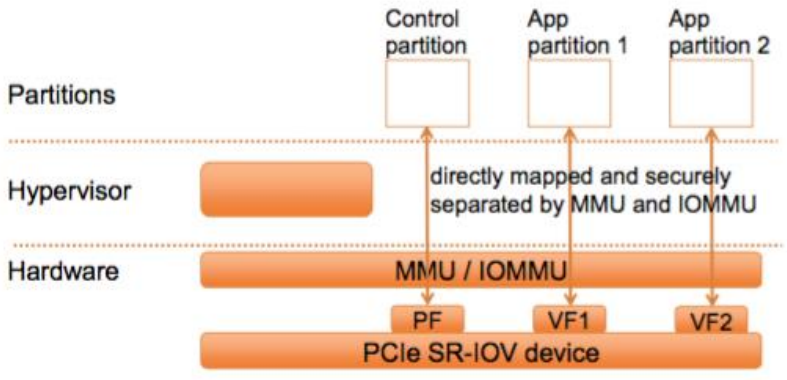

Abbildung 2 Architektur mit MMU & IOMMU

Des Weiteren wurden „safety net“ Ansätze mit Projektpartnern SYSGO, Airbus Defence & Space und AbsInt entwickelt, die eine übermäßige Inanspruchnahme von Ressourcen einem Multi-Core System verhindern oder zumindest soweit einschränken sollten, dass die Sicherheit des Systems nicht gefährdet ist.

Ein Schwerpunkt der Arbeiten im Security Bereich lag in der Definition und Evaluierung der Kommunikationsarchitektur. Es wurden Software-Blöcke und Funktionen basierend auf den Requirements definiert und deren Einbindung in die Gesamtarchitektur untersucht. Des Weiteren wurde erörtert, wie diese Software-Blöcke realisiert und umgesetzt werden können. Die Systemarchitektur wurde anschließend einer Sicherheits-

betrachtung unterzogen. Durch diese Betrachtung (Risikoanalyse) konnten bestehende Schwachstelen in der Systemarchitektur identifiziert und behoben werden.

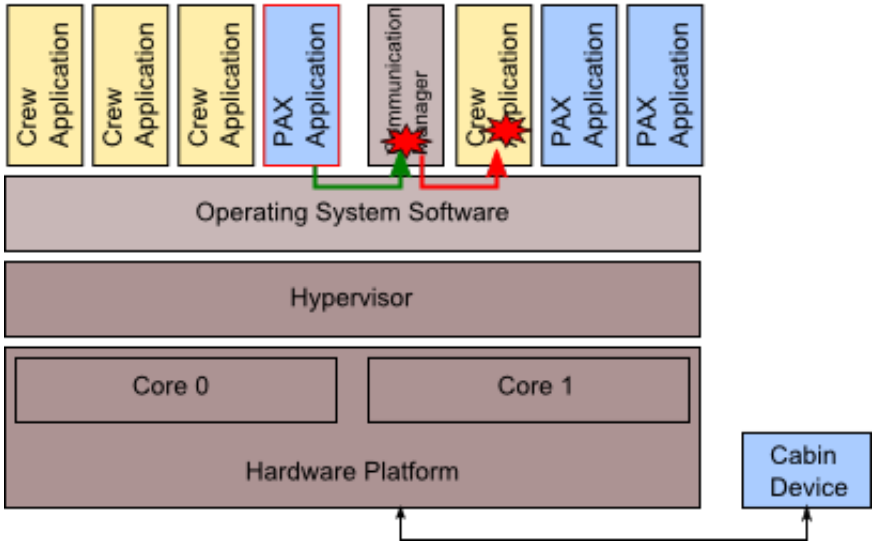

Abbildung 3 Angriffspfade als Bestandteil der Risikoanalyse

Ein zweiter Schwerpunkt lag auf der Realisierung eines Partitionierungskonzeptes zur Isolation des Zeitverhaltens von Anwendungen. Der Hauptteil der Arbeit wurde dabei in die Software Architektur eingebracht, nachdem Anpassungen im Betriebssystem notwendig waren. Nichtsdestotrotz stellt die Arbeit eine Verbindung der Themen Software Architektur, Safety Zertifizierung und Virtualisierung dar, da neben der Betriebs-systemumsetzung ebenfalls Erweiterungen des WCET Analyseframeworks aiT von AbsInt vorgenommen wurden und das Gesamtkonzept darauf abzielt zeitliche Einflüsse zwischen mehreren hart-echtzeitkritischen Anwendungen zu behandeln und damit eine entsprechende Partitionierung umzusetzen.

Eine entsprechende Implementierung wurde vorgestellt und eine Veröffentlichung bei der Dependable Systems and Networks Konferenz publiziert. Im Rahmen einer ersten prototypischen Evaluierung konnte die Wirksamkeit des Konzepts sowie der Implementierungen nachgewiesen werden.

Des Weiteren wurde das vorgeschlagene Partitionierungskonzept für die zeitliche Separierung von Anwendungen wurde wie geplant implementiert und evaluiert. Es wurden weitere Verbesserungen am grundlegenden Architekturmodell für WCET Analyse vorgenommen. Außerdem wurden die Laufzeitmechanismen verfeinert. Es wurden zwei Publikationen zum Thema eingereicht.

Das grundlegende Partitionierungskonzept wurde damit entsprechend der Planung umgesetzt und evaluiert. Gleiches gilt für die Average-Case Erweiterungen.

Ein weiterer Schwerpunkt war eine WCET Analyse (isWCET) in Verbindung mit Laufzeitüberwachung für Multi-core Systeme. Die Konzepte für Analyse und Überwachung. Die isWCET Analyse wurde in Zusammenarbeit mit AbsInt umgesetzt. Die Laufzeitüberwachung wurde initial in einem proprietären Betriebssystem, in SYSGOs PikeOS als auch in WindRiver OS implementiert. Im Rahmen dieser Arbeiten wurde eine Veröffentlichung auf der ECRTS14 eingereicht und angenommen. Titel der Veröffentlichung: Multi-core Interference-sensitive WCET Analysis Leveraging Runtime Resource Capacity Enforcement. Hiermit wurden die Arbeiten abgeschlossen.

Weitere technische Arbeiten wurden bei der Analyse verschiedener COTS Multi-core Plattformen geleistet. Hierbei wurde vor Allem der Freescale Prozessoren P5020 weiter untersucht. Im Speziellen wurde das Problem der gegenseitigen Beeinflussung an geteilten Ressourcen untersucht. Die durchgeführten Untersuchungen stehen dabei in direktem Zusammenhang zur obigen WCET Analyse.

Neben den technischen Arbeiten wurde zusätzlich der Avionikstandard DO178 im Hinblick auf Multi-core Zertifizierungs-aspekte untersucht. Die Ergebnisse als Input für die Bewertung der Safety-Konzepte genutzt.

Die serverseitige Security Architektur wurde final abgeschlossen und beschrieben. Hierbei wurden alle Elemente (Access Control Lists, Deep Packet Inspector, Hashed Message Authentication Codes) beschrieben und deren Zusammenspiel (sprich der Datenfluss innerhalb der Security-Komponente) beschrieben.

Die erarbeiteten Konzepte wurden prototypisch in einem Demonstrator umgesetzt. Im Vordergrund stand hierbei der Kabinenmanagement Server Demonstrator. Der Kabinen-management Server ist hierbei an ein Kabinenmanagement System angeschlossen und steuert dieses. Der Multicore basierte Server steuert hierbei die wichtigsten Kabinen Funktionen sowie übernimmt Aufgaben des Passagier Entertainments. Auf diese Weise konnte gezeigt werden, wie mit Hilfe der entwickelten Mechanismen parallel Funktionen verschiedener Security und Safety Klassen auf einem System sicher ausgeführt werden können. Der Demonstrator wurde auf dem ARAMiS Abschlussmeeting in Hamburg gezeigt.

Abbildung 4 Kabinenmanagement Demonstrator

Abschließend wurden die ursprünglichen Requirements herangezogen um die Konzepte und daraus resultierenden Systemarchitekturen zu validieren.

1.2 Notwendigkeit und Angemessenheit der Arbeiten

Der Beitrag zu den förderpolitischen Zielen des BMBF betrifft sichere eingebettete Systeme als wichtiger Bestandteil Deutschlands weltweit führender Industriezweige wie Automobilbau, Maschinen und Anlagenbau. Avionik taugliche Lösungen können dabei aufgrund ihrer hohen Kritikalität und Anforderungen hinsichtlich Sicherheit als Vorbild und Inspirationsquelle dienen. Insbesondere die Randbedingungen in der Luftfahrt hinsichtlich geringer Stückzahlen und langem Produktlebenszyklus (Verwendung von Commercial-off-the-Shelf Komponenten und Standard-kompatiblen Komponenten) zeigen Parallelen auf, die für den Maschinen- und Anlagenbau in ähnlicher Weise relevant sind und somit Verwendung finden könnten.

Das Teilvorhaben hat erfolgreich die Verwendbarkeit sowie benötigte Mechanismen von Multicore Prozessoren für einen Sicherheits- (Safety) sowie Datensicherheits- (Security) kritischen Anwendungsfall untersucht. Es hat sich hierbei gezeigt, was nötig ist um Multicore Prozessoren in einem solchen Umfeld einzusetzen. Dies wurde dabei von Software als auch von

Hardwareseite aus betrachtet. Zusätzlich wurde evaluiert, welchen Einfluss Multicore Systeme auf eine Gesamtarchitektur haben.

1.3 Fortschritte auf dem Gebiet des Vorhabens

EASA hat im Rahmen des Projektes MULCORS die Verwendung von Multi-Core Prozessoren genauer erforschen lassen. Die Ergebnisse verlangen Aspekte wir eine safety net. Der Ansatz in TP4 dürfte den Vorstellungen der EASA nahe kommen (als ein Teilaspekt!).

Erwähnenswert ist die Entwicklung des EASA Certification Review Items (CRI) zum Thema Multicore-Prozessoren (2-Kerne) (Guidance Document – Anleitung aus der Sicht der Sicherheits-behörde), welches die derzeitige Luftfahrtbranche zu dem Thema beschäftigt. Es sei angemerkt, dass die Luftfahrt-Industrie weiter an dem Thema arbeitet und sich Anfang Juli 2014 erstmals zu dem Thema bei der EASA in Köln getroffen hat.

1.4 Veröffentlichung der Ergebnisse

[1] Jan Nowotsch und Michael Paulitsch. Leveraging Multi-Core Computing Architectures in Avionics. EDCC European Dependable Computing Conference, 2013

[2] Jan Nowotsch, Michael Paulitsch, Arne Henrichsen, Werner Pongratz, and Andreas Schacht. Monitoring and WCET Analysis in COTS Multi-core-SoC-based Mixed-Criticality Systems. Workshop at Design, Automation and Test in Europe (DATE) Conference. March 2014.

[3] Daniel Muench, Michael Paulitsch, and Andreas Herkersdorf. Temporal Separation for Hardware-Based I/O Virtualization for Mixed-Criticality Embedded Real-Time Systems Using PCIe SR-IOV. In Proc. of the 27th International Conference on Architecture of Computing Systems (ARCS), Lübeck, Germany, February 25th – 28nd, 2014.

[4] Jan Nowotsch, Michael Paulitsch, Daniel Bühler, Henrik Theiling, Simon Wegener and Michael Schmidt. MULTI-CORE INTERFERENCE-SENSITIVE WCET ANALYSIS LEVERAGING RUNTIME RESOURCE CAPACITY ENFORCEMENT. In Proc of the 26th Euromicro Conference on Real-Time Systems (ECRTS14), July 8-11, 2014

[5] Stefan Burger, Kevin Müller, Oliver Hanka, Michael Paulitsch, Andrea Bastoni, Henrik Theiling, Matthias Heinisch. Implications of Multi-Core Processors on Safety-Critical Operating System Architectures. 10th annual workshop on Operating Systems Platforms for Embedded Real-Time applications. 2014.

[6] Daniel Münch, Michael Paulitsch, Oliver Hanka, Andreas Herkersdorf. MPIOV: Scaling Hardware-Based I/O Virtualization for Mixed-Criticality Embedded Real-Time Systems Using Non Transparent Bridges to (Multi-Core) Multi-Processor Systems. Design, Automation and Test in Europe (DATE2015). 2015

[7] Daniel Münch, Michael Paulitsch, Andreas Herkersdorf. Monitoring of I/O for Safety-Critical Systems Using PCI Express Advanced Error Reporting. International Symposium on Industrial Embedded Systems (SIES2015). 2015.

[8] Daniel Münch, Michael Paulitsch, Oliver Hanka, Andreas Herkersdorf. SgInt: Safeguarding Interrupts for Hardware-Based I/O Virtualization for Mixed-Criticality Embedded Real-Time Systems Using Non Transparent Bridges. International Conference on Architecture of Computing Systems (ARCS2015). 2015.

[9] Oliver Hanka, Franz Nuscheler, Matthias Heinisch, Michael Handwerker, Peter Klose, Stefan Schneele. Automatic head count determination on board a means of transport. Patentantrag: 14161682-1

[10] Oliver Hanka, Peter Klose. Inherent power-over-data bus signaling for secure operating mode switching. Patentantrag: 13173635-7

[11] Stefan Burger, Kevin Müller, Oliver Hanka, Michael Paulitsch, Andrea Bastoni, Henrik Theiling, Matthias Heinisch. Implications of Multi-Core Processors on Safety-Critical Operating System Architectures. 10th annual workshop on Operating Systems Platforms for Embedded Real-Time applications. 2014.

2 Audi Electronics Venture GmbH

2.1 Wissenschaftlich-technische Ergebnisse

Die AEV GmbH arbeitete an vielen Teilpaketen des Projekts ARAMiS mit, die Tätigkeiten umfassen hier die Aspekte Anforderungen und Szenarien, Systementwurf, Software sowie die domänenübergreifende Methoden- und Toolplattform.

Aus dem Blickwinkel eines internen Dienstleisters eines Automobilherstellers brachte die AEV GmbH zu Beginn Szenarien und Anforderungen als Basis für die weiteren Aktivitäten ins Projekt ARAMiS ein.

2.1.1 Beiträge für das „Teilpaket 2 – Systementwurf“

Im Rahmen des Arbeitspakets für den Systementwurf (TP2) wurde die Verantwortung für das Ergebnisdokument „E2.1.4.3 Report on the Specification of the Runtime Environment“ übernommen. Das Dokument beschreibt die Anforderungen an eine Middleware sowohl aus logischer als auch aus technischer Perspektive. Mittels einer Rating Matrix und den Anforderungen ist es möglich eine technische Perspektive abzuleiten. Der Inhalt des Ergebnis-dokuments wurde insbesondere mit Continental und dem KIT abgestimmt und überarbeitet.

Innerhalb des Arbeitspakets wurde zudem in Enterprise Architect ein Metamodell entworfen das Inhalte der logischen und technischen Sicht verdeutlicht. Zusätzlich wird ersichtlich wie die technische Sicht von der logischen Sicht, den Anforderungen sowie der Rating Matrix abgeleitet als zentrales Entscheidungselement werden kann. Berücksichtigt wird dabei sowohl die Automotive- als auch die Avionik-Domäne.

Eine engere Zusammenarbeit mit der Robert Bosch GmbH erfolgte für ein weiteres Ergebnisdokument das sich mit Mechanismen zur Fehlerisolation in Multicore-Systemen beschäftigt (E2.3.31).

Für das Arbeitspakets AP23 wurden weiterhin Arbeiten an dem Ergebnisdokument „E2.3.4.1 Optimierung der Safety-Eigenschaften der Architektur“ durchgeführt. Dazu wurde regelmäßig an Abstimmungsrunden sowie Telefonkonferenzen teilgenommen. Zusammen mit der AUDI AG wurden Anwendungsfälle für das Thema Fail-Safe / Fail-Operational definiert und in das Dokument eingearbeitet. Mögliche Lösungsansätze wurden mit den involvierten Projektpartnern des Dokuments diskutiert und in einer überarbeiteten Version des Dokuments berücksichtigt. Die Konzeptideen für eine mögliche

Umsetzung von Fail-Operational beinhaltet die Veröffentlichung „Fail-Operational in Safety-Related Automotive Multi-Core Systems“, die auf dem IEEE Symposium for Industrial Embedded Systems (SIES) 2015 präsentiert wurde.

Eine Teilnahme an Abstimmungsrunden und Diskussionen an der TU München sowie bei der Infineon Technologies AG erfolgte für diverse Ergebnisdokumente aus den Arbeitspaketen AP2.1.3 und AP2.1.4.

Zuletzt wurden mehrere Review-Arbeiten für die Ergebnis-dokumente E2.1.3.1, E2.3.3.1 und E2.1.4.1 wobei diese Dokumente auf Multicore-spezifische Aspekte überprüft wurden.

2.1.2 Beiträge für das „Teilpaket 4 – Software“

Die AEV GmbH brachte für das Teilpaket Software (TP4) ihre Erfahrung und etwaige Anforderungen aus der Software-entwicklung und der Zusammenarbeit zwischen unterschiedlichen Entwicklungspartnern ein. Besonderer Fokus wurde auf die Themen Mixed-Criticality-Schedulingverfahren sowie Virtuali-sierungsmechanismen in der Domäne Automotive gelegt. Ein weiterer Schwerpunkt war das Thema Software-Parallelisierung in AUTOSAR-basierten Softwarearchitekturen.

Erste Diskussionen zu Mixed-Criticality-Schedulingverfahren im automobilen Umfeld erfolgten mit der Universität Paderborn und der Symtavision GmbH. Im Arbeitspaket AP4.4 wurde dazu im Rahmen des Dokuments E4.4.4.2 „Design of Mixed-Criticality Scheduling Concepts“ ein Teilkapitel über Herausforderungen, Laufzeit und Scheduling beigesteuert. Die gesammelten Ideen resultierten zudem in einer Publikation auf dem Embedded Software Engineering Kongress 2012. Inhaltlich wurden hier Scheduling-Verfahren für das Timing von Funktionen mit unterschiedlichen sicherheitskritischen Anforderungen (Mixed Criticality) präsentiert.

Eine weitere Veröffentlichung erfolgte auf der „safetronic 2012“-Tagung mit Fokus auf sicherer Software und Hardware im Automobil. Hier wurde erneut mit der Symtavision GmbH kooperiert und ein Vortrag zum Thema „Echtzeitfähige, dynamische Softwarearchitekturen für hochintegrierte Steuergerät nach ISO 26262“ gehalten. Dieses Thema wurde anschließend weiterverfolgt so dass auf dem SafeTRANS Industrial Day 2013 mit dem Schwerpunkt „Future reference architecture for embedded systems in safety critical environments” erneut ein Beitrag publiziert wurde. Unter dem Titel „Software Architecture Methods and Mechanisms for Timing Error and Failure Detection According to ISO 26262: Deadline vs. Execution Time Monitoring“ wurden hier Scheduling-Verfahren für das Timing von Funktionen mit

unterschiedlichen sicherheitskritischen Anforderungen (Mixed Criticality) präsentiert.

Ein weiteres Thema mit dem sich die AEV in dem Teilpaket 4 beschäftigte war die Verteilung von AUTOSAR-Software auf mehrere Cores. Dazu wurde mit der AUDI AG, der EFS GmbH sowie der BMW AG kooperiert so dass die Ergebnisse auf dem AUDI-Demonstrator umgesetzt werden konnten. Der Beitrag ist in dem Ergebnisdokument E4.1.1.1 unter dem Abschnitt „AUTOSAR Code to Multicore“ zu finden und beschreibt Konzepte zur Software-Migration von Singlecore- auf Multicore-Systeme. Des Weiteren wurde in E4.1.1.3 im Teilkapitel „Deployment of AUTOSAR Runnables on Multicore“ beschrieben wie AUTOSAR Code auf Runnable-Ebene auf unterschiedliche Cores verteilt werden können. Die Arbeiten an der Parallelisierung einer Teilfunktion aus dem TP6-Demonstrator fortgesetzt. Die Ergebnisse wurden in das Ergebnisdokument „E4.1.1.5 – Evaluation Overall Concepts Identification of Problems and Solutions“ eingearbeitet. Eine Publikation erfolgte auch auf dem SAE World Congress 2015 mit dem Titel „Software Parallelization in Automotive Multicore-Systems“.

In hochintegrierten Multi-Core-Systemen ist ein entscheidender Faktor die Segregierung von Applikationen mit unterschiedlichen sicherheitskritischen Anforderungen. Dazu wurden Ideen und Konzepte erarbeitet die im Ergebnisdokument E4.1.1.2 in dem Teilkapitel „Concepts for Isolation of Different ASIL Level SW Components“ beschrieben wurden. Hier werden Ansätze zur zeitlichen, räumlichen und Software-basierten Isolation sowie zur Kommunikationsintegrität erläutert.

Der Fokus auf Middlewares wurde auch im Teilpaket 4 weiterverfolgt und in Zusammenarbeit mit der Elektrobit GmbH im Dokument „E4.1.2.1 Evaluation and Modification of Middleware and Operating Systems“ die AUTOSAR-Basissoftware „tresos“ von Elektrobit evaluiert. Die Evaluierung erfolgte dabei anhand von vorgegebenen Kriterien. Mit der Middleware ist das Thema Virtualisierung eng verknüpft so dass zusammen mit Infineon und der AUDI AG Untersuchungen zu bestehenden Virtualisierungs- und Isolationskonzepten für die Automobilindustrie durchgeführt wurden. Die Ergebnisse wurden auf dem Embedded Software Engineering Kongress 2013 in dem Vortrag „Virtualisierungskonzepte für eingebettete Multicore-Systeme“ präsentiert. Eine interne Konzeptausarbeitung in Kooperation mit Infineon beschäftigte sich ebenfalls mit den Virtualisierungsmöglichkeiten in Multicore-Systemen. Für die interne Verwertung wurden Hardwarearchitekturen zum Einsatz in Hochintegrations-Steuergeräten untersucht die eine Virtualisierungsunterstützung bieten. Zudem erfolgten diesbezüglich Abstimmungen mit Toolherstellern und es wurde eine Evaluierung eines Hypervisors

für eingebettete, sicherheitsrelevante Systeme gestartet die derzeit weitergeführt wird.

Weitere Tätigkeiten umfassten eine regelmäßige Teilnahme an Telefonkonferenzen und Review-Arbeiten zu den ausgearbeiteten Dokumenten. Für das Thema Funktionale Sicherheit nahm die AEV regelmäßig an internen und externen Abstimmungs- und Koordinationstreffen zur Verbesserung und Anpassung der Funktionalen Sicherheit in Cyber-Physical-Systems (CPS) teil.

2.1.3 Beiträge für das „Teilpaket 5 – Domänenspezifische Methoden- und Toolplattform“

Im Rahmen der Arbeiten in TP5 wurden die Audi-internen Entwicklungswerkzeuge und Methoden hinsichtlich der Relevanz für ARAMiS betrachtet und dazu Werkzeugsteckbriefe erstellt. Die Zusammenfassung zu den AEV-Tools umfasste die Kategorien Architektur, Timinganalyse, AUTOSAR-Tooling, Modellierung, Compiler, Debugger, Codeanalyse und Testtools. Die AEV beteiligte sich zudem bei dem Deployment von den Funktionen auf die technische Architektur und stellte ihre Vorgehensweise anhand von Methodentemplates des Fraunhofer IESE dar.

Weitere Aufwände umfassten Review-Arbeiten innerhalb der verschiedenen Arbeitspakete. Zudem beteiligte sich die AEV an regelmäßigen Abstimmungsrunden und Diskussionen zur Erstellung des TP-übergreifenden Metamodells.

Es ist beabsichtigt Ergebnisse aus TP5 in den AEV Entwicklungsworkflow „enprove“ zu übernehmen.

2.2 Notwendigkeit und Angemessenheit der Arbeiten

Aufgrund der steigenden Komplexität der Software in heutigen und zukünftigen Steuergeräten sind Multicore-Systeme langfristig unverzichtbar. Insbesondere bei autonomen und automatisierten Fahrzeugen entstehen zusätzliche Risiken für die Fahrzeuginsassen und die Umgebung so, dass die Einhaltung des Standards für Funktionale Sicherheit (ISO 26262) garantiert werden muss. Die Verwendung von Multi-Core-Prozessoren bringt neue, sicherheitsrelevante Herausforderungen mit sich die es zu untersuchen galt. Dazu gehört neben der Implementierung von mehreren Funktionen unterschiedlicher Safety-Level auch die Wiederverwendbarkeit von bestehender Software in Hochintegrationssteuergeräte. Ein weiterer Aspekt ist, dass durch die Fahrzeugvernetzung sowohl mit Geräten in der direkten Umgebung (Smartphones etc.) als auch die Anbindung ans Internet neue Sicherheitsrisiken mit sich bringt. Das Fahrzeug als geschlossenes System wird demnach langfristig durch ein offeneres, vernetztes System ersetzt. Die gegenseitige

Beeinflussung von Safety und Security ist folglich ein entscheidender Punkt für zukünftige Fahrzeugarchitekturen.

Die Arbeiten der AEV innerhalb des ARAMiS-Projekts beschäftigten sich aus diesem Grund mit der Konzeptentwicklung für Funktionale Sicherheit in Hochintegrationssteuergeräten. Um die Leistungsfähigkeit der Prozessoren optimal ausnutzen zu können, wurden Untersuchungen hinsichtlich Parallelisierung von Steuergeräte-Software durchgeführt. Der dritte Punkt umfasste die Evaluierung von Virtualisierungskonzepten um mehrere Betriebssysteme zeitgleich auf demselben Steuergerät auszuführen.

Die Untersuchungen wurden in Kooperation mit der AUDI AG am Beispiel eines Fahrwerksteuergeräts durchgeführt.

2.3 Fortschritte auf dem Gebiet des Vorhabens

Fortschritte auf dem Gebiet der Software-Parallelisierung, Virtualisierung und Safety Engineerings sind in die Projektergebnisse mit eingeflossen. Die Inhalte der Arbeiten sind in den Publikationen zu finden.

2.4 Veröffentlichung der Ergebnisse

Zu Präsentationszwecken können die im Rahmen des Projekts erarbeiteten Ergebnisse genutzt werden die in den folgenden Publikationen nachzulesen sind:

- Embedded Software Engineering Kongress 2013
  - Virtualisierungskonzepte für eingebettete Multicore-Systeme
- SAE World Congress 2014
  - Efficient Virtualization for Functional Integration on Modern Microcontrollers in Safety-Relevant Domains
  - Adapted Development Process for Security in Networked Automotive Systems
  - Timing Analysis and Tracing Concepts for ECU Development
- SAE World Congress 2015
  - Non-intrusive Tracing at First Instruction
  - Software Parallelization in Automotive Multicore Systems

- IEEE Symposium on Industrial Embedded Systems (SIES) 2015
  - Fail-Operational in Safety-Related Automotive Multi-Core Systems

3 Robert Bosch GmbH

3.1 Wissenschaftlich-technische Ergebnisse

Im Folgenden werden wichtige Ergebnisse des Projekts kurz dargestellt, wobei der Fokus auf den Arbeitspaketen liegt, zu denen die Robert Bosch GmbH wesentliche Beiträge geleistet hat. Die vollständigen Ergebnisse sind im gemeinsamen Abschlussbericht „ARAMiS – Final Report“ dokumentiert.

Im Teilprojekt 1 „Anforderungen und Szenarien“ entstand, basierend auf der agendaCPS, ein domänenübergreifendes SmartMobility-Szenario, das von einer „CPS-Taskforce“ in Abstimmung mit den Avionik- und Automotive-Domänen erarbeitet wurde. Darauf aufbauend entstanden spezifische Szenarien aus den einzelnen Domänen. Die Robert Bosch GmbH hat dabei insbesondere mit „Drehbüchern“ aus dem Bereich „Engineering for Multicore“ und „Zero Impact Car“ beigetragen. In den Drehbüchern werden die Szenarien beispielhaft durch mittelfristig zu erwartende oder gewünschte Situationen und Abläufe realistisch beschrieben.

Im Rahmen des Szenarios „Engineering for Multicore“ wird beschrieben, wie die Anzahl von Steuergeräten im Fahrzeug reduziert werden kann, indem mehrere zuvor unabhängige Systeme auf einem leistungsfähigen Rechner integriert werden. Das Ziel dabei ist unter anderem, Bauraum und Gewicht einzusparen, was zu einer Reduktion des Energieverbrauchs und zur Verringerung der CO2-Emissionen führt. Daneben können auch Kosten z.B. für Hardware und Verkabelung eingespart werden.

Aus den beschriebenen Szenarien und Drehbüchern wurden jeweils eine Reihe spezifischer Use Cases abgeleitet. Daraus wurden Anforderungen an zukünftige Systeme, wie z.B. bezüglich Sicherheit, Zuverlässigkeit, Echtzeitfähigkeit und Verbrauchs-Effizienz abgeleitet.

Die dokumentierten Anforderungen aus Teilprojekt 1 wurden zur Steuerung und Fokussierung der Arbeiten in den anderen Teilprojekten herangezogen.

Das Teilprojekt 2 „Systementwurf“ befasste sich unter anderem mit dem Thema Virtualisierung in Multicore Systemen. Dabei wurden die Anforderungen an einen Hypervisor zur Virtualisierung von Systemen mit harten Echtzeitanforderungen und stark limitierten Hardware-Ressourcen erarbeitet. Die Arbeitsergebnisse zur Virtualisierung bei eingebetteten Systemen auf Hardware-, Software- und Systemebene sind in Kapitel 7 „Embedded

Virtualization“ des gemeinsamen Abschlussberichts zusammengefasst.

Zu Beginn des Projekts waren keine Virtualisierungslösungen kommerziell verfügbar, welche die gestellten Anforderungen auch nur annähernd erfüllten und einen Einsatz in Echtzeit-Regelsystemen in der Automotive-Domäne möglich erscheinen ließen. Inzwischen wird jedoch, basierend auf den Projektergebnissen, an der Entwicklung kommerzieller Produkte gearbeitet.

Ein weiterer Schwerpunkt lag auf dem Thema „Safety/ Zertifizierbarkeit“. In der Arbeitsgruppe „Systemtechnische und domänenspezifische Kapselungsmechanismen“ wurden Mechanismen zur Fehlerisolation in Multicore-Systemen, wie z.B. Virtualisierung, Memory Monitor und Safety MPU, beschrieben und bewertet. In einer weiteren Gruppe wurde an einer Optimierung der Safety-Architektur gearbeitet, unter anderem. zum Thema „Sicherheits-Architektur von Microcontrollern für Anwendungen in der Automotive-Domäne“. Durch dezentrale Maßnahmen wird die Überwachung und Sicherstellung der geforderten Safety-Eigenschaften auf Gesamtsystemebene wesentlich vereinfacht. Die Arbeitsergebnisse zu diesen Themen sind im gemeinsamen Abschlussbericht in Kapitel 3 „Certification and Safety Aspekts“ zu finden.

Im Rahmen der Arbeiten zur Systemarchitektur wurden wichtige Kriterien wie Performanz, Funktionale Sicherheit, Verfügbarkeit und Kosten für Trade-Off Analysen zur Bewertung verschiedener Architekturansätze identifiziert und beschrieben. Im gemeinsamen Abschlussbericht sind die Ergebnisse in Kapitel 4.11 „Trade-Off Analysis“ dokumentiert.

Das Teilprojekt 3 „Hardware“ begann mit einer Evaluierung der für verschiedene Virtualisierungslösungen erforderlichen Hardwareunterstützung. Dabei wurden die Unterschiede zwischen „Full virtualization“, „Paravirtualization“, „Hardware assisted virtualization“ und „Hybrid virtualization“ beschrieben und die Komponenten CPU, Speicher und Peripherie näher betrachtet. Darauf aufbauend wurden Anforderungen an eine zukünftige Hardware erarbeitet, durch die insbesondere für Systeme der Automobil-Domäne mit harten Echtzeitanforderungen performante und sichere Virtualisierungslösungen ermöglicht werden sollen.

Im Arbeitspaket „Unterstützung von Scheduling auf Multicore bei Virtualisierung“ wurden Anforderungen an die Hardware beschrieben, welche für ein effizientes Scheduling bei virtualisierten Systemen mit harten Echtzeit-Anforderungen erfüllt werden müssen. Weiterhin wurden Defizite identifiziert, z.B. bei der Kontextumschaltung, um Anforderungen an zukünftige Prozessoren zu spezifizieren. Die Arbeitsergebnisse zum Thema Scheduling bei virtualisierten Systemen sind im gemeinsamen

Abschlussbericht in Kapitel 7.5 „Mixed Criticality Scheduling“ zu finden.

Im Teilprojekt 4 „Software“ wurde am Beispiel einer Motorsteuerung gezeigt, wie Legacy Code auf ein Multicore-System portiert werden kann, siehe Kapitel 5.2 im gemeinsamen Abschlussbericht. Dabei können auch AUTOSAR- und Nicht-AUTOSAR Anteile gemischt werden, abhängig von Kundenanforderungen und dem Migrationsgrad zugelieferter Fremdsoftware. Dazu wurden die Anforderungen der Softwarekomponenten bezüglich Kommunikation und zeitlichem Verhalten analysiert und werkzeugunterstützt bei der Migration zu AUTOSAR und der gleichzeitigen Portierung auf Multicore-Systeme berücksichtigt. Es wurden vor allem Fragen der Datenkonsistenz bearbeitet, da viele der bei Single-Core gültigen impliziten Annahmen bezüglich Timing und Priorität bei einer Verteilung auf mehrere Cores nicht mehr gültig sind. Die kritischen Stellen müssen identifiziert und entsprechend multicore-fähig gemacht werden. Die erarbeitete Methodik beginnt mit einer Analyse der vorhandenen Maßnahmen zur Sicherstellung der Datenkonsistenz und des zeitlichen Verhaltens, gefolgt von einer Abschätzung der Änderungen beim Ressourcenverbrauch durch die Migration, bis hin zu den notwendigen Refactoringmaßnahmen und der Verifikation.

Zum Thema Parallelisierung von Software wurden am Beispiel einer Motorsteuerung verschiedene Möglichkeiten der Verteilung von Legacy Software auf mehrere Cores analysiert und miteinander verglichen. Dabei sollte aus Kostengründen und zur Risikominimierung der Umfang von Änderungen in der Software möglichst klein gehalten werden. Die Verteilungsstrategien „Task Distribution“, „Task Splitting“ und „Task Chaining“ werden beschrieben und verglichen.

In einer anderen Arbeitsgruppe wurden Mechanismen zur sicheren zeitlichen und räumlichen Segregierung von Anwendungen in Multicore-Systemen untersucht und verglichen, z.B. zur sicheren Trennung von Anwendungen mit unterschiedlichem ASIL oder zur Absicherung gegenüber Fremdsoftware. Im Einzelnen wurden die zeitlichen Schutzmechanismen „Execution Time Protection“, „Resource/Interrupt Locking Time Protection“, „Inter-Arrival Time Protection“, „Deadline Supervision“ und „Alive Supervision“ beschrieben und bewertet. Zur räumlichen Segregierung wurden insbesondere Mechanismen zum Speicherschutz mit MPUs (Memory Protection Units) analysiert und beschrieben, wobei die Vorgehensweise in AUTOSAR-Systemen detailliert herausgearbeitet wurde. Weiterhin wurden die möglichen Ursachen für Speicherschutzfehler und Vermeidungsstrategien beschrieben. Die Arbeitsergebnisse sind in Kapitel 5.3 „Domain Driven Segregation of Automotive Software“ des gemeinsamen Abschlussberichts nachzulesen.

Als Beispiel für eine Mixed Criticality Anwendung wurde eine Motorsteuerung untersucht. In solchen Systemen existiert eine Mischung aus kooperativem und preemptivem Rate Monotonic Scheduling, mit der Gefahr der Prioritätsinversion. Diese und weitere Fragestellungen im Zusammenhang mit Scheduling bei Mixed Criticality Systemen wurden analysiert und bewertet. Für die einzelnen Problemfelder wurden Gegenmaßnahmen vorgeschlagen.

Für TP6 „Demonstratoren“ hatte die Robert Bosch GmbH ursprünglich keine eigenen Beiträge geplant. Wir wurden jedoch von anderen Partnern gebeten, unsere Kompetenz auf dem Gebiet der Virtualisierung durch beratende Unterstützung der Arbeiten am Hochintegrations-Demonstrator (Plattform C) beizutragen. Dabei haben wir die Inbetriebnahme eines Hypervisors und insbesondere Fragen des Zugriffs auf von den Gast-Partitionen gemeinsam genutzte Ressourcen (z.B. CAN-Bus) unterstützt.

Zusammenfassend kann festgestellt werden, dass im ARAMiS-Projekt alle in der Vorhabensbeschreibung genannten Arbeitspakete bearbeitet und die Ziele weitgehend erreicht wurden. Die Aufgabenverteilung zwischen den Partnern und die Zusammenarbeit in Arbeitsgruppen wurden wie geplant umgesetzt. Der domänenübergreifende Ansatz hat wesentlich dazu beigetragen, die Lösungen durch das Einbringen und die Weiterentwicklung von zuvor nur domänenintern in Betracht gezogenen Konzepten auf eine breitere Basis zu stellen.

Obwohl im Rahmen der geplanten Arbeiten für die wesentlichen Herausforderungen beim Einsatz von Multicore-Hardware in Mixed Criticality Systemen prinzipielle Lösungen aufgezeigt werden konnten, hat sich im Rahmen der industriellen Umsetzung jedoch gezeigt, dass durch die dramatisch steigende Komplexität die Entwicklung solcher Systeme mit den aktuell verfügbaren Werkzeugen und Methoden an ihre Grenzen stößt. Zur Beherrschung dieser Herausforderungen werden auf dem Weg hin zu effizienteren Entwicklungsprozessen weitere Arbeiten notwendig sein. Ein Nachfolgeprojekt ist in Vorbereitung.

3.2 Notwendigkeit und Angemessenheit der Arbeiten

Aufgrund des Neuigkeitsgrades vieler der Themenfelder im Umfeld von eingebetteten Multicore-Systeme konnte nur eingeschränkt auf Vorarbeiten zurückgegriffen werden, so dass im Mittel in der Summe etwa zwei Personen über den gesamten Projektzeitraum mit den Arbeiten wie Recherchen zum Stand der Technik, Abstimmung mit den Partnern und insbesondere Erarbeitung neuer Lösungen und ihrer Dokumentation beschäftigt waren. Eine geringere Beteiligung wäre nicht sinnvoll gewesen, da dann für die

internen Projekte wichtige Aspekte nicht hätten bearbeitet und Synergien zwischen den Mitarbeitern nicht ausreichend hätten entstehen können.

Durch die Förderung konnten zwei zusätzliche Mitarbeiter mitfinanziert werden, so dass laufende andere wichtige Projekte und Arbeiten weiter durchgeführt werden konnten und in der Summe zwei Mitarbeiter die Arbeiten für ARAMiS ausführen konnten. Die für die Organisation, und damit auch für den Standort Deutschland wertvollen und für die Zukunftssicherung notwendigen Arbeiten wären ohne eine Förderung und die zusätzlichen Mitarbeiter nicht möglich gewesen. Ein wertvoller Zusatznutzen ergibt sich aus der Zusammenarbeit mit anderen Firmen und Organisationen im Rahmen des öffentlich geförderten Projekts, die auf andere Art in diesem Umfang kaum hätte organisiert werden können

3.3 Fortschritte auf dem Gebiet des Vorhabens

Nachdem in ARAMiS gezeigt wurde, dass Multicore-Prozessoren in sicherheits- und echtzeitkritischen Anwendungen nutzbringend eingesetzt werden können, hat das geplante Nachfolgeprojekt „ARAMiS II – Efficient Use of Multicore for safety-critical Systems in Mobility Domains“ zum Ziel, die Steigerung der Rechenleistung durch parallele Multi-/Manycore-Architekturen für die Mobilitätsdomänen Automobil, Avionik und Bahn noch effizienter zu erschließen, um so eine weitere Erhöhung von Sicherheit, Verkehrseffizienz und Komfort zu ermöglichen. Erfahrungen aus den ersten Serienprojekten mit Multicore-Technologie zeigen, dass für die effiziente Entwicklung solcher Systeme im Gegensatz zum Single-Core Fall deutlich feingranularere Informationen über die Softwarestruktur ausgetauscht werden müssen. Daher sind abgestimmte, wohldefinierte Modellierungsansätze sowie eine enge Zusammenarbeit zwischen OEM und Zulieferer notwendig. Der Zusammenschluss von deutschen Automobilherstellern, Systemlieferanten und Toolhersteller in diesem Projekt soll garantieren, dass Anforderungen aus der gesamten Wertschöpfungskette Berücksichtigung finden und die jeweiligen Kompetenzen in die zu erarbeitenden Lösungen einfließen. Dadurch ist die Weiterentwicklung der in ARAMiS erarbeiteten Konzepte in entsprechende Produkte mit einer hohen Marktattraktivität und -akzeptanz sichergestellt. Durch die Definition gemeinsamer Konzepte und deren Einbringung in laufende und zukünftige Standardisierungen von automobiler Software ist der Weg zu einer internationalen Verbreitung - in Verbindung mit einem entsprechenden Vorsprung der deutschen Industrie - geebnet.

3.4 Veröffentlichung der Ergebnisse

Die ARAMiS-Projektergebnisse werden in einem gemeinsamen Abschlussbericht und in Buchform veröffentlicht. Weitere Veröffentlichungen durch die Robert Bosch GmbH sind nicht geplant, da wir uns auf die industrielle Umsetzbarkeit konzentrieren, unsere Beiträge zu den Projektergebnissen fließen aber selbstverständlich in wissenschaftliche Arbeiten vor allem der Partner aus dem universitären Umfeld ein.

Aufgrund der vorwiegend methodisch und wissenschaftlich orientierten Natur unserer Projektbeiträge wurden von der Robert Bosch GmbH keine Erfindungen oder Schutzrechte beantragt. Die Projektergebnisse aus den Bereichen Hardware-, Software- und Safety-Architekturen, Methoden und Tools fließen jedoch bereits heute in die Entwicklung von elektronischen Steuergeräten für automobile Systeme ein.

Die geplante Veröffentlichung der ARAMiS-Ergebnisse, die an vielen Stellen Anregungen für darauf aufbauende Masterarbeiten und Promotionen enthält, wird zur Befruchtung der deutschen Forschungslandschaft beitragen.

4 Daimler AG

4.1 Wissenschaftlich-technische Ergebnisse

Es wurden verschiedene Szenarien, welche als Input für die Demonstratoren in TP6 (und TP5) dienen sollen, entworfen. In diesem Rahmen hat die Daimler AG (Analyse) zusammen mit der Universität Stuttgart für die Domäne Automotiv das Szenario „Kontinuität“ entworfen.

Des Weiteren wurde aktiv mit BMW das Szenario „Intelligentes Infotainment“ entworfen.

Ziel von „Intelligentem Infotainment“ ist:

- Vernetzung vom Fahrzeug u.a. mit „Smart-Phones“
- Personalisierung der Fahrzeugeinstellungen
- Domänenübergreifendes Infotainment / Mobilität
- Verteilte Datenhaltung / Anbindung an „Cloud Dienste“

Zusammengefasst wurden diese im „Drehbuch Businesstrip“.

Die Szenarien für TP5 und TP6 wurden in Enterprise Architect (EA) dokumentiert ebenso wie die Erweiterung der bestehenden automotive Anforderungen.

Aus den Anforderungen aus TP 1 wurde eine Logische Rechnerarchitektur definiert und Anforderungen an diese Rechnerarchitektur erstellt.

Die Definition der Rechnerarchitektur wurde mit den beteiligten Projektpartnern diskutiert und abgestimmt.

Die Anforderungen sind in den zugehörigen Ergebnisdokumenten dokumentiert. Abhängig von den Anforderungen wurde ein Architekturbild erstellt, welche eine Aufteilung der Kritikalitätsbereiche für Virtualisierung festlegt. Insbesondere wurden Anforderungen aus den zugehörigen Anwendungsfällen des Szenarios „Intelligentes Infotainment“ und bestehenden Automotive Richtlinien sowie ISO Standards abgeleitet. Diese wurden im Automotive „Enterprise Architect“ Modell eingebracht.

Des Weiteren wurde ein Infotainment Computermodel für die Automotive Domäne erstellt. Dies fand im Wesentlichen in Abstimmung zwischen der Avionik- und der Automotive-Domäne statt.

Für die TP3 wurde eine Untersuchung bzgl. State-of-the-art im Bereich GPUs und Virtualisierung durchgeführt. Hierzu wurde gemeinsam mit Freescale die i.MX Architektur auf ihre

Möglichkeiten untersucht. Diese wurden mit aktuellen GPUs und den Konzepten von Nvidia und Intel verglichen.

Die Ergebnisse hierzu wurden in das Ergebnisdokument eingetragen. Des Weiteren wurde ein Review dazu durchgeführt und die Verbesserungsvorschläge eingearbeitet.

Zur Definition von Tätigkeiten im Rahmen von AP 4.4 wurden zuerst gemeinsam mit den Projektpartnern die Problemfelder identifiziert woraufhin die folgenden Lösungskonzepte abzielen.

Für die Virtualisierung von GPUs wurden aus den vorhergehenden TP1 und TP2 Arbeiten die relevanten Anforderungen ermittelt und eine Untersuchung der wesentlichen Probleme bzgl. der aktuellen GPU-Architekturen durchgeführt. Hierbei wurden die gravierenden Probleme ermittelt, welche eine sichere GPU Anbindung an mehrere virtuelle Partitionen betrifft.

Des Weiteren wurde eine Konzeptuntersuchung bzgl. der Anforderung für GPU Scheduling durchgeführt sowie eine Untersuchung der Vorhersagbarkeit von Laufzeiten bzgl. OpenGL ES.

Im Rahmen von TP 4.3 wurde eine Analyse der Anforderungen aus ISO 26262 am Beispiel von Vector Microsar Safe durchgeführt. Hieraus wurden Anforderungen für den zukünftigen produktiven Einsatz ermittelt.

Weiter wurde eine Softwarearchitektur für GPU-Virtualisierung erstellt und eine Testimplementierung zur Verifikation erzeugt. Verschiedene GPU-Virtualisierungskonzepte wurden diskutiert und Teilaspekte davon anhand der Testimplementierung evaluiert. Mit Hilfe von Performancetests wurde die Effizienz für parallele GPU Zugriffe von verschiedenen Applikationen ermittelt. Des Weiteren wurden Konzepte zur Trennung von funktionalen und HMI Anteilen der Kombiinstrumente erstellt. Im Rahmen dessen wurden Kriterien zur Verteilung der Gesamtfunktionalität auf die verschiedenen Virtualisierungsdomänen festgelegt. Ein Grobentwurf zur Partitionierung der Anteile aus Headunit und Kombiinstrument für ein kombiniertes Steuergerät wurde erstellt.

Für die Zielplattform Freescale I.MX6 AI wurde die Anbindung der 2D GPU für die Verwendung im Umfeld Virtualisierung untersucht. Dabei wurde insbesondere das „bitblitting“-Verhalten der 2D-GPU mittels Performancetests betrachtet. Ein Aufbau von Automotive-Testszenarien zur Untersuchung der Leistungsfähigkeit der Systemarchitektur für Virtualisierung wurde erstellt. Dies ermöglicht eine realitätsnahe Evaluation der GPU-Virtualisierungskonzepte. Weiter wurde das Timing-Verhalten von OpenGL-ES-Befehlen untersucht. Hierbei wurden Konzepte entwickelt, welche ein einfaches Abschätzen der Laufzeit ermöglichen. Es wurden Performancetests bzgl. der graphischen Leistungsfähigkeit der ARM-Plattform erstellt.

Für eine effiziente Darstellung von 2D/3D Inhalten ist ein Compositing auf Seiten des Graphikservers, welcher die Verwaltung aller Grafikkontexte vornimmt, notwendig. Hierzu wurde ein Konzept entwickelt, welches eine effiziente Verwaltung der graphischen Inhalte der Anwendungen vornimmt und die sichtbaren Teilbereiche/Teilfenster darstellt. Dazu wurde eine Komponenten Compositing entworfen und testweise auf dem i.MX6 in Betrieb genommen. Mit geeigneten Tests wurde die Tauglichkeit nachgewiesen. Für eine Weiterentwicklung wurden Performancetests durchgeführt und weiterhin ein Algorithmus abhängig von den sichtbaren Anwendungsfenstern entwickelt.

Für die Umsetzungen des Algorithmus wurde auf Grundlage einer Ruby Script Implementierung, welche eine schnelle Testumsetzung ermöglicht, erste Evaluationen durchgeführt.

Des Weiteren wurde das Konzept um 2D Funktionalitäten für dynamisches Erzeugen und Manipulieren von Fenstern erweitert. Dies beinhaltet unter anderem auch die Animation von 2D Fenstern. Für die Einbindung eines CPU-Schedulings, welches von der UNI Stuttgart erstellt wird, wurden bereits Schnittstellen definiert. Dies ermöglicht eine einfache Integration auf Seiten eines Graphikservers. Es wurden weiter Regeln „zur Verteilung der Anwendungsfunktionalitäten zwischen den verschiedenen Virtualisierungsdomänen“ erstellt.

Um die Darstellung sicherheitskritischer Anwendungen zu gewährleisten, wurde die Entwicklung von Algorithmen, welche das Übermalen von überlappenden Fensterinhalten vermeiden, durchgeführt. Exemplarisch wurden Schnittstellen zwischen Automotive-Anwendungen und grafischer Visualisierung festgelegt, um eine übersichtliche Darstellung der Arbeitsweise der Virtualisierung zu ermöglichen. Teile der vormals genannten Ergebnisse wurden in das Ergebnisdokument E4.4.2 eingefügt.

Um eine bessere Aussagefähigkeit über die Funktionalität von AUTOSAR in Zusammenhang mit der Überwachung von Videodatenströmen mittels CRC Prüfung zu erhalten wurde die Zertifizierbarkeit des Gesamtsystems über ISO 26262 evaluiert.

Es wurde eine beispielhafte Konfiguration einer SWC mittels AUTOSAR zur Evaluierung der 2D Performance erstellt. Des Weiteren wurde untersucht in wie weit die i.MX6 Hardwareschnittstelle zur Überprüfung des Framebuffer-Inhaltes mittels CRC sich eignet.

Hierzu wurde ein alternativer graphischer 2D Pfad, welcher die Einsparung einer expliziten Überwachungshardware vorsieht, auf seine Tauglichkeit im Zusammenhang mit ASIL relevanten Ausgaben untersucht.

Hierbei wurde zum einen eine eingeschränkte Tauglichkeit attestiert. Für den Anwendungsfall 2D Präsentation für

sicherheitsrelevante wurde eine Tauglichkeit attestiert, jedoch ist eine Nutzung von 3D nicht möglich. Dies bedeutet wir haben nur einen eingeschränkten abgesicherten Pfad. Des Weiteren ist für die 2D Darstellung mit erheblichen Einschränkungen im graphischen Design zu rechnen, die ebenfalls die Tauglichkeit einschränken.

Damit man den nix-fused Daemon für Testszenarien besser konfigurieren kann, wurde eine DBus-anbindung implementiert. Mittels eines kleinen Tools, lässt sich der Daemon nun flexibel Konfigurieren. Für die große Menge an Verzeichnissen/Daten, welche in nix-fused verarbeitet werden müssen, wurde die Integration von sqlite in Betracht gezogen und mittels eines Pilotprojekts evaluiert. Da nix-fused einen rewrite in Qt5 erhalten hat, mussten einige Kernklassen in die neue Implementierung übertragen werden.

Für das Entwickeln eines flexiblen Deploymentlösung auf dem Fahrzeugbackend wurde mittels nix-fuse ein Teststand erstellt. Die Evaluationen für die Performance wurden durchgeführt. Die Evaluation beinhaltet verschiedene Testszenarien die sich in erster Linie auf fahrzeugspezifische Use Cases bezieht. Als erstes wurde ein Testszenario für ein dynamisches Nachladen von Kartendaten aufgebaut und ausgeführt. Hierbei zeigen sich vielversprechende Ergebnisse. Als zweites Szenario wurde das Ausführen von Programmen entworfen, welches darauf abzielt lokal nicht verfügbare Programme dynamisch („on demand“) vom Backend nachzuladen.

Da die Präsentation von sicherheits- / nicht sicherheitskritschen Fensterinhalten in IC/HU entsprechend der Sicherheitskritikalität gewährleistet werden muss, wurde ein Konzept für eine Zugriffkontrolle der Anwendungen auf die Anzeige erstellt. Eine Anwendung erhält nur dann Zugriff auf den Bereich einer Anzeige, wenn sie dafür eine Berechtigung besitzt. Die Zuweisung von Berechtigungen für den konfliktfreien Zugriff auf die Anzeigen kann zwischen den Anwendungen ausgetauscht werden, wenn kein Bedarf für die Darstellung in dem Anzeigenbereich vorliegt. Hierbei kann eine wesentlich flexiblere und vor allem sicherere gemeinsame Nutzung der Anzeigen von IC und HU ermöglicht werden.

Für die sichere Darstellung der Anwendungen, welche sich auf jeweils einer virtuellen Maschine befinden, wird eine Forwarding-Lösung für die Übertragung der Grafikdaten auf die Server-Partition verwendet, die einen sicheren Zugriff auf die GPU ermöglicht. Hierzu wurden erste Benchmark Simulationen mittels Benchmark Applikationen „GLmark ES“ und „eglgears“ durchgeführt. Des Weiteren wurden Implementierungen von Demonstratorapplikationen wie Speedometer mittels Forwarding auf der Zielplattform erstellt. Auf Grundlage dieser Evaluationen

wurden Optimierungen der Forwarding-Lösung durchgeführt und dadurch Performancesteigerungen von bis zu 70% erreicht.

Die Arbeiten wurden im Rahmen der Deliverables 5.1 – 5.5 durchgeführt. Arbeiten an dem Deliverable D5.2 waren iterativ, da mit der Integration weiterer Werkzeugklassen neue Anforderungen in die Spezifikation ergänzt werden mussten. Weitere wichtige Schritte waren das Deliverable D5.4 „Tool Integration in the ARAMiS RTP“ (AP2) und Vorbereitungen auf das Deliverable D5.5 „User Guideline for the Seamless Methodology, 2nd Version“ (AP5.2).

Die Aktivitäten im Rahmen des TP wurden mit geeigneten Prozessschritten strukturiert.

- Erhebung von Anforderungen mittels einer Befragung und eines Soll-/Ist-Vergleichs (AP 1)
- Erstellung der Erweiterung an die Erweiterungen der RTP (D5.1)
- Priorisierung der Anforderungen
- Erstellung einer Spezifikation auf Grundlage der RTP (D5.2)
- Erstellung von User Guidelines für eine konsistente Methodik (D5.3)
- Erstellung einer prototypischen Werkzeugkette (D5.4)
- Fortsetzung der Erstellung von User Guidelines für eine konsistente Methodik (D5.5)

Ziel des AP1 für das Deliverable D5.1 war es die Werkzeuge mit ihren im Rahmen von ARAMIS geplanten Erweiterungen zu erfassen und in einen projektweiten Zusammenhang zu stellen.

Im ersten Schritt, dem Task 5.1.1, wurde, unter anderem mit Hilfe der aus TP1 gewonnen Ergebnisse, ein Fragebogen erstellt. Dieser wurde zunächst TP5 intern kommuniziert, und nach präziser Überarbeitung projektweit ausgerollt. Im Anschluss wurden die erfassten Fragebögen ausgewertet. Die Ergebnisse der Auswertung dienen als Input für den Folgeprozess des TPs und sind ein fester Bestandteil des Deliverables 5.1.1.

Die Fertigstellung des Deliverables D5.1 zu Anforderungen an die Reference Technology Platform (RTP) war eines der zentralen Themen. Die RTP wurde fest in der Vorhabensbeschreibung verankert als eine Vorarbeit aus vorherigen Projekten wie z.B. CESAR und SPES 2020. Hiermit sollte es ermöglicht werden, im Rahmen von ARAMiS Multicore Systeme durchgängig modellbasiert und werkzeuggestützt zu betrachten. Die RTP stellt dabei technisch die Mechanismen zur Verfügung, um eine verteilte heterogene Werkzeuglandschaft aufeinander abzustimmen, so

dass diese durchgängig angewandt werden können. Im Rahmen des Deliverables wurde ein Überblick über die RTP gegeben und Anforderungen an die RTP aus Sicht von Multicore gestellt.

Die Anforderungen bezogen sich auf die Methodik und Werkzeuge der RTP. Bei der Methodik lag der Schwerpunkt auf der Betrachtung von Multicore spezifischen Anforderungen und deren Durchgängigkeit im RTP-Entwurfsraum. In Bezug auf die Werkzeuge wurden für die in ARAMiS enthaltenen Domänen Multicore-geeignete Werkzeugketten beschrieben und Anforderungen an diese erfasst.

Für D5.2 „Spezifikation Interoperabilität für die Integration der Werkzeuge“ wurden die bereits existierenden spezifizierten Use-Cases und Requirements aus der ersten Iteration übertragen und darauf basierend ein Metamodell (Struktur- und Sequenz-diagramme) erstellt. Besonderer Fokus bei der Erweiterung lag auf den notwendigen Artefakten für Zeitanalysen, Nebenläufigkeits-analysen, deren Kombination sowie Fragestellungen zur Unterstützung von Segregation für Multi-Core Systeme. Zwei Themen Entscheidungen hatten hierbei wesentlichen Einfluss auf die Weiterentwicklung.

- Erstens wurde festgelegt, dass alle Artefakte im Enterprise Architect erstellt werden sollen. Auch galt es eine Durchgängigkeit bei der Modellierung zu erreichen. Dazu war es notwendig, die bereits existierenden Requirements und Use Cases aus D5.1 in Enterprise Architect zu spezifizieren und entsprechend anzupassen. Es wurden zudem neue Requirements ergänzt, die sich aus der Zusammenarbeit mit der Absint zur Weiterentwicklung des Werkzeugs Astrée (für Nebenläufigkeitsanalysen) ergeben haben.
- Zweitens wurden Abnahmekriterien für die Requirements teilweise in Form von Testfällen definiert, um eine gemeinsame Diskussions- und Verhandlungsgrundlage mit den Werkzeugherstellern über notwendige zu erweiternde / neue Funktionen der betroffenen Werkzeuge zu erreichen.

Für das Deliverable D5.2 wurde zudem ein Ansatz entwickelt, der es erlaubte die Schnittstellen von Methoden, die in ARAMiS entwickelt / erweitert wurden, modellbasiert zu beschreiben und mit dem ARAMiS RTP Metamodell in Beziehung zu setzen. Dies erlaubt das Untersuchen von möglichen Verzahnungen von Methoden für den methodischen Leitfaden, der im Deliverable D5.5 dargestellt wurde.

Diese wichtigsten Use-Case für eine industrielle Anwendung wurden für konkrete Werkzeugszenarien spezifiziert und erstellt. Hierbei wurden konkret für die beschriebenen Szenarien die ausgetauschten Informationen beschrieben. In einem weiteren

Schritt wurden für die Multicore Use Case Erweiterungen des Interoperability Metamodels das RTP für ARAMiS beschrieben. Die konkreten Informationen aus den gewonnenen Szenarien wurden anschließend auf das ARAMiS RTP Metamodel aus Schritt übertragen, um sicherzustellen, dass das Metamodell die notwendigen Inhalte aus den Szenarien beinhaltet. Diese erste Version wurde (schon) in dem Deliverable D5.2 beschrieben und diente als Ausgangspunkt für eine iterative Weiterentwicklung des Metamodels. Konkret wurde die Aktualisierung in einem halbjährlichen Zyklus bedarfsorientiert während der Projektlaufzeit von ARAMiS durchgeführt.

Im Rahmen der Arbeiten an dem Deliverable D5.3 wurde ein Fragebogen für die erstellten/eingesetzten Methoden erarbeitet. Schwerpunkte lagen in der Beschreibung der Schnittstelle und der Erarbeitung von Voraussetzungen, um mit der Methode arbeiten zu können. Ziel war und ist es, zum einen Leitfäden für den Umgang mit den Methoden abzuleiten sowie eine gesamte Methodik zu erarbeiten. Im Berichtszeitraum wurde der Fragebogen an die Projektpartner verteilt und initiale Versionen der Methoden beschrieben.

In einer Vorabstimmung für das Deliverable D5.4 wurden die Möglichkeiten einer Implementierung des spezifizierten Metamodells in den Werkzeugen der Kooperationspartner abgestimmt, Dazu wurden durch OFFIS und Daimler erste Sondierungsgespräche mit der Uni Stuttgart, Uni Kiel sowie Symtavision geführt. Für die Umsetzung der Implementierung und der Einschätzung der qualitativen Aussagekraft der Ergebnisse durch die Analysewerkzeuge wurde damit begonnen die Ergebnisse der verschiedenen Werkzeuge anhand einer industriellen Software zu vergleichen. In dem Zusammenhang wurden die Entwicklungen der Nebenläufigkeitsanalysewerkzeuge Gropius, MEMICS und Bauhaus fortgesetzt, um in einer speziellen Tool-Konstellation, die gemeldeten Nebenläufigkeiten der Werkzeuge zu vergleichen und zu bewerten. Zur Validierung von Ergebnissen durch Bauhaus konnten Ergebnisse durch Gropius und dem Tool Paar Gropius-MEMICS verglichen werden.

Mit den Ergebnissen konnten die Arbeiten an dem Deliverable D5.4 „Tool Integration in the ARAMiS RTP“ fortgesetzt werden. Es wurden die Werkzeuge der verschiedenen Partner konsolidiert und basierend auf den in D5.2 entwickelten Use Cases angewandt. Die Schwerpunkte standen dabei auf

- Feinabstimmung des Metamodels
- Umsetzung/Implementierung von Use-Case
- Validierung von Use-Case anhand eines Industrieprojektes

Das Ergebnis war ein Prototyp in Form einer Werkzeugkette der Partner ISTE / Uni Stuttgart, CAU Kiel und Symtavision. Neben

der Werkzeugkette wurden im Rahmen des Deliverable D5.4 explizit Artefakte im Rahmen einer Gap Analyse identifiziert, die bis dahin noch nicht durch das ARAMiS RTP Metamodel beschrieben waren. Die fehlenden Artefakte sind Ergebnis für eine weitere Iteration des Metamodells. Basierend auf dieser aktuellen Version des Metamodells wurde durch die Partner angeregt, eine zusätzliche Iteration im Anschluss an das Deliverable durchzuführen.

Wegen den nun vorliegenden Ergebnissen der ersten Analysen mit dem Prototyp, konnten weitere neue Werkzeuge mit demselben Industrieprojekt untersucht und verglichen werden.

- Ein erster Vergleich außerhalb von ARAMiS erfolgte mit den Ergebnissen kommerzieller Anbieter wie Bugfinder (Mathworks) und Astrée (Absint).
- Eine weitere Evaluation außerhalb von ARAMiS wurde mit dem Analysewerkzeug Goblint (TUM) in Zusammenarbeit mit den Partnern aus MBAT durchgeführt. Die Ergebnisse wurden mit den MBAT Partnern besprochen und dienten für weitere Tests.

Die Umsetzung der Use-Case und der Aufbau des Prototyps zeigten, dass die Werkzeugkette eine Optimierung der Werkzeuge selbst nach sich zog. Eine entsprechende Evaluierung wurde vor allem zusammen mit Gropius (CAU/Uni Kiel) vorgenommen und brachte zahlreiche neue Möglichkeiten und Ziele für die Entwicklung des Werkzeuges mit sich. Diese wurden teilweise in ARAMiS verfolgt, werden jedoch auch nach Projektende weiterverfolgt werden.

Ein weiterer Vergleich zwischen SymTA/S (Symtavision) und ORCA (OFFIS) wurde an einer anderen Stelle des Prototyps durchgeführt. Dazu wurde eine Gap Analyse des Metamodells erstellt. Die dabei vorgeschlagenen Erweiterungen des ARAMiS RTP Metamodelles wurden mit beteiligten Partnern abgestimmt und nach Korrekturen in das Metamodell integriert. Dieses erweiterte Metamodell wurde verwendet, um ORCA in die Werkzeugkette zu integrieren. Ziel war es die von beiden Werkzeuge erzeugten Daten für Gropius, Bauhaus, … qualitativ zu vergleichen und zu bewerten. Daneben war es eine Möglichkeit die einfache Handhabung des Metamodells für eine weitere Integration eines Werkzeuges zu überprüfen.

Im Umfeld des Automotive Demonstrators VTC aus AP61 wurde eine Beschreibung der Ziele erstellt. Hierbei lag der Fokus insb. auf der Beschreibung der Testszenarien, welche für die Evaluation verwendet werden sollen. Dies erfolgte in erster Linie in Abstimmung mit den beteiligten Projektpartnern.

Es wurde ein Aufbau für die Demonstratorplattform B (VCT-B), auf Grundlage eines Cockpits, welcher für Demonstrationszwecke

verwendet wird, erstellt. Aufgrund der Festlegung der Storyline und der Showcases wurden Testanwendungen festgelegt, welche in einer Entwicklungsumgebung für die Applikationsentwicklung, basierend auf den Szenarien, erstellt wurden. Des Weiteren wurde ein Konzept zur Festlegung der Darstellung von Warn- und Informationsmeldungen im CGI-Studio erstellt.

Für die Demonstratorplattform B wurde für die Anbindung der CAN-Busse eine Raspberry PI Plattform als Bridge von CAN Signalen auf Input Events über Ethernet auf die i.MX6 Plattform eingesetzt. Diese Anbindung der CAN-Busse direkt an die i.MX6 Plattform war nicht möglich, da ZBE und MRSM an zwei verschiedenen Bussen hängen und der i.MX6 Board aber nur eine CAN-Schnittstelle bei gleichzeitiger Verwendung des Ethernetports bietet. Das Raspberry PI Boards agiert dabei als eine CAN-Bridge, welche die beiden, 500kbit und 125kbit Busse, zu einem zusammenführt. Das Anbinden der CAN-Busse mittels Raspberry PI wurde getestet und optimiert.

Für die Demonstratorplattform B wurde eine Tabletvisualisierung umgesetzt, welche den Status des CANBusses darstellt. Für die Busvisualisierung wurde eine Node.JSApplikation entworfen.

Des Weiteren wurde eine Android App entwickelt, welche eine Anbindung eines Tablets (Nexus 7) ermöglicht. Mittels dieser App können die Inputevents ebenfalls verwendet werden um eine Steuerung des Tablets mittels der Fahrzeugtasten zu ermöglichen. Das Forwarding-Konzept für das Weiterleiten von OpenGL Befehle und Window-Managing Befehlen wurde in die Demonstrator-plattform integriert und getestet. Hierzu wurde für den im Demonstrator verwendeten Hypervisor eine Shared-Memory (SHM) Lösung mittels PikeOS Kernelmodule implementiert, welche den Austausch der Daten über einen isolierten SHM-Bereich zwischen Client-Partition und Server-Partition ermöglicht. Hierzu wurden dann Benchmarktests durchgeführt, welche nach Einbringen von Optimierungen vielversprechende Ergebnisse lieferte. Es konnte mit Hilfe der Messungen nachgewiesen werden, dass das OpenGL-Forwarding unter Optimierung eine effiziente Darstellung von Applikation, welche auf verschiedenen Partitionen laufen, ermöglicht.

4.2 Notwendigkeit und Angemessenheit der Arbeiten

Bezugnehmend auf der förderpolitischen Definition der Vorhabensbeschreibung für die Domäne Automotive sind folgende Punkte relevant. Die detaillierte technische Beschreibung der Ergebnisse ist in Kapitel 4.1 zu finden.

Es ergeben sich u.a. folgende potentielle Themenschwerpunkte:

- Konzeption Automotive-tauglicher Integrationsplattformen zur Funktionspartitionierung unter Berücksichtigung der aktuellen und zukünftigen Fahrzeugarchitekturen. → Die wesentlichen Ergebnisse wurden in TP4, TP6 erarbeitet.
- Analyse von Safety/Zertifizierungsaspekten unter Nutzung von Multicore im Kontext funktionaler Hochintegration für verschiedene Fahrzeugdomänen. → Die wesentlichen Ergebnisse wurden in TP5 bearbeitet.
- Analyse der Systemarchitektur unter dem speziellen Aspekt der Safety und Zertifizierbarkeit für Fail Operational-Systeme. Zertifizierbarkeit der Ausfallsicherheit / Hochverfügbarkeit der Funktion des Gesamtsystems durch Kombination von Hardware- und reiner Software-Redundanz. Erarbeitung eines Konzeptes unter Gewährleistung der Einhaltung der Echtzeitanforderungen der Funktion bei Verlagerung. → Die wesentlichen Ergebnisse wurden in TP4, TP5 bearbeitet.
- Darstellung von Abhängigkeiten zur optimierten Auslastung einer Multicore Architektur (Visualisierung, Optimierungsvorschläge). Klärung der Auswirkung der Multicore Architektur auf den Software-Entwicklungsprozess. Methoden der Migration existierender Software-Assets auf Multicore Architekturen. → Die wesentlichen Ergebnisse wurden in TP5 bearbeitet (die optimale Auslastung einer Multicore Architektur wurde nicht betrachtet). Im Rahmen von TP4 bzw. TP6 wurde ein GPU Scheduling im Rahmen der Demonstrator Erstellung durchgeführt.

Bewertung bestehender Virtualisierungstechnologien unter den Aspekten Safety und Security zur Unterstützung der funktionalen Hochintegration in verschiedenen Fahrzeugdomänen. Erarbeitung von Konzepten zur Erfüllung der automotiven Anforderungen. → Die wesentlichen Ergebnisse wurden in TP4, TP6 (Fokus Safety) bearbeitet.

4.3 Fortschritte auf dem Gebiet des Vorhabens

Die Prüfung von Werkzeugen und die Umsetzung von Multicore Themen wird mittelfristig bei allen Werkzeuganbietern vollständig evaluiert werden. Ergebnisse werden in eine unternehmenseigene „Testdatenbank“ einfließen. Dies schafft zukünftig ökonomische Vorteile zum Beispiel bei der Wahl von Methoden oder Werkzeugen. Das schafft langfristig Vorteile bei der Wahl von Methoden oder Werkzeugen und somit einer Erhöhung der Effizienz. Das Thema soll nach 2016 weiterverfolgt werden.

In der Kooperation mit der Uni Kiel wird die Weiterentwicklung von Werkzeugen wie Gropius (Uni Kiel) und MEMICS (Daimler) gefördert. Es kann anhand von industriellen Projekten (Projektsourcen, OSEK, Autosar) die Entwicklung vorangetrieben werden. Gropius soll voraussichtlich Anfang 2016 genauere Aussagen zu möglichen Race-Conditions ermöglichen.

Verbesserung der Methoden und Werkzeuge: Im Rahmen von TP5 findet eine weitere Evaluierung von Werkzeugen (Gropius, Astrée, …) statt. Dies wurde und wird exemplarisch mit zwei ausgewählten Industrieprojekten aus der Entwicklung der Daimler AG auch für andere Werkzeuge durchgeführt.

Ziel ist es, für kommende Serienprojekte den Toolherstellern automotiv-spezifisches Feedback und Ergebnisse zu zuführen und eine effizientere Arbeitsweise zu ermöglichen.

Dazu soll eine geeignete Vorgehensweise für den Einsatz von Methoden und Werkzeugen im innovativen Umfeld als Qualitätsmanagementmethode entworfen und eingeführt werden. Das soll Mitte 2016 erfolgen.

Spezifikation: Durch die Erstellung des Prototyps ist die Grundlage für eine zukünftige Produktweiterentwicklung eines Analysetools geschaffen worden.

Forschung: Die in ARAMiS gesammelten Erfahrungen haben zu einem weiteren öffentlich geförderten Anschluss-Projekt geführt. Im Rahmen des ITEA-3 geförderten Forschungsprojektes ASSUME und gegebenenfalls ARAMiS II wird Daimler mit weiteren Partnern an der Erweiterung von Multicore Analysen arbeiten (ab 2015).

Im Zusammenhang mit TP4/TP6 können das Headunit (HU) und das Instrument Cluster (IC) in naher Zukunft in einer Hardware konsolidiert werden. Dies stellt eine Kostenreduzierung dar, wobei die Höhe noch nicht exakt eingeschätzt werden kann. Auf Grundlage der ARAMiS Ergebnisse wird dies für zukünftige Serienprojekte evaluiert. Die Ergebnisse wurden dem verantwortlichen Serienbereich übergeben.

Im Rahmen der GPU Virtualisierung mittels RT-Linux wird an einer Evaluation in Kooperation mit der Universität Stuttgart (IPVS) im Anschluss zu dem Projekt weitergearbeitet, da dies für zukünftige Lösungen mit Virtualisierung unabdingbar ist und derzeit keine Produkte existieren, welche den Automotive Anforderungen dazu gerecht werden.

4.4 Veröffentlichung der Ergebnisse

Bereits während der Laufzeit von ARAMiS, hat Daimler die erarbeiteten Ergebnisse präsentiert. Im Rahmen Konferenzen und

Kongressen hat Daimler zudem wesentliche Erkenntnisse und Werkzeuge vorgestellt. Zudem wurde in kleineren Arbeitsgruppen und Workshops das Thema vermittelt. Unternehmensintern wurden ebenfalls Veranstaltungen wahrgenommen.

Folgende Veröffentlichungen wurden im Rahmen von ARAMiS publiziert:

[1] Simon Gansel, Christian Maihoefer, Stephan Schnitzer, Frank Duerr, Kurt Rotherme
Towards Virtualization Concepts for Novel Automotive HMI Systems
International Embedded Systems Symposium, IESS 2013
Paderborn, Germany, June 17-19, 2013

[2] Stephan Schnitzer, Simon Gansel, Frank Dürr, Kurt Rothermel
Concepts for Execution Time Prediction of 3D GPU Rendering
9th IEEE International Symposium on Industrial Embedded Systems
Pisa, Italy, June 18-20, 2014

[3] Simon Gansel, Stephan Schnitzer, Ahmad Gilbeau-Hammoudy, Viktor Friesen, Frank Dürr, Kurt Rothermel, Christian Maihöfer
An Access Control Concept for Noval Automotive HMI Systems
ACM Symposium on Access Control Models and Technologies (SACMAT)
London, Canada, June 25-27, 2014,

[4] Dirk Nowotka, Johannes Traub
Formal Verification of Concurrent Embedded Software
International Embedded Systems Symposium, IESS 2013
Paderborn, Germany, June 17-19, 2013

[5] Thorsten Ehlers, Dirk Nowotka, Philipp Sieweck
Finding race conditions in real-time code by using formal software verification
12th International Conference on Formal Modeling and Analysis of Timed Systems
Florence, Italy, September 8-10, 2014

[6] Thorsten Ehlers, Dirk Nowotka, Philipp Sieweck
Communication in massively-parallel SAT Solving
IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2014)
Limassol, Cyprus, November 10-12, 2014

[7] Tayfun Gezgin, Stefan Henkler, Ingo Stierand, Achim Rettberg, Carl von Ossietzky
Evaluation of a State-based Real-Time Scheduling Analysis Technique
12th IEEE International Conference on Industrial Informatics
Porto Alegre, Brazil, July 27-30, 2014

[8] Philipp Reinkemeier, Heinz Hille, Stefan Henkler
Towards Creating Flexible Tool Chains for the Design and Analysis of Multi-Core Systems
SE-WS 2014, Kiel, Germany, Februrary 25-26, p. 99-108

[9] Simon Gansel, Stephan Schnitzer, Riccardo Cecolin, Frank Duerr, Kurt Rothermel, Christian Maihoefer
Efficient Compositing Strategies for Automotive HMI Systems
SIES'2015: 10th IEEE International Symposium on Industrial Embedded Systems
Siegen, Germany, June 8-10, 2015

[10] Simon Gansel, Stephan Schnitzer, Ahmad Gilbeau-Hammoud, Viktor Friesen, Frank Dürr, Kurt Rothermel, Christian Maihöfer and Ulrich Krämer
Context-aware Access Control in Novel Automotive HMI Systems
ICISS 2015: Eleventh International Conference on Information Systems Security, Kolkata, India, December 17-20, 2015

5 Fortiss GmbH

5.1 Wissenschaftlich-technische Ergebnisse

Im Rahmen des ARAMiS Projektes hat die fortiss GmbH Themen in folgenden Teilarbeitspaketen geleistet: TP1 (Anforderungen und Szenarien), TP5 (Durchgängige Entwicklungsmethodik und Anbindung an RTP) und TP6 (Demonstratoren).

TP1

Ziel des TP1 war es Anforderungen an Multicore-Systeme und zugehörige Technologien abzuleiten und zu validieren.

Im Rahmen dieses Teilprojekts war fortiss GmbH daran beteiligt Anwendungsszenarien für die Mobilitätsdomänen Automotive, Avionik und Bahn zu erstellen. Aus diesen Anwendungsszenarien konnten auf der einen Seite die späteren Anforderungen für die folgenden Teilprojekte abgeleitet werden, auf der anderen Seite wurde mit diesen Szenarien die Verbindung zur AgendaCPS hergestellt.

TP5

Das Ziel des TP5 war es zum einen Methoden und Werkzeuge zur Entwicklung von Multi-core basierten System bereitzustellen und zum anderen eine Werkzeugplattform mit Interoperabilitätskonzept zu entwickeln, wo diese Werkzeuge und Methoden integriert werden können.

Bei der Entwicklung der Werkzeugplattform war fortiss GmbH als Dokumentverantwortlicher für das D5.5 direkt beteiligt. Im Rahmen dieses Dokumentes wurden Verfahren überlegt, wie man feststellt ob zwei Werkzeuge miteinander kombiniert werden können.

Die Methode, die fortiss GmbH im Rahmen dieses Teilprojekts weiterentwickelt hat, ist die der Zustandsraumexploration (Design Space Exploration (DSE)). Unter dem Begriff Design Space Exploration verstehen wir einen Prozess bei dem es darum geht systematisch Lösungen in einem Lösungsraum zu finden, die bestimmte Eigenschaften erfüllen. Dieser Ansatz kann bei Systementwicklung als Unterstützung bei Entscheidungsfindungen benutzt werden (z.B.: wie allokiere ich die Software am besten auf meiner Hardwareplattform).

Um den Nutzen des Ansatzes zu demonstrieren haben wir Instanzen davon in das Forschungswerkzeug AutoFOCUS3 (AF3) integriert. AutoFOCUS3 ist eine modellbasierte Entwicklungsumgebung, die die Entwicklung von reaktiven Echtzeitsystemen entlang unterschiedlicher Abstraktionsebenen erlaubt. Eine der ersten Erweiterungen war es das Datenmodell für die Technische Architektur in AF3 so anzupassen, dass man damit Multicore Architekturen darstellen kann. Das erweiterte Datenmodell hat es dann im nächsten Schritt möglich gemacht die

schon vorhandenen DSE Methoden zur Synthese von Deployments und Schedules für den Multicore-Kontext anzupassen. Daraufhin gab es Arbeiten in zwei unterschiedliche Richtungen.

Zum einen wurde eine neue DSE Methode ins AF3 entwickelt und in das Werkzeug integriert, die es erlaubt hat nicht nur Deployments und Schedules zu generieren, sondern auch die zugehörige technische Architektur einer Multi-core Architektur auf Basis eines Komponentennetzwerks zu synthetisieren. Die Ergebnisse dieser Methode, bestehend aus dem Tripel (Deployment, Schedule, technische Architektur), waren pareto-optimiert bezüglich Kosten, Zeitverhalten und Energieverbrauch.

Zum anderen wurden die bestehenden Methoden weiter an Anforderungen der Mobilitätsdomänen angepasst. Basierend auf der ISO2626-2 wurden neue Deployment-Constraints in AF3 integriert, die ein Deployment garantieren, dass standardkonform ist. Damit sind diese Deployments „correct by design“ und können somit in Safety Cases als „Beweismittel“ verwendet werden.

Mit diesen ganzen Neuerungen gibt AutoFOCUS3 dem Benutzer relativ viele Freiheiten, welche Constraints er bei z.B. der Deploymentsynthese verwenden soll. Für den Fall, dass das Syntheseverfahren keine gültige Lösung findet und es nicht sofort ersichtlich ist, an welchen Constraints es liegt, wurde noch die UnSAT Core Funktion in die DSE Verfahren integriert. Diese Funktionalität macht es möglich den Benutzer darauf hinzuweisen, welche Constraints nicht erfüllbar waren.

Im weiteren Verlaufe des Projektes wurden Teile des AutoFOCUS3 Meta-Modells auf Meta-Modelle von gängigen Standards (ISO2626-2 und AUTOSAR) und Technologien (PikeOS) abgebildet. Damit wurde gezeigt, dass man AF3 in einen industriellen Entwicklungsprozess integrieren könnte.

Zusätzlich hat fortiss GmbH gemeinsam mit dem Fraunhofer IESE eine Umfrage zum Thema Werkzeugunterstützung für Design Space Exploration durchgeführt. Aus dem Forschungsaspekt hat diese Umfrage mehrere positive Effekte für fortiss. Zum einen können die Ergebnisse dieser Umfrage als Input für weitere Forschung verwendet werden und zum anderen wurde auf der Grundlage dieser Umfrage zusammen mit IESE ein Paper verfasst, welches auf die Problemstellungen, mit denen sich fortiss beschäftigt, aufmerksam macht. Zu aller Letzt wird durch das Paper Aufmerksamkeit auf das Forschungswerkzeug AutoFOCUS3 gezogen, da es als ein Beispiel für ein Werkzeug aufgeführt wird, welches Design Space Exploration unterstützt.

TP6

Das Ziel von TP6 war es zum einen die Architekturvorlagen und Methoden aus TP2-TP4 und zum anderen die Werkzeuge und Methoden aus TP5 zu validieren.

Im Rahmen dieses Teilprojektes haben wir einen Demonstrator gebaut, bei dem wir die Modelle von AutoFOCUS3 nahtlos auf die Hardware bringen konnten. Als Betriebssystem haben wir PikeOS benutzt. Bei der Benutzung von dem Freescale Board und Pike OS haben wir damit Teilergebnisse aus TP3 (Multicore-fähige embedded Hardware) und aus TP4 (Virtualisierung mit PikeOS) validiert. Im AutoFOCUS3 haben wir den zugehörigen Applikationscode und Konfigurationsdateien generiert. Insbesondere für die Synthese von Konfigurationsdateien wurden unterschiedliche Design Space Exploration Methoden (Allokation, Scheduling, etc.) verwendet. Damit haben wir unsere Methoden aus TP5 validiert.

5.2 Notwendigkeit und Angemessenheit der Arbeiten

Komplexe Fragestellungen, wie sie im Rahmen dieses Projektes angegangen wurden, erfordern die enge Zusammenarbeit von Unternehmen und Forschungseinrichtungen, um ganzheitliche, Disziplinen-übergreifende Herangehensweisen zu ermöglichen. Insbesondere die Entwicklung von Verfahren der Zustandsraumexploration, sowie der Aufbau eines Demonstrators, in dem die entwickelten Verfahren und Konzepte dargestellt und demonstriert werden können ist sehr ressourcenintensiv. Allerdings können nur solche Demonstrator-basierten Lösungen zeigen, dass die entwickelten Verfahren, Konzepte und Werkzeuge den Anforderungen der Industrie genügen und somit zielgerichtet zur Erhöhung der Wettbewerbsfähigkeit der deutschen Industrie beitragen können.

5.3 Veröffentlichung der Ergebnisse

[1] S. Voss, S. Zverlov *et al.*, “Design space exploration in autofocus3 - an overview,” in *IFIP First International Workshop on Design Space Exploration of Cyber-Physical Systems*. Springer, 2014

[2] B. Schaetz, S. Voss, S. Zverlov, “Automating design-space exploration: Optimal deployment of automotive sw-components in an iso26262 context,” in *Design Automation Conference (DAC), 2015 52st ACM/EDAC/IEEE*, 2015

[3] P. Diebold, C. Lampanosa *et al.*, “Practitioners’ and researchers’ expectations on design space exploration for multicore systems in the automotive and avionics domains – a survey,” in *EASE Proceedings*. ACM Digital Library, 2014

[4] S. Zverlov and S. Voss, “Synthesis of pareto efficient technical architectures for multi-core systems,” in *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*. IEEE, 2014, pp. 366–371

[5] S. Zverlov, M. Khalil, M. Chaudhary, “Pareto-efficient deployment synthesis for safety-critical applications in seamless model-based development” in ERTS Proceedings, 2016

[6] Birgit Penzenstadler, Jonas Eckhardt, Wolfgang Schwitzen, Maria Victoria Cengarle, Sebastian Voss: Inter-Domain Requirements and their Future Realisability: The ARAMiS Cyber-Physical Systems Scenario, Technical Report, TUM 2011

6 Fraunhofer AISEC und Fraunhofer IESE

6.1 Wissenschaftlich-technische Ergebnisse

Dieser Abschnitt bietet einen Kurzüberblick über die von Fraunhofer AISEC und Fraunhofer IESE im Rahmen von ARAMiS erzielten Ergebnisse. Genauere Ausführungen lassen sich im gemeinsamen ARAMiS Abschlussbericht finden, insbesondere in den Kapiteln „Safety and Certification Aspects“, „Security and Multicore“, sowie „Evaluation Strategy“.

6.1.1 Fraunhofer AISEC

Das Fraunhofer AISEC agierte innerhalb von ARAMiS in enger Kooperation mit zwei Lehrstühlen für Informationssicherheit an der TU München, und zwar an den Fakultäten für Elektrotechnik und Informationstechnik. Diesem Verbund oblag im Wesentlichen die Verantwortung für die Sicherheit (im Sinne der Informationssicherheit oder „Security“, nicht „Safety“) innerhalb des Projektes. Daher war das Fraunhofer AISEC in den Teilprojekten TP2, TP3 und TP4 jeweils stark in den Arbeitspaketen 2 engagiert, die jeweils Sicherheitsaspekte bearbeiteten. Die Leitung der Security-Arbeitspakete oblag den drei im Verbund agierenden Institutionen. Im Speziellen war das Fraunhofer AISEC für die Leitung der Arbeitspakete 2.2 und 4.2 verantwortlich.

Die Konzeptarbeiten in Teilprojekt TP2 umfassten ein Zusammenspiel aus Software und Hardware. Hierbei war das Fraunhofer AISEC in AP2.2 hauptsächlich für die Teile der Sicherheitsarchitektur verantwortlich, die die Einbindung eines HSM in das Multiprozessorsystem beschreiben. Im Anschluss an die Konzeptarbeit wurden auf Basis der Ergebnisse aus E2.1 in Abstimmung mit TP3.2 und TP4.2 die Architektur des ARAMIS CPS aus Security-Sicht definiert und abstrahiert. Auf Basis der Architektur erfolgte die Angriffsanalyse und Aufstellung der Angriffsbäume. Die Struktur der Bedrohungsanalyse wurde erarbeitet.

Im Bereich der Informationssicherheit konzentrierten sich die Arbeiten des Fraunhofer AISEC auf so genannte Hardware Secure Elements (HSM). Die spezielle Aufgabenstellung aus ARAMiS bestand in der Tatsache, dass ein HSM in ein Cyber-Physical System, also ein eingebettetes System mit teilweise kritischen Steuerfunktionen, mit Mehrfach-Prozessor integriert werden musste. Dabei mussten verschiedene auf einer solchen Plattform integrierte Systeme die HSM-Ressourcen nutzen können. Dies

stellte eine Herausforderung dar und machte durchgängige Integrationskonzepte mit Software- und Hardwareunterstützung notwendig.

Im Rahmen des TP3.2 beschäftigte sich das Fraunhofer AISEC hierbei mit der Feinspezifikation und den Implementierungsaufgaben bezüglich des virtualisierten HSMs. Die Architekturvorgaben des HSMs aus TP2.2 wurden konkretisiert und mit möglichen Umsetzungen auf FPGA-Basis abgestimmt. Schlussendlich wurde ein Xilinx Virtex5 FPGA als Plattform ausgewählt.

In TP4.2 lag eine Hauptaufgabe lag darin, den Treiber zum Ansprechen der verschiedenen HSM-Kontexte aus den virtuellen Maschinen (Partitionen) heraus zu realisieren. Dieser Treiber ist zweiteilig: In der Partition selbst stellt sich der Treiber als Device dar, über das man mit dem HSM kommunizieren und die unterstützte Funktionalität verwenden kann. Hierzu zählt beispielsweise die Durchführung kryptografischer Operationen oder die Überprüfung von Signaturen. Der zweite Teil des Treibers ist in PikeOS integriert, dem zugrundeliegenden Betriebssystem, das den Betrieb der verschiedenen Partitionen erlaubt. PikeOS fängt die Zugriffe der Partitionen auf das HSM-Device ab und erweitert sie um ein Label, das beschreibt, aus welcher Partition heraus der Zugriff auf das HSM erfolgt. Diese Information ist für das HSM wichtig, damit es die Zugriffe auf den der Partition zugewiesenen Speicherbereich im HSM einschränken kann, in dem sensitive Informationen gespeichert sind, auf die keine der anderen Partitionen Zugriff erlangen darf. Die Entwicklung und der Test des Treibers wurden erfolgreich durchgeführt. Die Testmethoden und Ergebnisse sind in E4.2.3.1 genauer dargestellt und umfassen neben den von AISEC durchgeführten Tests ebenso die Testmethoden und -ergebnisse der anderen Partner in TP4.2.

Arbeiten in Bezug auf architekturunabhängige seitenkanalresistente Kryptografie durch Nutzung der Multicore-Architektur wurden ebenfalls durchgeführt. Hier wurde am Fraunhofer AISEC eine neuartige Implementierung von AES entwickelt, welche die Operationen innerhalb einer Runde über mehrere Cores randomisiert parallel rechnet. Dies soll eine Resistenz gegen softwarebasierte Timing-Angriffe, wie sie im Rahmen des BMBF-Projekts RESIST am Fraunhofer AISEC durchgeführt wurden, liefern. Prototypische Implementierungen haben allerdings gezeigt, dass der Synchronisationsaufwand für nur einen Block zu hoch ist. Deshalb wurde hier in Betracht gezogen, mehrere Blöcke gleichzeitig auf diese Weise zu verarbeiten, um die Performanz zu erhöhen

Das Konzept der seitenkanalresistenten Kryptografie wurde um einen speziellen CTR-Modus erweitert. Eine prototypische

Umsetzung dieses Multicore-CTR-Modus wurde in OpenSSL getestet. Bereits nach der ersten Optimierungsphase wurden Durchsatzraten von rund 1MB/s erreicht. Im Vergleich zu nur einem Block aus den ersten Tests ist das ein enormer Performancegewinn, allerdings im Vergleich zu der in OpenSSL integrierten ungehärteten Single-Core-Lösung immer noch um ca. den Faktor 200 langsamer. Hier wurde der Bedarf nach weiteren Forschungsarbeiten, nicht nur an alternativen Synchronisations-mechanismen, sondern auch an optimierter Datenverwaltung deutlich.

Außerdem war das Fraunhofer AISEC an Teilprojekt TP6 zur Erstellung eines Demonstrators (Freescale-basierten Automotive Plattform) beteiligt. Es erfolgte ein Laboraufbau mit HSM (FPGA) und i.MX6 SabreLite und die Ergebnisse von TP3 und TP4 konnten erfolgreich in den Demonstrator integriert werden. Dieser wurde auf dem ARAMiS Abschluss Workshop am 29.04.2015 der Öffentlichkeit präsentiert.

Insgesamt hat so das Fraunhofer AISEC im Laufe des Projekts ARAMiS domänenübergreifende Sicherheitslösungen im Bereich der Embedded Systems erforscht und umgesetzt. Das virtualisierte Hardware-Security-Modul, das sicheren und angriffsresistenten Speicherplatz für sicherheitskritische Daten bietet, die im Bereich Embedded Systems verwendet werden (z.B. Credentials, kryptografisches Material oder Payment-Daten), ist von seiner Architektur her in allen betrachteten Domänen – also Automotive, Avionik und Bahn, sowie weiteren wie Industrie 4.0 – einsetzbar.

Die Weiterentwicklung von Virtualisierungstechnologie im Bereich eingebetteter Systeme mit Hinblick auf Informationssicherheit, vor allem auch in Verbindung mit HSMs als Vertrauensanker, stellt eine Grundlage dar, um umfassend vernetzte Cyber-Physical Systems zu realisieren. Kryptografische Daten und Zertifikate, die zum Beispiel für die Absicherung der Kommunikation und die Authentifizierung der Systeme benötigt werden, müssen in einer Architekturlandschaft mit einer Vielzahl vernetzter, verteilter Systeme angriffssicher gespeichert werden können. Ferner muss es möglich sein, von verschiedenen virtuellen Laufzeitumgebungen sicher isoliert darauf zugreifen zu können. Diese Voraussetzung ist ebenfalls unabdingbar im Bereich von Industrie 4.0, wo die in ARAMiS entwickelten Lösungen ebenfalls Grundlage für die Entwicklung von Sicherheitstechnologien sein können.

Vor diesem Hintergrund wurde nicht nur neue Hardware in Form eines virtualisierbaren HSMs entwickelt, sondern es wurde auch bestehende Software zur Virtualisierung von Systemen, die durch die hohen Ressourcen von Multicore CPUs ermöglicht wird, an die neuen Anforderungen angepasst. Das Zusammenspiel aus Hard- und Software bildet die Basis der universell einsetzbaren

Sicherheitsarchitektur, die eine deutliche Erhöhung der Sicherheit bietet.

Alle entwickelten Komponenten wurden eingehend validiert und in Demonstratoren verwendet.

6.1.2 Fraunhofer IESE

Das Fraunhofer IESE leitete gemeinsam mit SYSGO das vierte ARAMiS-Teilprojekt TP4 „Software“. Analog dazu waren die meisten Aufwände des Fraunhofer IESE ebenfalls TP 4 zugeordnet. Die restlichen Aufwände des Fraunhofer IESE befanden sich in den Teilprojekten TP0 (AP0.3) und TP2; einige wenige waren in TP3 und TP5.

Im Kontext von TP4 lag der Hauptfokus des Fraunhofer IESE auf der Bearbeitung des Themenbereichs Software Safety (TP 4.3). Die Vorgehensweise war dabei in drei Schritte unterteilt und orientierte sich am Entwicklungsprozess eines sicherheitskritischen Systems. Der erste Schritt bestand in der Identifikation multicore-spezifischer Herausforderungen bei der Zertifizierung von sicherheitskritischen Systemen. Solche Herausforderungen manifestieren sich in der Regel an schwierig zu behebenden Fehlerbildern oder an der schwierig nachzuweisenden Validität der implementierten Fehlermaßnahmen. Als multicore-spezifisch gelten die Herausforderungen hierbei nur, wenn diese erst durch den Einsatz von Multicore-Technologie entstehen oder dadurch stark an Bedeutung gewinnen.

Ausgehend von gemeinsam genutzten Ressourcen des Mikroprozessors (z.B. Cache) und des Mikrocontrollers (z.B. Speicherbus, Speichermodule, E/A Geräte) wurden Szenarien analysiert, wie die gegebene Architektur das Verhalten der Applikationssoftware beeinflussen kann. Ein solches Szenario wird durch drei aufeinanderfolgende Schritte beschrieben. Der erste Schritt beschreibt den so genannten Auslöser. Der Auslöser ist eine Aktion einer Softwarekomponente oder ein externes Ereignis, welche(s) das System in einen unvorteilhaften Zustand bringt. Die Beschreibung dieses Zustands ist der zweite Teil der Beschreibung des Szenarios. Der letzte Teil des Szenarios beschreibt schlussendlich den negativen Effekt auf die betroffene Softwarekomponente und die Abweichung des Ist-Verhaltens der Komponente vom erwarteten Soll-Verhalten.

Die Unterteilung des Szenarios in Auslöser, (Fehler)Zustand und (Fehler)Auswirkung hat einen Hauptvorteil. Alle drei Schritte bieten unterschiedliche Möglichkeiten zur Vermeidung oder zur Erkennung des Szenarios. Kann man durch Designmaßnahmen den Auslöser verhindern, kann das gesamte Szenario verhindert werden. Eine Erkennung des fehlerhaften Zustands erlaubt es

immerhin Maßnahmen einzuleiten, bevor die betroffene Softwarekomponente eine Abweichung vom Sollverhalten zeigt. Ist weder die Vermeidung des Auslösers noch die Behandlung des Zustands möglich, bleibt die Erkennung und Behandlung der Fehlerauswirkung als letzter Lösungsweg. Durch die Unterteilung des Szenarios in unterschiedliche Teilschritte kann die Auswirkung geplanter Maßnahmen besser beschrieben und präziser bewertet werden.

Auf Basis einer exemplarischen Interferenzanalyse anhand der ARAMiS Multicore-Referenzarchitektur, die in TP2 erarbeitet wurde, wurde eine Analysemethode entwickelt, welche in E4.3.1.1 beschrieben und auf dem 25. IEEE International Symposium on Software Reliability Engineering veröffentlicht wurde.

Der zweite Schritt in AP 4.3 beinhaltete die Identifikation und Entwicklung von Sicherheitsmaßnahmen und Sicherheitskonzepten zur Bekämpfung der im ersten Schritt identifizierten relevanten Fehlerbilder oder Fehlerursachen. Dieser Schritt beinhaltete ebenfalls die Bewertung der Ansätze aus den parallel laufenden APs 4.1 und 4.4 im Hinblick auf deren Wirksamkeit zur Bewältigung der sicherheitsrelevanten Herausforderung.

Der dritte Schritt behandelte dann abschließend die Erstellung eines Sicherheitsnachweises. Die dabei entstehenden Evidenzen sollen belegen, dass die identifizierten und entwickelten Maßnahmen und Konzepte ausreichend sind, um die Fehlerbilder eines Multicore-Systems adäquat zu adressieren. Diese Nachweise sollen die Abnahme und Zertifizierung von Multicore-Systemen ermöglichen bzw. erleichtern.

Die Ergebnisse der Analysemethode in Bezug auf die ARAMiS-Referenzarchitektur sowie Maßnahmen und Strategien zum Schutz gegen die erkannten Interferenzen wurden in einer webbasierten Segregation Knowledge Base für die Projektöffentlichkeit zugänglich gemacht.

Zusätzlich wurde auch im Kontext der Segregation Knowledge Base aufgezeigt, wie anhand von Sicherheitsmechanismen und adressierten Interferenzen ein Protection Case erstellt werden kann, der ähnlich dem Safety Case eine Argumentation darstellt, um eine Zertifizierung/ Qualifizierung zu erreichen und somit den Nachweis von Sicherheit ermöglicht.

Da die funktionale Sicherheit eine systemweite Eigenschaft ist, die nicht ausschließlich auf Softwareebene betrachtet werden kann, war das Fraunhofer IESE ebenfalls in den entsprechenden Arbeitspaketen auf System- (AP 2.3) und Hardwareebene (AP 3.3) involviert.

Im Bereich Software war das Fraunhofer IESE ebenfalls im Bereich Architekturen aktiv (AP 4.1). Hier wurde ein Framework entwickelt, welches die Simulation und Bewertung von

sicherheitsrelevanten Systemfunktionen auf Architekturebene beschreibt und so dazu beiträgt, geeignete Konzepte, zum Beispiel zur Segregierung von Funktionen, auszuwählen und zu bewerten.

Dies ermöglicht es, Fehler, die durch reguläres Testen nur selten gefunden werden können, sicher zu identifizieren. Hierzu zählen zum Beispiel potenzielle Deadlocks, Fehler bei der Nutzung von Semaphoren und Spinlocks sowie Fehler, die zu Dateninkonsistenzen führen. Normalerweise treten diese Fehler nur unter speziellen Bedingungen auf. Ein Beispiel hierfür ist ein spezielles Timing, das nur in bestimmten Systemkonfigurationen auftritt. Diese Fehler können daher nicht mit herkömmlichen Testmethoden gefunden werden.

Dazu wurde unser Simulationsframework FERAL (Framework for Simulator Coupling on Requirements and Architecture Level) um die Fähigkeit zur Simulation abstrakter Konzepte erweitert. Diese Erweiterung ermöglicht es bspw. Java-Datenstrukturen zu nutzen, um Cache-Speicher oder Schedulingtabellen in der Simulation effizient zu realisieren. Hierbei war es wichtig, ein Ausführmodell zu erstellen, das auch parallele Operationen mit nicht atomarer Zeitdauer ermöglicht. Dies ist zum Beispiel bei der Simulation von Zugriffen auf eine Crossbar, aber auch bei der Simulation von Betriebssystemfunktionen erforderlich.

Mit diesem Framework ist es möglich, eine Architektur auf funktionaler Ebene zu realisieren, die sowohl das Verhalten von Hardware als auch das Verhalten von Softwarekomponenten abbildet. Dies ist wichtig, da Softwarefunktionen häufig Eigenschaften der Hardware, wie zum Beispiel fehlende Cache-Kohärenz, ausgleichen müssen.

Um eine überwachte Ausführung des C-Codes zu ermöglichen, wurde im Rahmen des Projekts die LLVM in den Simulator integriert. Ursprünglich wurde die LLVM als Compiler-Infrastruktur entwickelt; sie ermöglicht jedoch auch das Kompilieren von C und C++-Code in Bytecode und dessen Ausführung durch eine virtuelle Maschine. Wir nutzen diese Technologie, um Variablenzugriffe und Funktionsaufrufe aufzuzeichnen und nachzuverfolgen.

Hierfür wurde die virtuelle Maschine der LLVM, der LLI, entsprechend erweitert. Durch diese Erweiterung werden Hinweise auf Synchronisationsfehler aufgezeichnet; hierzu gehört neben Variablenzugriffen auch die Nutzung von Synchronisierungs-mechanismen.

Im Bereich der empirischen Unterstützung (AP0.3) bestand die Aufgabe des Fraunhofer IESE darin, die empirische Bewertung der Projektergebnisse in verschiedenen Arten von Studien über den kompletten Entwicklungszyklus zu unterstützen. Insbesondere

wurden die in ARAMiS durchgeführten Studien zentral geplant, koordiniert und am Ende gesammelt und zusammengefasst. Außerdem wurden die Projektpartner bei der Planung, Durchführung und Analyse von empirischen Studien vom Fraunhofer IESE unterstützt.

Zur initialen Schärfung des Themas Multicore wurden in systematischen Literatur Reviews (SLRs) Metriken, Qualitätsmodelle, Methoden und Techniken gesucht, die zur Messung und Erreichung bestimmter Qualitätsattribute beitragen. Zusätzlich erarbeitete und verbreitete das Fraunhofer IESE ein Glossar zum Thema „Empirische Methoden“. Das Glossar ist ein Teil von D0.2.

Es wurden über die komplette Projektlaufzeit viele verschiedene empirische Studien betreut, wobei im Folgenden nur einige prominente Beispiele etwas detaillierter beleuchtet werden:

Eine Umfrage zur „Priorisierung von Design Space Exploration Goals“ in Zusammenarbeit mit der fortiss GmbH wurde vom Fraunhofer IESE entworfen und durchgeführt. Der Fragebogen wurde in einer Fokusgruppe am 29. Juli 2013 in München evaluiert. Die Ergebnisse wurden analysiert und dokumentiert und als Paper veröffentlicht. In Zusammenarbeit mit TUM(SSE) und Audi wurden die Ergebnisse der Studie zu den TP1-Ergebnissen veröffentlicht, vorgestellt und diskutiert.

Für TP6 wurde übergreifend über alle Demonstratoren vom Fraunhofer IESE ein Konzept zur Validierung entwickelt, abgestimmt und eingesetzt. Dieses Konzept wurde in Enterprise Architect in Zusammenarbeit mit der TUM(SSE) umgesetzt und in dem Zusammenfassungs-Deliverable wurden alle Studien erläutert. Mithilfe dieser Umsetzung war es möglich, die Demonstratoren und deren Evaluationsergebnisse direkt mit anderen ARAMiS-EA-Ergebnissen zu verknüpfen.

In TP5 wurde, wie in der Vorhabensbeschreibung vorgesehen, eine durchgängige und domänenübergreifende Methodenplattform für den Entwurf von multicore-basierten Systemen bereitgestellt. Dazu wurden zuerst Anforderungen erhoben und anschließend wurden diese in entsprechende Konzepte, wie zum Beispiel Templates und eine integrierte Softwarelösung, umgesetzt. Das Fraunhofer IESE arbeitet an der Integration von systematisch erhobenem Methodenwissen, welches teilweise im Projekt in einer Methodenplattform entwickelt wurde.

Eine finale Version eines Templates zur Methodenbeschreibung wurde durch das IESE erstellt, nach einer Pilotierung innerhalb von TP5. Es wurde eine Strategie zur Verteilung des Templates erarbeitet und von TP5 aus an die anderen TPs zum Ausfüllen verteilt. Des Weiteren unterstützte das IESE mehrere Partner bei der Benutzung des Templates.

In Zusammenarbeit mit der TU Kaiserslautern wurde ein Prototyp eines Methoden-Repositorys entwickelt und mit einigen Partnern und Ergebnissen des Projektes befüllt, um die Durchgängigkeit zu zeigen.

6.2 Notwendigkeit und Angemessenheit der Arbeiten

Die Notwendigkeit der verschieden von den beiden Fraunhofer Instituten gemachten Arbeiten ergibt sich aus der Zielsetzung des Gesamtprojekts. Die beschriebenen Arbeiten zu Safety und Security lagen im Kernbereich des Projekts, da beides entscheidend für sicherheitskritische Systeme ist. Die sich aus den Herausforderungen und Lösungskonzepten ergebenden neuen Methoden müssen entsprechend in einen ganzheitlichen und domänenübergreifenden Ansatz eingegliedert werden. Empirie ist notwendig um den Nutzen der in ARAMiS entwickelten Methoden, Szenarien, etc. zu zeigen. Der Umfang ist angemessen, da zusammen mit der TU Kaiserslautern alle Evaluationen in dem Projekt (und damit aller einzelnen Partner) unterstützt wurden.

6.3 Fortschritte auf dem Gebiet des Vorhabens

Es sind während der Projektlaufzeit keine Ergebnisse von dritter Seite bekannt geworden, die an dieser Stelle hervorzuheben wären.

6.4 Veröffentlichung der Ergebnisse

[1] Daniel Adam, Carsten Rolfes, Sergey Tveryshev, Timo Sandmann: Two Architecture Approaches for MILS Systems in Mobility Domains (Automotive, Railway and Avionics); in: Proceedings of the 10th HiPEAC conference MILS workshop

[2] O. Khalid, C. Rolfes, A. Ibing: On Implementing Trusted Boot for Embedded Systems, IEEE Int. Symposium on Hardware-Oriented Security and Trust, 2013

[3] Phillip Diebold, Constanza Lampasona, Davide Taibi: “Moonlighting SCRUM: an agile method for distributed teams with part-time developers working during non-overlapping hours”, in Proceedings of The Eighth International Conference on Software Engineering Advances (ICSEA 2013), October 2013

[4] Jasmin Jahic, Thiyagarajan Purusothaman, Markus Damm, Thomas Kuhn, Peter Liggesmeyer, Christoph Grimm: “Automatic Test Coverage Measurements to support Design Space Exploration”, First International Workshop on Design Space Exploration of Cyber-Physical Systems (IDEAL) 2014, Springer

[5] Phillip Diebold, Laurent Dieudonné, Davide Taibi: “Process Configuration Framework Tool”, in 39th Euromicro Conference on Software Engineering and Advanced Applications, 2014.

[6] Davide Taibi, Valentina Lenarduzzi, Laurent Dieudonné, Christiane Plociennik: “Towards a Classification Schema for Development Technologies: an Empirical Study in the Avionic Domain”, in International Journal on Advances in Software, vol 8, No. 1 & 2, 2015

[7] Zimmer, B.; Dropmann, C.; Hanger, J.U., "A Systematic Approach for Software Interference Analysis," in *Software Reliability Engineering (ISSRE), 2014 IEEE 25th International Symposium on* , vol., no., pp.78-87, 3-6 Nov. 2014

[8] Jahic, J.; Kuhn, T., "Analysis of Functional Software Dependencies through Supervised Execution," in *Software Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on* , vol., no., pp.128-129, 3-6 Nov. 2014

7 Intel Deutschland GmbH

7.1 Wissenschaftlich-technische Ergebnisse

Das ARAMiS Projekt ermöglichte es Intel Ergebnisse mit Hilfe mehrerer technologischer Errungenschaften zu erzielen. Die folgende Liste beschreibt die Technologien im Detail:

VCT-Demonstrator

Der in TP6 entwickelte Virtualized Car Telematics (VCT)-Demonstrator zeigt die Realisierung einer gemeinsamen, leistungsfähigen Plattform für offene, nutzerorientierte, vernetzte Systeme und auf Sicherheit fokussierte, kontextsensitive Fahrzeugsysteme unter Nutzung von Virtualisierungstechnologien auf Multicore-Plattformen.

PCIe, SR-IOV, VT-d

Die Arbeiten an virtualisierten Schnittstellen für Peripherie-komponenten und rekonfigurierbaren Coprozessoren demonstrieren die effektive Einsetzbarkeit von Virtualisierungs-technologien aus dem Server- und Desktop-Bereich für I/O- und Coprozessor-Virtualisierung in eingebetteten Systemen; dabei muss insbesondere auf die Interoperabilität der einzelnen Lösungen sowie auf den Grad der Implementierung von Standards geachtet werden – heutige Standards beinhalten bereits für die Zielanwendung geeignete Fähigkeiten, die aber nicht in allen Implementierungen umgesetzt sind (z.B. PCIe QoS).

Virtualisiertes Coprozessor-Interface

Mehrere Ergebnisse innerhalb von ARAMiS zeigen deutlich die Notwendigkeit von hardwareunterstützter Virtualisierung, um Anforderungen an die Performance erfüllen zu können. Darüber hinaus ist die zusätzliche hardwarebasierte Steuerung der Nutzung von geteilten Ressourcen (z.B. DMA) notwendig, um Fairness und temporale Segregierung beim Zugriff mehrerer virtueller Maschinen gewährleisten zu können. Diese Anforderungen wurden in der Entwicklung eines virtualisierten Coprozessor-Interfaces umgesetzt, welches die Basis für mehrere der im Projekt umgesetzten technischen Arbeiten und Demonstratoren bildet.

Streaming-Case Study Video-Tracking

Die Virtualisierung von Coprozessoren hat (je nach Anwendung) höhere Anforderungen an Bandbreite und Arbitrierung, teilweise bedingt durch einen üblicherweise umfangreicheren Kontext pro VM. Spezialisierte Entwurfsverfahren (beispielsweise der Einsatz von Streaming-Modellen) können diese Anforderungen reduzieren.

Speziell der Einsatz von Streaming ist für die im Projekt betrachtete Anbindung (PCIe) von Peripherie und Coprozessoren vorteilhaft, da hier ggf. erhöhte Latenzen nicht signifikant ins Gewicht fallen.

GPU-Virtualisierung im VCT-Demonstrator

Im Verlauf der Arbeiten an der Virtualisierung der grafischen Ausgabekomponenten des VCT-Demonstrators hat sich gezeigt, dass eine Verlagerung des Ressource-Sharings auf höhere Abstraktionsebenen für komplexe Peripheriekomponenten (z.B. GPUs) unter bestimmten Voraussetzungen vorteilhaft sein kann, um die Komplexität der Implementierung zu reduzieren.

Zusammengefasst erfordern zukünftige Mobilitätsszenarien eine erhöhte Vernetzung der Teilnehmer und (teilweise auch dadurch bedingte) erhöhte Rechenleistung der eingesetzten Plattformen bei gleichzeitiger Beibehaltung der Anforderungen an funktionale Sicherheit. Dies erfordert einen Fokus zukünftiger Plattformen auf eine Kombination aus Multicore-Prozessoren, dedizierten Beschleunigern, Virtualisierung, und auf Vorhersagbarkeit optimierte Verbindungstechnologien.

Intel nahm die Führungsrolle bei der Definition von Schlüssel-szenarien und Anwendungsfällen für die Bereiche Transportation, In-Vehicle Infotainment und Virtualized Car Telematics ein. Gleichzeitig wurde zu der Definition der Gesamtarchitektur und Validierungsansätze beigetragen.

Der Forschungsschwerpunkt lag auf Interfaces die Virtualisierung von Coprozessoren und Peripherie ermöglichen z.B. rekonfigurierbaren Beschleuniger und CAN Controllern. Forschungsergebnisse beinhalten ein wiederverwendbares PCIe Interface, das in der Lage ist mehrere VMs über Single Root I/O Virtualization (SR-IOV) zu unterstützen, sowie die Übertragung und Evaluierung von Virtualisierungstechniken, aus dem Desktop und Datacenter Umfeld, in eingebettete und mixed-criticality Systeme.

Intel übernahm zudem eine der Führungsrollen bei der Entwicklung des VCT-Demonstrators. Hierbei wurde die Multicore Rechenplattform und eine transportable Stand-Alone Version des Demonstrator-Setups bereitgestellt.

Die fortschreitende Verbreitung von Multi-Core Plattformen in eingebetteten Systemen erfordert die Identifikation von notwendigen Forschungsschwerpunkten und Ansätzen, um ihren Einsatz insbesondere in Bereichen zu ermöglichen, die erhöhte Anforderungen an die Sicherheit und Zuverlässigkeit stellen.

Obwohl die Vorteile und Herausforderungen von Multi-Core nicht exklusiv auf einen bestimmten Kreis von Anwendungen einzuschränken sind, setzt ARAMiS nicht umsonst einen

besonderen Fokus auf den Einsatz dieser Systeme im Bereich der Mobilität. Im Rahmen des Projekts konzentrierte sich Intel hier genauer auf In-Car-Infotainment und Kommunikations-architekturen im Fahrzeug, auch mit Blick auf weiterführende Themen wie Fahrassistenzsysteme bis hin zum Autonomen Fahren.

In diesem Umfeld bietet das Konzept der Virtualisierung eine Schlüsseltechnologie zur Beherrschbarkeit von Multi-Core in sicherheitskritischen Anwendungen, um die verschiedenen Anforderungen von heterogenen und mixed-criticality-Systemen abdecken zu können. Virtualisierung an sich ist keine völlig neue Technologie; interessant ist vielmehr die Fragestellung, in wie weit und mit welchen Modifikationen sich existierende Virtualisierungstechnologien, wie sie in Servern und Desktop-Systemen eingesetzt werden, auf eingebettete und sicherheitskritische Systeme übertragen lassen. Hierfür liefert ARAMiS sowohl Ansätze zur Validierung als auch bereits eine stabile Basis für zukünftige Entwicklungen, welche wichtige Impulse für sicherheitskritische Plattformen geben können.

In der Tat hat Intel seit Kurzem Multi-Core-Plattformen für In-Car-Infotainment und andere automobile Anwendungen im Portfolio, welche sich bereits erfolgreich bei Partnern und Kunden im Einsatz befinden.

7.2 Notwendigkeit und Angemessenheit der Arbeiten

Basierend auf während der Projektlaufzeit identifizierten Einsparungen bei der Erstellung des VCT-Demonstrators sowie aufgrund von Änderungen in der Aufwandsplanung, die sich durch interne Restrukturierung ergaben und in den entsprechenden Aufwandsänderungen kommuniziert wurden, ergab sich insgesamt eine Reduktion des Kostenaufwands.

Die erzielten Einsparungen hatten keinen messbaren Einfluss auf die Umsetzung des Projektplans. Insgesamt wurden die im Projektplan und der Vorhabensbeschreibung avisierten Ziele unter Berücksichtigung der kostenneutralen Verlängerung des Projektes erfüllt.

Die Entwicklung von Multicore-Systemen im Automotive Bereich wurde hauptsächlich motiviert durch die Nachfrage nach zusätzlicher Rechenleistung, Reduktion der Gesamtanzahl benötigter ECUs und Systemkomplexität, mit dem Ziel künftige Anwendungen wie Fahrerassistenzsysteme, hoch automatisierte und autonome Fahrersysteme zu ermöglichen. Dies führte zu Fortschritten in Technologien wie Virtualisierung und Beschleu-nigung von Anwendungen durch Coprozessoren.

Die Forschung war gleichermaßen motiviert durch das Nichtvorhandensein von Off-The-Shelf Lösungen. Dies war der Grund für Intel die Gelegenheit zu nutzen um seine Expertise innerhalb des Automotive Bereichs zu erweitern.

Die Ergebnisse des Projekts sind nur für das Konsortium und die breite Forschung von Bedeutung, sondern auch von großem Nutzen für Intel. Sie wurden bereits an die wichtigen Geschäftsfelder übergeben und halfen bei der Definition von Anforderungen für künftige Arbeiten. Andere Projekte von Intel nutzen die neugewonnene Expertise und der Demonstrator wurde öffentlich auf IT Messen vorgestellt.

Während der Beteiligung am Projekt wurde insbesondere auf Kosteneffizienz wert gelegt. In enger Absprache mit dem ARAMiS Koordinator war Intel in der Lage die Anzahl an Arbeitsstunden und Ausgaben für Hardware sowie Software zu optimieren. So wurde z.B. der Fahrzeugdemonstrator als Labor-Setup anstatt als Fahrzeug repliziert. Dabei wurde Open-Source Software und interne Lizenzabkommen anstatt von teuren Alternativen genutzt. Dies erlaubte eine Verlängerung des Projektes um über 3 Monate

7.3 Fortschritte auf dem Gebiet des Vorhabens

Es sind keine Ergebnisse von dritter Seite bekannt geworden, die an dieser Stelle hervorzuheben wären.

7.4 Veröffentlichung der Ergebnisse

Das Projekt führte zu mehreren Publikationen und Besuchen bei IT Messen. Unter der Vielzahl an Publikationen war Intel an zweien als Co-Autor beteiligt:

[1] „A Flexible Interface Architecture for Reconfigurable Coprocessors in Embedded Multicore Systems using PCIe Single-Root I/O Virtualization“, written by Oliver Sander, Steffen Bähr, Enno Lübbers, Timo Sandmann, Duy Viet Vu and Jürgen Becker.

[2] „Hardware virtualization support for shared resources in mixed-criticality multicore systems“, written by Oliver Sander, Timo Sandmann, Duy Viet Vu, Steffen Baehr, Falco Bapp, Juergen Becker, Hans-Ulrich Michel, Dirk Kaule, Daniel Adam, Enno Luebbers, Juergen Hairbucher, Andre Richter, Christian Herber and Andreas Herkersdorf.

Gleichzeitig nahm Intel an der weltweit meistbesuchten IT Messe, der CEBIT 2015 teil. Hier zog der Demonstrator ein großes Interesse auf sich, welches von der Presse über Podcasts zu Interviews reichte.

8 Karlsruher Institut für Technologie

8.1 Wissenschaftlich-technische Ergebnisse

Eine detaillierte Beschreibung der im Folgenden auszugsweise vorgestellten wissenschaftlich-technischen Ergebnisse ist im gemeinsamen Abschlussbericht (Final Report vom 21.10.2015) des Projekts zu finden.

8.1.1 TP 1 Anforderungen und Szenarien

Für die in TP 1 definierten ARAMiS-weiten Anforderungen und Szenarien wurden die Abstraktionsebenen und Sichtweisen aus SPES2020 analysiert und für die Verwendung in ARAMiS entsprechend adaptiert.

Neben der Herausforderung die Lücke zwischen den übergeordneten Szenarien inkl. ihren abstrakten Anforderungen und den technischen Anforderungen auf Rechnerarchitekturebene zu schließen, war das KIT außerdem an der Ausgestaltung des domänenübergreifenden CPS Szenarios beteiligt, das eine Verbindung der domänenspezifischen Szenarien miteinander erreichen konnte.

Für die eigenen Schwerpunkthemen aus dem Automotive-Umfeld wurden spezifische Anforderungen für die Anbindung von Coprozessoren abgeleitet.

8.1.2 TP 2 Systementwurf

Im Teilprojekt TP 2 war das KIT hauptsächlich an den Themenbereichen Systemarchitektur und Modellierung, Security-Konzept und Monitoring-Konzept (Safety) beteiligt. Auf Systemebene wurden die Aspekte Dekomposition auf funktionaler Ebene und Allokation von logischen Komponenten behandelt. Im Weiteren wurde die Modellierung von typischen Multicore-Architekturen für die verschiedenen Domänen aus technischer Perspektive durchgeführt, die eine ganzheitliche Betrachtung auf Systemebene ermöglicht. Dies erfolgte inklusive der Extraktion von erforderlichen Details, die insbesondere für Multicore-Architekturen und deren Einsatz in sicherheitskritischen, eingebetteten Systemen relevant sind. Neben den klassischen Software-Architekturen wurden außerdem virtualisierte Systeme betrachtet und entsprechende Modellierungsansätze hierzu erarbeitet. Die Modellierung der Hard- und Software-Architekturen und Schaffung einer einheitlichen Architekturbeschreibung bezogen auf die

Multicore-relevanten Architekturaspekte bildeten auf diese Weise die Grundlage für weitere Architekturanalysen sowie die Übergabe der im Teilprojekt TP 2 gewonnenen Erkenntnisse und definierten Anforderungen an die Teilprojekte TP 3 (Hardware) und TP 4 (Software).

Später erfolgte ein Rückfluss der auf Hard- und Software-Ebene gewonnenen Erkenntnisse für eine entsprechende Überarbeitung und Optimierung der Systemarchitekturen aus TP 2. Als wichtige Erkenntnis stellte sich heraus, dass auf Grund der unterschiedlichen Anforderungen sowie Hardware-, Software- und Systemarchitekturen in den einzelnen Domänen sowie Anwendungsfeldern der Entwurf einer einzigen, einheitlichen Multicore-Systemarchitektur nicht zielführend sein kann. Vielmehr erfordert die Komplexität aktueller und zukünftiger Steuergeräte in den Mobilitätsdomänen die Verwendung verschiedener, bedarfsgerecht angepasster Systemarchitekturen, auch auf Grund der Notwendigkeit der Einbindung von Legacy-Anwendungen und der Integration in bestehende Systeme. Insbesondere vor dem Hintergrund dieser im Projekt gewonnenen Erkenntnis stellt die übergreifende modellbasierte Aufbereitung der Multicore Herausforderungen einen wesentlichen Mehrwert dar. Dementsprechend sind diese Aspekte in die Konzepte im Bereich Monitoring und Anbindung von gemeinsam genutzten Coprozessoren eingeflossen, um durch einen generischen Ansatz die Verwendungsmöglichkeiten im Zusammenhang mit verschiedenen Systemarchitekturen je nach Anwendungsfall zu auszuweiten.

In den Teilbereichen Security (AP 2.2) und Safety (AP 2.3) auf Systemebene hat sich das KIT sowohl hinsichtlich der übergeordneten, systemweiten Aspekte, als auch mit individuellen Konzepten eingebracht. Hierbei entstand ein Securitykonzept zur Anbindung der gemeinsam genutzten Coprozessoren aus TP 3 zur Beschleunigung von kryptografischen Operationen an die in AP 2.2 entwickelte Architektur eines Hardware Security Moduls.

Im Bereich Safety wurden grundlegende Recherchen und Analysen von State-of-the-Art Lösungen redundanter Systeme bzgl. Safety und Zertifizierbarkeit nach gültigen Normen durchgeführt. Der Focus der Analysen lag dabei insbesondere auf den Unterschieden zwischen Single- und Multicore-Architekturen und den sich daraus ergebenden Konsequenzen für Implementierung und Zertifizierung von sicherheitskritischen und redundant auszulegenden Systemen. Die Ergebnisse sind in den Entwurf des Monitoring-Konzepts für Multicore-Architekturen eingeflossen, welches eine zweifache Überwachung eines Triple Modular Redundancy (TMR) Systems auf Basis einer Multicore-Architektur beschreibt. Die Ergebnisse der Untersuchungen bzgl. der Auswirkungen durch den Einsatz von Multicore-Architekturen in sicherheitskritischen Systemen haben klar die Notwendigkeit

eines zweistufigen Ansatzes für das Monitoring gezeigt. Diese ergibt sich aus der erforderlichen Überwachung und Zuteilung von gemeinsam genutzten Ressourcen wie Bussystemen und Peripherie-Komponenten sowie der Entdeckung von Fehlern aufgrund gemeinsamer Ursachen (Common Cause Failures). Im Kontext dieser Untersuchungen wurden außerdem Mechanismen zur Fehlerisolation identifiziert, die im Teilprojekt TP 3 prototypisch implementiert und evaluiert wurden. Die hierbei gewonnenen Erkenntnisse aus AP 3.3 wurden anschließend im Rahmen der Optimierung der Safety Eigenschaften auf Systemebene zurück gespiegelt und entsprechend dokumentiert.

8.1.3 TP 3 Hardware

Das KIT hat sich im Rahmen des Teilprojekts TP 3 mit heterogenen Hardware-Architekturen und dabei insbesondere mit der Anbindung von Coprozessoren und Hardware-Beschleunigern beschäftigt. Dies erfolgte sowohl für klassische, partitionierte als auch für virtualisierte Systeme, insbesondere vor dem Hintergrund von mixed-criticality Anwendungen. In den Arbeitspaketen AP 3.1 und AP 3.4 wurden entsprechende Anbindungskonzepte sowie verschiedene Mechanismen zur Ressourcenteilung entworfen. Unter Berücksichtigung der in TP 2 ermittelten Parameter wurde ein SystemC-Modell entwickelt, das eine Systemsimulation mit dem Hauptaugenmerk auf das Scheduling der Coprozessor-Nutzung ermöglicht. Damit wurde eine Exploration des Entwurfsraumes betrieben, die als Grundlage für weitere Arbeiten des KIT in diesem Bereich dienen konnte.

Für die Anbindung von Coprozessoren für AP 3.1 und AP 3.4 wurde eine auf der Intel Ivy Bridge Architektur basierende Rechner-Plattform aufgebaut und mit einer Xilinx FPGA-Karte erweitert. Diese Plattform diente während der Projektlaufzeit als Entwicklungs- und Evaluationsplattform. Für die Evaluation von virtualisierten Systemen wurde neben KVM und dem Windriver Hypervisor auf den quelloffenen Hypervisor NOVA zurück-gegriffen. Implementierung und Evaluierung konzentrierten sich hier zunächst auf die funktionalen Zugriffe auf virtualisierte Coprozessoren unter Umgehung der Virtualisierungsschicht sowie die Behandlung von Interrupts, die direkt in die jeweiligen virtuellen Maschinen weitergeleitet werden. Des Weiteren wurden Zugriffe zu Konfigurationszwecken speziell berücksichtigt, da sie im Vergleich zu funktionalen Zugriffen andere Anforderungen haben. Aus diesen frühen Evaluationen auf Basis des NOVA Hypervisors wurden Erkenntnisse bzgl. Latenz und Overhead durch Virtualisierung gewonnen, die für die Themen Interruptbehandlung und DMA im Zusammenhang mit Coprozessoren in Multicore-Systemen von entscheidender Bedeutung für die Weiterführung und Implementierung der Konzepte waren. So wurden die

erstellten Konzepte zur Anbindung von Coprozessoren an Multicore-Architekturen um Direct Memory Access (DMA) mit Bandbreitenmanagement erweitert (siehe Abbildung 5).

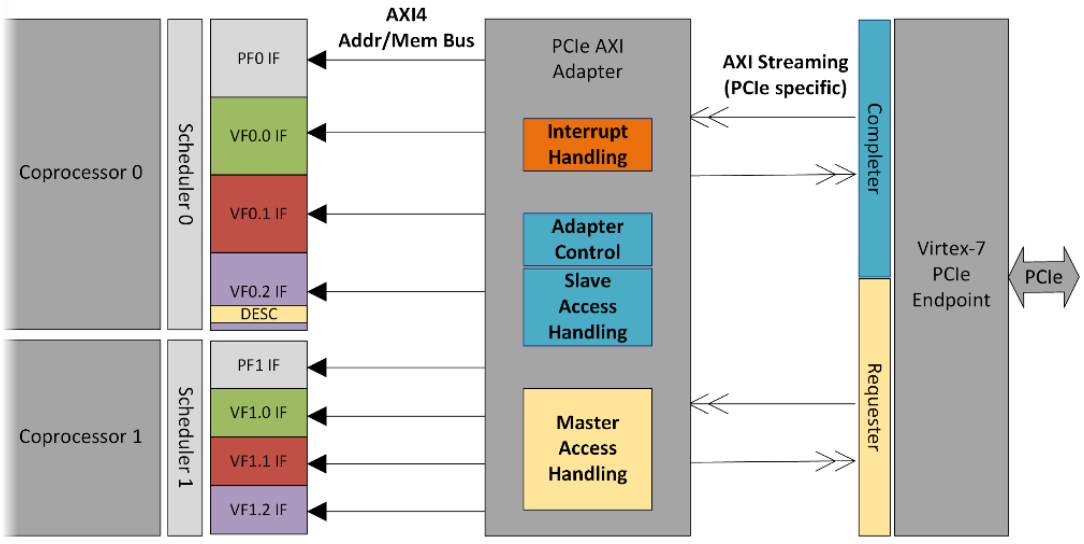

Abbildung 5 Systemarchitektur für die Anbindung von Coprozessoren in Multicore-Systemen

Dieses umfasst ein deterministisches Scheduling-Verfahren, welches die verfügbare Bandbreite für sämtliche Speicherzugriffe auf oder durch die Coprozessoren verteilt. Damit lassen sich verschiedenen Coprozessoren unterschiedliche Budgets an konsekutiven Daten-Paketen zuordnen, um eine Priorität der einzelnen Coprozessoren abzubilden, so dass für den einzelnen Coprozessor die Vorhersagbarkeit der Speicherzugriffslatenz erreicht wird. Der Fokus lag, basierend auf den Anforderungen aus TP 2, für das gesamte Konzept auf einer sowohl hoch performanten und generischen als auch deterministischen Umsetzung des Ressourcensharings in mixed-criticality Multicore Systemen. Die Implementierungen wurden anhand des Laboraufbaus (Intel Ivy Bridge Architektur mit FPGA-Karte) für die Evaluation systematisch getestet und insbesondere in Hinsicht auf Determinismus in Multicore Systemen und Performanz bewertet. Die Anforderungen aus TP 2 wurden dabei den jeweiligen Teilen der Implementierungen zugeordnet sowie ihre Erfüllung validiert und dokumentiert.

Das entwickelte Konzept basiert auf PCIe-SR-IOV-fähigen generischen Coprozessor-Interfaces und ermöglicht somit bereits durch das Design selbst eine nahtlose Integration in virtualisierte Systeme. Dies umfasst neben der Slave-Anbindung von Coprozessoren das Interrupt-Handling und hochperformante DMA-Transfers inkl. dem entwickelten Bandbreitenmanagement,

welches für den Einsatz in virtualisierten Systemen verfeinert wurde (siehe Abbildung 6).

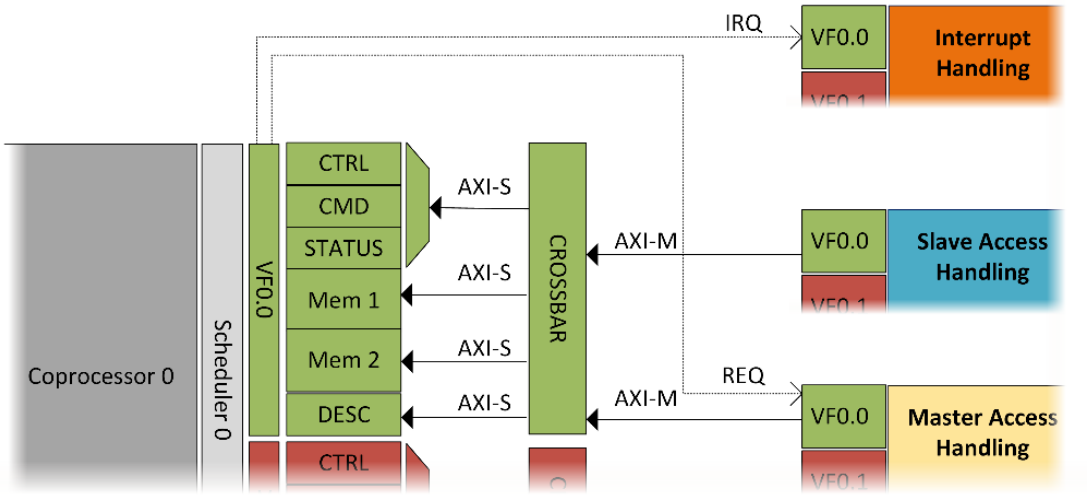

Abbildung 6 Schnittstellen-Architektur für Coprozessoren in virtualisierten Multicore-Systemen

Durch die direkte Zuweisung von Coprozessor-Schnittstellen an virtuelle Maschinen wird im Zusammenhang mit dem Segregierungskonzept der Schnittstellen und dem o.a. Bandbreitenmanagement ein mixed-criticality Multicore System ermöglicht. Die entwickelten Konzepte erlauben es, verschiedene Varianten des AXI Protokolls zu nutzen und somit die entwickelte Architektur an das Interface des entsprechenden Coprozessors anzupassen. Die Architektur des Coprozessor-Interfaces wurde zusammen mit dem Projektpartner Intel für einen Konferenzbeitrag auf der *IEEE ICFPT 2014* dokumentiert und dort präsentiert. Weiterhin wurden detaillierte Test-Cases zur Validierung der Performance-Daten des Coprozessor-Interfaces sowie zur Abdeckung von Requirements aus TP 2 definiert und dokumentiert.

Insgesamt konnte durch die Evaluierung der Anbindungskonzepte gezeigt werden, dass die erzielten Latenzen für verschiedene interne Bussysteme unabhängig von der Nutzung der Virtualisierung sind. Gleichzeitig wird ein Netto-Datendurchsatz von bis zu 2.6 GiB/s per DMA erzielt, nahe dem theoretischen Maximum von PCI Express. Bei der Nutzung des Bandbreitenmanagements zeigten die Messungen, dass die jeweiligen Coprozessoren die ihnen zugewiesene, garantierte Bandbreite auch erhielten. Es konnte eine vorhersagbare und deterministische Verteilung der Bandbreite auf die Coprozessoren nachgewiesen werden.

Für rekonfigurierbare Coprozessoren wurde ein Konzept zur Verwendung in mixed-criticality Multicore-Architekturen entwickelt,

das die Nutzung der partiellen Rekonfiguration ermöglicht. Dieses Konzept erlaubt dabei einerseits eine bedarfsabhängige, anwendungstransparente partielle Rekonfiguration für nicht kritische Coprozessoren, dessen Fokus auf der flexiblen und effizienten Ausnutzung der rekonfigurierbaren Ressourcen liegt. Andererseits ist es für sicherheitsrelevante Anwendungen nur möglich die partielle Rekonfiguration explizit anzufordern, so dass eine vorhersagbare maximale Dauer gegeben ist. Es ist darüber hinaus vorgesehen mittels standardisierter Schnittstellen und einfacher Test-Logik die Funktionsfähigkeit der Coprozessoren nach der partiellen Rekonfiguration zu überprüfen. Damit wird ein deterministisches Verhalten der sicherheitsrelevanten Anwendungen selbst im Falle einer partiellen Rekonfiguration sichergestellt. Die Interface-Erweiterungen der o.a. Architektur, die die Unterstützung für partielle Rekonfiguration anbieten, sind in Abbildung 7 zu sehen. Das Ergebnis hierzu wurde in einer Veröffentlichung zusammengefasst, die auf der *ReConFig 2014* Konferenz präsentiert wurde.

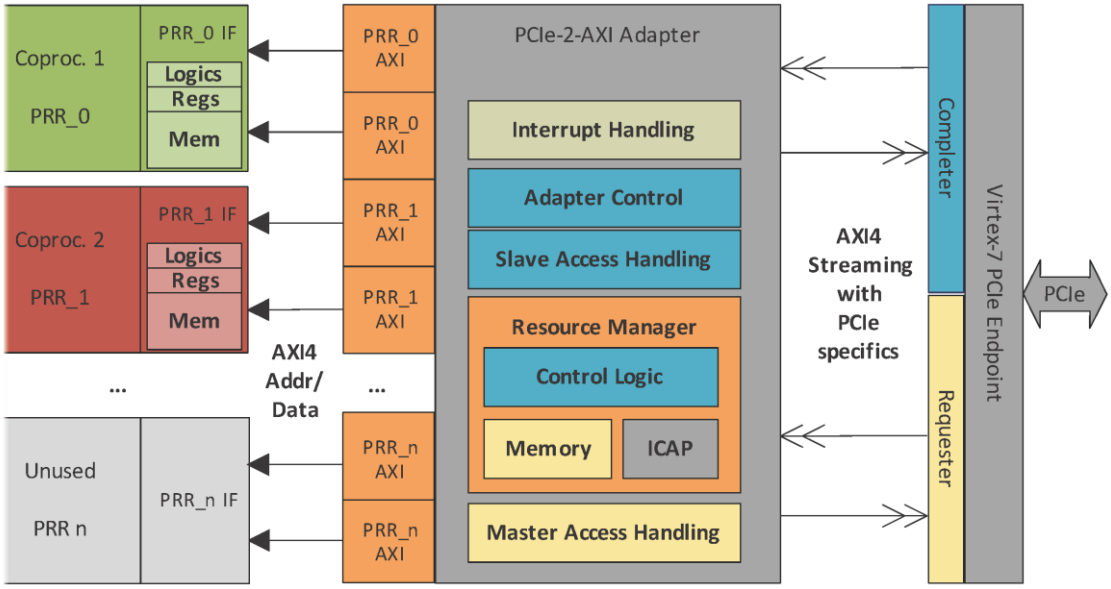

Abbildung 7 Erweiterte Interface-Architektur mit Unterstützung für partielle Rekonfiguration

Die Implementierung zur Anbindung von Coprozessoren an Multicore-Architekturen wurde außerdem für den Einsatz im VCT-Demonstrator von TP 6 erweitert und dort integriert. Zusätzlich wurde die heterogene Multicore-Plattform „Xilinx Zynq“ evaluiert bezüglich Performanz und Latenz für sicherheitskritische Anwendungen auf Basis von Multicore. Die gewonnenen Erkenntnisse für eine Hardware- / Software-Partitionierung einer heterogenen Multicore-Architektur wurden für den Einsatz im Automotive Umfeld zum Zweck der kryptografischen Absicherung von Car-2-X Szenarien prototypisch umgesetzt.

Im Rahmen des Arbeitspaketes AP 3.2 wurde ein Konzept für die Anbindung von rekonfigurierbaren Coprozessoren an ein Hardware Security Module (HSM) erarbeitet, basierend auf den in AP 2.2 erarbeiteten Systemarchitekturkonzepten für Security. Im allgemeinen Fall kann der Coprozessor als HSM-Erweiterung gesehen werden, wie in Abbildung 8 verdeutlicht.

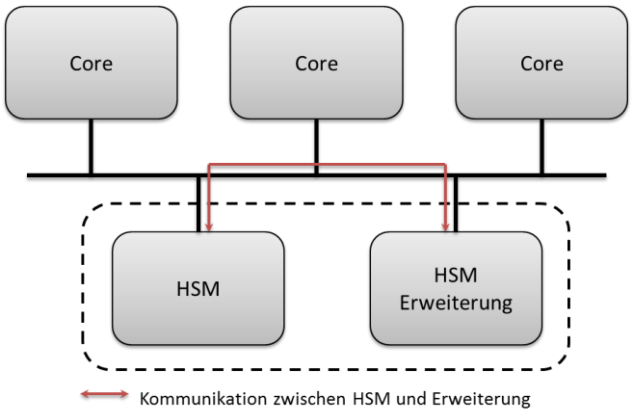

Abbildung 8 HSM-Erweiterung durch Coprozessoren

Damit lassen sich rechen- und kommunikationsintensive Aufgaben aus dem HSM auslagern und somit Größe und Komplexität des HSM reduzieren. Das wirkt sich positiv auf die Zuverlässigkeit und Vertrauenswürdigkeit des HSMs aus. Durch die Rekonfigurierbarkeit des Coprozessors lassen sich unterschiedliche kryptographische Algorithmen platzsparend implementieren.

Zu Veranschaulichung ist in Abbildung 9 der Aufbau eines Systems mit einem Car-2-X Crypto Coprozessor gezeigt. Der Coprozessor bekommt die für die Ver- und Entschlüsselung notwendigen Schlüssel, Zertifikate usw. aus dem HSM. Diese Kommunikation findet eher selten statt. Bei dem intensiven Datenaustausch zwischen Anwendungssoftware, dem Coprozessor und dem Car-2-X IO Modul wird das HSM nicht mehr einbezogen, sodass sowohl die Latenz der Kommunikation als auch der Rechendurchsatz durch das HSM nicht beeinflusst werden.

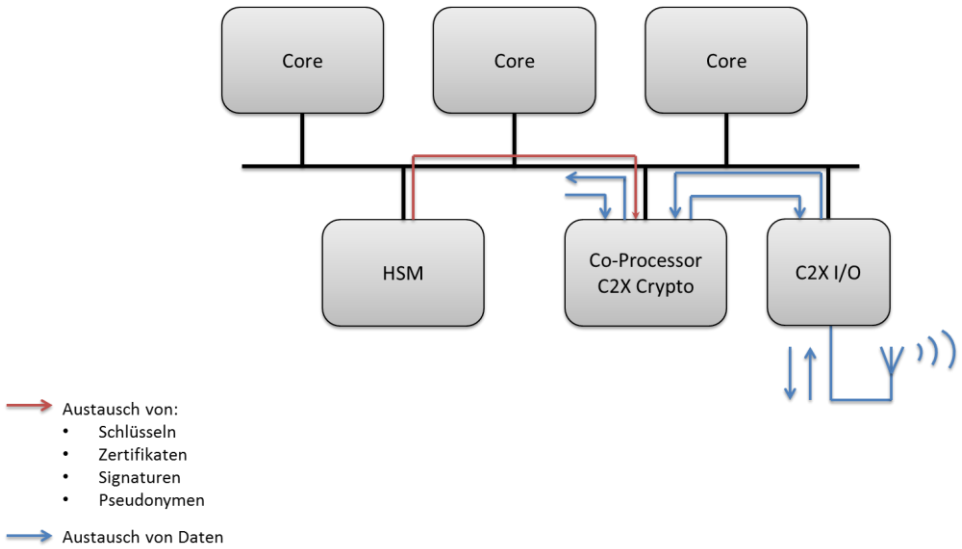

Abbildung 9 Kommunikation des HSMs mit einem Coprozessor am Fallbeispiel Car-2-X Crypto

Im Themenbereich Safety (AP 3.3) wurde seitens des KIT eine architekturspezifische Analyse aktueller Commercial Off-The-Shelf (COTS) Multicore-Architekturen bzgl. ihrer Eignung für sicherheitskritische und redundant abzusichernde Systeme durchgeführt. Hier lag der Focus der Analysen auf der Möglichkeit, das in AP 2.3 entworfene Konzept zum Monitoring technisch realisieren zu können. Die Ergebnisse flossen dabei in die Abbildung des Konzepts auf die Ebene einer technischen Architektur ein und zurück nach TP 2. Der resultierende Entwurf eines entsprechenden Systems ist in Abbildung 10 schematisch dargestellt. Dieser berücksichtigt architekturspezifischen Eigenschaften wie das verwendete Bussystem für die Anbindung gemeinsam genutzter Ressourcen und deren Verwaltung. Um Aussagen über die Fehlererkennungsmöglichkeiten treffen zu können, wurden verschiedene Arten von Fehlern für Tests implantiert (Timingfehler, Berechnungsfehler, etc.). Darüber hinaus werden Common Cause Failures des FPGAs von einem externen Microcontroller überwacht.

Die Konzepte der Kontroll- und Monitoring Architekturen wurden im weiteren Verlauf des Projekts um die Implementierung für ein hierarchisches Timing-Überwachungskonzept erweitert, wie Abbildung 11 entnommen werden kann.

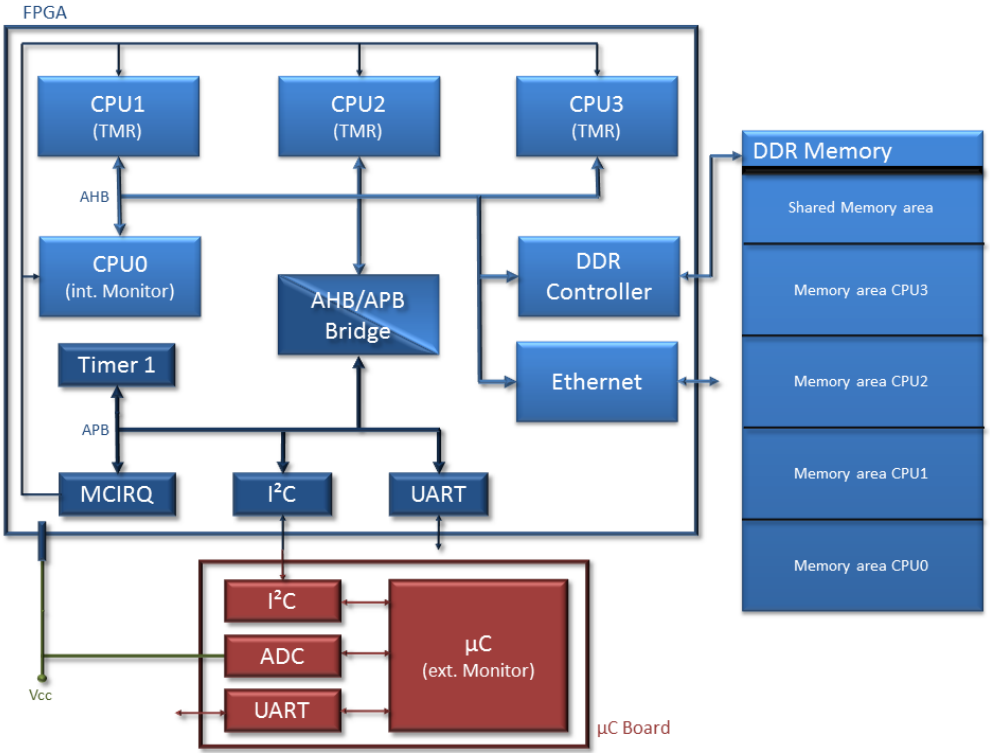

Abbildung 10: Technische Architektur des Monitoring-Konzepts eines 3-fach redundanten Systems auf Multicore-Basis

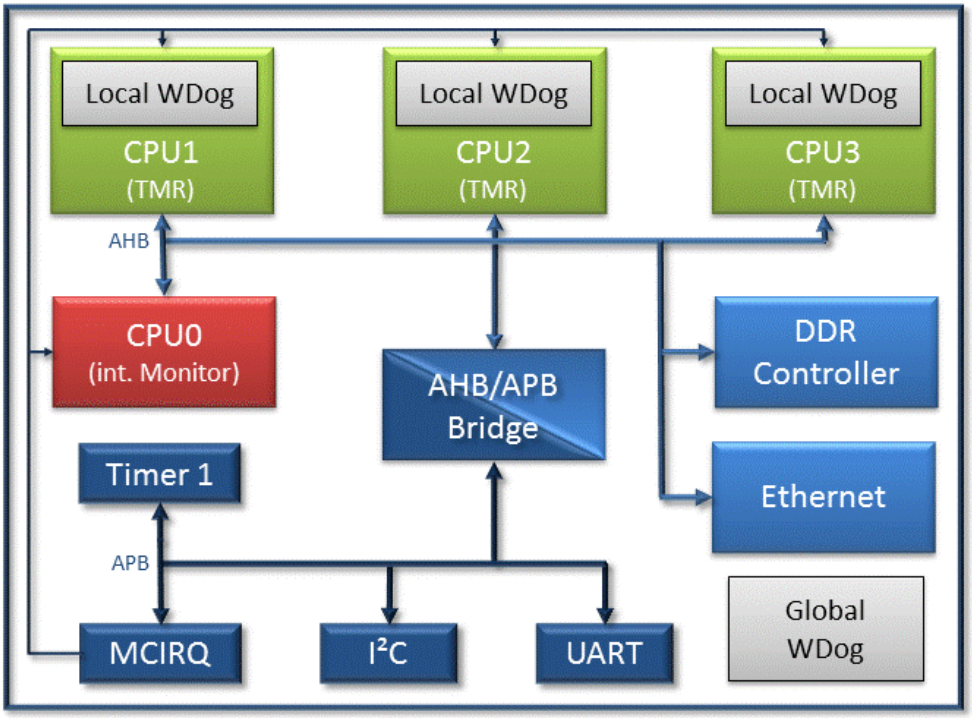

Abbildung 11 Multicore mit hierarchischem Watchdog

Durch lokale Überwachungseinheiten können Core-lokale Fehler auch lokal abgefangen und behandelt werden, mit dem Ziel übergreifende Fehler im Gesamtsystem zu vermeiden. Dies basiert auf den in TP 2 erarbeiteten Fehlerisolationsmechanismen. Das entwickelte hierarchische Watchdog-Konzept bezieht sich dabei auf die Erkennung und Eindämmung von zeitlichem Fehlverhalten auf unterschiedlichen Ebenen.

Auf Basis der Erkenntnisse der Anforderungs- und Fehlerbilderanalysen aus TP 2 ist ein weiteres Monitoring-Konzept für die Überwachung von DMA Zugriffen entstanden. Dieses Konzept beinhaltet die Möglichkeit, die Auslastung von DMA Controllern einer COTS Plattform zu evaluieren. Auf diese Weise ist es möglich, die Zugriffe und die Aktivierungsquellen der DMA Transfers aufzuzeichnen und mit den hier gewonnenen Informationen eine Überbeanspruchung des DMA Controllers zu erkennen und darauf zu reagieren. Die Umsetzung dieses in Abbildung 12 gezeigten Konzepts wurde prototypisch auf der i.MX 6 COTS Plattform von Freescale vorgenommen. Sämtliche Evaluationsergebnisse im Bereich Safety wurden mit den Safety-Mechanismen auf Software-Seite aus TP 4 abgeglichen und auf die Systemarchitektur im Rahmen des Optimierungsprozesses in TP 2 zurückgeführt.

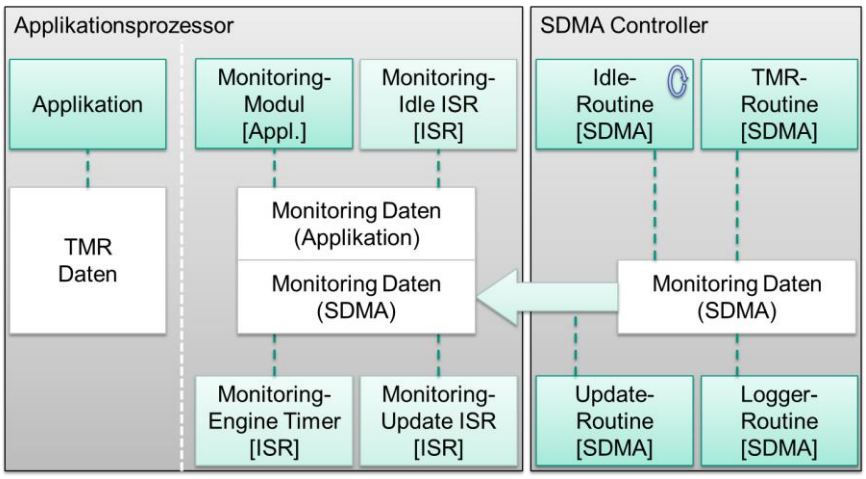

Abbildung 12 DMA Monitoring Konzept

8.1.4 TP 4 Software

Im Rahmen des Teilprojekts TP 4 beschäftigte sich das KIT innerhalb des AP 4.4 mit Aspekten der Entwicklung von Treibern für virtualisierte Coprozessoren sowie deren Optimierung hinsichtlich von Fast-Boot-Anforderungen. Auf Basis der in AP 3.1 und AP 3.4 verwendeten Hardware-Plattform sowie des quelloffenen Microhypervisors NOVA wurden Konzepte für die Anbindung von hardwareunterstützten virtualisierten Copro-

zessoren an die Software erarbeitet, prototypisch implementiert und evaluiert. Diese Arbeit diente auch als Grundlage für die Demonstrator-Entwicklungen in AP 6.1, wo der WindRiver Hypervisor eingesetzt wurde. Weiterhin wurden Konzepte für die Anbindung von hardwareunterstützten virtualisierten Coprozessoren mit PCIe SR-IOV-Anbindung erarbeitet, prototypisch für KVM / Linux implementiert und evaluiert. Um die hardwareseitige SR-IOV Unterstützung von Xilinx Virtex-7 FPGAs auf der eingesetzten Intel IA-32 Plattform zu ermöglichen, wurde das evaluierte Hypervisor-System KVM dabei entsprechend angepasst. Die Implementierung der im TP 6 VCT-Demonstrator eingesetzten Hypervisor-Lösung des Partners WindRiver wurde erweitert und an die Implementierungen aus TP 3 und AP 4.4 angepasst sowie getestet.

8.1.5 TP 5 Durchgängige Entwicklungsmethodik und Anbindung an RTP

Das KIT hat für der Erfassung aktuell eingesetzter Werkzeuge und Methoden Werkzeugsteckbriefe erstellt und die verwendeten Werkzeugketten im Rahmen von TP 5 dokumentiert. Die Ergebnisse von TP 5 wurden begutachtet und entsprechend den im Projekt gewonnenen Erkenntnissen zur Anwendung von Modellierungsmethoden und Werkzeugen ergänzt.

8.1.6 TP 6 Demonstratoren

Für die Demonstration der eigenen Projektbeiträge war das KIT hauptsächlich am VCT-Demonstrator aus AP 6.1 beteiligt, um die Konzepte und Implementierungen des Schwerpunkthemas Anbindung von gemeinsam genutzten Coprozessoren am Beispiel eines virtualisierten Steuergerätes zu demonstrieren. Zu diesem Zweck entstand ein Use Case, der den Einsatz eines kryptografischen Coprozessors zur Absicherung von Car-2-X Nachrichten gewährleistet und im VCT-Demonstrator als gemeinsam genutzte Ressource von Partitionen unterschiedlicher Kritikalität eingebunden wurde.

Für die Kommunikationsinfrastruktur, die Simulation der Car-2-X Demonstration und die Visualisierung der eigenen Beiträge wurde ein Konzept erarbeitet und implementiert sowie im Zusammenspiel mit den Komponenten der anderen beteiligten Partner abgestimmt und getestet. Die Hypervisor-Schnittstelle wurde für die eigenen Beiträge angepasst. Basierend auf den Evaluationserkenntnissen wurde eine Abstimmung mit den beteiligten Partnern und entsprechende Anpassung der Hardware-Implementierung aufgrund der durch die Hypervisor-Architektur gegebenen Rahmenbedingungen durchgeführt. Insgesamt ergibt sich der in

Abbildung 13 dargestellte Aufbau an Komponenten im Demonstrator.

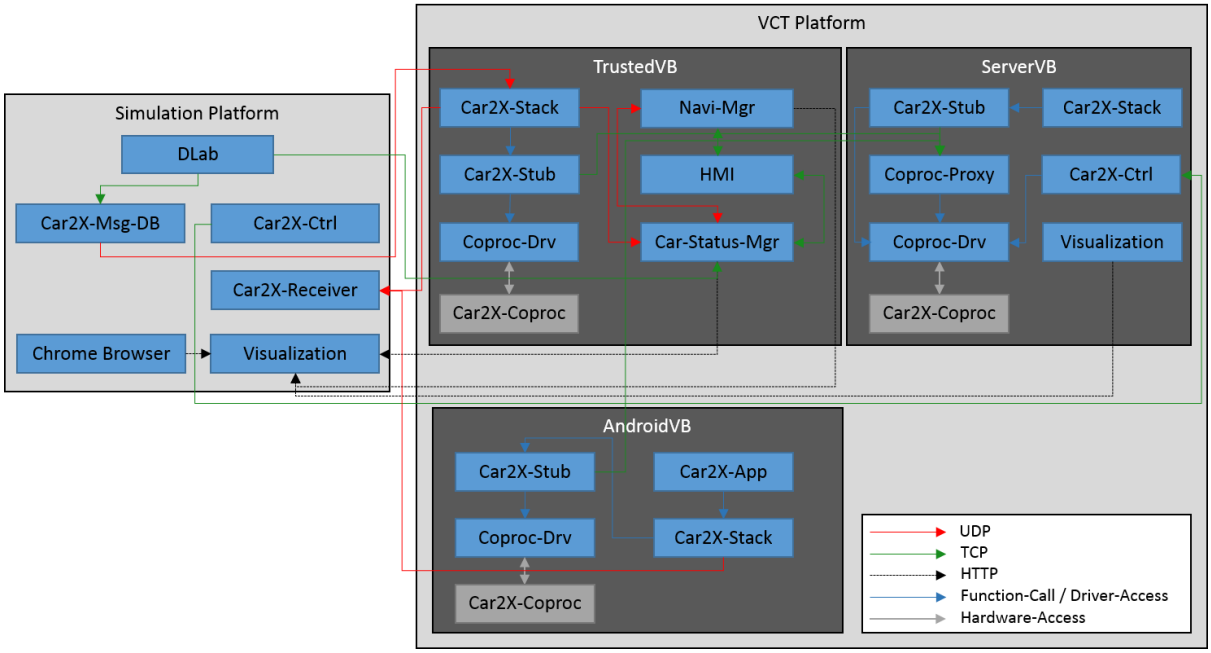

Abbildung 13 VCT Demonstrator Komponenten vom KIT

Um die Validierung der Ergebnisse auf eine einheitliche und Demonstrator-übergreifende Art und Weise sicherstellen zu können, wurde ein Validierungsprozess für die ARAMiS Demonstratoren entwickelt. Die Validierung wurde anschließend auf Basis der in TP 2 erfassten Requirements durchgeführt und dokumentiert.

8.2 Notwendigkeit und Angemessenheit der Arbeiten

Multicore-Systeme müssen für den Einsatz in den Mobilitätsdomänen Automobil, Avionik und Bahn weitreichende und spezifische funktionale und nicht funktionale Anforderungen erfüllen, die über diejenigen des General Purpose Computings weit hinausgehen: Echtzeitfähigkeit, Leistungsfähigkeit, Zuverlässigkeit und Verfügbarkeit, zertifizierbare Funktionssicherheit (Safety), Sicherheit gegen Angriffe (Security), Kompatibilität zu bestehenden Konzepten und Energieeffizienz. Die Übertragung von Lösungen aus anderen Domänen, die Multicore-Systeme bereits erfolgreich eingesetzt haben, war nur begrenzt möglich. ARAMiS hatte deshalb zum Ziel, für die Mobilitätsdomänen spezialisierte Architekturen und Methoden zu entwickeln. Auf Grund der nur sehr eingeschränkten Übertragbarkeit von bereits existierende Erfahrungen aus Singlecore-Lösungen, bestand die Notwendigkeit zur Betrachtung

des gesamten Entwicklungsprozesses und insbesondere die Zusammenarbeit verschiedener Partner über die Grenzen der Domänen hinweg. Insbesondere um die Koexistenz von Funktionen unterschiedlicher Kritikalität auf einem hochintegrierten Multicore-System-on-Chip zu ermöglichen, bedarf es spezieller Schutzmaßnahmen, die u.a. über eine Virtualisierungsschicht zur Verfügung gestellt werden können. Auch hier erforderte die Komplexität eine intensive Zusammenarbeit zwischen Hardware-Herstellern, Basissoftware-Entwicklern und Forschungspartnern, um die Herausforderung der Erfüllung der oben genannten Anforderungen ermöglichen zu können.

8.3 Fortschritte auf dem Gebiet des Vorhabens

Als Fortschritt des Stands der Technik während der Projektlaufzeit von ARAMiS ist für die Automotive Domäne die Erweiterung des AUTOSAR 4.0 Standards für Multicore (AUTOSAR – Guide to Multi-Core Systems) zu nennen, die eine Grundlage für die zukünftige Anpassung von AUTOSAR-konformen Architekturen für die Verwendung von Multicore-Hardware für bestimmte Szenarien liefert. Die Konzepte sind entsprechend in die Arbeiten und Demonstratoren von ARAMiS eingeflossen.

In der Domäne Avionik hat sich die Sichtweise von Zertifizierungsbehörden bzgl. der grundsätzlichen Verwendbarkeit von Multicore-Architekturen erweitert, so dass unter bestimmten Bedingungen der Einsatz von Dualcore-Prozessoren als prinzipiell möglich eingestuft wurde (Position Paper CAST-32 – Multi-core Processors).

Das parallel gelaufene Projekt SPES_XT beschäftigte sich mit der Entwicklung einer Methodik zur durchgängig modellbasierten Entwicklung von eingebetteten Systemen. SPES XT Ergebnisse sind während der Laufzeit von ARAMiS im Teilprojekt TP 2 (Systemarchitektur) als Basis für Modellierungsaktivitäten eingeflossen und haben durch ARAMiS eine Weiterentwicklung bzgl. der direkten Anwendbarkeit für Multicore-Architekturen erfahren.

8.4 Veröffentlichung der Ergebnisse

Im Rahmen von ARAMiS konnten einige wissenschaftliche Publikationen platziert und somit der Öffentlichkeit zugänglich gemacht werden (Auszug):

[1] **Architectural Measures Against Radiation Effects in Multicore SoC for Safety Critical Applications**
O. Sander, F. Bapp, T. Sandmann, D. V. Vu, S. Baehr, J. Becker – IEEE 57th Midwest Symposium on Circuits and Systems (MWSCAS 14), 2014

[2] **The promised Future of Multicore-Processors in Avionics Systems**
O. Sander, F. Bapp, T. Sandmann, L. Dieudonne, J. Becker

[3] **Hardware virtualization support for shared resources in mixed-criticality multicore systems**
O. Sander, T. Sandmann, D. V. Vu, S. Baehr, F. Bapp, J. Becker, H. U. Michel, D. Kaule, D. Adam, E. Lubbers, J. Hairbucher, A. Richter, C. Herber, A. Herkersdorf – Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014, S. 1-6, 2014

[4] **Adapting Commercial Off-The-Shelf Multicore Processors for Safety-Related Automotive Systems Using Online Monitoring**
F. Bapp, O. Sander, T. Sandmann, D. V. Vu, S. Baehr, J. Becker – SAE 2015 World Congress and Exhibition, 2015

[5] **Virtualization Support for FPGA-based Coprocessors Connected via PCI Express to an Intel Multicore Platform**
V. Vu, T. Sandmann, S. Baehr, O. Sander, J. Becker – Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2014 IEEE 28th International, 2014

[6] **Enabling Partial Reconfiguration for Coprocessors in Mixed Criticality Multicore Systems Using PCI Express Single-Root I/O Virtualization**
D. V. Vu, O. Sander, T. Sandmann, S. Baehr, J. Heidelberger, J. Becker – 2014 International Conference on Reconfigurable Computing and FPGAs (ReConFig'14), 2014

[7] **A Flexible Interface Architecture for Reconfigurable Coprocessors in Embedded Multicore Systems using PCIe Single-Root I/O Virtualization**
O. Sander, T. Sandmann, S. Baehr, D. V. Vu, E. Lubbers, J. Becker – The 2014 International Conference on Field-Programmable Technology (ICFPT 2014), 2014

[8] **Embedded Virtualization Approaches for Ensuring Safety and Security within E/E Automotive Systems**
D. Reinhardt, D. Adam, E. Lubbers, R. Amarnath, R. Schneider, S. Gansel, S. Schnitzer, C. Herber, T. Sandmann, H. U. Michel, D. Kaule, D. Olkun, M. Rehm, J. Harnisch, A. Richter, S. Baehr, O. Sander, J. Becker, U. Baumgarten, H. Theiling – Embedded World Conference, 2015

[9] **Two Architecture Approaches for MILS Systems in Mobility Domains (Automobile, Railway and Avionik)**
D. Adam, S. Tverdyshev, C. Rolfes, T. Sandmann, S. Baehr, O. Sander, J. Becker, U. Baumgarten – International Workshop on MILS: Architecture and Assurance for Secure Systems (MILS 2015), 2015

[10] **On-Demand Reconfiguration for Coprocessors in Mixed Criticality Multicore Systems**
D. V. Vu, O. Sander, T. Sandmann, J. Heidelberger, S. Baehr, J. Becker – 7th International Workshop on Dependable Many-Core Computing (DMCC 2015), 2015

9 Liebherr-Aerospace Lindenberg GmbH

9.1 Wissenschaftlich-technische Ergebnisse

9.1.1 TP1 Anforderungen und Szenarien

Als wesentliches Ergebnis von TP1 ist die Ermittlung der Anforderungen an Multicore-Systeme und die daraus resultierende Darstellung in entsprechende Szenarien zu nennen. Um dies zu erreichen, arbeitete Liebherr-Aerospace im TP1-Kernteam mit um auf diese Weise die avionikspezifischen Erfahrungen in Form von entsprechenden Anforderungen für sicherheitskritische Systeme in das Teilprojekt einbringen zu können. Ein weiterer LLI-Beitrag stellte das Einbringen der in SPES2020 entwickelten Methodologie-Matrix dar, die als Basis für die Reference Technology Platform (RTP)-Methodologie in ARAMiS dient.

Das Ziel von TP1 war die Ableitung und Validierung der Anforderungen an Multicore-Systeme, Virtualisierungslösungen und die dabei eingesetzten Engineeringprozesse, einschließlich Zertifizierung der funktionalen Sicherheit, wie sie sich aus zukünftigen Anwendungsszenarien in den Mobilitätsdomänen Automotive, Avionik und Bahn ergeben. Es wurde durch die geleisteten Arbeiten erreicht.

9.1.2 TP2 Systementwurf

Liebherr-Aerospace führte neben anderen Beiträgen und Aufgaben die Koordination des Task 2.1.3 „Logische Rechnerarchitektur und Funktionen der Laufzeitumgebung“ durch. Im Rahmen dieser Tätigkeit entstanden die Ergebnisdokumente E2.1.3.1 – E2.1.3.8, die die folgenden essentiellen Themen beschrieben: konfigurierbare Modellen für (logische) System-Architekturen, notwendige Runtime-Environment Spezifikation, logische architekturale Pattern zur Sicherstellung der Sicherheitsfähigkeiten Multicore-basierten Systemen, Virtua-lisierungsmechanismen aus logischer Ebene, Verteilungstechniken für Software Applikationen auf Multicore-basierten Rechner, und Segregations- und Timing-Analysemethoden für sicherheitskritische Funktionen in verteilten Systeme. Diese Dokumente wurden sowohl durch Partner- als auch durch Liebherr-Beiträge für das Deliverable D2.1 realisiert.

Im Rahmen der Entwicklung des Ergebnisdokuments E2.1.4.1 „Description of the technical architecture based on configurable models“ für das Deliverable D2.2 wurde von Liebherr-Aerospace

eine Beschreibung sowie der Bewertung verschiedener Design Pattern für die technische Architektur von Multicore-basierten Durchführungsplattformen mit unterschiedlicher Core-Anzahl erstellt. Dabei wurde auch ein konfigurierbares Modell für die Formalisierung der Fähigkeiten von Multicore-basierten Durchführungsplattformen entwickelt. Es wurden ebenfalls Requirements an die anderen TPs formuliert.

Die für das Ergebnisdokument E2.1.4.3 „Report on the Specification of the Runtime Environment“ gelieferten Anforderungen an eine technische Architektur sowie die entsprechenden Metamodelle stellten einen weiteren Beitrag an das Deliverable D2.1 durch Liebherr-Aerospace dar.

Für das Deliverable 2.4 „System Architecture“ beschrieb LLI im Ergebnisdokument E2.3.2.1 „Control and Monitor Architectures“ verschiedenen Kategorien von Monitoren, die in der Avionik Anwendung finden. Eine Klassifizierung von Monitoren gemäß des Avionik Standard ARINC 653 wurde erarbeitet. Bei der Erstellung des Ergebnisdokuments E2.3.3.1 „Specification of Fault Isolation Mechanism in Multicore Systems“ erfolgte eine Mitarbeit durch LLI durch Einführung mehrerer Crossreferenzen bzw. Korrektur vorhandener Referenzen.

Für das E2.2.1.1 „Specification of security Architecture“ für Deliverable D2.3 beschäftigte sich LLI mit der Analyse von Zielen und Anforderungen, welche die potentiellen Bedrohungen auf Applikationen beschreiben, die sich einen Multicore-Prozessor teilen. Dafür wurden entsprechende Anforderungen weiterbearbeitet.

Als Mitglied des TP2-Core-Teams brachte Liebherr-Aerospace die avionikspezifischen Sichtweisen zu den Themen funktionalen Sicherheit und Integrität mit Lockstep/Monitoring sowie Energie-Management und dessen Einfluss auf die Systemarchitektur ein.

Für das Deliverable D2.5 erfolgten durch LLI Ergänzungen am Dokument E2.1.4.1 „Technical Architecture“ und ein ausführlicher Beitrag für das Ergebnisdokument E2.3.4.1 „Optimization of the Architectural Safety Properties“. LLI definierte und arbeitete den Use-Case „Corruption of the Segregation Mechanism due to SEE“. Dafür wurden drei haupt- sowie mehrere Unter-Fehler-Szenarien erstellt. Diese Szenarien erlauben das Verhalten beim Auftreten verschiedener Ereignissen wie z.B. Single Event Effect (SEE), Multiple Bit Upset (MBU), usw. bezüglich Memory Management Unit (MMU) sowie Memory Protection Unit (MPU) zu beschreiben.

Im Rahmen von AP2.1 wurden ebenfalls das Architektur- und sicherheitsrelevante Verhaltensdesign für die LLI-Plattform-Architektur im Besonderen für Segregation und Determinismus durchgeführt (in Abstimmung mit Arbeiten in TP4 und TP6). Bei der Weiterentwicklung sowohl der logischen als auch der

technischen Architektur für die LLI-System-Architektur wurde der Fokus auf deren Einsatz für den Liebherr-Demonstrator gelegt.

Das Ziel von TP2, die Entwicklung von Systemarchitekturen und darauf zugeschnittene Hardware/Software-Partitionierungen für zukünftige Multicore-Systeme, wurde weitestgehend erreicht. Die Zulassungsproblematik ist noch nicht umfassend gelöst.

9.1.3 TP 3 Hardware

Die Mitarbeit am Teilprojekt 3 durch Liebherr-Aerospace erfolgte mit dem Ziel, passende Konzepte für Avionik-sicherheitskritische Systeme mit innovativer und heterogener Hardware-Architekturen zu analysieren und ggf. Verbesserungen zu spezifizieren. Die Schwerpunkte waren dabei die Tätigkeiten in den entsprechenden Sub-Task des AP3.1, u.a. bei der Analyse bestehender Multicore-Hardware-Architekturen.

Ein wichtiges Betätigungsfeld war die Analyse mehrerer Multicore-Prozessoren bezüglich ihrer Einsetzbarkeit. Dabei wurden Kriterien für die Bewertung von MC-Prozessoren definiert und ausgearbeitet um auch ihre Eignung für einen Einsatz in der Avionik-Domäne feststellen zu können. Darauf aufbauend startete die Auswahlphase der Multicore-Prozessoren, u.a. durch konkreten technischen Austausch mit den avioniktauglichen Prozessorherstellern Partner in ARAMiS Infineon und Freescale. Im Laufe des Projektes wurden diese Kriterien weiter vertieft, im Besonderen für Mitigation / Safety-Mechanismen für Segregation, Fehlerdetektion, -Isolation und -Korrektur, u.a. wegen Radiation und Störfällen. Es wurden ebenfalls Anforderung bzw. Verbesserungsvorschläge für Multicore-Prozessoren an die Chip-Hersteller erarbeitet.

Für das Ergebnisdokument E3.1.1.1 „Requirements Specification for Multi-Core Hardware“ wurde an der Ermittlung der notwendigen Requirements an Multicore-Prozessoren unter schwerpunktmäßiger Betrachtung der „Avionic Safety“ gearbeitet.

Mit der Evaluierung von Multicore-SoC, u.a. für Flight Control System (vor allem für den LLI-Demonstrator), wurden die Eigenschaften mehrerer Multicore-Plattformen erfasst und geliefert. Konkret wurden dabei die Architekturen AURIX von Infineon sowie McKinley von Freescale untersucht. Die daraus resultierenden Anforderungen an passende Chips wurden im Dokument E3.1.2.1 als Liebherr-Beitrag dokumentiert. Es wurden auch weiter Prozessoren genannt, die durch die Chip-Hersteller beschrieben werden sollten.

Um die Hardware-Separation und –Virtualisierungsmöglichkeiten zu untersuchen, wurden für mehrere Multicore-Architekturen entsprechende Analysen durchgeführt.

Die Formulierung und Klassifizierung von Anforderungen an Multicore-Prozessoren, mit Schwerpunkt auf „Avionic Safety“, wurde in das Ergebnisdokument E3.1.2.2 „Classification of Existing and Future Requirements“ eingebracht.

Arbeiten und Überlegungen zur Safety- und Zertifizierungs-problematik wurden im Rahmen von AP3.3 als TP-übergreifen durchgeführt. Dabei wurde ebenso der Einfluss auf die Multicore-Auswahlkriterien und auf die Verwendung der Multicore-Prozessoren sowie die notwendigen SW-Mitigations untersucht.

Durch die geleisteten Arbeiten konnten die Ziele von TP3, die Erarbeitung von Hardware-Architekturkonzepte und Methoden für eine optimale Nutzung der Multicore-Technologie unter den in TP1 und TP2 erarbeiteten Vorgaben, erreicht werden.

9.1.4 TP4 Software

Für künftige Entwicklungen von Avionik-Systemen bei Liebherr-Aerospace wird es unerlässlich sein, Software-Architekturkonzepte und –Methoden, die die Nutzung von Multicore-Technologien und Virtualisierung beinhalten, zu erarbeiten. Um diese Ziele erreichen zu können, stellte die Mitarbeit am Teilprojekt TP 4 „Software“ einen der Schwerpunkte von Liebherr-Aerospace im Rahmen von ARAMiS dar.

Liebherr-Aerospace arbeitete im Rahmen von AP4.1 an der Erstellung eines eigenen Middleware-Konzeptes. Dabei wurde die Integration eines ebenfalls entwickelten schlanken Ressource-Management-Layers in das Gesamtkonzept der Middleware, begleitet von prototypischen Tests sowie Programmierungen, durchgeführt.

Im Einzelnen wurde bei der Konzipierung eines Segregations-konzeptes für Avionik-Software ebenfalls Ressource-Sharing-Management (RSM) für Multicoresysteme untersucht und entsprechende Konzepte (mit Vor- und Nachteilen) formuliert. Dabei wurden die Anforderungen aus der Avionik an eine Plattform, die für die höchsten Kritikalitätsstufen angemessen sind, berücksichtigt. Die auf diese Weise erarbeitetet Ergebnisse wurden als Beitrag von Liebherr-Aerospace in das Ergebnisdokument E4.1.1.2 „Multicore Segregation Concepts“ eingebracht.

Liebherr-Aerospace übernahm die Verantwortung für das im Rahmen von D4.2 zu erstellenden Ergebnisdokuments E4.1.1.5 „Evaluation Overall Concepts Identification of Problem and Solutions“. Neben Organisation- und Koordinierungsaufgaben lieferte LLI auch eigene Fachbeiträge. U.a. wurde ein Detailkonzept der Liebherr-Softwarearchitektur für eine deterministische, sicherheits-, modular- multicore-basierte

Plattform ausgearbeitet. Es wurden Herausforderungen für die Segregation ausgearbeitet und formuliert, sowie auch mögliche Lösungsansätze erarbeitet. Dabei wurde auch beachtet, die in den Standards DO178B/C beschriebenen Anforderung zu erfüllen. Die Hauptherausforderung an die Segregation, um dies Standards zu erfüllen, besteht darin, dass auf eine Hardware-Ressource nur von einer Softwarekomponente zugegriffen werden darf. Eine Ausnahme stellt die Möglichkeit dar, wenn garantiert werden kann, dass sich die Software-Komponenten gegenseitig nicht beeinflussen. In der entwickelten LLI-Plattform wurde diese Möglichkeit durch eine Zeit- und Budget-Aufteilung in einer dedizierten Architektur umgesetzt. Somit könnte ein Einsatz von Multicore-Technologie in der Avionik ermöglicht werden. Für die Weiterentwicklung der LLI-Multicore-Plattform kam der McKinley MPC5746M von Freescale zum Einsatz. Eine Integration des Liebherr-Aerospace Operating System (LAOS) in das „Distributed Modular Platform"-Konzept (DiMoP) wurde erfolgreich durchgeführt. In diesem Zusammenhang wurden verschiedene Erweiterungen realisiert, wie z.B. das Anlegen von 4 Beispielapplikationen auf den entsprechenden Application-Cores. Es wurde auch eine Unterscheidung zwischen interner Core-Kommunikation und der Kommunikation über Core-Grenzen im Rahmen der Inter-Core/Inter-Partition-Kommunikation (ICC/IPC) umgesetzt. Um das zu erreichen, war es notwendig, das eingesetzte LAOS durch die Implementierung eines spezifischen Kommunikationslayers „Basic Input Output Functions (BIOF)" zu erweitern.

Außerdem erstellte Liebherr eine Beschreibung des Standards ARINC 653 die die aktuellen und wichtigsten Charakteristika, relevant für die Multicore-Problematik, beschreibt, sowie eine Beschreibung der für den Einsatz von Multicore-Prozessoren notwendigen Erweiterungen. Das Ergebnis wurde als Beitrag in das Ergebnisdokument E4.1.2 „Evaluation and Modification of Middleware and Operating Systems" eingebracht.

Für die Liebherr-Plattform wurden im Rahmen der AP4.2 verschiedene Verbesserungen/Optimierungen für die Inter-Core Kommunikation per „Shared memory" (mehrere Shared Memory, Update Flags, etc.) vorgenommen.

Für das Ergebnisdokument E4.3.1.1 „Requirements for the Certification of Multicore Architectures" realisierte Liebherr das Fachreview für das Kapitel „Compliance Viewpoint" und brachte seine Fachkompetenzen bei Workshops für die Realisierung einer Analyse der DO178C ein. Im Rahmen von E4.3.3.1 beteiligte sich Liebherr am Thema „Segregation" durch eine Beschreibung der Mechanismen in Bezug auf Safety-Herausforderungen. Es wurden ebenfalls im Rahmen von Reviews Verbesserungsvorschläge, basierend auf Liebherr-Erfahrungen bezüglich sicherheitskritischer Luftfahrtsystemen, eingebracht.

Im Rahmen von AP4.4 wurde das Isolationskonzept der Liebherr-Multicore-Plattform entwickelt und während der Projektlaufzeit weiter optimiert.

Die Erarbeitung von Software-Architekturkonzepte und-methoden, unter Nutzung der Multicore-Technologie und Virtualisierung gemäß den in TP1 und TP2 erarbeiteten Vorgaben, wurde als Ziel für TP4 erreicht.

9.1.5 TP 5 Durchgängige Entwicklungsmethodik

Die in den TP2-TP4 als notwendig identifizierten Werkzeuge, Methoden sowie Anforderungen an deren Schnittstellen stellen die Basis für die innerhalb von TP5 zu leistenden Arbeiten dar. Ziel des TP5 war deshalb die Bereitstellung einer durchgängigen, domänenübergreifenden Methoden- und Werkzeugplattform für den Entwurf Multicore-basierter Systeme. Die dabei entstehende Reference Technology Platform (RTP) soll insbesondere eine Unterstützung der Nachweisführung zur Zulassung ermöglichen.

Dazu lieferte Liebherr-Aerospace für die Erstellung des Deliveable D5.1 „Requirements for the Reference Technology Platform (RTP)“ den ausführlich beantworteten Fragebogen zum Thema Tools sowie entsprechende Anforderungen aus Sicht von LLI, incl. eines Basisvorschlags für ein Metamodell. In analoger Weise wurde bei der Formulierung der Anforderungen an die Methoden verfahren (Fragebogen, Erstellung von „Steckbriefen“).

Im Weiteren untersuchte und, daraus resultierend, formulierte Liebherr einen Entwicklungsprozess, der kompatibel mit den Avionikprozessen für eine optimierte Verwendung von Multicore-Prozessen (gezielt auf optimiertes Deployment) sein musste. Diese Ergebnisse wurden in das Deliverable D5.3 „User Guideline for the Seamless Methodology“ eingebracht. Es wurde auch eine Beschreibung für Methoden für die Behandlung von nicht-funktionalen Anforderungen und Aspekte geliefert. Eine Entwicklungsmethodologie für Multicore-Systeme wurde ebenfalls fertiggestellt. Insgesamt stellten die von Liebherr-Aerospace Beiträge den größten Anteil in diesem Dokument dar.

Für das Deliverable D5.4 „Tool Integration in the ARAMiS RTP“ arbeitete Liebherr-Aerospace am Thema „Domain Specific Platform“ mit. Dabei wurden die Tools bzw. die Toolkette, die üblicherweise in der Avionik-Domäne verwendet und die direkt durch die Zwänge der Multicore-Technologie beeinflusst werden, beschrieben und die Zusammenhänge dargestellt.

Liebherr-Aerospace unterstütze die Arbeiten für das Deliverable D5.5 „User Guideline for the Seamless Methodology“ durch ein Arbeitsdokument „Quickstart with SPEM“. Ein weiterer Beitrag bestand in einer beispielhaften Beschreibung der Anwendung der

RTP-Matrix in der Avionik-Domäne („An example of application of the RTP to the avionic-domain“). Für das gemeinsam mit TU Kaiserslautern erzeugte Kapitel „Process Configuration Framework“ beschreibt Liebherr-Aerospace den Anwendungsfall und die Avionik-Prozesse, basierend auf der TP5-Methodologie.

Die Erarbeitung von Software-Architekturkonzepte und-methoden, unter Nutzung der Multicore-Technologie und Virtualisierung gemäß den in TP1 und TP2 erarbeiteten Vorgaben, wurde als Ziel für TP5 erreicht.

Das Ziel von TP5 war die Bereitstellung von Werkzeugen und Methoden, die den Entwurf von Multicore-basierten Systemen unterstützen sowie die Bereitstellung einer Werkzeugplattform mit Interoperabilitätskonzept für die Integration von Werkzeugen, die den Entwurf von Multicore-basierten Systemen unterstützt, wurde prinzipiell erreicht.

9.1.6 TP6 Demonstratoren

Liebherr-Aerospace arbeitete innerhalb des Teilprojektes TP6 am Arbeitspaket AP6.3 mit. Die Mitarbeit sollte dazu dienen, mittels eines dafür entwickelten Demonstrators die Verteilung von Funktionen auf einer Multicore-Plattform ebenso wie die in den Teilprojekten TP2,3 und 4 entwickelten Konzepte und Technologien darstellen zu können. Die Konzepte zu Determinismus, Segregation und Safety, sollen damit demonstriert und erprobt werden können.

Um eine geeignete Multicore-Plattform zu finden, wurden Auswahlkriterien ermittelt sowie eine Recherche der potentiell für die Avionik-Domäne bzw. Liebherr geeigneten Multicore-Mikrocontroller durchgeführt. Es wurden mehr als 10 Multicore-Prozessoren aus verschiedenen Halbleitern evaluiert. Als am besten geeignete Multicore-Controller wurden dabei der MPC5746M („McKinley“) von Freescale sowie der Tricore AURIX TC2775 von Infineon ermittelt.

Es wurde ein Konzept des Demonstrators „Flight-Control-System (FCS)“ entwickelt, dass im Collison Avoidance System Demonstrator von Airbus D&S angebunden wurde. Die Modellierung der Anforderungen und Use-Cases erfolgte dabei durch das Tool Enterprise Architect.

Nach Auswahl des geeigneten Multicore-Prozessors, den MPC5746M (3 Kerne) von Freescale aus Verfügbarkeitsgründe (ideale Wahl wäre der AURIX TC2775 von Infineon), erfolgte die Inbetriebnahme sowie der Realisierung der in TP2 und TP4 erarbeiteten Konzepten. Daraus wurde die Liebherr-Plattform „Distributed Modular Platform (DiMoP)“ realisiert. Die für diese Plattform konzipierten Applikationen wurden entwickelt,

insbesondere wurden basierend auf dem ARINC-653 Standard die Schnittstelle und Konzepte (Konfiguration, Space Partitioning, Task Scheduling) auf Multicore-prozessoren implementiert. Als Multiprocessing-Konzept wurde von Liebherr-Aerospace ein „synchronized Asymetric Multi-Processing“ (sAMP) entschieden, das die Vorteile für Segregation und Determinismus des AMP nützt und die Nachteile (u.a. Performanzreduzierung wegen multilateralen Synchronisierungen) stark vermindert. In diesem Konzept ist die Benutzung der Cores spezialisiert: ein Core wird für das IO-Management reserviert, die anderen Cores führen die Applikationen durch und kommunizieren mit dem IO-Core für jeden Zugriff auf die Peripherie. Das von Liebherr-Aerospace entwickelte Operating System (LAOS) wurde auf die Applications-Cores erfolgreich portiert.

Mit der erfolgreichen Validierung der in TP2-TP4 entwickelten Architekturvorlagen und Methoden für Multicore-basierte Hard- und Software sowie die Validierung der in TP5 entwickelten Werkzeuge und Methoden für die Unterstützung von Multicore-basierten Systemen, wurde das festgelegte Ziel erreicht.

9.2 Notwendigkeit und Angemessenheit der Arbeiten

Bedingt durch die Komplexität des Themas, ergibt sich die Notwendigkeit, dass sich Firmen aus den unterschiedlichen Mobilitätsdomänen sowie akademische Partner im Rahmen eines Forschungsprojektes zusammenschließen um zukunftsträchtige Lösungen erarbeiten zu können. Aus diesem Grund ist eine Förderung durch das BMBF angesichts der Wichtigkeit Multicore-Technologie für die Zukunft des Standorts Deutschlands angemessen.

Der Umfang der Förderung ist durch die sehr umfangreiche, anspruchsvolle sowie zukunftsweisende Aufgabenstellung zu rechtfertigen.

Umfangreiche und komplexe Arbeiten waren notwendig um die Ziele zu erreichen. Dabei wurde die Zertifizierungsproblematik noch nicht vollständig gelöst und es bietet sich an in einem Nachfolgeprojekt wieder aufgegriffen zu werden

9.3 Fortschritte auf dem Gebiet des Vorhabens

Die Ergebnisse, allein realisiert oder gemeinsam mit verschiedenen Partner aus den Avionik, Automotive und Chip-Hersteller Domänen, sowie aus der Akademia, bringen Liebherr-Aerospace Lindenberg wesentliche Fortschritte für eine effiziente und vorschriftgemäße Realisierung von Produkten basierend auf Multicore-Prozessoren.

In erste Linie ermöglicht die genaue Identifizierung der potentiellen Probleme bei der Verwendung von Multicore-Prozessoren für sicherheitskritische Domäne, im Besonderen für die Avionik, die Unterstützung der korrekten Designentscheidungen, auf System, Equipment und Software Ebenen. Unter anderem können jetzt nicht nur zuverlässige Software Architekturen für Multicore entwickelt werden, sondern auch korrekte und effiziente System-Architekturen durch Verteilung von sicherheitskritischen Applikationen auf optimierte Multicore-basierte Hardware Architektur auf die Equipment-Ebene realisiert werden.

Für eine komplette, durchgängige übergeordnete Methodologie für die Entwicklung von Systemen mit Multicore-prozessoren werden noch gewisse Untersuchungen benötigt. Die fundamentalen Schritte und Aspekten wurden aber festgelegt, wie zum Beispiel eine bessere detaillierte Formalisierung der Anforderungen, notwendig für Systeme basierend auf Multicore-System-on-Chip Komponenten mit zahlreichen gemeinsamen Ressourcen.

Viele Ergebnisse aus ARAMiS bilden eine solide Basis für konstruktive Argumente für die Avionik-Zertifizierungsaktivitäten, besonders wichtig, weil derzeit zu wenig und sehr (zu) strenge „Recommendations“ Seitens der Behörden angekündigt worden sind.

9.4 Veröffentlichung der Ergebnisse

Liste der erfolgten Veröffentlichungen:

[1] „The Promised Future of Multicore Processors in Avionic Systems“
Autoren: O. Sander (KIT), F. Bapp (KIT), L. Dieudonné (Liebherr-Aerospace), T. Sandmann (KIT), J. Becker (KIT)
Präsentiert durch L. Dieudonné bei der DGLRK2014 Konferenz

[2] „A Classification Schema for Development Technologies“
Autoren: D. Taibi (TU Kaiserslautern), L. Dieudonné (Liebherr-Aerospace)
Präsentiert durch Davide Taibi bei der ICSEA-2014-Konferenz

[3] „How to select the most appropriate technology using the Reference Technology Platform and the Process Configuration Framework“
Autoren: P. Diebold (Fraunhofer IESE), D. Taibi (TU Kaiserslautern), L. Dieudonné (Liebherr-Aerospace)
Präsentiert durch Davide Taibi bei der Euromicro/SEAA-2014-Konferenz

Geplante zukünftige Veröffentlichungen:

[1] ARAMiS-Buch für 2016 geplant

10 OFFIS e.V.

10.1 Wissenschaftlich-technische Ergebnisse

Ein wesentliches Ziel von OFFIS im Projekt ARAMiS war es den Brückenschlag zwischen abstrakteren Modellierungstechniken für Multicore Systeme via Taskmodellen über Ressourcenmodelle für virtuelle Architekturen hin zu konkreten Multicore Architekturen zu erlauben und dafür entsprechende kompositionale Timing Analysen bereit zu stellen. Die hierbei anfallenden Methoden und Werkzeuge sollten durchgängig in die Reference Technology Platform (RTP) integriert werden.

Des Weiteren hatte OFFIS als TP5 Leader neben Daimler das Ziel die RTP aus den Projekten SPES2020 und CESAR für ARAMiS zu zuschneiden. Bei letzterem Ziel hat OFFIS eine stärker konsolidierende Rolle im Projekt eingenommen, insbesondere hinsichtlich eines ARAMiS spezifischen Metamodelles, welches die Basis für eine ARAMiS spezifische Anpassung des RTP darstellt.

Die technischen Ergebnisse, die OFFIS im Rahmen von ARAMiS erzielt hat, sind in folgenden Abschnitten des Abschlussberichtes aufgegangen:

- Abschnitt 4.5 Segregation and Timing Analysis
- Abschnitt 4.6 Contract Based Design (Decomposition)
- Abschnitt 7.4.1 Compositional Real-Time Analysis for Systems with a Hypervisor
- Abschnitt 13.2.4 Contract-Based Design and Virtual Integration Testing
- Abschnitt 13.2.5.2 Computational Timing and Scheduling Analysis in Orca
- Abschnitt 13.2.7 Preemption Analysis in Orca

Des Weiteren wurde in Kooperation mit Daimler eine Strategie für die Integration von Methoden und Werkzeugen erarbeitet, die in Abschnitt 13.3 *Integrating Methods & Tools* des Abschlussberichts dargelegt wird. Dieser Strategie folgend wurde das Werkzeug Orca um eine Schnittstelle erweitert, die es erlaubt Informationen über nebenläufige Ausführungen von Tasks an Data Race Analysen anderer Projektpartner weiter zu reichen (vgl. Abschnitte 13.4.2 und 13.4.3 des Abschlussberichtes).

Bezogen auf die einzelnen Teilprojekte hat OFFIS folgende Beiträge geleistet:

10.1.1 TP2 Systementwurf

Im Rahmen des TP2 hat OFFIS an der Definition der logischen Architektur mitgewirkt. Insbesondere hat OFFIS hierbei zusammen mit Liebherr die Arbeiten an dem Ergebnisdokument E2.1.3.6 *Strategies and Criteria for Allocating Safety Critical Functions in Distributed Architectures* koordiniert. des Weiteren wurden Vorarbeiten aus dem SPES2020 Projekt bzgl. eines ARAMiS Metamodelles eingebracht und als Grundlage für die Modellierung der logischen Architektur verankert. Diese Arbeiten wurden über die Laufzeit des TP2 hinweg mit dem ARAMiS RTP Metamodell aus TP5 synchronisiert, sodass die in TP2 entwickelten Architekturmodelle mittels der RTP beschreibbar waren.

10.1.2 TP3 Hardware

Im Rahmen des TP3 sollte eine Evaluation der Spezifika der Hardware Architekturen der Halbleiterhersteller erfolgen, insbesondere hinsichtlich der Performanz und des Grades an Determinismus bzgl. des Interconnects, der Interrupt Logik, der Unterstützung des OS hinsichtlich Monitoring, Unterstützung für Virtualisierung, etc. Im Rahmen dieser Evaluation war OFFIS verantwortlich für die Ergebnisdokumente E3.1.2.1 *Hardware: Architecture and Methods* sowie E3.1.2.2 *Classification of Existing and Future Requirements*. Zu diesem Zwecke wurde ein Fragenkatalog erarbeitet, um diese Spezifika in einer strukturierten Art und Weise zu sammeln und gegenüberstellen zu können. Diese Evaluation stellte die Basis für eine Auswahl von Hardware Architekturen dar sowie die Grundlage für die Entwicklung von Konzepten zur deterministischen Ausführung von Softwarefunktionen.

10.1.3 TP4 Software

Im Rahmen des TP4 war OFFIS an dem Arbeitspaket Virtualisierung beteiligt, präziser an dem Ergebnisdokument E4.4.7.1 *Concepts for Real-Time-Analysability of Multi-Core Virtualization Environments*. Zu diesen Ergebnissen steuerte OFFIS ein Konzept bei, mit dessen Hilfe verschiedene Applikationen, die auf einer virtualisierten Multicore Plattform ausgeführt werden, kompositional analysiert werden können hinsichtlich ihres zeitlichen Verhaltens. Dazu wurde ein formales Rahmenwerk erarbeitet, sowie entsprechende Modellierungskonzepte. Beide Aspekte sind eingeflossen in die prototypische Umsetzung eines Werkezeuges für die Durchführung dieser Analyse.

10.1.4 TP5 Durchgängige Entwicklungsmethodik und Anbindung an RTP

Das TP5 stellte den Schwerpunkt der Arbeiten von OFFIS im ARAMiS Projekt dar. Neben Daimler hat OFFIS dieses Teilprojekt geleitet und die Arbeiten an den Deliverables D5.2 *Interoperability Specification by means of a Common MetaModel Specification* und D5.4 *Tool Integration in the ARAMiS RTP* koordiniert.

Ein wesentlicher Bestandteil der erzielten Ergebnisse ist ein ARAMiS spezifisches RTP Metamodell, welches fortwährend mit Arbeiten aus TP2 abgeglichen und entsprechend erweitert wurde. Dieses Metamodell bildete die Grundlage für Konzepte zur Methoden und Werkzeug Integration und wurde von anderen Partnern im TP5 zu eben diesem Zweck genutzt. Insbesondere die durch die Werkzeugintegration erzielten Resultate sind Bestandteil des Kapitels 13 des Abschlussberichtes.

Hinsichtlich der von OFFIS entwickelten Methoden und Werkzeuge wurde im Rahmen des TP5 eine Methode zur durchgängigen Betrachtung von Segregationseigenschaften entwickelt, die in einem prototypischen Werkzeug zur kompositionalen Echtzeitanalyse integriert wurde. Dieses Werkzeug wurde unter Anwendung des Konzeptes zur Methoden und Werkzeugintegration mit Data Race Analyse Werkzeugen wie Bauhaus und Gropius vernetzt (Abschnitt 13.4.2 des Abschlussberichtes), was zu einer Reduktion der Anzahl der „false positives“ der Data Race Analyse führte. Diese Aktivität wurde maßgeblich von Daimler getrieben in Kooperation mit den Partnern Universität Stuttgart, Universität Kiel, Symtavision und OFFIS.

10.2 Notwendigkeit und Angemessenheit der Arbeiten

Das Projekt hatte zum Ziel Multicore-Technologie und Virtualisierung für Mobilitätsdomänen nutzbar zu machen. Insbesondere die nicht funktionalen Anforderungen (Echtzeitfähigkeit, Safety, Security und Performanz) unter den domänenspezifischen Qualitätsanforderungen bildeten hierbei eine wesentliche Herausforderung. Das gesteckte Ziel konnte nur durch ein Konsortium mit namhaften Vertretern von Systemherstellern aus den Anwendungsdomänen, Tool/Softwareherstellern, Halbleiterherstellern sowie Forschungseinrichtungen geschaffen werden.

Die für die Forschungsarbeiten aufgewandten Ressourcen entsprechen in etwa dem Umfang wie sie in der Vorhabensbeschreibung – und ergänzt durch die Antragsunterlagen zur Projektverlängerung – aufgeführt und begründet worden sind. Bei der Durchführung der Arbeiten wurde auf einen effizienten Einsatz der Ressourcen geachtet. Gemäß diesen Angaben ergibt sich die Notwendigkeit der durchgeführten Arbeiten als auch der eingesetzten Ressourcen.

10.3 Fortschritte auf dem Gebiet des Vorhabens

Bezogen auf kompositionale Echtzeitanalysen hat sich der Stand der Technik seit Beginn des Projektes nicht wesentlich geändert. Wie in Abschnitt 13.2.5 *Timing and Scheduling Analysis* des Abschlussberichts diskutiert, lassen sich Echtzeitanalysen bezügliche analytischer Verfahren und berechnender Verfahren klassifizieren. In der Klasse der analytischen Verfahren gibt es eine Reihe von Echtzeitanalysen, die auch eine kompositionale Analysestrategie erlauben. Basierend auf entsprechenden Publikationen sind auch Werkzeuge entstanden, die kommerziell vertrieben werden oder frei erhältlich sind. Hier sind hauptsächlich SymTA/S und die Real-Time Calculus Toolbox zu nennen. Im Gegensatz dazu sind kompositionale berechnende Verfahren für die Echtzeitanalyse nicht bekannt, außer der Methode die im Rahmen des ARAMiS Projektes von OFFIS vorangetrieben wurde.

In einem anderen Projektkontext (Designing for Adaptability and evolutioN in System of systems Engineering, DANSE) wurden verwandte Arbeiten durchgeführt, die sich mit der Analyse des Impacts von Änderungen des zu analysierenden Systems und/oder dessen Anforderungen beschäftigen. Der dort verwendete Analyseansatz ist kompatibel zu dem im ARAMiS Projekt entwickelten Ansatz. Diese Arbeiten werden in folgender Publikation diskutiert:

Tayfun Gezgin, Stefan Henkler, Ingo Stierand, Achim Rettberg; **Impact Analysis for Timing Requirements on Real-Time Systems**; The 20th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2014).

Außerdem wurden, ebenfalls im DANSE Projekt, Abstraktionstechniken für zustandsbasierte kompositionale Scheduling-Analysen untersucht. Im Sinne der Effizienzsteigerung der im ARAMiS Projekt entstandenen Analyse, werden derzeit derartige Abstraktionstechniken auf ihre Anwendbarkeit auf eben diese Analyse untersucht. Folgende Publikation beschreibt diese Abstraktionstechniken:

Tayfun Gezgin, Stefan Henkler, Achim Rettberg, Ingo Stierand; **Abstraction Techniques for Compositional State-based Scheduling Analysis**; Brazilian Symposium on Computing System Engineering (SBESC) 2012.

Des Weiteren wurde an der Verzahnung von Echtzeitanalysen und Sicherheitsaspekten geforscht. In der im folgenden Abschnitt unter Punkt 4 genannten Publikation wird der Einfluss der Integration eines Hardware Security Moduls in ein System auf dessen Echtzeiteigenschaften betrachtet unter Nutzung unterschiedlicher Interfaces des Moduls (Low-Pin-Count Interface und Memory-Mapped I/O).

10.4 Veröffentlichung der Ergebnisse

Im Rahmen der Laufzeit des ARAMiS Projektes wurden von OFFIS insgesamt 10 wissenschaftliche, begutachtete Papiere auf Peer-Review-Konferenzen veröffentlicht:

[1] Philipp Reinkemeier, Ingo Stierand; **Compositional Timing Analysis of Real-Time Systems based on Resource Segregation Abstraction**; ARAMiS special session of IESS 2013, Paderborn in Deutschland

[2] Philipp Ittershagen, Philipp A. Hartmann, Kim Grüttner, Achim Rettberg; **Hierarchical Real-Time Scheduling in the Multi-Core Era - An Overview**; ARAMiS special session of IESS 2013, Paderborn in Deutschland

[3] Maher Fakih, Kim Grüttner, Achim Rettberg, Martin Fränzle; **Exploiting Segregation in Bus-Based MPSoCs to Improve Scalability of Model-Checking-Based Performance Analysis for SDFAs**; ARAMiS special session of IESS 2013, Paderborn in Deutschland

[4] Sunil Malipatlolla, Ingo Stierand; **Evaluating the Impact of Integrating a Security Module on the Real-Time Properties of a System**; IESS 2013, Paderborn in Deutschland

[5] Sunil Malipatlolla; **A Novel Approach for a Hardware-based Secure Process Isolation in an Embedded System**; International Symposium on Security in Computing and Communications (SSCC) 2013, Indien

[6] Ingo Stierand, Sunil Malipatlolla; **Exploiting Functional Models to Assess the Security Aspect in Embedded System Design**; International Symposium on Security in Computing and Communications (SSCC) 2013, Indien

[7] Philipp Reinkemeier, Heinz Hille, Stefan Henkler; **Towards Creating Flexible Tool Chains for the Design and Analysis of Multi-Core Systems**; Envision Workshop im Rahmen der SE2014 Konferenz, Kiel in Deutschland

[8] Ingo Stierand, Philipp Reinkemeier, Purandar Bhaduri; **Virtual Integration of Real-Time Systems based on Resource Segregation Abstraction**; FORMATS 2014, Florenz in Italien

[9] Tim Schmidt, Kim Grüttner, Rainer Dömer, Achim Rettberg; **A Program State Machine Based Virtual Processing Model in SystemC**; EWiLi'14, The 4th Embedded Operating Systems Workshop

[10] Philipp Reinkemeier, Albert Benveniste, Werner Damm, Ingo Stierand; **Contracts for Schedulability Analysis**; FORMATS 2015, Madrid in Spanien

Des Weiteren wurde ein Poster im Rahmen der SBCCI Konferenz vorgestellt:

- Stefan Henkler, Philipp Reinkemeier, Achim Rettberg, Ingo Stierand, **Seamless Development of Multicore Systems**, SBCCI 2012, Brasilia in Brasilien

Im Rahmen derselben Konferenz wurde ein eingeladener Vortrag zu ARAMiS relevanten Inhalten gehalten:

- Prof. Dr. Achim Rettberg, **A Revolutionary Change in Embedded System Design**, SBCCI 2012, Brasilia in Brasilien

Außerdem entstand ein technischer Report, in dessen Rahmen erarbeitet wurde wie die Methode zur Betrachtung von Segregationseigenschaften verzahnt werden kann mit einer Erweiterung der Systemlevel-Design Sprache SystemC, die im Hause OFFIS entwickelt wurde:

- Philipp Reinkemeier, Philipp Ittershagen, Ingo Stierand, Philipp A. Hartmann, Stefan Henkler, Kim Grüttner; **Seamless Segregation for Multi-Core Systems**; OFFIS Technical Report 2013

11 Symtavision GmbH

11.1 Wissenschaftlich-technische Ergebnisse

11.1.1 Timing-Analyse von Multicore Systemen

Im Rahmen von ARAMiS hat Symtavision sein Timing-Analyseframework für Multicore erweitert und ausgebaut. Mit diesen Timing-Analysen sind Aussagen über die Zeiteigenschaften von Multicore-Systemen möglich.

Symtavision hat neben „normalen“ Multicore-Systemen auch den speziellen Einsatz-Bereich in Gateways untersucht. Solche Gateway-Verbindungen werden nicht nur für Busse verwendet, sondern werden zukünftig mit anderen Fahrzeugfunktionen auf Multi-Core-Steuergeräten integriert. Hierfür wurde die Analyse um die Unterstützung von Polling-Mechanismen erweitert, welche eine Modellierung von dynamischer, datenabhängiger Ausführung ermöglicht.

Hiermit wurden folgende Ziele aus der VHB abgedeckt:

- AP4.1: Es wurde eine Laufzeitanalyse für Multicore-Software sowie Echtzeit-Analyse für Multi-Core Betriebssysteme und Middleware erzeugt, zur Entwicklung von Multi-Core Software-Architekturen.

Symtavision hat zusammen mit Firma Daimler und der Uni Stuttgart die Kopplung von SymTA/S und Bauhaus zur verbesserten Erkennung von Data-Races umgesetzt. Hierfür wurde von Symtavision eine Datenkonsistenzanalyse für Multicore-Systeme entwickelt, welche anhand von Modellinformationen, Simulationen und/oder Trace-Daten erkennen kann, ob und wo es im System zu Inkonsistenten Datenzugriffen kommen kann, die durch die Nebenläufigkeit von Multicore-Systemen entstehen. Die Kopplung mit Bauhaus ermöglicht die Kombination von Code-Analyse mit der modell-/simulationsbasierten Analyse, um den Entwickler von Multicore-Systemen schneller und zielgerichteter auf die wichtigsten Probleme in seinem System hinzuweisen.

Hiermit wurden folgende Ziele aus der VHB abgedeckt:

- AP5.1: Die Integration der modellbasierten Timing-Analyse mit einer codebasierten Data-Race-Analyse stellt einen Beitrag zur durchgängigen, domänenübergreifenden Werkzeugplattform dar, die ein wichtiges Problem von Multicore-Systemen (Data Races) adressiert.

- AP5.2: Die kombinierte Behandlung von Timing- und Data-Race-Analyse stellt eine Methodik zum effizienteren Aufspüren von Data-Races dar.

Symtavision hat zudem eine Analyse der Vorhersagbarkeit von Virtualisierungslösungen durchgeführt. Diese kam zu dem Schluss, dass eine Vorhersagbarkeit generell möglich ist, wenn entsprechende Kriterien (insb. temporale Isolation) berücksichtig werden und die benötigten Informationen zur Virtualisierungs-architektur zur Verfügung stehen.

Hiermit wurden folgende Ziele aus der VHB abgedeckt:

- AP4.4: Die Untersuchung der Analysierbarkeit von Virtualisierungslösungen hat gezeigt, unter welchen Bedingungen diese für timing-kritische Anwendungen eingesetzt werden können.

11.1.2 Timing-Analyse von Ethernet-Netzwerken

Bei der Analyse von Multicore-Systemen und –Gateways wurde schnell klar, dass diese eine entsprechend schnell Anbindung benötigen. Ethernet wird zukünftig (da besteht kaum Zweifel) in allen Automobilen eine zentrale Rolle spielen. Insbesondere die Einführung von Multi-Core-Architekturen stellt erhöhte Anforderungen an die Kommunikations-Datenrate, welche sich mit Ethernet erfüllen lassen.

Symtavision hat daher die Timing-Analyse-Methoden im Rahmen von ARAMiS auf Ethernet ausgeweitet. Abbildung 14 zeigt die Modellierung von Ethernet in Symtavisions Timing-Analysewerkzeug SymTA/S.

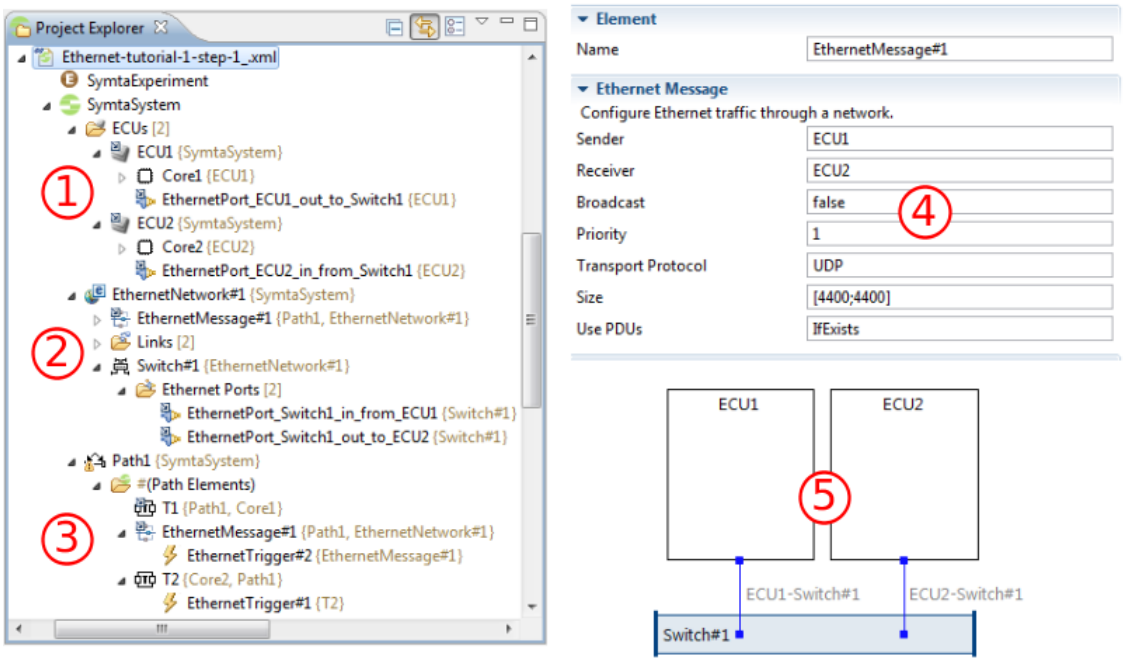

Abbildung 14 Ethernet-Modellierung in SymTA/S

Diese Ethernet-Analyse unterstützt neben Standard-Ethernet nach IEEE-802.1Q auch Ethernet-AVB nach IEEE-802.1Qav sowie Time-Triggered Ethernet nach SAE Standard AS6802, sodass auch die Echzeit-relevanten Standards für durch eine erhöhte Vorhersagbarkeit, was für Regelungs-Anwendungen sehr wichtig ist, unterstützt werden.

Ein Nebenergebnis ist zudem die Erweiterung der Timing-Analysemethoden um die Analyse von LIN-Bussen. Hierdurch können Systeme nun komplett modelliert und analysiert werden, was insbesondere für Ende-zu-Ende-Betrachtungen essentiell ist.

Hiermit wurden folgende Ziele aus der VHB abgedeckt:

- AP4.3: Die entwickelte Timing-Analyse von Ethernet-Netzwerken ermöglicht eine Analyse von Timing-Verhalten von für Multicores essentiellen Verbindungsnetzen. Hierdurch wurde eine Technik bereitgestellt zum Entwurf wesentlich robusterer Systeme.
- AP2.1: Durch die Integration der Ethernet-Analyse mit der Multicore-Analyse zu einer Systemanalyse wird der effiziente Entwurf von Netzwerk- und Systemarchitektur ermöglicht.

11.1.3 Weitere Ergebnisse

Die Timing-Analyse wird stark vereinfacht durch die Integration in bestehende Prozesse und Modelle, für welche Symtavision entsprechende Schnittstellen bereitgestellt hat: Unterstützung von AUTOSAR inkl. Multi-Core-Import, sowie der Import von Multicore-Traces aus Messungen auf der Zielplattform sind hier die wichtigsten Ergebnisse.

Weiterhin hat Symtavision zur Koordination von Methoden in der AUTOSAR Timing User Group mitgewirkt und sich in ARTE Meetings eingebracht. Hierdurch konnten die Erkenntnisse von ARAMiS direkt in den entsprechenden Gremien genutzt werden.

Hiermit wurden folgende Ziele aus der VHB abgedeckt:

- AP5.1: Die Integration mit insb. AUTOSAR-Modellen ermöglicht die Integration der Timing-Analyse in eine durchgängige, domänenübergreifenden Werkzeugplattform.
- AP5.2: Symtavision hat im Rahmen von AUTOSAR Methoden zur Analyse des Timings von Multicore-Systemen beigesteuert.

Ein wichtiges Nebenergebnis von ARAMiS ist die Verbesserung der Analyse-Performanz des Timing-Analysewerkzeuges. Hier wurde im Rahmen von ARAMiS eine Vervielfachung der Analysegeschwindigkeit bei gleichzeitiger starker Reduktion des Speicherverbrauchs erzielt. So sind die entwickelten Analysemethoden auch in großen Multicore- und Ethernet-Systemen einsatzfähig.

Hiermit wurden folgende Ziele aus der VHB abgedeckt:

- TP6: Die Performanzverbesserungen haben die Arbeit an Demonstratoren (auch mit Beispielen von Kunden) mit realistischen Problemgrößen ermöglicht und stellen sicher, dass die Ergebnisse (aller Beiträge von Symtavision) in der Praxis eingesetzt werden können.

Weiterhin wurde ein Engineering Projekt-Server angeschafft, damit große Analyse (mit realistischen Problemgrößen) mit den entwickelten Analysemethoden durchgeführt werden konnten.

Reisekosten wurden für die Teilnahme an Projektmeetings sowie die Vorstellung von Ergebnissen auf Konferenzen aufgewendet.

11.2 Notwendigkeit und Angemessenheit der Arbeiten

Die Förderung im Rahmen von ARAMiS hat Symtavision ermöglicht, Lösungen zu entwickeln, die es aus eigener Kraft nicht hätte umsetzen können. Die eingesetzten Personalaufwände

waren notwendig, um die geplanten Ziele zu erreichen, also die Entwurf, Implementierung, Test der Lösungen durchzuführen, aber auch die Markteinführung vorzubereiten.

11.3 Fortschritte auf dem Gebiet des Vorhabens

Die Timing-Analyse von Multi-Core-Systemen wird mittlerweile auch von anderen Firmen angeboten. Simulationsbasierte Verfahren werden von Inchron (chronSIM) und TimingArchitects (TA Simulator) angeboten. Diese setzen z.T. auf eine detailliertere Modellierung als die von Symtavision entwickelte Lösung, bieten aber weder die Möglichkeit, ganze Systeme zu analysieren, noch die Möglichkeit einer Worst-Case-Analyse.

Auf dem Gebiet der Ethernet-Timing-Analysen gibt es mittlerweile das Tool Pegase von RealTime-at-Work, welches ähnlich wie die von Symtavision entwickelte Lösung die Simulation und Analyse von Ethernet-Netzwerken ermöglicht. Diese bieten jedoch nicht die Möglichkeit, ganze Systeme inkl. Steuergeräte und anderer Busse (CAN, FlexRay) zu analysieren.

11.4 Veröffentlichung der Ergebnisse

Im Rahmen von ARAMiS konnte Symtavision einige wissenschaftliche Publikationen platzieren (siehe Zwischenberichte) und somit öffentlich zugänglich machen (Auszug):

[1] "Safeguarding Development of Real-Time Software from Plan to Integration", Dr. Simon Schliecker, Embedded Platforms Conference at "Electronica", 2014, Munich

[2] "Partitioning vs. protection - What's "better" for mixed-criticality HW/SW architectures?", Dr. Kai Richter, Safetronic – Functional Safety for Automotive, 2014, Stuttgart, Germany

[3] "Exploiting shaper Context to improve Performance Bounds of Ethernet AVB Networks", Axer, Philip and Thiele, Daniel and Ernst, Rolf and Diemer, Jonas, 51st Design Automation Conference (DAC), 2014, San Francisco, USA

[4] "What's the Bus Load? Real-Time Metrics for Automotive Ethernet Networks", Dr. Simon Schliecker, Jonas Diemer, Dr. Kai Richter, ERTS², 2014, Toulouse, France

[5] "Ethernet Performance Metrics for In-Vehicle Applications", Dr. Simon Schliecker, Embedded World, 2014, Nürnberg

[6] Presentation/Paper: "Avoiding Risks in First-Generation Multi-Core Designs through Timing-Aware Software Development", Dr. Simon Schliecker, Embedded World, 2014, Nürnberg

[7] "Real-Time Metrics for Automotive Ethernet Architectures", Jonas Diemer, Kai Richter, Simon Schliecker, 14. Internationales Stuttgarter Symposium, 2014, Stuttgart

[8] "Echtzeit-Bewertung von Ethernet-Konfigurationen", Dr. Kai Richter, Dr. Simon Schliecker, Jonas Diemer, Hanser Automotive Networks: Special 2013

[9] “E/E and Software Development Process for Timing-Aware Design”, Dr. Simon Schliecker, Maurice Sebastian, ATZ extra 10/2013

[10] "Gemeinsam echtzeitfähige Ethernet-Architekturen im Fahrzeug schaffen", Dr. Kai Richter, VDI Elektronik im Fahrzeug, Baden-Baden

[11] "Software architecture methods and mechanisms for timing error and failure detection according to ISO 26262: Deadline vs. Execution Time Monitoring", Dr. Kai Richter, Safetronic, Stuttgart

[12] "Echtzeit-Softwareentwicklung von der Planung bis zur Integration", Dr. Simon Schliecker, ESE Congress, Sindelfingen

[13] "Quantifying the Timing Quality of Ethernet-based Network Configurations - What is the “Bus Load” of my Ethernet network?" Simon Schliecker, Jonas Diemer, Bosch Ethernet Day 2013

[14] "Mit Timing-Modellen sicher auf Multi-Core umsteigen – erst virtuell, dann richtig!", Simon Schliecker, Kai Richter, Hanser automotive 10/2013

[15] "Software Architecture Methods and Mechanisms for Timing Error and Failure Detection According to ISO 26262: Deadline vs. Execution Time Monitoring", Marek Jersak, Karsten Schmidt, SAE World Congress 2013, Detroit, USA

[16] "Design guidelines for ISO 26262 compliant software architectures", Kai Richter, CTI conference on ISO 26262, Detroit, USA

[17] "Leveraging Multicore Design Options for Safety-Critical Automotive Applications", Dr. Kai Richter, Embedded Word 2013

12 SYSGO AG

12.1 Wissenschaftlich-technische Ergebnisse

12.1.1 Überblick des Beitrags von SYSGO

SYSGO hat das Echtzeitbetriebssystem PikeOS ins ARAMIS-Projekt eingebracht, welches ein zertifizierter Hypervisor für harte Echtzeitsysteme ist, der Partitionen zur Verfügung stellt, in denen unterschiedlich kritische Software untergebracht werden kann. Jede Partition ist von jeder anderen prinzipiell sowohl bzgl. Zeit als auch bzgl. Speicher und anderer Ressourcen, abgetrennt. Jedwede Kommunikation zwischen Partitionen muss in der PikeOS-Konfiguration feingranular freigeschaltet werden, etwa wenn Partitionen miteinander arbeiten wollen. Ebenso muss jede Zugriffsmöglichkeit auf Speicher, Geräte-Hardware oder CPU-Zeit vorkonfiguriert werden. Die Konfiguration ist zur Laufzeit fest, d.h. PikeOS garantiert, dass es keine Erweiterungen der Rechte zur Laufzeit gibt. Diese harte Partitionierung ist das wichtigste Element bei der Kombination verschiedenkritischer Software auf einem einzigen System.

PikeOS bietet seit Version 3.2 Unterstützung von SMP-Hardware an. Diese äußerst komplexe Unterstützung wurde und wird kontinuierlich weiterentwickelt, indem wir in Forschungsprojekten die Probleme genauer untersuchen, um Lösungen zu finden. Das Hauptproblem sind zwischen Kernen geteilte Ressourcen wie Speicher, Caches, Busse, Geräte. Im ARAMIS-Projekt haben wir einige neue Aspekte beleuchtet und an Lösungen gearbeitet.

Innerhalb der PikeOS-Partitionen sind sowohl harte Echtzeitprogramme ausführbar, etwa unter Benutzung der nativen PikeOS-Benutzerschnittstelle (*Personality*) oder auch ARINC653- oder POSIX-Programme, als auch komplette komplexe Betriebssysteme wie Linux, die paravirtualisiert werden, und sich um Aufgaben kümmern können, welche typischerweise weniger kritisch für den Betrieb des Systems sind.

SYSGO hat im ARAMIS-Projekt in TP2, TP3, TP4 und TP6 mit den anderen Projektpartnern zusammengearbeitet. Unser Hauptfokus lag als Software-Hersteller auf TP4. In TP2 und TP3 haben wir die Kooperation mit anderen Partnern genutzt zur Analyse von Problemen und Kommunikation, um den Informationsfluss in unsere TP4-Aufgaben zu gewährleisten, etwa bei der Diskussion über mögliche systemweiten Lösungen oder über die Rolle der Hardware vs. Software. In TP6 haben wir Demonstratoren mit PikeOS unterstützt. Wie habe dazu mehrere Workshops besucht, PikeOS geliefert und bei der Integration geholfen. Ebenso wurde ein Workshop organisiert zur Einführung in PikeOS. In TP6 war es für uns wichtig, Feedback zurückfließen

zu lassen in unsere Arbeitspakete in TP4, auch um evtl. Anpassungen liefern zu können für die Integration in die Demonstratoren.

12.1.2 Portierungen

In TP4 haben wir PikeOS auf im Projekt benutzte Hardware-Plattformen portiert. Diese Arbeit umfasste die Unterstützung der Sabre-Boards der ARM-i.mx6-Familie, die hauptsächlich im Automotive-Bereich benutzt wurden, wie auch die Anpassung von PikeOS an die PowerPC QorIQ-Familie mit den Vertretern P4080, P2041 und P5040, sowie auch die Portierung von PikeOS auf das FPGA-basierte Xilinx Zynq 7000 Zedboard. Zur Unterstützung der TU Braunschweig wurde PikeOS auf die Sparc LEON3-basierte IDAMC-Plattform portiert.

Die Ports wurden nach der Fertigstellung den Partnern zur Verfügung gestellt und sind zur Verwertung in unser PikeOS-Produkt zurückgeflossen.

12.1.3 Synergien zwischen Sicherheitsstandards

Um die Zertifizierbarkeit zu erhöhen bzw. die Effizienz des Zertifizierungsprozesses zu verbessern, haben wir mehrere Sicherheitsstandards auf Synergien hin untersucht: den IT-Sicherheitsstandard CommonCriteria, die Funktionssicherheits-standards IEC61508, DO178b, sowie IEC62433. Außerdem wurde ein Top-Down- mit einem Bottom-Up-Fehlermodell für PikeOS erstellt und verglichen. Hieraus haben wir wertvolle Erkenntnisse für zukünftige Zertifizierungen gewonnen. In diesem Zusammenhang haben wir auch Kontakt zu einer Prüfstelle und Zertifizierungsstelle aufgenommen.

12.1.4 Speicherbandbreitenbegrenzung

Wir haben auf dem P4080 ein neuartiges System zur Separierung von verschiedenkritischen Partitionen auf SMP-Systemen entwickelt. Ein Theoriepapier wurde zusammen mit EADS-IW, Cassidian und AbsInt veröffentlicht. Die zugrundeliegende Beobachtung ist, dass moderne SMP-Systeme keine direkte Messung von Laufzeitbeeinflussungen von einem auf den anderen Kern erlauben. Die wäre nötig, um bei parallel ausgeführten Partitionen zu bestimmen, wie sich die Ausführungszeit durch die gegenseitige Beeinflussung auf Bussen, Caches, usw. ändert, um darauf dynamisch zu reagieren, etwa, indem die weniger kritische Partition gedrosselt wird, um einer höherkritischen Partition die Erledigung der Aufgabe zu garantieren. Während diese direkte Messung nicht möglich ist, haben moderne Mikrocontroller Hardwarezähler, die Zugriffe auf den Speicher zählen. Da Speicherzugriffe im Normalfall die Laufzeit dominieren, war unsere Idee, aus dieser Information mittelbare Schlüsse auf die Beeinflussung der Laufzeit zu ziehen, um entsprechend damit die Drosselung der minderkritischen Partitionen vorzunehmen.

SYSGOs Aufgabe in der Kooperation war es, PikeOS um eine Unterstützung von den relevanten Performance-Zählern zu erweitern, was wir im Laufe des Projektes für die QorIQ P4080-Hardware umgesetzt haben. Darauf aufbauend wurde die Zeitpartitionierung des PikeOS-Kerns erweitert, so dass sie bei Überläufen den Speicherbandbreitenbudgets die entsprechende Partition abbrechen, bzw. einen Fehler auslösen kann. Mit diesem Ansatz wird es erstmals denkbar, parallel ablaufenden verschiedenkritischen Partitionen auf einem SMP-System zu vereinen.

12.1.5 Grafikvirtualisierung

SYSGO hat im ARAMIS-Projekt an der Virtualisierung von Grafikhardware gearbeitet. Bei dem Zugriff auf die Grafikhardware hat man das Problem, dass die Beschleunigungshardware, welche nötig ist, um gute Performance bei der Darstellung zu erlangen, für unterschiedlich kritische Systeme freigeschaltet werden soll, ohne jedoch eine Beeinflussung der kritischeren durch minderkritische Partitionen zu erleiden.

Hierzu wurde OpenGL virtualisiert, um interferenzfrei verschiedenen Partitionen Zugriff auf die OpenGL-Grafik zu erlauben. Mehrere Partitionen greifen auf OpenGL zu, und eine Zwischenschicht tunnelt die Zugriffe auf die eine zugrundeliegende Hardware. Das System macht es grundsätzlich möglich, parallel auf die Hardware zuzugreifen, und ebenso, bestimmte Bildbereiche oder Ebenen zuzuordnen. Allerdings bietet die heutige Hardware keine Schutzfunktionen hierzu, so dass das aktuelle Verfahren seine Grenzen hat, z.B. wenn Shader-Programme auf die Hardware geschrieben werden müssen. Die Prüfung, dass solche komplexen Programme die Hardware nicht über Gebühr belegen, ist nicht effizient und nicht in 100% der Fälle möglich. Hier wünschen wir uns als Softwarehersteller eine entsprechende Hardwareunterstützung in zukünftigen Architek-turen.

Das System wird nach dem Ende des ARAMIS-Projektes weiter verwertet, um unser Produkt für entsprechende Anwendungen anbieten zu können.

12.1.6 TPM- und HSM-Module für Secure-Boot

Für Security-Anwendungen hat SYSGO einen I2C-Treiber ent-wickelt, über den TPM und HSM ins PikeOS-System eingebunden wurden.

Darauf aufbauend wurde Secure-Boot auf den entsprechenden Plattformen implementiert. Ebenso wurde Secure-Boot auf der QorIQ-Plattform in Betrieb genommen. Auch diese Erweiterungen plant SYSGO in kommende Produktversionen aufnehmen

12.1.7 Security-Audit-Infrastruktur

Eine Security-Audit-Architektur wurde entwickelt, mit der sich PikeOS ab Systemstart auditieren lässt. Auch diese Erweiterung steht für kommende Produkte zur Verfügung.

Es wurden Synergien mit Tracing entdeckt, die wir im Produkt weiter ausbauen wollen, etwa in Form eines Lightweight-Tracing-Systems. Andererseits hatten die Security-Erweiterungen auch eigene Anforderungen, etwa, dass vom frühesten Start-Zeitpunkt des PikeOS-Systems die Audit-Dienste verfügbar sein sollen.

12.1.8 Fastboot

In einigen Anwendungsbereichen, etwa in einem Automobil-System, gibt es die Anforderung, Teiles des Systems sehr schnell hochzufahren, während andere, komplexe Teile länger zum Hochfahren brauchen dürfen.

Zur Integration auf einem einzigen Mikrocontroller haben wir einen Fastboot-Mechanismus weiterentwickelt, damit PikeOS in sehr kurzer Boot-Zeit Grundfunktionen zur Verfügung stellt, etwa für CAN-Kommunikation, während komplexe Teile, etwas eine Rückfahrkamera-Funktion, erst später hinzukommen.

12.1.9 Dynamische Ressourcenverteilung

Für dynamische Aufgaben haben wir PikeOS um eine verbesserte Umschaltmöglichkeit von Zeitpartitionsschemata auf SMP-Systemen erweitert. Diese Systeme können nun auf allen Kernen synchronisiert das Zeitpartitionsschema umschalten, wenn zur Laufzeit eine entsprechende Reaktion gefordert ist.

12.1.10 Isolation von Peripheriegeräten

Zur verbesserten Isolation von Peripherie in SMP-Systemen hat SYSGO auf dem PowerPC QorIQ Unterstützung der PAMU entwickelt. Die ist eine MMU-artige Hardware-Unterstützung, mit der DMA-fähige Peripherie-Geräte virtualisiert und deren Speicherzugriffe kontrolliert werden können. Die Implementierung ergänzt die in PikeOS schon vorhandene IOMMU-Implementierung unseres x86-Ports. Die QorIQ-Implementierung soll ebenfalls in unser Produkt einfließen.

12.1.11 Unterstützung anderer Projektpartner

Im ARAMIS-Projekt haben wir einen Workshop zur Einführung in PikeOS veranstaltet, an dem die beteiligten und interessierten Projektpartner teilgenommen haben. Dieser zweitägige Workshop hat in Mainz stattgefunden. Der Workshop fand in Kooperation mit OpenSynergy statt, die Im direkten Anschluss den dritten Tag mit einer Einführung ins AUTOSAR-Betriebssystem Coqos gegeben haben, welches als Hypervisor PikeOS verwendet.

SYSGO hat im weiteren Verlauf an mehreren Workshops verschiedener Arbeitspakete teilgenommen.

12.2 Notwendigkeit und Angemessenheit der Arbeiten

Das ARAMIS-Projekt erfüllt zentrale förderpolitische Ziele des Förderprogrammes „IKT 20120 – Forschung für Innovationen“ des BMBF. Dabei sichert ARAMIS Deutschland eine Führungsposition durch den Einsatz von IKT in komplexen Multicore-Systemen, welche auf einer Kombination von Elektronik und Software als Basistechnologien aufbauen.

Der Einsatz von Multicore-Technologien bieten vielfältige Chancen, das ansonsten physikalisch begrenzte Wachstum der Leistungsfähigkeit elektronischer Systeme weiterzuführen. Hier hat ARAMIS einen Beitrag geleistet, diese System besser zu verstehen, und Lösungen zu entwickeln, wie mit den Schwierigkeiten solch komplexer Systeme umzugehen ist, insbesondere mit Vorhersagbarkeit von Echtzeitsystemen, die für die Zertifizierung im Bereich der Funktions- wie der IT-Sicherheit unerlässlich ist.

SYSGO hat in diesem Bereich das ARAMIS-Projekt nutzen können, um an den schwierigen Themen bei der Virtualisierung von Multicore-Architekturen weiterzuarbeiten. Unsere Ergebnisse zeigen uns, dass das Projekt in diesem Bereich gute Fortschritte ermöglicht hat. Das begründet sich wesentlich darin, dass die Mittel effizient genutzt wurden und im Projektkonsortium eine sehr gute Kommunikation und Zusammenarbeit in den komplexen Problemszenarien möglich war.

12.3 Fortschritte auf dem Gebiet des Vorhabens

Als Verbesserungen des Stands der Technik, die SYSGO erreicht hat, ist zunächst das speicherbandbreitenbasierte Scheduling zu erwähnen, welches es zum ersten Mal denkbar macht, auf Multicore-Systemen mehrere verschiedenkritische Applikationen parallel auszuführen. Dieses Ergebnis haben wir zusammen mit EADS-IW, Cassidian und AbsInt in einer wissenschaftlichen Veröffentlichung zusammenfassen können.

Weitere Fortschritte ergeben sich aus SYSGO-Sicht aus dem besseren Verständnis der Beziehung von Funktions- und IT-Sicherheit, die wir in mehreren Teilprojekten erarbeitet haben im Zusammenhang mit TSP und HSM. Diese Erkenntnisse werden wir in zukünftigen Zertifizierungen benutzen können.

Auf dem Gebiet der Grafikvirtualisierung haben wir durch das ARAMIS-Projekt nun besser verstanden, welche Teile der Virtualisierung Software effizient handhaben kann, und für welche Teile eine Hardware-Unterstützung wünschenswert wäre. Zudem haben wir gezeigt, dass Lösungen überhaupt möglich sind.

12.4 Veröffentlichung der Ergebnisse

SYSGO veröffentlicht seine Ergebnisse als Industriepartner in erster Linie durch Gespräche und Vorträge auf Industriemessen, damit die Veröffentlichung einen Kundenbezug erhält, aus dem wir uns natürlich kommerzielle Erfolge versprechen. Als große Messen ist die jährliche „Embedded World“ in Nürnberg für uns sehr wichtig, an der wir jedes Jahr teilnehmen. Ein weiterer wichtiger Kontakt- und Veröffentlichungspunkt sind unsere jährlichen „SYSGO-Days“, auf denen wir auch Vorträge über den Stand unserer Forschungsprojektarbeit halten. Neben diesen zwei wichtigsten Veranstaltungen gab es zahlreiche weitere Messen, auf denen wir mit potentiellen Kunde über ARAMIS und seine Ergebnisse gesprochen haben.

Daneben hatten wir im ARAMIS-Projekt das Glück, zusammen mit EADS-IW und AbsInt wissenschaftliche Arbeiten zu veröffentlichen, ebenso wie zusammen mit Industrie-Partnern Berichte über die Arbeit. Im Einzelnen gab es folgende Veröffentlichungen.

[1] Jan Nowotsch, Michael Paulitsch, Daniel Buhler, Henrik Theiling, Simon Wegener, Michael Schmidt: *Multi-core Interference-Sensitive WCET Analysis Leveraging Runtime Resource Capacity Enforcement*. ECRTS 2014: 109-118

[2] Selma Said, Rolf Ernst, Sascha Urig, Benoit Dupont de Dinechin, Henrik Theiling: *The Shift to Multicores in Real-Time and Safety-Critical Systems*, CODES+ISSS 2015, Amsterdam, Niederlande.

[3] Dominik Reinhard, et. al.: *Embedded Virtualization Approaches for Ensuring Safety and Security within E/E Automotive Systems*, Embedded World 2015, Nürnberg.

13 Technische Universität Braunschweig

13.1 Wissenschaftlich-technische Ergebnisse

In dem ARAMiS Projekt lagen die Schwerpunkte der TU Braunschweig auf der sicheren Kommunikation in einem Network-on-Chip (NoC) und der sicheren Anbindung von Peripherie (bspw. Speicher, externe Kommunikation), sowie der Bereitstellung von Analyse-Artefakten für die neuentwickelten Mechanismen. Die Arbeiten hierzu wurden auf mehrere Arbeitspakete des Projektes verteilt, um sowohl die Entwicklung neuer Mechanismen (TP2, TP3 und TP4) als auch die Integration zu einer Gesamtarchitektur (TP2), deren Analyse der Vorhersagbarkeit (TP5) und Umsetzbarkeit in realen Beispielen (TP6) abzudecken.

Hierbei sind ein Verfahren zur globalen Steuerung von Netzwerkzugriffen (*Resource Brokering*), ein Speichercontroller für gemischt kritische Systeme und eine Erweiterung der vorhandenen NoC Architektur auf der IDAMC Plattform entstanden. Ziel war die sichere und performante Isolation von gemischt kritischen Anwendungen in einem Multiprozessor System-on-Chip (MPSoC). Die Performance der Isolation spielte dabei eine wichtige Rolle, da nur so die effiziente Integration von vielen verschiedenen Funktionen in einem MPSoC erreicht werden kann, wie es für zukünftige Automotive, Avionik und Bahn-Systeme gefordert sein wird. Die entwickelten Verfahren wurden in der IDAMC Architektur implementiert und evaluiert. Im Rahmen von ARAMiS wurde für die IDAMC Architektur zusätzlich eine Implementation des PikeOS Betriebssystems der Firma Sysgo geschaffen. Durch die erfolgreiche Umsetzung der Mechanismen in einer Gesamtarchitektur und einer prototypischen Implementierung, sowie den Analysemethoden zum Beweis der Vorhersagbarkeit, erfüllen die Ergebnisse das zentrale Ziel von ARAMiS. Das zentrale Ziel war hierbei die Erstellung von Konzepten, welche den sicheren Einsatz von Multicorearchitekturen in den Mobilitätdomänen Automobil, Avionik und Bahn ermöglichen, um die Sicherheit, Verkehrseffizienz und den Komfort zu erhöhen. Die folgenden Absätze geben einen Überblick über die geleisteten Arbeiten. Eine detaillierte Beschreibung dieser ist dem Abschlussbericht und den Ergebnisdokumenten zu entnehmen.

Zu Beginn des Projektes wurde in dem Teilprojekt TP2 eine Analyse des aktuellen Stands der Technik in Bezug auf Netzwerkarchitekturen, deren Eignung für sicherheits-relevante eingebettete Systeme und den Einfluss gemeinsam genutzter Ressourcen erstellt. Dies beinhaltete auch diverse Sicherheitskonzepte (primär in Bezug auf ein sicheres

Zeitverhalten) und Monitorarchitekturen, welche in einem System integriert werden können, um die Vorhersagbarkeit des Zeitverhaltens sicher zu stellen. Die initiale Analyse zeigte dabei, dass bereits vorhandene Konzepte einige Nachteile haben und sich nicht im vollen Umfang für Multiprozessorsysteme mit Sicherheitsanforderungen in eingebetteten Systemen einsetzen lassen. Der wichtigste Punkt hierbei war die effiziente Umsetzung von Kapselungsmechanismen, um Garantien für das Zeitverhalten geben zu können. Alle betrachteten Verfahren nutzten statische Annahmen über das Systemverhalten und somit auch statische Mechanismen. Da das Verhalten eines Systems bzw. dessen Anwendungen aber hoch dynamisch ist, mussten starke Überabschätzungen des Verhaltens gemacht werden. Dies resultierte in Architekturen, welche zwar im schlimmsten Fall (worst-case) das gewünschte Verhalten zeigten, jedoch im Normalfall einen Großteil der Systemressourcen nicht nutzen konnten und Leistung/Performance verschenkten. Zusätzlich fokussierten die bereits vorhandenen Ansätze nur die Vorhersagbarkeit der sicherheitsrelevanten Funktionen und vernachlässigten das Verhalten aller unkritischen Funktionen. Auf diese Weise wurden in vielen Fällen die unkritischen Funktionen unnötigerweise durch die sicherheitsrelevanten Funktionen nachteilig beeinflusst, was die Performance des Systems zusätzlich verschlechterte. Diese ersten Untersuchungen bildeten die Grundlage zur Definition erster Architekturansätze und die Anforderungen für die Arbeiten in TP3 und TP4 zur Entwicklung von Hardware- und Softwarelösungen.

Auf Grundlage dieser Untersuchungen, entwickelte die TU Braunschweig in den einzelnen TPs die oben erwähnten Konzepte, um kontrollierte Dynamik in einem System zu zulassen und neue Quality-of-Service Mechanismen zu entwickeln, welche die Benachteiligungen der unkritischen Funktionen minimieren.

Das *Resource Brokering* beschreibt dabei ein Konzept zur Steuerung der gemeinsamen Verwendung von Ressourcen, wie beispielsweise eines On-Chip Netzwerkes oder Speichers. Dazu wird in einem Netzwerk eine zusätzliche Kontrollebene integriert, wie in Abbildung 15 dargestellt. In der Abbildung existieren drei Anwendungen auf den Tiles T1-T3. Diese koordinieren ihre Zugriffe zu einem Speicher (DRAM) über einen Resource Manager (RM), um unnötige Interferenzen zu vermeiden.

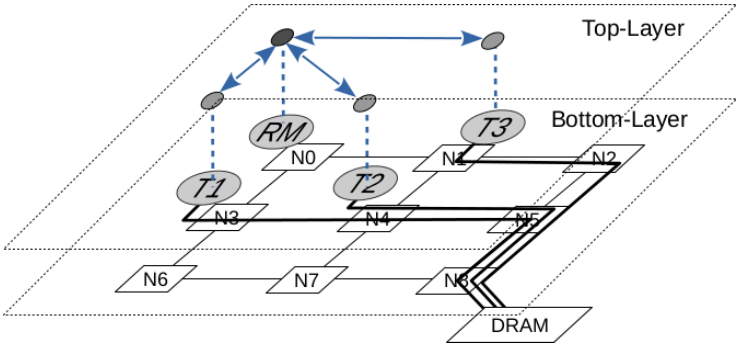

Abbildung 15 Resource Broker Layer

Die zusätzliche Kontrollebene kann dabei in Software oder Hardware umgesetzt werden. Die Umsetzung in Software erlaubt es, ein beliebiges COTS On-Chip Netzwerk zu nutzen und in diesem eine (globale) Steuerung zu integrieren. Durch die Steuerung können so Vorhersagbarkeit und Zuverlässigkeit zeitgleich mit einer verbesserten Auslastung der Ressourcen ermöglicht werden. Hierzu koordiniert die Steuerung die Zugriffe auf die Ressource derart, dass unnötige Blockierungen vermieden oder aber auf ein höheres Level verschoben werden, um eine Propagation der Blockierungen über die Ressource (z.B. im Netzwerk) zu verhindern. Das Konzept wurde in einem realen System, dem IDAMC, umgesetzt. Hierbei wurde eine Implementierung basierend auf dem PikeOS Betriebssystems zusammen mit Sysgo Germany erarbeitet.

Für On-Chip Netzwerke wurden neue Konzepte für das Bereitstellen von Quality-of-Service (QoS) erarbeitet. Hierbei wurde ein generelles Konzept für das Scheduling von Zugriffen auf geteilten Ressourcen entwickelt, welches Monitoring nutzt, um feingranular Interferenzen zu kontrollieren. Basierend auf diesen Arbeiten wurde das Netzwerkinterface und das Router-Design des IDAMC angepasst und durch eine End-zu-End Steuerung von (virtuellen) Kanälen erweitert. Hierdurch wird es ermöglicht, garantierte Antwortzeiten für sicherheitsrelevanten Verkehr zu geben, während der negative Einfluss auf die Performance von unkritischem Verkehr (verglichen mit klassischen Verfahren wie Time-Triggered Architekturen oder einfacher Priorisierung des kritischen Verkehrs) reduziert wurde. Die Grundidee ist in Abbildung 16 dargestellt. Hierbei werden für die verschiedenen QoS-Klassen (z.B. garantierte Bandbreite (GT), garantierte Latenz (GL) und keine Garantien (BE)) unterschiedliche Puffer an einer Ressource bereitgestellt. Eine Ressource kann dabei das Netzwerkinterface oder die Ports eines Netzwerkrouters sein. Entgegen dem klassischen Ansatz werden nicht die kritischen Anwendungen priorisiert, sondern die unkritischen. Damit diese nun nicht ungehindert die kritischen Anwendungen beeinflussen, überwachen Monitore die Interferenz, welche die unkritischen

Anwendungen auf die Kritischen erzeugt. Überschreitet diese Interferenz einen Schwellwert, werden die kritischen Anwendungen in dem Scheduler (MCSP) priorisiert, bis wieder ein valider Systemzustand erreicht ist.

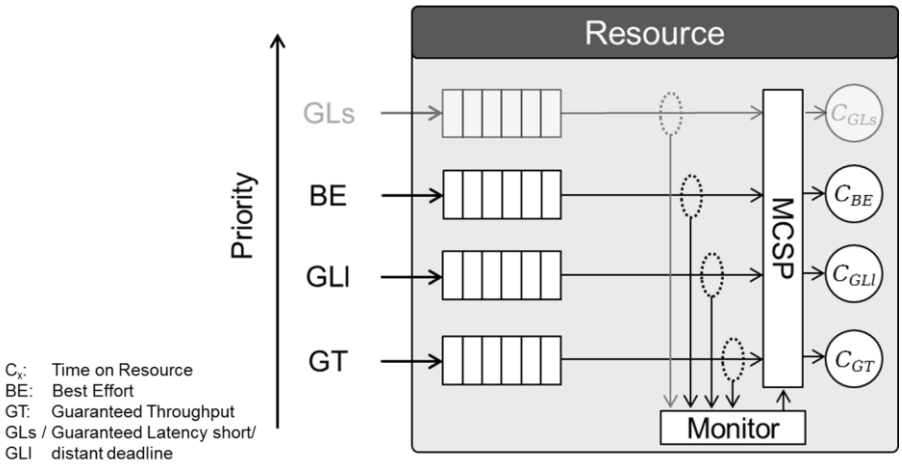

Abbildung 16 Workload basierte Priorisierung

Dieses Konzept der dynamischen Priorisierung ermöglicht es, die Performance der unkritischen Anwendungen im Vergleich zu klassischen Verfahren zu verbessern. Zeitgleich führen die Monitore zu einer begrenzten Interferenz und somit zu einer ausreichenden Isolation zwischen kritischen und nicht kritischen Anwendung (wie z.B. in der ISO 26262 gefordert), was die Vorhersagbarkeit des Verhaltens der kritischen Anwendungen ermöglicht. Der Performancegewinn für die unkritische BE Klasse ist für einige Benchmarks in Abbildung 17 dargestellt. Hierbei werden die klassische Priorisierung der kritischen Anwendungen (SP), die dynamische Umpriorisierung basierend auf simplen Zugriffszählern (SP_{rev}) und der neue Ansatz (MCSP) für die in der Grafik dargestellt Lastverteilung verglichen. Wie der Grafik zu entnehmen ist, führt die dynamische Umpriorisierung zu einer verbesserten Performanz. Zusätzlich kann durch das Verwenden von Monitoren der Performanz Gewinn im Vergleich zu simplen Zugriffszählern weiter erhöht werden. Die liegt daran, dass die Monitore die Interferenz feingranularer erfassen, als schlichte Zugriffszähler die können.

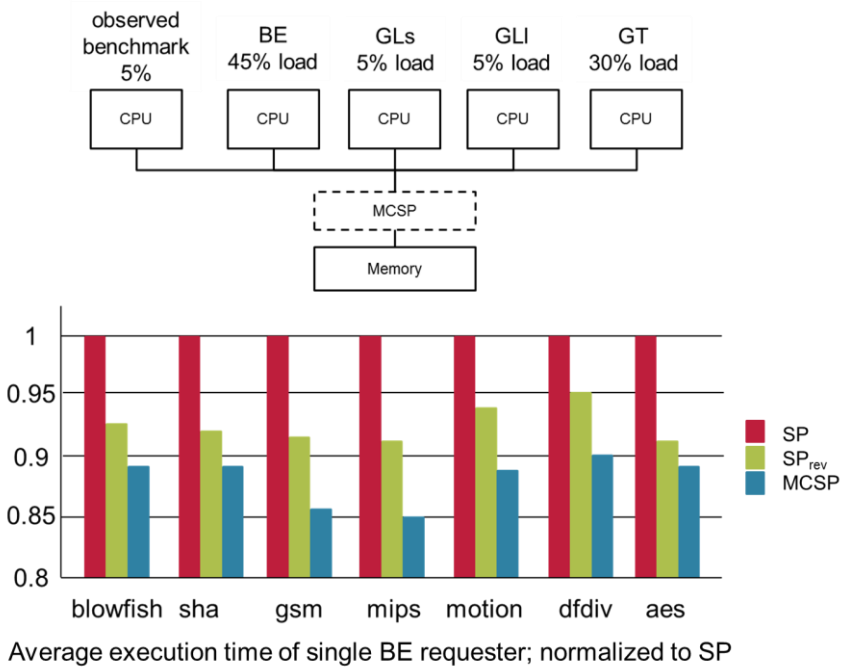

Abbildung 17 Effekt der dynamischen Priorisierung

Für die Anbindung von anderen gemeinsam genutzten Komponenten, wie DDR-Speicher, wurden verschiedene Arbitrierungsverfahren untersucht und Konzepte für vorhersagbare Ressourcen-Controller entwickelt. Diese Konzepte nutzen Monitor-Mechanismen, um effizient QoS-Garantien zu ermöglichen. Basierend auf diesen Konzepten, wurden verschiedene Konzepte für Speichercontroller entwickelt. Aus den Konzepten entstanden verschiedene Implementierungen eines vorhersagbaren Speichercontrollers für die IDAMC Plattform, welche in gemischt kritischen Systemen genutzt werden können. Die Grundarchitektur nutzt dabei eine Kombination aus Priorisierung und fester Zuweisung von Task zu Speicherbänken (bank privatization). Dabei kann eine Task eine oder mehrere Bänke zugeteilt bekommen. Durch die feste Zuweisung entstehen mehrere Partitionen im Speicher (sog. virtual devices; VD). Zwischen den einzelnen VDs wird eine Zeitsteuerung (time division multiplex; TDM) genutzt, um den Zugriff zu steuern. Auf diese Weise sind die einzelnen VDs zeitlich isoliert und damit unabhängig. Ein einfacher Scheduler für dieses Vorgehen für zwei Kritikalitätsklassen (kritisch und unkritisch) ist in Abbildung 18 zu sehen.

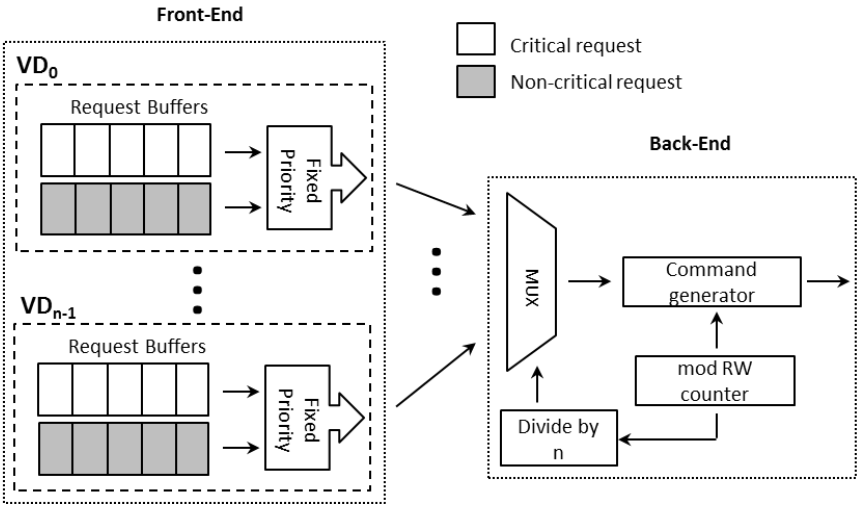

Abbildung 18 Speicher-Scheduler

Der Scheduler kann in ein Front-End und Back-End geteilt werden. Im Front-End existieren für jedes VD zwei Wartschlangen für die verschiedenen Kritikalitäten. Durch einen Arbiter mit fester Priorität werden die Anfragen an das Back-End geleitet. Dabei erhalten kritische Anfragen eine hohe Priorität. Im Back-End existiert ein Zähler, der basierend auf einer Rundengröße (RW) einen Muxer und den Command-Generator steuert. Auf diese Weise kann das Back-End einem bestimmten Muxer-Eingang feste Adressen im Speicher zuweisen, diese für einen gewissen Zeitraum zugreifbar machen und so die Isolation sicherstellen. Die Integration des Schedulers in einen Speicher-Controller ist in Abbildung 19 dargestellt. Der gesamte Speicher-Controller besteht dabei aus sechs verschiedenen Blöcken: den Ports, einer Adressübersetzung, Datenpuffern, dem Scheduler, einem Access Controller und dem Daten-Manager. Für jedes VD gibt jeweils zwei Lese- und Schreibports (für jede Kritikalität jeweils einen). Der kritische Port wird dabei einer einzelnen Task zugewiesen, wobei der nicht kritische Port auch von mehreren Anwendungen geteilt werden kann. Da der kritische Port eine höhere Priorität hat, hat das Verhalten der Tasks am nicht kritischen Port bzw. derer Anzahl keinen Einfluss auf das Verhalten der kritischen Task. Die Adressübersetzung wandelt logische Adressen in ein für den Speicher benötigtes Format. Die Übersetzen Adressen werden zusammen mit den Daten in den Datenpuffern zwischengehalten, bis diese vom Scheduler zur Verarbeitung gewählt werden. Die tatsächliche Verarbeitung (bzw. Weiterleitung) der Daten erfolgt dann über den Daten Manager, welcher die Zugriffe und Antworten zum/vom Speicher den richtigen Puffern zuweist. Damit die Verarbeitung (Scheduling, Pufferzuweisung) unabhängig vom jeweiligen Speichertyp geschehen kann, regelt der Access Controller die physikalische Kommunikation mit dem Speicher. Die Evaluation und mehr Details zu dem Speichercontroller können

dem Ergebnisdokument E3.4.5.2 oder dem Projekt-Abschlussbericht entnommen werden.

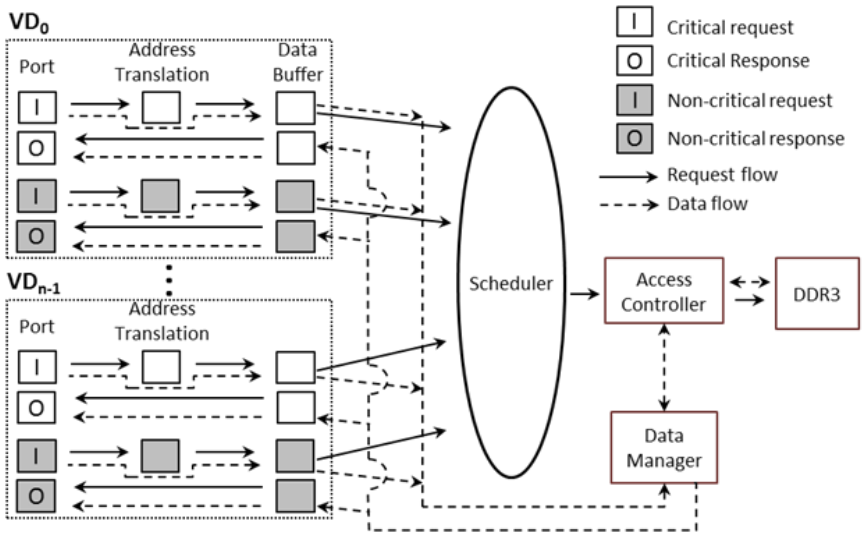

Abbildung 19 Speicher-Controller

Nebst dem Entwickeln der einzelnen Mechanismen in TP3 und TP4, wurden diese zusammen in eine Systemarchitektur integriert (TP2) und als Ganzes in der IDAMC Architektur umgesetzt (TP6). Auf diese Weise entstand eine prototypische Implementierung einer auf HW und SW Co-Design basierenden Architektur, welche zeigt, dass und wie verschiedenste Mechanismen in zukünftigen Systemen kombiniert werden müssen, um ein sicheres aber zeitgleich performantes Systemverhalten zu erzeugen. Die grundlegende Architektur der Plattform ist in Abbildung 20 dargestellt.

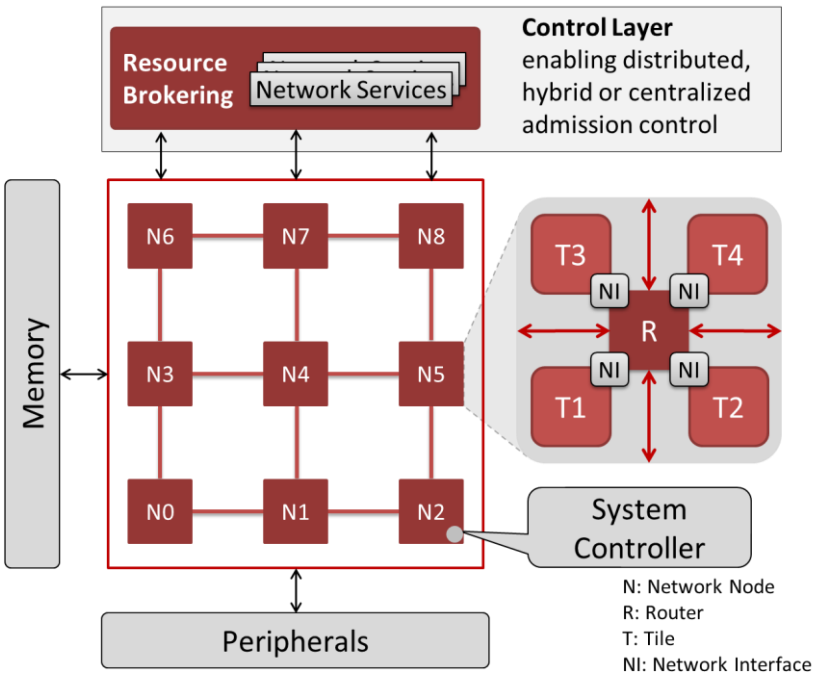

Abbildung 20 IDAMC - Netzwerk-Architektur

Die Plattform fasst die Architekturkonzepte des Kontroll-Layers, der QoS-Router, der QoS-Ressourceninterface und der Netzwerkinterface zusammen. Das Zusammenspiel dieser Mechanismen ermöglicht das effiziente und dynamische Bereitstellen von Quality-of-Service Garantien für bestimmte Applikationen oder Prozessorkerne im System. Der Kontroll-Layer ermöglicht es, die jeweiligen Zugriffskontrollen und QoS-Mechanismen der Interfaces zum Netzwerk und Ressourcen, sowie in den Routern, dynamisch zur Laufzeit anzupassen. Eine detaillierte Beschreibung und Evaluation ist im Abschlussbericht und Ergebnisdokumenten zu finden.

Neben der Entwicklung der Mechanismen und deren Zusammenführung zu einem Gesamtarchitekturkonzept für gemischt kritische Multicore Systeme, wurden Konzepte und Methodiken für die Beweisbarkeit der Vorhersagbarkeit erstellt (TP5). Hierzu wurden formale Analysemethoden für alle von der TU Braunschweig vorgeschlagenen Mechanismen entwickelt. Zusätzlich wurde der Einfluss der Mechanismen auf das Gesamtsystem untersucht, um so eine Systemlevel-Analyse zu ermöglichen. Die Analysemethoden wurden in dem pyCPA Framework zur kompositionellen Performanz Analyse umgesetzt. Auf diese Weise können die Eigenschaften der vorgeschlagenen Mechanismen in Bezug auf die Vorhersagbarkeit formal bewiesen werden.

Eine ausführliche Beschreibung aller Arbeiten ist dem Abschlussbericht zu entnehmen.

13.2 Notwendigkeit und Angemessenheit der Arbeiten

Das ARAMiS Vorhaben beschäftigte sich mit der sicheren Nutzung der Multicore Technologie für Embedded Systems und erfüllt somit zentrale förderpolitische Ziele des Förderprogramms „IKT 2020 – Forschung für Innovationen“ des BMBF. Dabei schaffen die Ergebnisse von ARAMiS die wissenschaftlichen, technischen und wirtschaftlichen Grundlagen für den Erhalt und zum weiteren Ausbau der führenden Position der deutschen Wissenschaft und Industrie.

Durch die von der TU Braunschweig entwickelten Software- und Hardware-Architekturvorschläge, sowie deren formalen Analysemethoden, ist es möglich, insbesondere im Schwerpunktthema Embedded Systems, einen größeren Entwurfsraum in kürzerer Zeit zu evaluieren. Die entwickelten Mechanismen, insbesondere das dynamische Verwalten von Ressourcen in einem Network-on-Chip (Resource Brokering) und die Isolation von verschiedenen Kritikalitäten durch die Netzwerkrouter und Interface, tragen durch den modularen und isolierten Aufbau dazu bei, zukünftig zuverlässige und analysierbare Plattformen zu entwickeln. Durch den Einsatz einer modularen Architektur, zusammen mit einer kompositionellen, formalen Analyse, ist ein schnellerer und ressourcenschonenderer Forschungs- und Entwicklungsprozess ermöglicht. Dadurch wird die, im BMBF-Programm geforderte, Steigerung von Entwicklungsproduktivität und Produktqualität erreicht. Die Ergebnisse bzw. Mechanismen wurden in einer Prototypenplattform implementiert und evaluiert, was die generelle Anwendbarkeit und Umsetzbarkeit der Ergebnisse von ARAMiS beweist. Des Weiteren werden bereits Ergebnisse in ersten Kooperationen mit Unternehmen und anderen Forschungsprojekten außerhalb des Projektes eingesetzt, was zusätzlich zeigt, dass das ARAMiS Projekt bzw. die geleisteten Arbeiten ein Erfolg waren.

13.3 Fortschritte auf dem Gebiet des Vorhabens

Es sind keine Ergebnisse von dritter Seite bekannt geworden, die an dieser Stelle hervorzuheben wären.

13.4 Veröffentlichung der Ergebnisse

Während der Projektlaufzeit wurden Veröffentlichungen und studentische Abschlussarbeiten erstellt, um die Ergebnisse des Projekts der (wissenschaftlichen) Öffentlichkeit zu präsentieren. Zusätzlich sind derzeit einige Veröffentlichungen in Vorbereitung, welche direkt auf die Ergebnisse von ARAMiS aufbauen.

Bisher erfolgte Veröffentlichungen:

[1] IDAMC: A NoC for Mixed-Criticality Systems
RTCSA 2013, Teipei, Taiwan, 20.08.2013,
Konferenzvortrag

[2] Mixed Criticality Aware Memory Controller for NoCs
Nano-Tera/ARTIST International Summer School 2013, Aix-les-Bains, France, 09-13.09.2013
Poster

[3] Supervised resource sharing in NoCs
Nano-Tera/ARTIST International Summer School 2013, Aix-les-Bains, France, 09-13.09.2013
Poster

[4] Supervised Sharing of Virtual Channels in Networks-on-Chip
Nano-Tera/ARTIST International Summer School 2013, Aix-les-Bains, France, 18-20.06.2014
Konferenzvortag

[5] A Mixed Critical Memory Controller Using Bank Privatization and Fixed Priority Scheduling
RTCSA 2014, Chongqing, China, 20-22.08.2014
Konferenzvortrag

[6] ARAMiS Project Booth
EMC2 Consortium Conference, Oldenburg, Germany, 30.09.-01.10.2014
Poster

[7] Workload-aware shaping of shared resource accesses in mixed-criticality systems
Codes+ISSS 2014, New Delhi, India, 12-17.10.2014
Konferenzvortrag

[8] Real-Time DRAM Throughput Guarantees for Latency Sensitive Mixed QoS MPSoCs
SIES, 2015, Siegen, Germany, 08-10.06.2015
Konferenzvortrag

[9] Flexible TDM- Based Resource Management in On-Chip Networks
RTNS2015, 04-06.11.2015
Konferenzvortrag

Aktuell laufende Veröffentlichungen, die im direkten Zusammenhang mit den in ARAMiS gewonnen Ergebnissen stehen:

[1] Slack-Based Resource Arbitration for Real-Time Networks-On-chip,
Date 2016, Dresden, Germany, 14-18.03.2016
Konfernezvortrag

[2] Dynamic Admission Control for Real-Time Networks-On-Chips
ASP-DAC 2016 Macao SAR, China, 25-28.01.2016
Konferenzvortrag

[3] Predictable and Dynamic Traffic Rate Control for Networks-on-Chips
DAC 2016, Austin, TX, USA
Konferenzvortrag in Einreichung

[4] Handling Large Granularity DRAM Requests in FPGA-Based Networks-on-Chip
DAC 2016, Austin, TX, USA
Konferenzvortrag in Einreichung

[5] Efficient Latency Guarantees for Mixed-criticality Networks-on-Chip
ISCA 2016 Seoul, Korea
Konferenzvortrag in Einreichung

Studentische Abschlussarbeiten, die im Rahmen des ARAMiS Projektes entstanden sind:

Datum	**Titel**	**Art**
04.2013	Implementation, Integration and Testing of a Mixed-criticality Aware Predictable Memory Bank Access Scheduler for Many-core Processors	Masterarbeit
10.2014	Implementation, integration and testing of a mixed-criticality aware, predictable sharing of a graphic processing unit for many-core processors	Masterarbeit
11.2014	Implementation, integration and testing of a mixed-criticality aware, predictable network-on-chip router for many-core processors	Masterarbeit
03.2015	Dynamische Verwaltung von virtuellen Kanälen im IDA NoC Netzwerk Interface	Bachelorarbeit

14 Technische Universität Kaiserslautern

14.1 Wissenschaftlich-technische Ergebnisse

Die TU Kaiserslautern war mit zwei Arbeitsgruppen an ARAMiS beteiligt, die inhaltlich weitgehend unabhängig voneinander an verschiedenen Themengebieten gearbeitet haben. Insgesamt fanden alle Arbeiten der TU Kaiserslautern in enger Zusammenarbeit mit dem Fraunhofer IESE statt.

Seitens der Arbeitsgruppe „Software Engineering: Dependability“ (Prof. Liggesmeyer) wurde inhaltlich im Wesentlichen an zwei Themenkomplexen gearbeitet: Modellierung und Simulation (im Rahmen von AP4.1) und Safety und Multicore (im Rahmen von AP4.3). Die geringen Aufwände in AP2.3 sind hierbei als Brücke von der Systemebene zu AP4.3 zu verstehen.

In AP4.1 wurde zusammen mit dem Fraunhofer IESE an einer Erweiterung des hauseigenen FERAL Simulationsframeworks gearbeitet. Basis hierfür war das Compilerframework LLVM und SystemC. Hierbei wurde der LLVM Interpreter LLI ähnlich wie ein Instruction Set Simulator verwendet; der verwendete Befehlssatz entspricht hierbei der LLVM-IR (intermediate representation), einem assembler-ähnlichen Zwischencode, der allerdings noch auf komplexen Datenstrukturen wie structs oder arrays operiert. Da die LLVM-IR auch Metadaten enthält die die Beziehung zum ursprünglichen Hochsprachen-Quellcode (etwa C, C++ oder Java) erhält kann man durch Überwachung der Code-Interpretation in LLI Analysen wie Statement- oder Branch-Coverage durchführen ohne den Code instrumentieren zu müssen.

Die SystemC Anbindung wurde so realisiert das Variablenzugriffe in LLI abgefangen und in SystemC TLM Transaktionen übersetzt wurden. Hierzu wird LLI zusätzlich zum LLVM-IR Code eine Tabelle mit Variablennamen und zugehörigen Adressen übergeben. So wird der eigentliche Speicherzugriff dann über die erzeugte Transaktion mit der passenden Adresse im SystemC Modell realisiert; wiederrum ohne den Code instrumentieren zu müssen. Technisch wurde die Anbindung mittels Java Native Access realisiert.

Die ursprüngliche Idee bezüglich der SystemC Anbindung war dabei über Simulation und auf Basis von LLVM-IR Code Architektur-Evaluation im SystemC Modell zu betreiben, also z.B. die Speicherhierarchie zu modellieren und etwa verschiedene Cache-Strategien bzgl. ihres Multicore-Verhaltens zu evaluieren. Die konkreten Speicheradressen können hierbei aus späteren

Kompilierungsschritten (also von LLVM-IR zum Maschinencode der Zielarchitektur) extrahiert werden.

Allerdings hat sich herausgestellt das die Anzahl der Speicherzugriffe zwischen LLVM-IR und dem Zielcode sich hierfür doch zu sehr unterscheiden. Hierfür wurde auf Basis von Benchmark-Code die Anzahl der Speicherzugriffe basierend auf LLVM-IR mit der Anzahl der Speicherzugriffe im weiter auf verschiedene Zielarchitekturen (z.B. x86, ARM, PowerPC) kompilierten Code verglichen. Hierbei ergaben sich teilweise erhebliche Unterschiede dahingehend das im Maschinencode mehr Zugriffe gemacht wurden (bis zu einem Faktor von 4, in Einzelfällen sogar mehr), insbesondere wenn Compileroptimierung verwendet wurde.

Dies betrifft aber nicht Zugriffe auf Kontrollregister (z.B. von Peripherie); diese sind unabhängig von Zielarchitektur und Compileroptimierung. Daher haben wir uns in der Anwendung unseres Ansatzes auf die Modellierung und Simulation von I/O Zugriffen konzentriert (siehe Abbildung 21). Hierzu wurde, um Regelkreise mit analogen Komponenten modellieren zu können, auch SystemC AMS verwendet (in Abbildung 21 der „Motor“ Block). Gerade für eingebettete Systeme ist dies wichtig, da deren Eingaben oft von vorherigen Ausgaben abhängen – am offensichtlichsten wenn Regelkreise implementiert werden. Eine Code-Coverage Analyse von derartiger eingebetteter Software benötigt daher auch ein Modell der Regelstrecke für ein realistisches Ein-/Ausgabe Verhalten. Durch den Wechsel des Fokus von Architekturevaluation zu I/O Verhalten haben wir die Dokumentation dieser Arbeiten auch von E4.1.1.4 nach E4.3.2.1_E4.3.3.1 verlegt.

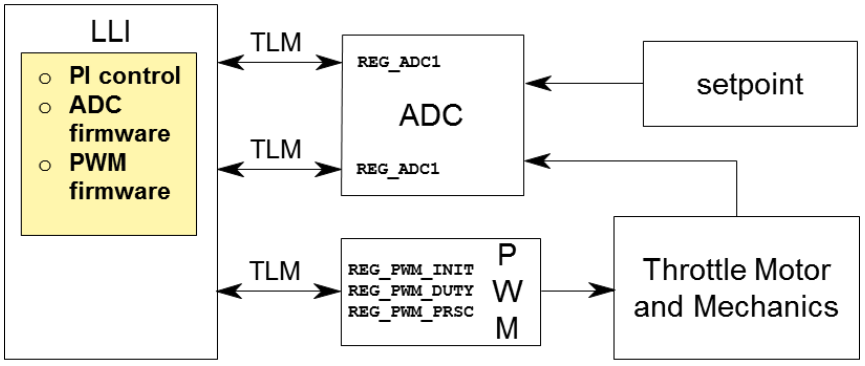

Abbildung 21 Beispielsystem: PWM-Regelung eines Motors

Die Arbeiten in AP4.3 wurden unter Federführung des Fraunhofer IESE durchgeführt und hatten das Ziel, die entscheidenden multicore-spezifischen Elemente zu identifizieren, die bei der Entwicklung eines *sicherheitskritischen* multicore-basierten Systems zu beachten sind. Hierzu wurden die wesentlichen

Schritte bei der Zertifizierung bzw. Qualifizierung eines Systems anhand eines abstrakten Multicore Systems betrachtet:

- Im Rahmen der *Risikoanalyse* wurde untersucht welche multicore-*spezifischen* Fehler allgemein in einem Multicore System auftreten können. Da die Basis hierfür ein abstraktes System ohne Anwendungskontext war, entspricht dies nicht genau einer klassischen Risikoanalyse bei einem konkreten System; insbesondere wurden keine Kritikalitäten zugeordnet. Diese Analyse hat Eingang in D4.4, E4.3.1.1 gefunden.
- Auf der Risikoanalyse aufbauend wurde (zusammen mit mehreren Projektpartnern) ein allgemeines Multicore-Sicherheitskonzept entwickelt. Hierbei handelt es sich um eine Sammlung von Maßnahmen zur Vermeidung verschiedener multicore-spezifischer Fehlerszenarien; insbesondere bezüglich I/O-Sharing und temporaler Segregierung. Die Ergebnisse sind in D4.4, E4.3.2.1_E4.3.3.1 zu finden.
- Als letzter Schritt folgt der Safety-case, also der Nachweis der Wirksamkeit des Sicherheitskonzepts. Da auch hier (ähnlich wie bei der Risikoanalyse) ein allgemeiner Safety-case für ein abstraktes Multicore-System nicht möglich ist, wird zunächst das generelle Vorgehen der Erstellung eines Safety-cases mit der goal structuring notation (GSN) besprochen, und dann anhand eines Beispiels demonstriert (siehe D4.4, E4.3.4.1).

Des Weiteren hatte die TU Kaiserslautern seit Anfang 2013 zusammen mit Sysgo die Leitung des Teilprojekts 4 (Software) inne. Hierbei war insbesondere das Ziel, über die reine administrative Arbeit hinaus auch inhaltlich in dem doch recht großen TP 4 zu einem Konsens zu kommen. Dies ist unserer Meinung auch gelungen in dem Sinne das trotz der weit gefächerten Themen in TP 4 (Safety, Security, Applikation, Betriebssysteme und Virtualisierung) sich doch ein gemeinsames Verständnis zur Problematik des Einsatzes von Multicore-Systemen in sicherheitskritischen Anwendungen entwickelt hat, und es auch weitgehend Einigkeit über die Vor- und Nachteile entwickelter und bestehender Lösungen und Lösungsansätze gibt.

Die Arbeiten der Arbeitsgruppe „Software Engineering: Processes and Measurement Research“ (Prof. Rombach) waren in AP0.3 „Empirische Bewertung der methodischen Projektergebnisse“ und TP5 “Durchgängige Entwicklungsmethodik und Anbindung an RTP” angesiedelt.

In TP0.3 wurde das Fraunhofer IESE (Tasklead) bei der Durchführung von Studien unterstützt. Dabei wurden mehrere Studien direkt von uns betreut und durchgeführt. Dies waren

überwiegend Studien in TP4, wodurch die Überlappungen zu den Arbeiten des Fraunhofer IESE minimiert wurden. Zwischen diesen beiden Partnern wurden die empirischen Kompetenzen, der Status der Studien bzw. der Unterstützungsarbeiten periodisch ausgetauscht. Des Weiteren wurde die ARAMiS Abschlussumfrage von uns konzipiert und umgesetzt. Eine Zusammenfassung aller empirischen Ergebnisse ist in D0.2 zu finden.

Das Hauptergebnis in TP5 ist das von uns implementierte Process Configuration Framework (PCF). Das PCF implementiert vom Fraunhofer IESE entwickelte Konzepte zur Verknüpfung von verschiedenen Methoden, Tools und Techniken auf Basis eines methoden-Repository. Das PCF ist web-basiert, und wurde verwendet um die Durchgängigkeit der Methoden innerhalb des Projektes aufzuzeigen. So wurden teilweise Projektergebnisse zusammen mit Industriepartnern (z.B. LLI) eingefügt und die Durchgängigkeit der Methoden evaluiert. Mit diesen Arbeiten wurde gezeigt, wie sich Methoden, Tools oder Technologien untereinander, aber auch in existieren Entwicklungsprozesse und Tools durchgängig integrieren lassen.

14.2 Notwendigkeit und Angemessenheit der Arbeiten

Der Förderumfang war im Wesentlichen auf die Beschäftigung eines wissenschaftlichen Mitarbeiters pro Arbeitsgruppe für die Projektlaufzeit ausgelegt, was für eine sinnvolle Beteiligung an einem solchen Verbundprojekt letztlich das Minimum darstellt.

Die Arbeiten im Bereich Safety lagen im Kernthema von ARAMiS, und waren insbesondere notwendig um die wesentlichen Sicherheitsherausforderungen von Mehrkernprozessoren zu klassifizieren. Mit den Simulationsarbeiten wurden Wege aufgezeigt, Sicherheitskonzepte für Mehrkernprozessoren simulativ zu evaluieren.

Empirie ist notwendig um den Nutzen der in ARAMiS entwickelten Methoden, Szenarien, etc. zu zeigen. Der Umfang ist angemessen, da zusammen mit dem IESE alle Evaluationen in dem Projekt (und damit aller einzelnen Partner) unterstützt wurden.

14.3 Fortschritte auf dem Gebiet des Vorhabens

Es sind während der Projektlaufzeit keine Ergebnisse von dritter Seite bekannt geworden, die an dieser Stelle hervorzuheben wären.

14.4 Veröffentlichung der Ergebnisse

[1] Phillip Diebold, Constanza Lampasona, Davide Taibi: “Moonlighting SCRUM: an agile method for distributed teams with part-time developers working during non-overlapping hours”, in Proceedings of The Eighth International Conference on Software Engineering Advances (ICSEA 2013), October 2013

[2] Jasmin Jahic, Thiyagarajan Purusothaman, Markus Damm, Thomas Kuhn, Peter Liggesmeyer, Christoph Grimm: “Automatic Test Coverage Measurements to support Design Space Exploration”, First International Workshop on Design Space Exploration of Cyber-Physical Systems (IDEAL) 2014, Springer

[3] Phillip Diebold, Laurent Dieudonné, Davide Taibi: “Process Configuration Framework Tool”, in 39th Euromicro Conference on Software Engineering and Advanced Applications, 2014.

[4] Davide Taibi, Valentina Lenarduzzi, Laurent Dieudonné, Christiane Plociennik: “Towards a Classification Schema for Development Technologies: an Empirical Study in the Avionic Domain”, in International Journal on Advances in Software, vol 8, No. 1 & 2, 2015

15 Technische Universität München (EISEC, LIS, INSEC)

15.1 Wissenschaftlich-technische Ergebnisse

Es wurden Konzepte zur hardwareunterstützten Virtualisierung von CAN Controllern erarbeitet. Diese erlauben eine direkte Einbindung der Kommunikationsschnittstelle in Virtuelle Maschinen ohne zusätzlich den Hypervisor zu involvieren (siehe Abbildung 22). Basierend auf den Konzepten wurde ein Prototyp entwickelt, der im ARAMiS Demonstrator für „Virtualized Car Telematics - VCT“ zum Einsatz kommt. Im Demonstrator wurden virtualisierte Linux-Gastsysteme über die etablierte SocketCAN Schnittstelle an den virtualisierten CAN Controller angebunden und das Gesamtsetup evaluiert.

Die Konzepte wurden außerdem erweitert, um Virtualisierung auch auf Netzwerkebene einsetzen zu können. Des Weiteren wurden Lösungen zur Integration von neuen Netzwerktechnologien, insbesondere AVB Ethernet erarbeitet.

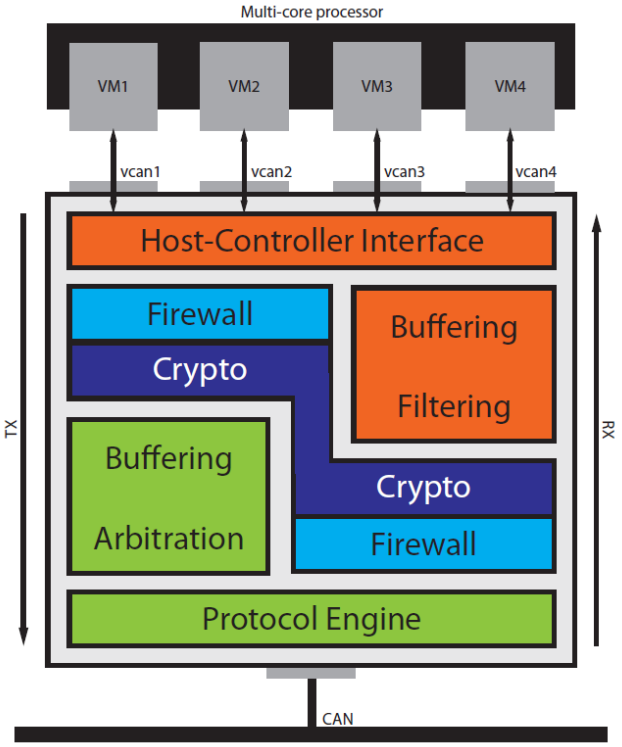

Abbildung 22 Gesamtarchitektur virtualisierter CAN Controller mit Security Erweiterungen

Darüber hinaus wurde die Anbindung von virtualisierten I/O Geräten in Multicore Steuergeräten untersucht. Insbesondere x86 Plattformen, wie sie z.B. in Infotainment-Systemen zum Einsatz kommen können, standen im Fokus. Als prototypischer Aufbau wurde eine Xeon-x86 Plattform mit speziell ausgewählten Komponenten kompiliert, welche Hardwareunterstützung für CPU-, Speicher- sowie I/O-Device-Virtualisierung bereitstellen. Durch den Einsatz dieser Hardware wurde die Grundvoraussetzung für latenzarme Virtualisierung im Embedded-Bereich erfüllt.

Untersuchungen am Prototyp haben gezeigt, dass mutwillig erzeugte Interferenzen zwischen verschiedenen Partitionen von Multicore-Prozessoren dennoch zu Performance-Problemen führen können. Wie in Abbildung 23 dargestellt, wurden Monitoring-Ansätze konzeptioniert und implementiert, die das Problem durch Kombination von Hardware- und Software-Erweiterungen eindämmen und somit vorhersagbare I/O Performance ermöglichen. Der Themenkomplex Virtualisierung von Netzwerkschnittstellen wird im partnerübergreifenden Abschlussbericht in Kapitel II/9 detailliert beleuchtet.

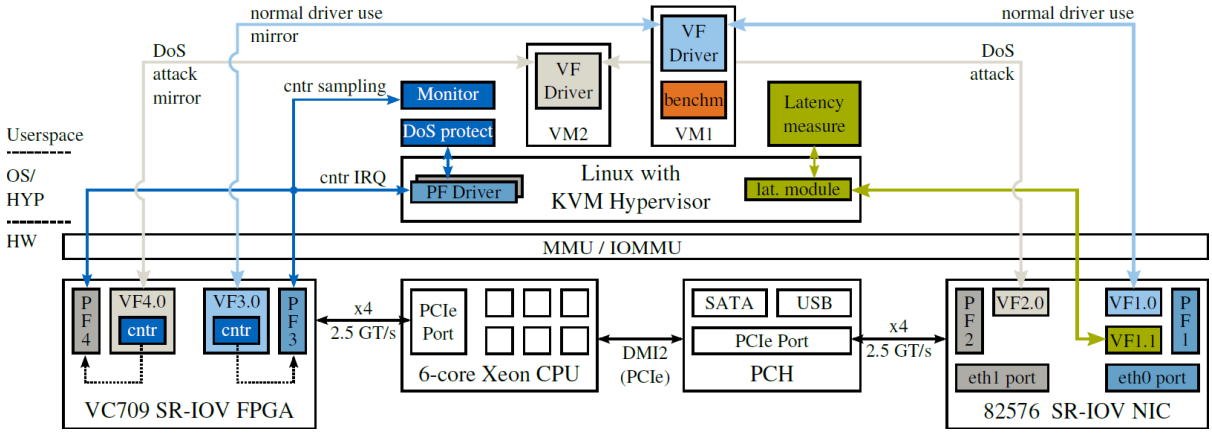

Abbildung 23 Prototypischer Aufbau zu Performance-Zählern im I/O Device am Beispiel einer Netzwerkkarte

Als Grundlage für die Entwicklung der Sicherheitsarchitektur wurde eine systematische Bedrohungsanalyse durchgeführt. Mögliche Angriffsvektoren wurden durch die sogenannte Attack Tree Methode modelliert. Die Wurzel eines Baumes repräsentiert dabei ein Angriffsziel wie z.B. einen kryptographischen Schlüssel. Für einen erfolgreichen Angriff ergeben sich so verschiedene Pfade, die durch geeignete Maßnahmen abgesichert werden müssen. Die Auswahl der Maßnahmen wird durch eine Bewertung der möglichen Risiken unterstützt. Hierfür wurde im Projekt das Common Vulnerability Scoring System (CVSS) angewendet. Für die Abwehr von Angriffen werden verschiedene Maßnahmen ausgearbeitet:

Um die Sicherheit gegen Angriffe über Kommunikationsschnittstellen in virtualisierten Systemen sicherzustellen, wurden zwei Konzepte entwickelt: Erstens wurde ein Authentifizierungs- und Verschlüsselungssystem für CAN Kommunikation entwickelt wie in Abbildung 22 dargestellt. Zweitens wurde eine Kombination aus Firewall und Intrusion Detection entwickelt, die unerlaubte Zugriffsmuster detektiert und unterbindet.

Kryptographische Operationen sowie die Speicherung von Schlüsseln werden durch ein Hardware Security Module (HSM, siehe Abbildung 24) bereitgestellt. Im Projekt wurde ein Konzept für solch ein HSM erarbeitet, das in einer Multicore-Umgebung eingesetzt werden kann und das die Anforderungen der Mobilitätsdomänen abdeckt. Dabei wurde besonders die Kompatibilität zu bestehenden Industriestandards und Software-Bibliotheken beachtet. Um die Integrität der ausgeführten Systemsoftware sicherstellen zu können, sieht das Konzept die Unterstützung von Trusted Boot vor. Das HSM wurde prototypisch auf einer FPGA Plattform implementiert und in den „Virtualized Car Telematics - VCT“ Demonstrator integriert.

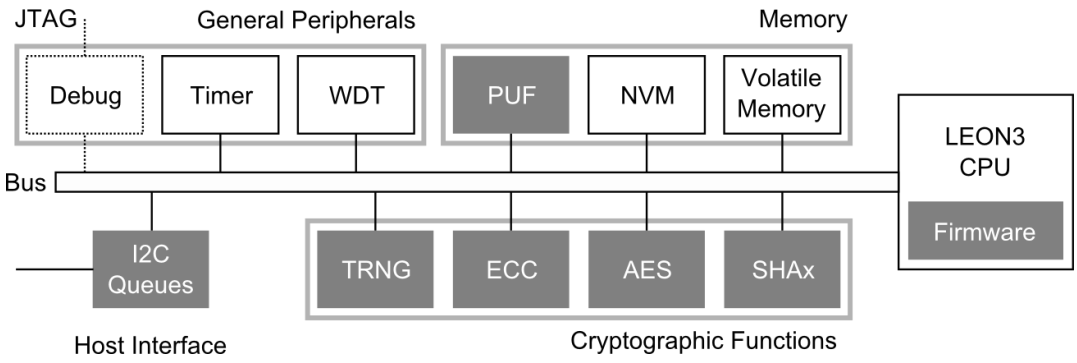

Abbildung 24 Interner Aufbau des ARAMiS HSMs

Zur Laufzeit wird das Systemverhalten mit Hilfe eines verteilen Security-Monitors protokolliert (siehe Abbildung 25). Insbesondere werden dabei die CPU-Auslastung, Speicherzugriffe und der Netzwerkverkehr überwacht. Dies erlaubt das Erkennen von Anomalien die auf einen Angriff hinweisen können. Das Thema Security wird im Partnerübergreifenden Abschlussbericht in Kapitel II/11 eingehend diskutiert.

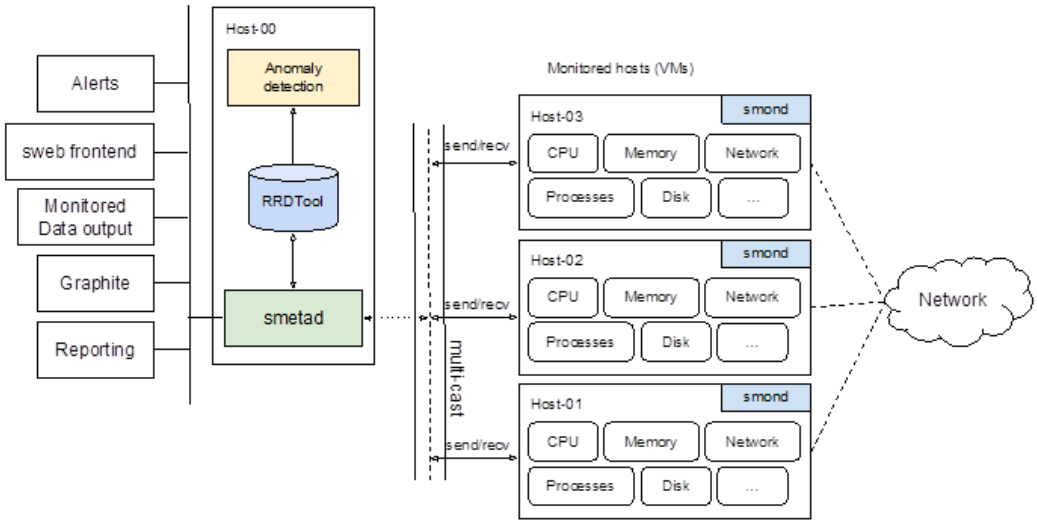

Abbildung 25 Architektur des Security Monitors

15.2 Notwendigkeit und Angemessenheit der Arbeiten

Von der TU München wurden im Projekt ARAMiS neuartige Konzepte zum Einsatz von Multicore Systemen als Grundelemente von cyber-physischen Systemen in den Mobilitätsdomänen untersucht und prototypisch entwickelt.

Ein Schwerpunkt lag auf dem Erkennen und der Abwehr von Cyber-Angriffen durch verschiedene Maßnahmen auf Hardware- und Software-Ebene. Dieses Thema hat einen sehr großen Einfluss auf die Anwendbarkeit und den zukünftigen wirtschaftlichen Erfolg von cyber-physischen Systemen, bei der der Aspekt Sicherheit von zentraler Bedeutung ist.

Der zweite große Themenkomplex befasste sich mit der Virtualisierung von Hardware-Schnittstellen in einem Multicore Szenario und den damit verbundenen Konsequenzen für die Echtzeitfähigkeit und Zuverlässigkeit. Um für den Übergang auf Multicore-Technologie gerüstet zu sein, ist es für die deutsche Industrie sehr wichtig, sich eingehend und rechtzeitig mit den damit verbundenen Problemen zu beschäftigen. Zum einen ist es unter Berücksichtigung einer zu erwartenden Verdrängung von Einzelprozessoren vom Markt entscheidend, funktionale und nicht-funktionale Eigenschaften von Applikationen auf Multicore-Plattformen weiterhin gewährleisten zu können. Gleichzeitig sollten aber auch mögliche Vorteile durch die weitergehende Integration nutzbar gemacht werden. Nur so ist es möglich, dass die deutsche Industrie mit der Weiterentwicklung in der Prozessortechnologie schritthalten kann und neue Technologien in den Mobilitätsdomänen eingesetzt werden können. Dies ist ein

wichtiger Beitrag zur Sicherstellung der langfristigen Innovationsfähigkeit des Wirtschaftsstandtortes Deutschland.

15.3 Fortschritte auf dem Gebiet des Vorhabens

Mit den erzielten Ergebnissen wurden wichtige Beiträge zum Einsatz von Multicore-Prozessoren im verschiedenen Safety-Kritischen Mobilitätsdomänen geleistet. Mit der Weiterentwicklung der Prozessortechnologie in Richtung Multicore und dem zunehmenden Anteil von Software an der Wertschöpfung ist dies von großer Wichtigkeit für die deutsche Wirtschaft, um international wettbewerbsfähig zu bleiben.

Unter Berücksichtigung solcher technischen Weiterentwicklungen werden darauf aufbauende eingebettete bzw. cyber-physische Systeme möglich, die über einzelne Transportdomänen hinausgehende Mobilitätslösungen erlauben.

Infolge der zunehmenden Vernetzung, beispielsweise in Form solcher cyber-physischen Systeme, ist mit einer erhöhten Anzahl von Angriffen sowie mit einer Steigerung der Professionalität dieser Angriffe zu rechnen. Aus diesem Grund wurden im Projekt ARAMiS Bedrohungsszenarien für derartige Systeme systematisch analysiert. Aufbauend auf den gewonnenen Erkenntnissen wurden Technologien entwickelt, um potentielle Angriffe abzuwehren. Im Rahmen des Projektes wurden Grundlagen auf Architekturebene geschaffen, die das Thema IT-Sicherheit für cyber-physische Systeme in den Mobilitätsdomänen schon bei der Entwicklung berücksichtigen.

15.4 Veröffentlichung der Ergebnisse

[1] C. Herber, A. Richter, T. Wild, A. Herkersdorf, "Real-Time Capable CAN to AVB Ethernet Gateway Using Frame Aggregation and Scheduling", Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 61-66, March 9-13, 2015

[2] C. Herber, A. Richter, T. Wild, A. Herkersdorf, "Deadline-Aware Interrupt Coalescing in Controller Area Network (CAN)", The 11th IEEE International Conference on Embedded Software and Systems, Paris, France, 701-708, August 20-22, 2014

[3] C. Herber, A. Richter, T. Wild, A. Herkersdorf, "A Network Virtualization Approach for Performance Isolation in Controller Area Network (CAN)", The 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Berlin, Germany, 215-224, April 15-17, 2014

[4] O. Sander, T. Sandmann, V. Vu Duy, S. Bähr, F. Bapp, J. Becker, H.-U. Michel, D. Kaule, D. Adam, E. Lübbers, J. Hairbucher, A. Richter, C. Herber, A. Herkersdorf, "Hardware Virtualization Support for Shared Resources in Mixed-Criticality Multicore Systems", Design, Automation & Test in Europe Conference & Exhibition (DATE 2014), Dresden, Germany, March 24-28, 2014

[5] Richter, C. Herber, H. Rauchfuss, T. Wild, A. Herkersdorf, "Performance Isolation Exposure in Virtualized Platforms with PCI Passthrough I/O Sharing", Architecture of Computing Systems-ARCS 2014, Lübeck, Germany, (Awarded ARCS 2014 Best Paper Award), 171-182, February 25-28, 2014

[6] Herber, A. Richter, H. Rauchfuss, A. Herkersdorf, "Spatial and Temporal Isolation of Virtual CAN Controllers", Workshop on Virtualization for Real-Time Embedded Systems (VtRES) 2013, Taipei, Taiwan, 7-13, August 21, 2013

[7] Herber, A. Richter, H. Rauchfuss, A. Herkersdorf, "Self-Virtualized CAN Controller for Multi-Core Processors in Real-Time Applications", Architecture of Computing Systems-ARCS 2013, Prague, Czech Republic, 244-255, February 19-22, 2013

[8] Herkersdorf, H.-U. Michel, H. Rauchfuss, T. Wild, "Multicore Enablement for Automotive Cyber Physical Systems", Special issue of journal 'it - Information Technology' 6/2012, December, 2012

[9] M. Hiller, D. Merli, F. Stumpf, and G. Sigl, "Complementary IBS: Application SpecificError correction for PUFs," in IEEE Symposium on Hardware-Oriented Security and Trust (HOST), Jun. 2012.

[10] M. Hiller, F. De Santis, D. Merli, and G. Sigl, "Reliability bound and channel capacity of IBS-based fuzzy embedders," in NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Jun. 2012.

[11] Hiller, M.; Sigl, G.; Pehl, M., "A new model for estimating bit error probabilities of Ring-Oscillator PUFs," in Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2013 8th International Workshop on , vol., no., pp.1-8, 10-12 July 2013

[12] S. Belaid, F. De Santis, J. Heyszl, S. Mangard, M. Medwed, J.-M. Schmidt, F.-X. Standaert and S. Tillich, Towards Fresh Re-Keying with Leakage-Resilient PRFs: Cipher Design Principles and Analysis, in the proceedings of PROOFS 2013 (Security Proofs for Embedded Systems), Santa-Barbara, California, August 2013 (slides). Extended version in the Journal of Cryptographic Engineering, vol 4, num 3, pp 157-171, September 2014, Springer.

[13] H. Seuschek, P. Khurana, and G. Sigl, "HiPeC - High Performance Cryptographic Service for Heterogeneous Network-on-Chip Systems" in 13th International IFAC Conference on Programmable Devices and Embedded Systems, PDeS 2015. International Federation of Automatic Control, May 2015, pp. 43-48.

[14] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, and F. Roli, "Support vector machines under adversarial label contamination," Journal of Neurocomputing, Special Issue on Advances in Learning with Label Noise, Aug. 2014, in press.

[15] H. Xiao and C. Eckert, "Indicative support vector clustering with its application on anomaly detection," in IEEE 12th International Conference on Machine Learning and Applications (ICMLA'13), Miami, Florida, Dec. 2013.

[16] T. Kittel, S. Vogl, T. K. Lengyel, J. Pfoh, and C. Eckert, "Code validation for modern OS kernels," in Workshop on Malware Memory Forensics (MMF), Dec. 2014.

16 Technische Universität München (SSE)

16.1 Wissenschaftlich-technische Ergebnisse

TUM-SSE nahm im Projekt ARAMiS unterschiedliche Rollen abhängig vom jeweiligen Teilprojekt ein. TUM-SSE war in den Teilprojekten TP1, TP5 und TP6 aktiv.

Im Folgenden werden die wissenschaftlich-technischen Ergebnisse pro Teilprojekt beschreiben.

16.1.1 TP1 Anforderungen und Szenarien

Organisation und Abstimmung

Die TUM-SSE übernahm in diesem Teilprojekt zusammen mit AUDI die Organisation und Teilprojektleitung. Dazu wurde zu Beginn des Projekts ein Modell für die Zusammenarbeit in TP1 sowie die Struktur und Verantwortlichkeiten für die Ergebnisdokumente (D1.1, D1.2) festgelegt. Als wesentliche gemeinsame Meilensteine des Fertigstellungsgrads dieser Dokumente wurden die Versionen 0.5, 1.0 und 2.0 vereinbart, die im Mai, August und Dezember 2012 fällig werden sollten. Die geplante Interaktion der Arbeitspakete AP1.1 und AP1.2 und die Lage der Meilensteine auf der Zeitachse sind in Abbildung 26 dargestellt.

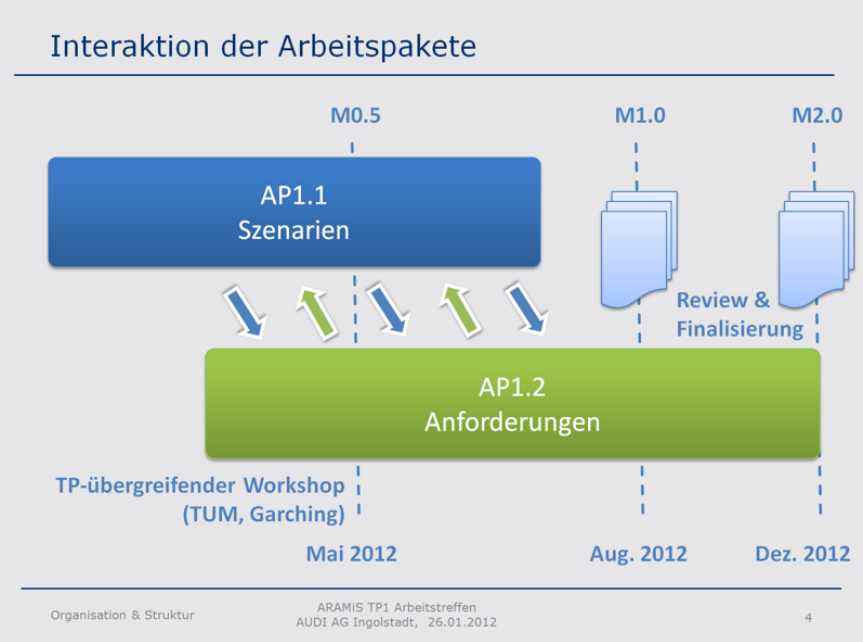

Abbildung 26 Interaktion der Arbeitspakete und Meilensteine der Deliverables

Die enge Abstimmung mit TP2 zur nahtlosen Integration und Dissemination der TP1-Ergenisse in das Gesamtprojekt sowie die frühe Abstimmung mit TP6 zur Berücksichtigung Demonstrator-relevanter Szenarien und Anforderungen wurden als zwei kritische TP-übergreifende Schnittstellen in der Projektkommunikation identifiziert. Aufgrund der beträchtlichen Größe des Projekts ARAMiS und um den dabei zu erwartenden Kommunikations- und Abstimmungsaufwand beherrschbar zu machen wurden drei Ebenen der Abstimmung und Kommunikation mit dem Projekt vereinbart:

- **TP1-Kernteam** (Setup: ca. 10 Personen aus Domänen, OEMs und Tier-1 Zulieferern, TUM-SSE, FORTISS, UniStg, TUM-LIS [Common])
- **TP1-Team** (Setup: ca. 90 Personen, alle Ansprechpartner und deren Vertretung bei den ARAMiS-Partnern, die gemäß VHB Aufwände für TP1 geplant haben)
- **Gesamtprojekt**

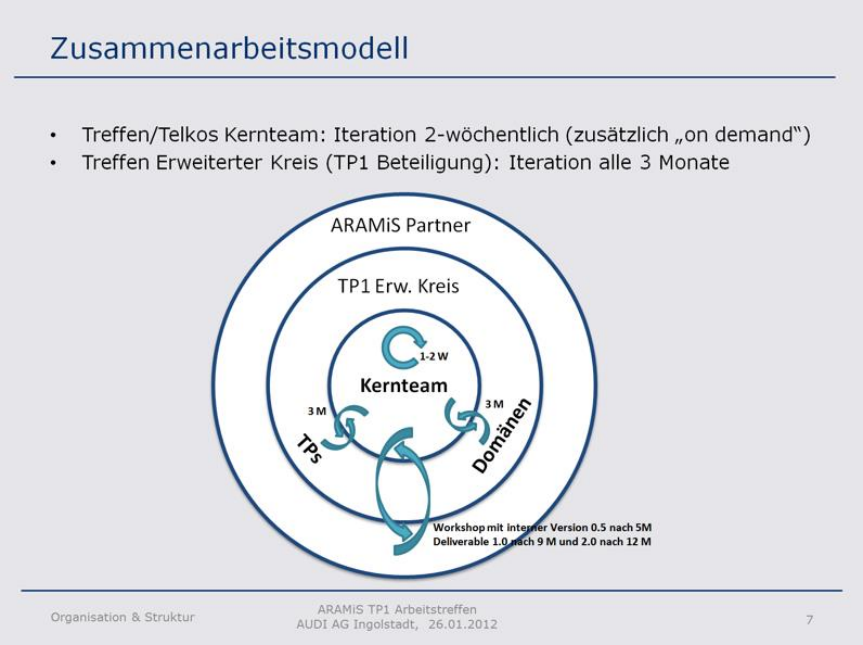

Abbildung 27 Zusammenarbeitsmodell (Ebenen und Intervalle) in TP1

Wie in Abbildung 27 dargestellt erfolgte die Abstimmung im TP1-Kernteam regelmäßig (ca. 2-wöchentlich oder wöchentlich bei Bedarf) hauptsächlich durch Telefonkonferenzen aber auch durch Workshops zu spezifischen Themen nach Bedarf. Die Abstimmung mit dem erweiterten TP1-Team und dem Gesamtprojekt erfolgt entsprechend weniger häufig.

Ferner wurde schon früh unter anderem auf Wunsch einiger OEMS und Tier-1 Zulieferer beschlossen, das Requirements Engineering in ARAMiS *modellbasiert* durch das UML/SysML Werkzeug *Enterprise Architect* durchzuführen.

Szenarien und Anforderungen Workshop

Ein wichtiger Meilenstein in TP1 war der *ARAMiS Szenarien und Anforderungen Workshop* am 23.05.2012 an der TUM am Campus Garching bei München. Dieser projektübergreifende eintägige Workshop mit mehr als 30 ARAMiS-Partnern aus Industrie und Forschung wurde von TUM-SSE Mitarbeitern organisiert und durchgeführt. Auf dem Workshop wurden die ersten Versionen von Arbeitsdokumenten (Version 0.5) sowohl aus den Domänen als auch domänenübergreifend (siehe weiter unten, CPS-Szenario) verfügbar gemacht und durch die jeweiligen Hauptautoren vorgestellt.

Content-Model und Enterprise Architect Plugin

TUM-SSE war auch inhaltlich wesentlich an TP1 durch die Erarbeitung eines ARAMiS-spezifischen sog. *Content-Models* und dessen Umsetzung im UML/SysML Werkzeug *Enterprise Architect*

(EA) beteiligt. Das Content-Model dient als gemeinsame Basis, um die Struktur der wesentlichen Artefakte beim modellbasierten Requirements Engineering für die Szenarien und Anforderungen in ARAMiS festzulegen. Während sich das Content-Model auf konzeptueller Ebene bewegt, wurde dieses durch die Implementierung eines sog. *Plugins* für das Werkzeug Enterprise Architect durch TUM-SSE Mitarbeiter für die Verwendung im Projekt ARAMiS konkret operationalisiert.

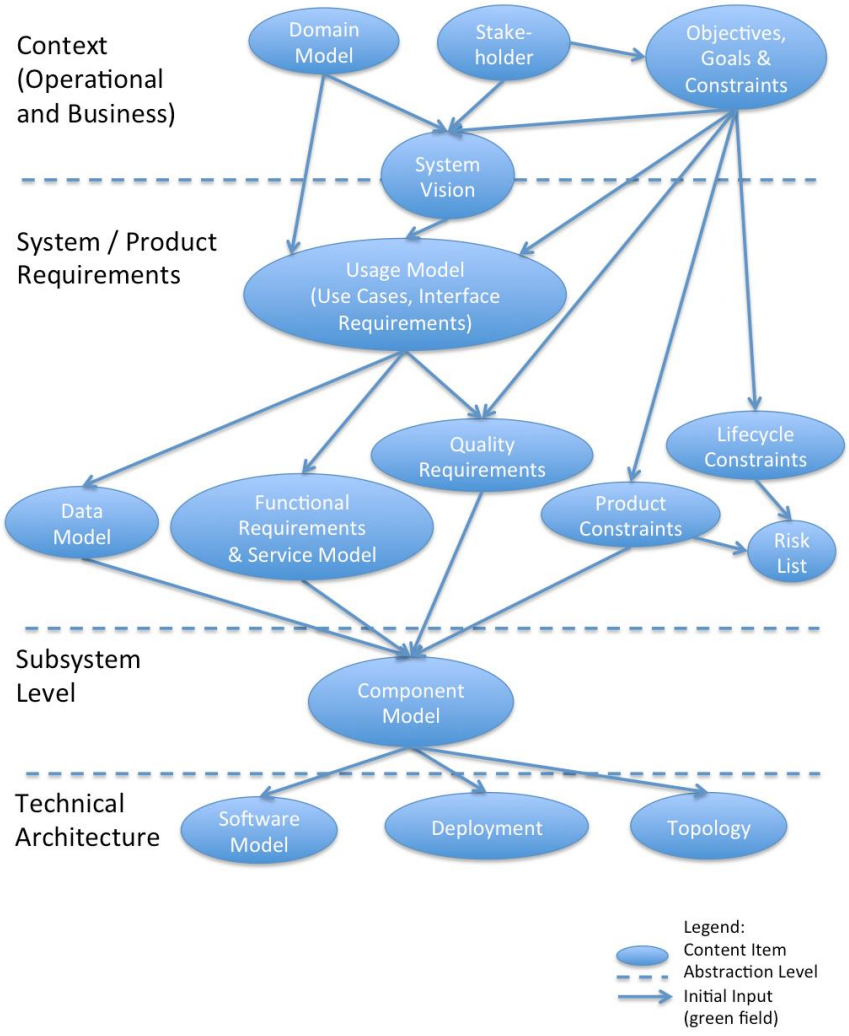

Abbildung 28 Konzeptuelles ARAMiS Content-Model

Abbildung 28 zeigt das angepasste konzeptuelle ARAMiS Content-Model und Abbildung 29 zeigt dessen Umsetzung als UML-Metamodell in Enterprise Architect. Dabei ist der obere Teil von Abbildung 28 im Fokus von TP1. Der untere Teil der Grafik

steht mehr für TP2 und die an Technologien orientierten Teilprojekte TP3 und TP4 im Zentrum.

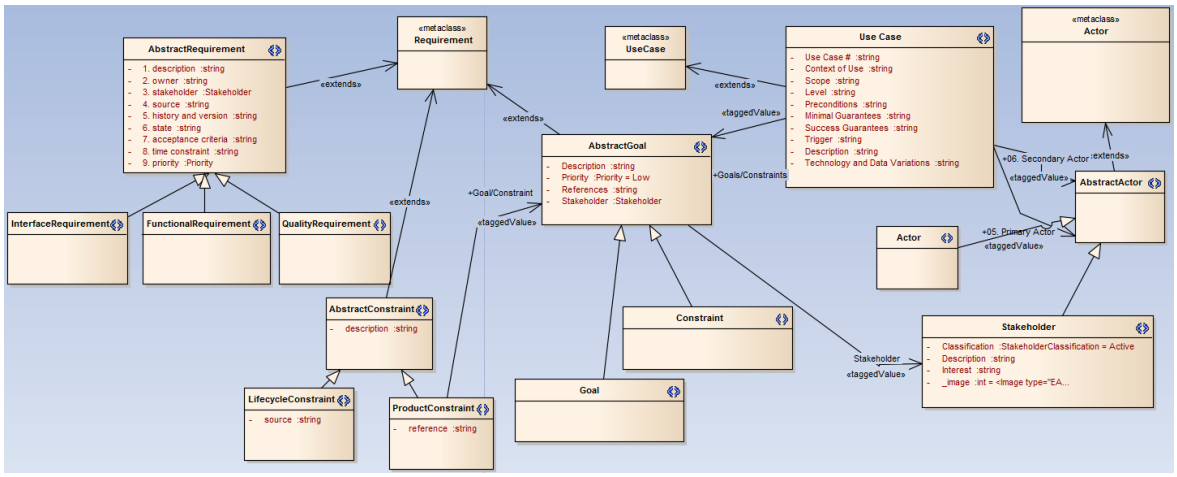

Abbildung 29 Technische Umsetzung des ARAMiS Content-Model als UML-Metamodell in Enterprise Architect

Sowohl das Content-Model als auch das dazu gehörende EA-Plugin wurden in mehreren Iterationen durch Telefonkonferenzen und Workshops insbesondere mit den Domänen sowie TP2 und TP6 abgestimmt. Die enge Abstimmung mit TP2 und TP6 ermöglicht eine einfache Weiterverwendung und Bezugnahme von TP1 Szenarien und Anforderungen beim Systementwurf (TP2) und den Demonstratoren (TP6) im weiteren Projektverlauf.

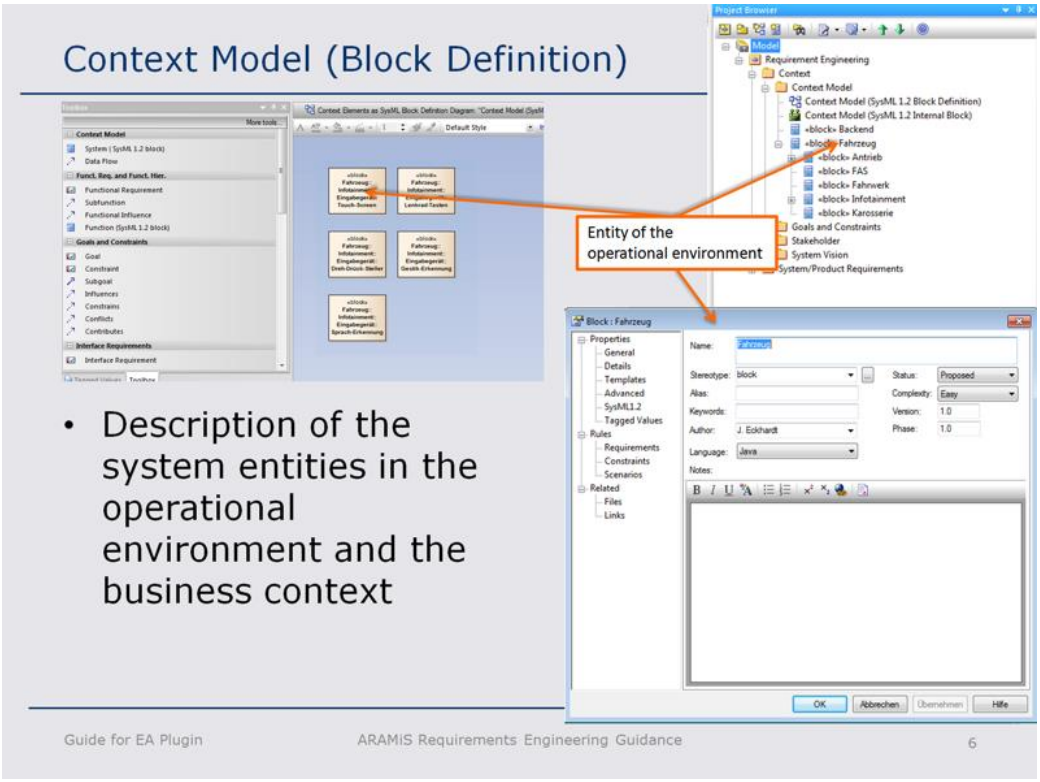

Abbildung 30 Auszug aus der ARAMiS-Guideline EA-Plugin

Für die einfache Verwendung des Content-Models und des EA-Plugins durch einen erweiterten Kreis von Projektpartnern wurde von TUM-SSE ein Dokument *ARAMiS-Guideline-EA-Plugin* (siehe Abbildung 30) erstellt und in Q2 2012 verteilt. In Abbildung 30 ist auch die weitreichende Unterstützung des Content-Models durch das Werkzeug Enterprise Architect beispielhaft erkennbar.

Veröffentlichung auf RESS12

Ein weiteres wichtiges wissenschaftliches Ergebnis von TUM-SSE in TP1 war die Veröffentlichung des Content Models und der dazu gehörenden Requirements Engineering Methodik auf der internationalen Konferenz RESS12 (Chicago) mit dem Beitrag *A Requirements Engineering Content Model for Cyber-physical Systems.*

Übergreifendes ARAMiS CPS-SmartMobility Szenario

TUM-SSE arbeitet zusammen mit Partnern in einer *CPS-Taskforce* (TUM-SSE, FORTISS, KIT, Intel, EADS IW, CASSIDIAN, BMW, CONTINENTAL) an einem (domänen-)übergreifenden ARAMiS CPS-Szenario mit besonderer Betonung der ARAMiS-relevanten Aspekte wie Smart Mobility, Multicore, Infrastruktur und hierarchische Vernetzung der ursprünglich isolierten Mobilitätsdomänen.

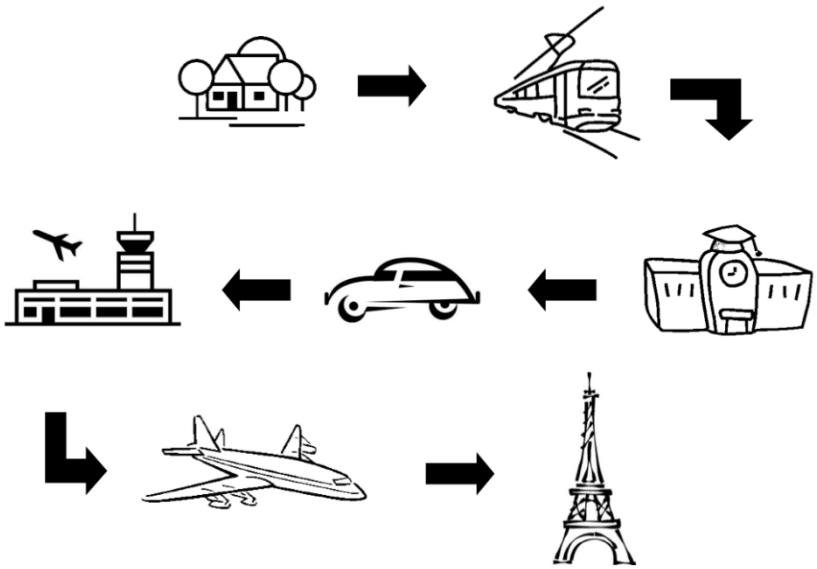

Abbildung 31 Übersicht über das übergreifende CPS-SmartMobility Szenario

Dazu wurde bereits früh (Feb. – Mär. 2012) eine erste System-Vision in Form eines Dokuments erstellt und in weiteren Iteration als Version 0.5 auf dem Workshop am 23.05.2012 (siehe oben, Szenarien und Anforderungen Workshop) vorgestellt und projektübergreifend diskutiert.

Beim halbtägigen CPS-Infrastructure-Workshop am 28.06.2012 bei Intel in Feldkirchen bei München wurde das Szenario um weitere Infrastruktur-Themen angereichert und an einigen Stellen erweitert.

Abbildung 31 zeigt die eine Übersicht über den Ablauf des Szenarios, das im Wesentlichen die Reise von ‚Ms Weber' von München nach Paris beschreibt. Ms Weber wird dabei durch eine Reihe neuartiger Dienste, die von vernetzten CPS erbracht werden, unterstützt. Abbildung 32 zeigt die anfangs identifizierten Ziele und Rahmenbedingungen bei der Entwicklung des übergreifenden CPS-SmartMobility-Szenarios.

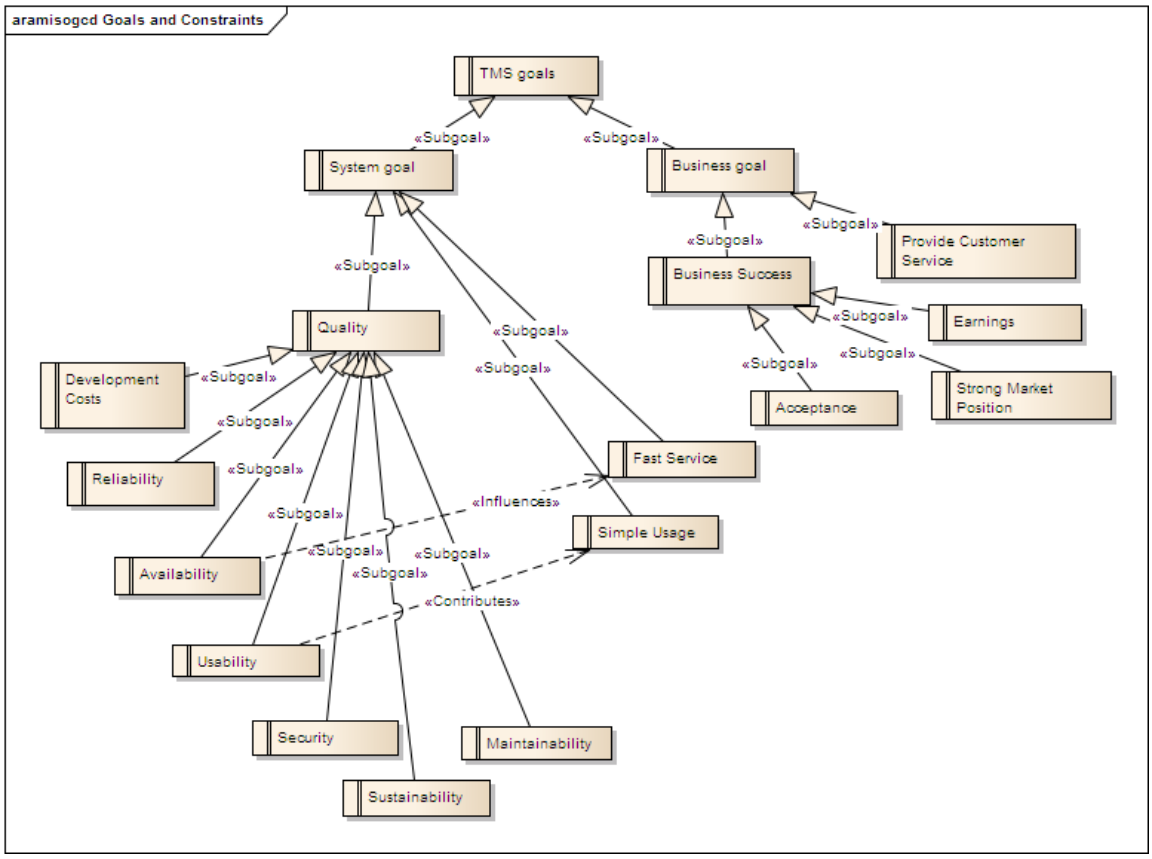

Abbildung 32 Ziele und Rahmenbedingungen des übergreifenden CPS-SmartMobility Szenarios

Veröffentlichungen zum ARAMiS RE Content Model

Mitarbeiter von TUM-SSE haben weitere Evaluierungs-Studien für das ARAMiS RE Content-Model konzipiert und durchgeführt. Dabei sind die folgenden zwei Veröffentlichungen auf Konferenzen entstanden:

- Understanding the Impact of Artefact-based RE - Design of a Replication Study (ESEM'13)
- Two Replication Studies for Evaluating Artefact Models in RE: Results and Lessons Learnt (RESER'13)

Veröffentlichung ARAMiS Inter-Domain CPS Scenario

Mitarbeiter von TUM-SSE haben eine Machbarkeitsbewertung der in TP1 entwickelten domänenübergreifenden Szenarien (Inter-Domain Scenarios) durch Experteninterviews durchgeführt. Diese Machbarkeitsbewertung wurde zusammen mit einer Beschreibung der Herausforderungen beim Requirements Engineering für solche domänenübergreifenden Szenarien und der Beschreibung der ARAMiS Inter-Domain CPS Szenarien selbst zusammen mit FORTISS auf einer Konferenz veröffentlicht:

- Inter-Domain Requirements and their Future Realisability: The ARAMiS Cyber-Physical Systems Scenario (IWCPS'13)

Abschluss der Ergebnisdokumente

Der abschließende Review aller TP1 Ergebnisdokumente laut Projekthandbuch wurde unter Leitung AUDI/TUM-SSE durchgeführt. Der Review wurde abgeschlossen und alle Reviewkommentare wurden qualitätsgesichert durch Excel-Listen mit Erledigungsstatus eingepflegt.

Vorbereitung und Durchführung der Ergebnisdissemination im Projekt

Mitarbeiter von TUM-SSE haben zusammen mit Fraunhofer IESE (AP0.3) eine Workshop-Reihe für die anderen ARAMiS TPs (TP2-TP6) vorbereitet und durchgeführt. Diese weitere Ergebnisverbreitung im Projekt wurde auf Anregung von AUDI/TUM-SSE beim PMT-Treffen im Frühjahr 2013 in Karlsruhe beschlossen. Unter anderem wurden hier von TUM-SSE und Fraunhofer IESE Foliensätze, interaktive Tooldemonstrationen und Fragebögen erstellt, um die Inhalte von TP1 einerseits in den anderen TPs *verständlich zu präsentieren* und andererseits auch *evaluieren* zu können.

Veröffentlichung einer Studie zum ARAMiS RE Content Model

Mitarbeiter von TUM-SSE haben in Zusammenarbeit mit Fraunhofer IESE (AP0.3) eine Studie zur Evaluierung des ARAMiS RE Content-Model im zweiten Halbjahr 2013 durchgeführt. Der Fokus dieser Studie lag auf der Frage wie man in großen Projekten Ergebnisdissemination und –Evaluation durchführt, d.h. es liegt ein Studiendesign vor, welches am Beispiel des ARAMiS RE Content Models und des ARAMiS Inter-Domain CPS Scenario aus TP1 zeigen soll, wie Evaluierungen in großen Projekten durchführt werden können. Wie geplant, wurde diese Studie auf der ESEM 2014 eingereicht und wurde im September 2014 von Mitarbeitern der TUM-SSE auf der Konferenz präsentiert:

- Constanza Lampasona, Philipp Diebold, Jonas Eckhardt and Rolf Schneider, Evaluation in Practice: Artifact-based Requirements Engineering and Scenarios in Smart Mobility Domains. 8th International Symposium on Empirical Software Engineering and Measurement (ESEM'14)

16.1.2 TP5 Durchgängige Entwicklungsmethodik und Anbindung an RTP

Methodenfragebogen

Zusammen mit anderen TP5-Partnern wurde ein Methoden-fragebogen in mehreren Iterationen abgestimmt. Dieser Frage-bogen sollte der Erfassung der von den ARAMiS Partnern

eingesetzten Methoden dienen und darüber hinaus, um eventuelle Defizite bei den aktuell eingesetzten Methoden für das Software Engineering für CPS festzustellen. Der Fragebogen wurde von TUM-SSE selbst auch ausgefüllt.

Werkzeugfragebogen

Parallel zum Methodenfragebogen (siehe oben) wurde zusammen mit TP5-Partnern ein Werkzeugfragebogen in mehreren Iterationen abgestimmt. Dieser Fragebogen sollte der Erfassung der von den ARAMiS Partnern eingesetzten Werkzeuge für (Coding, Modellierung, Dokumentation, Testen, Simulation usw.) dienen. Darüber hinaus, sollten eventuelle Defizite bei den aktuell eingesetzten Werkzeugen für das Software Engineering für CPS identifiziert werden. Der Fragebogen wurde von TUM-SSE selbst auch ausgefüllt.

Kapitel Anforderungen an Multicore-spezifische Methoden

Zusammen mit anderen Partnern hat TUM-SSE in D5.1 insbesondere an der Erstellung des Kapitels 4 mitgewirkt.

CADMOS Architekturbeschreibungssprache

TUM-SSE arbeitet im Rahmen von ARAMiS an der Multicore- und CPS-spezifischen Erweiterung des Open-Source Tools *CADMOS* (http://code.google.com/p/cadmos/). CADMOS wurde mit einer ersten Version einer Architekturbeschreibungssprache für parallele CPS erweitert. Diese Sprache basiert auf den stromverarbeitenden Funktionen der FOCUS Methode und ist technisch durch Xtext, Eclipse RCP und Java umgesetzt. Die Architekturbeschreibungs-sprache soll unter anderem bei der Erprobung von Methoden beim Bau der ARAMiS Demonstratoren bei einer möglichst einfachen Integration verschiedener Partner-spezifischer Tools helfen.

CADMOS Schedule Robustness Optimization

Für das Tool CADMOS (siehe oben) wurde ein Verfahren zur *Schedule Robustness Optimization* (siehe auch unten, Veröffentlichung auf FMICS2012) implementiert. So können verteilte parallele Schedules durch den gezielten Einsatz von Wartezeiten vor Tasks im Rahmen der Gegebenheiten maximal robust gegen zeitliche Schwankungen sowohl bei der Nachrichtenübertragung als auch bei Ausführungszeiten (sog. Jitter) gemacht werden.

Weiterentwicklung des Tools Cadmos

Gemäß VHB [1] hat TUM-SSE das Tool Cadmos weiterentwickelt. Dabei wurden auch Typparameter auf Komponentenebene eingeführt und die Sprache für den Aufbau komplexerer Kanal-Bündel bei der Komponenten-Replikation erweitert.

Ein direkt auf der Java-Virtual-Machine ausgeführten System-Simulators für die in Cadmos beschriebenen parallelen Softwaresysteme wird aktuell weiterentwickelt.

Eine Interview-basierte Erhebung von Timing-Anforderungen und deren Spezifikation, Analyse, Visualisierung und Deployment wurde in Zusammenarbeit mit AUDI vorbereitet.

Veröffentlichung auf FMICS2012

Ein weiteres wissenschaftliches Ergebnis von TUM-SSE in TP5 war die Veröffentlichung der *Schedule Robustness Optimization* Methode auf dem internationalen Workshop FMICS2012 (bei der Konferenz *Formal Methods 2012* in Paris) mit dem Beitrag *Optimizing the Robustness of Software against Communication Latencies in Distributed Reactive Embedded Systems.*

Methoden und Tools für Virtualisierung in Zusammenarbeit mit BMW

Sowohl die Methoden als auch das Tooling zum Thema „Gleichzeitiger Zugriff auf geteilte Ressourcen in Virtualisierungslösungen für den Einsatz in Echtzeit Multicore-Systemen im Automobil" wurden in Zusammenarbeit mit BMW bearbeitet.

Veröffentlichung SAE2013 zu Deployment und Cadmos

Die Präsentation für die SAE2013 (Detroit) Veröffentlichung wurde zusammen mit BMW und AUDI vorbereitet. Sowohl die Methodik als auch die Toolunterstützung für die modellbasierte Entwicklung für zeitkritische Multicore-basierte Automotive-Systeme und die Anbindung an Industrie-relevante Referenz-Technologie-Plattformen wir AUTOSAR werden in dieser Veröffentlichung beschrieben.

Deliverable 5.3

TUM-SSE wirkte darüber hinaus am Deliverable 5.3. „User Guideline for Seamless Methodology“ mit.

Weiterentwicklung des TMREC Ansatzes/Tools

Mitarbeiter der TUM SSE haben den TMREC Ansatz methodisch weiterentwickelt. Insbesondere wurde die Soundness (Wenn die Methode das Programm als korrekt bzgl. einer Spezifikation bezeichnet, ist das Programm auch tatsächlich korrekt) und die polynomielle Laufzeit in der Anzahl der Threads bewiesen. Außerdem wurden Teile der Methode prototypisch implementiert. Dies beinhaltet unter Anderem die Anbindung an das Werkzeug QARMC: Die Konvertierung der Methode und des zu überprüfenden Programms in ein von QARMC vorgegebenes Eingabeformat wurde entwickelt.

Weiterentwicklung des Tools Cadmos

Gemäß VHB [1] hat TUM-SSE das Tool Cadmos weiterentwickelt. Eine Interview-basierte Erhebung von Timing-Anforderungen und deren Spezifikation, Analyse, Visualisierung und Deployment wurde in Zusammenarbeit mit AUDI durchgeführt. Die Ergebnisse wurden bei AUDI präsentiert und diskutiert.

Deliverable 5.4

TUM-SSE wirkte darüber hinaus am Deliverable 5.4. „Tool Integration in the ARAMiS RTP“ mit. Hier wurde die Integration des ARAMiS Content Models (inkl. EA Plugin), des TMREC Ansatzes und des Cadmos Tools in die RTP beigetragen.

Weiterentwicklung des TMREC Ansatzes/Tools

Mitarbeiter der TUM SSE haben den TMREC Ansatz methodisch weiterentwickelt. Insbesondere wurde hier die Integration in eine übergeordnete durchgängige Entwicklungsmethodik anvisiert. Dazu wurden die Ein- und Ausgaben und außerdem die Pre- und Postconditions von TMREC basierend auf dem Meta-Modell modellbasiert beschrieben (in Enterprise Architect).

Deliverable 5.5

TUM-SSE wirkte darüber hinaus am Deliverable 5.5. „User Guideline für eine durchgängige Methodik 2.0 und Leitfaden für eine durchgängige Methodik und Anwendung der Werkzeuge“ mit. Hier wurde insbesondere in Zusammenarbeit mit Offis und Fortis die Planung des Deliverables durchgeführt und erste Prototypen basierend auf dem Meta-Modell erstellt. Außerdem wurde eine Strukturierung des Deliverables und eine erste übergreifende Integration der verschiedenen Tools und Methoden durchgeführt. Hierfür wurden zwei Workshops organisiert (einer von Fortiss und einer von der TUM-SSE).

Interoperabilität des TMREC Ansatzes/Tools

Mitarbeiter der TUM-SSE entwickelten den TMRec-Prototypen methodisch weiter. Insbesondere wurde hier untersucht, inwieweit sich TMRec mit anderen Werkzeugen der übergeordneten durchgängigen Entwurfsmethodik integrieren lässt. Dazu wurden die Ein- und Ausgaben, die Pre- und Postconditions von TMRec in der Sprache SPEM modellbasiert beschrieben (in Enterprise Architect). Ferner wurde der Prozess der Verwendung von TMRec in SPEM beschrieben.

Interoperabilität des Werkzeugs Cadmos

Mitarbeiter der TUM-SSE untersuchten, inwieweit sich die vorhandene Cadmos-Implementierung mit anderen Werkzeugen und Methoden integrieren lässt. Dazu wurde das Interface von Cadmos in der Sprache SPEM beschrieben (in Enterprise Architect).

Interoperabilität des Werkzeugs MTG Plugin

Mitarbeiter der TUM-SSE untersuchten ob sich das MTG Plugin dafür eignet andere zur Entwicklung und Dokumentation von Multicore Systemen benötigten Methoden und Werkzeuge zu beschreiben. Dazu wurde die Inputs und Outputs des MTG Plugins in der Sprache SPEM beschrieben (in Enterprise Architect).

Deliverable 5.5

TUM-SSE wirkte darüber hinaus am Deliverable 5.5. „User Guideline for a seamless methodology, 2nd version“ mit. Hier wurde insbesondere in Zusammenarbeit mit Offis und Fortiss das geplante Deliverable durchgeführt und die auf dem Metamodell basierten Prototypen auf die Integrationsfähigkeit untersucht. Außerdem wurde das Deliverable strukturiert. Hierfür fanden regelmäßige Telefonkonferenzen alle zwei Wochen statt.

16.1.3 TP6 Demonstratoren

TP6.2 Ideenworkshop AUDI, BMW, TUM-SSE

Zusammen mit AUDI und BMW fanden erste Abstimmungen bezüglich relevanter Methoden wie optimierte Schedules unter Beachtung der Kritikalität von Softwarefunktionen und für den Werkzeugeinsatz (AutoFOCUS3 und CADMOS) in TP6.2 Hochintegrationsszenario statt. Bei einem halbtägigen Ideenworkshop am 27.06.2012 bei der TUM am Campus Garching bei München wurden die vorhandenen Konzepte, Methoden und Werkzeuge vorgestellt und nächste Schritte besprochen.

Abstimmung zur Integration von Cadmos

TUM-SSE hat sich weiter mit BMW und AUDI über den Einsatz des Tools Cadmos und dessen mögliche Erweiterung für TP6.2. abgestimmt.

Erstellung des Evaluierungskonzepts für TP6

TUM-SSE hat sich insbesondere mit BMW und dem Fraunhofer IESE über die Konzeption der Evaluierung der Demonstratoren abgestimmt (TP6.2). Hieraus ist ein beispielhaftes Enterprise Architect Modell entstanden welches einen Ansatz verdeutlicht, wie man Requirements, Use Cases, Componenten und die Tests verlinkt. Des Weiteren ist hier in Zusammenarbeit mit dem Fraunhofer IESE und BMW F&T eine Methodik zur Erstellung der Testcases in AP6 entstanden. Dazu wurde in Enterprise Architect ein Vorgehen etabliert, welches zur Erstellung von Testcases verwendet werden soll. Außerdem wurde das Vorgehen in Telefonkonferenzen und in Workshops präsentiert und diskutiert. Dieses Vorgehen wurde im vierten Quartal 2014 weiterentwickelt und im Projekt disseminiert. Dazu wurde zum einen das (bereits in

Q2 und Q3 initial entwickelte) Enterprise Architekt Vorgehen weiterentwickelt und in Telefonkonferenzen vorgestellt. Zum anderen wurde das Vorgehen in dem Deliverable detailliert beschrieben. Außerdem hat die TUM-SSE an den regelmäßigen TP6 Telefonkonferenzen teilgenommen. Schließlich hat die TUM-SSE in Zusammenarbeit mit dem Fraunhofer IESE projektinternen Support zu dem Enterprise Architect Vorgehen gegeben.

16.2 Notwendigkeit und Angemessenheit der Arbeiten

Die Softwareentwicklung rückt für viele Firmen immer stärker in den Mittelpunkt der Produktentwicklung. Die Chancen in Bezug auf die Verbesserung von Markteintritts- und Wettbewerbschancen von deutschen Unternehmen, insbesondere KMUs, auf dem Gebiet eingebetteter Systeme rechtfertigen hier eine Förderung. Viele Firmen haben jedoch nicht die Kapazitäten oder nicht die Kompetenz die enormen Herausforderungen, die bei der Einführung der Techniken von innen heraus zu bewältigen. Individuelle firmeninterne Initiativen greifen meist zu kurz, da oftmals sehr spezifische und suboptimale Lösungen geschaffen werden. Für KMUs stellt diese Hürde ein unüberwindliches Hindernis dar. Das Projekt hat die Basis dafür geschaffen, dass modellbasierte Entwicklung im Bereich Multi-core Entwicklung in vielen Branchen erfolgreich in der deutschen Industrie eingeführt werden kann. Die Chancen in Bezug auf die Verbesserung von Markteintritts- und Wettbewerbschancen von deutschen Unternehmen, insbesondere KMUs, auf dem Gebiet eingebetteter Systeme haben hier eine Förderung gerechtfertigt.

Insbesondere hat die TUM in den Teilprojekten TP1, TP5 und TP6 ihre langjährigen Kompetenzen im Bereich Requirements Engineering, modellbasierte Entwicklung und empirischen Studien eingebracht. Beispielsweise basiert das in TP1 erstellte Content Model auf langjährigen Erfahrungen mit Artefakt-Modellen und der Erfahrung, die in Zusammenarbeit mit Industriepartnern gewonnen wurde.

Zur Erreichung der Ziele war ein Verbund von Partnern notwendig, die sich in der Ausrichtung und im vorhandenen Knowhow optimal ergänzen. Als nicht-profitorientierte Organisation war die Technische Universität München zwingend auf eine Förderung zur Bearbeitung dieses Projekts angewiesen.

16.3 Fortschritte auf dem Gebiet des Vorhabens

Die Verwendung von Multicore-Prozessoren für Realzeitanwendungen in Embedded Systems stellt Industrie und Forschung immer noch vor große Herausforderungen. Offene

Fragestellung sind hier in allen Software und System-Engineering Disziplinen. Beispielsweise werden immer mehr im Requirements Engineering komplexe und komposite Systeme untersucht (Systems of Systems). Die im Projekt ARAMiS entstandenen Ergebnisse sind ein solches SoS. Hierbei wurden Fortschritte in der Industrie und in der Wissenschaft auf Konferenzen (wie z.B. auf der International Requirements Engineering Conference) insbesondere im Bereich Elicitation von Requirements bis hin zu Traceability von Requirements vorgestellt. Die in TP1 erarbeiteten Ergebnisse betten sich hier in den Stand der Technik ein. Außerdem sind weitere offene Fragestellungen beispielsweise die Ermittlung der WCET aufgrund von Wettstreitigkeiten bei Zugriffen auf geteilte Ressourcen in solch einem System, die sichere Abstraktion/Abschottung der Plattform gegenüber nicht autorisierte Zugriffe und generell die effiziente Ausschöpfung der Potentiale von Nebenläufigkeit. Die in der Literatur genannten Konzepte und Methoden nehmen meist Vereinfachungen bei der Systemarchitektur an (keine geteilten Ressourcen, keine Rekonfiguration der Tasks während der Laufzeit) oder schalten für die Performance wichtige Features aus (Caches, Hardware-Beschleuniger).

16.4 Veröffentlichung der Ergebnisse

Viele der in diesem Projekt erarbeiten Ergebnisse wurden auf wissenschaftlichen Konferenzen veröffentlich. Die folgenden wissenschaftlichen Publikationen sind unter Mitarbeit von TUM-SSE entstanden:

[1] B. Penzenstadler, J. Eckhardt, A Requirements Engineering Content-Model for Cyber Physical Systems. 2nd Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems (RES4).

[2] V. Popa, W. Schwitzer, Optimizing the Robustness of Software against Communication Latencies in Distributed Reactive Embedded Systems. 17th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'12)

[3] B. Penzenstadler, D. Mendez Fernandez, J. Eckhardt, Understanding the Impact of Artefact-based RE - Design of a Replication Study. 7th International Symposium on Empirical Software Engineering and Measurement (ESEM '13)

[4] B. Penzenstadler, J. Eckhardt, D. Mendez Fernandez, Two Replication Studies for Evaluating Artefact Models in RE: Results and Lessons Learnt. 3rd International Workshop on Replication in Empirical Software Engineering Research (RESER '13)

[5] B. Penzenstadler, J. Eckhardt, W. Schwitzer, M. V. Cengarle, S. Voss, Inter-Domain Requirements and their Future Realisability: The ARAMiS Cyber-Physical Systems Scenario. International Workshop on Cyber-Physical Systems (IWCPS'13)

[6] María Victoria Cengarle, Jonas Eckhardt, Jürgen Hairbucher, Oliver Hanka, Stefan Kuntz, Birgit Penzenstadler, Oliver Sander, Wolfgang Schwitzer, Astrid Steingrüber. ARAMiS Scenarios and Requirements Technischer Report der TUM (I1334)

[7] Constanza Lampasona, Philipp Diebold, Jonas Eckhardt and Rolf Schneider, Evaluation in Practice: Artifact-based Requirements Engineering and Scenarios in Smart Mobility Domains. 8th International Symposium on Empirical Software Engineering and Measurement (ESEM'14)

[8] Diego Marmsoler, Alexander Malkis, Jonas Eckhardt, A Model of Layered Architectures. 12th International Workshop on Formal Engineering approaches to Software Components and Architectures (FECSA'15)

[9] Alexander Malkis, Multithreaded Cartesian Semantics of Multithreaded Recursive Programs Is Polynomial. Submitted to 40th International Symposium on Mathematical Foundation of Computer Science

[10] Vlad Popa, Wolfgang Schwitzer, Optimizing the Robustness of Software against Communication Latencies in Distributed Reactive Embedded Systems. Formal Methods for Industrial Critical Systems (FMICS'12)

[11] Wolfgang Schwitzer, Rolf Schneider, Dominik Reinhardt, Georg Hofstetter, Tackling the Complexity of Timing-relevant Deployment Decisions in Multicore-Based Embedded Automotive Software Systems (SAE'13)

[12] Lars Lucas, Tobias Schüle, Wolfgang Schwitzer. Self-timed Scheduling and Execution of Nonlinear Pipelines with Parallel Stages. Proc. International Conference on Multicore Software Engineering, Performance, and Tools (MUSEPAT2013), St. Petersburg, 2013.

17 Universität Stuttgart (IPVS)

17.1 Wissenschaftlich-technische Ergebnisse

17.1.1 Konzepte und Methoden für echtzeitfähiges 3D-Rendering auf gemeinsam genutzten GPUs

Grafikkarten (GPUs) stellen eine besondere Form eines Multicore-Systems dar. Sie arbeiten, vergleichbar zu Coprozessoren, asynchron zur Haupt-CPU der Plattform und sind speziell für schnelles Rendering von 2D/3D-Szenen ausgelegt. Da Rendering üblicherweise sehr gut parallelisierbar ist, verfügen GPUs meist über viele Kerne, deren Gesamtrechenleistung nicht selten die der CPU übersteigt. In aktuellen Fahrzeugen wird 3D-Rendering bereits häufig eingesetzt. Zukünftig wird dieser Einsatz weiter stark zunehmen und sich dabei auch das Einsatzspektrum vergrößern: Einerseits in den Bereich von sicherheitskritischen 3D-Animationen (beispielsweise im Kombiinstrument oder Head-up-Display), als auch in den Bereich der Drittanbietersoftware (z. B. aus einem Appstore). Gleichzeitig soll jedoch die Anzahl der dafür eingesetzten Hardwareplattformen reduziert werden, indem beispielsweise Kombiinstrument und Head-Unit konsolidiert werden. Um in solchen Szenarien sicherstellen zu können, dass sicherheitskritisches 3D-Rendering nicht durch sonstige Anwendungen beeinträchtigt wird, muss die verfügbare Rechenleistung der GPU effektiv und in Echtzeit von einem GPU-Scheduler verwaltet werden. Leider sind aktuell verfügbare GPUs nicht präemptiv, sodass eine Reihe von Konzepten und Mechanismen erforderlich sind, um 3D-GPU-Scheduling in Echtzeit durchführen zu können. Im Rahmen des Projekts wurden daher mehrere Konzepte entwickelt, welche im Folgenden beschrieben werden.

17.1.1.1 Laufzeitabschätzung von GPU-Befehlen

Da GPUs nicht präemptiv sind, benötigt der GPU-Scheduler zur Laufzeit die erwartete Ausführungszeit der GPU-Befehlsbatches, um Scheduling-Entscheidung treffen zu können. Hierzu wurde ein Konzept für OpenGL ES 2.0 erstellt, implementiert, evaluiert und als wissenschaftliche Publikation veröffentlicht.

17.1.1.2 GPU-Scheduling

Es wurde ein Konzept für einen GPU-Scheduler erstellt, welcher unter Berücksichtigung der Prioritäten und der gewünschten

Zielframeraten entscheidet, welcher GPU-Befehlsbatch als nächstes ausgeführt wird. Dieses Konzept wurde prototypisch implementiert und ist Gegenstand einer aktuell vorbereiteten wissenschaftlichen Publikation.

17.1.1.3 Display-Berechtigungen

In Kooperation mit Daimler (Info+Telematik) wurde ein Konzept erarbeitet, bei dem die verfügbare (gemeinsame) Displayfläche in Abhängigkeit von vergebenen Berechtigungen und dem Zustand des Fahrzeugs (bzw. Anwendungskontexten) gesteuert wird. Durch dieses Konzept kann sichergestellt werden, dass sicherheits-kritische Anwendungen an der entsprechenden Stelle auf dem Display Zugriff auf dedizierte Displaybereiche erhalten, welche im entsprechenden Kontext nicht von anderen Anwendungen genutzt werden können. Gleichzeitig ermöglicht das Zugriffskontrollmodell jedoch (entsprechenden Kontext vorausgesetzt) hohe Flexibilität bei der Berechtigungsvergabe. Diese Ergebnisse wurden im Rahmen einer wissenschaftlichen Publikation bereits veröffentlicht. Weitere Publikationen sind außerdem geplant.

17.1.1.4 Bitblitting

Anwendungen rendern typischerweise in sogenannte Offscreen-Buffer. Deren Inhalt wird bei Bedarf an die gewünschte Stelle des Display-Buffers kopiert (genannt Bitblitting) und damit für den Fahrer oder die Fahrgäste sichtbar. Da diese Kopiervorgänge recht häufig nötig und sehr ressourcenaufwändig sind, wurde in Kooperation mit Daimler ein optimiertes Bitblittingkonzept erstellt, implementiert, evaluiert und im Rahmen einer wissenschaftlichen Publikation veröffentlicht.

17.1.2 Mobile Code Offloading

Durch die Einführung von leistungsfähigen Multicore-Prozessoren innerhalb der Domäne Automotive findet eine (Hardware-) Konsolidierung statt, welche einhergehende Vorteile, beispielsweise die Reduktion von (Hardware-) Kosten und die Einsparung von Bauraum, mit sich bringt. Diese Vorteile können durch das komplementäre Konzept des so genannten Mobile Code Offloadings (MOC) weiter unterstützt werden. Die Grundidee ist dabei die Auslagerung (engl. Offloading) von ressourcen-intensiven Berechnungen auf leistungsfähige, hoch-effiziente und gemeinsam genutzte Rechnerinfrastrukturen, wie zum Beispiel in einem Cloud-Rechenzentrum. Das Konzept des MOCs bietet sich vor allem für mobile vernetzte Systeme innerhalb eines CPS an, die oft ein weiteres Ziel verfolgen: die Energieeffizienz.

Heutzutage stellt bei mobilen vernetzten Systemen der begrenzte Energiespeicher (Batterie) immer noch die Hauptproblematik dar, weshalb die lokale Ausführung von rechenintensiven Aufgaben zu einer schnellen Entladung der Batterie führt. Aus diesem Grund beschäftigt sich das MCO mit der verteilten Ausführung von (mobilen) Anwendungen, um den Energieverbrauch sowie die Ausführungszeit auf einem ressourcen-beschränkten Gerät zu reduzieren. Zu diesem Zweck werden ressourcen-intensive Anwendungsbereiche, wie zum Beispiel sehr lang andauernde Berechnungen, auf einem leistungsstarken Server in einem Cloud-Rechenzentrum ausgeführt. Hierfür erfolgt eine Migration des lokalen Ausführungszustandes zum entfernten Server und das anschließende lokale Warten auf das Ende der entfernten Ausführung. Nachdem das Ergebnis der entfernten Ausführung empfangen wurde, führt die lokale Recheneinheit die Ausführung fort. Hierdurch können sowohl der Energieverbrauch als auch die Ausführungszeit von Anwendungen signifikant reduziert werden.

17.1.2.1 Die Verwendung von Safe-points innerhalb des Mobilen Code Offloadings

Hinsichtlich einer verteilten Ausführung von Anwendungs-komponenten in einem CPS wird zuallererst eine effiziente Laufzeitumgebung benötigt, die bestimmte Kriterien, beispielsweise die Portabilität von Code (Plattformunabhängigkeit), mit sich bringt. Eine Java-basierte Laufzeitumgebung besitzt die gewünschten Eigenschaften und besitzt darüber hinaus eine breite Akzeptanz innerhalb der Anwendungsentwickler (vgl. Android Apps). Für die Live-Migration von Anwendungskomponenten zwischen einer lokalen und entfernten Recheneinheit wurde die Laufzeitumgebung um folgende Funktionalitäten erweitert: eine Monitoring-Komponente, eine Controller-Komponente sowie einer Safe-point-Komponente. Die Aufgaben der Monitoring-Komponente umfassen die Messung von aktuellen Umgebungsvariablen, beispielsweise die Bandbreite oder Latenz des Kommunikationsmediums zum Internet. Diese gemessenen Informationen dienen der Controller-Komponente als Entscheidungshilfe, der für eine optimale und zuverlässige Ausführung der Anwendung zwischen der lokalen und entfernten Recheneinheit verantwortlich ist. So entscheidet er anhand der aktuellen Verbindungsqualität, ob eine rechenintensive Anwendungskomponente auf der lokalen oder entfernten Recheneinheit ausgeführt wird. Zu diesem Zweck wird die lokale Ausführungsdauer der Anwendungskomponente mit der Übertragungszeit eines sogenannten Safe-points verglichen. Ein Safe-point umfasst in diesem Zusammenhang die benötigten Informationen, wie zum Beispiel Parameter, welche für eine entfernte Ausführung notwendig sind. Die Safe-point-Komponente trägt diese Informationen zu einem Safe-point zusammen.

17.1.2.2 Erhöhung der Robustheit durch die Verwendung von Safe-points

Eine wesentliche Herausforderung des Mobile Code Offloadings ist die Robustheit gegenüber dem Verlust der Kommunikations-verbindung während der entfernten Ausführung auf einer Recheneinheit in einem Cloud-Rechenzentrum. Diese Verbindung wird bei mobilen vernetzten Systemen wie Fahrzeugen typischerweise über Mobilfunktechnologien hergestellt, die für Verbindungs-abbrüche anfällig sind. In diesem Zusammenhang konnten innerhalb des Projektes ARAMiS erfolgreich Methoden und Konzepte entwickelt werden, um die Robustheit von Mobile Code Offloading Systemen gegenüber Verbindungsabbrüchen zu verbessern. Die grundlegende Idee basiert darauf, dass die lokale Recheneinheit nicht nur darauf wartet, einen einzigen Safe-point nach der Beendigung der entfernten Ausführung zu empfangen, sondern bereits während dieser weitere Safe-points (die Zwischen-ergebnisse darstellen) empfängt und lokal speichert. Das Safe-point-Konzept innerhalb des Mobilen Code Offloadings wurde innerhalb einer wissenschaftlichen Publikation veröffentlicht.

17.1.2.3 Erhöhung der Energieeffizienz durch die Verwendung eines stochastischen Vorhersage-Modells

Hierbei spielt die Integration eines stochastischen Vorhersage-Modells eine wichtige Rolle, anhand dessen die entfernte Recheneinheit entscheidet, zu welchem Zeitpunkt ein weiteres Zwischenergebnis zur lokalen Recheneinheit gesendet wird. Zu viele Zwischenergebnisse würden einen starken Anstieg des Energieverbrauches mit sich bringen. Zu wenige bedeuten jedoch ein sehr altes Zwischenergebnis im Falle eines Verbindungsabbruches und der lokalen Ausführungsfortsetzung basierend auf den empfangenen Zwischenergebnissen. Als stochastisches Vorhersage-Modell bietet sich eine Markov Kette an, die die zukünftige Verbindungsqualität aus der bisherig gemessenen Netzwerkqualität effizient und ressourcenschonend vorhersagt. Das Konzept der Verwendung eines stochastischen Vorhersage-Modells für die Erhöhung der Energieeffizienz innerhalb des Mobilen Code Offloadings wurde in eine wissenschaftliche Publikation veröffentlicht.

17.1.3 Demonstratoren

Die in Rahmen von ARAMiS erarbeiteten Konzepte wurden in drei Demonstratoren prototypisch umgesetzt. In Kooperation mit Daimler wurde ein Cockpit-Demonstrator für Virtualized Car Telematics (VCT) erstellt. GPU-Scheduling wurde separat in einem eigenen Aufbau, basierend auf derselben Hardwareplattform wie der Cockpit-Demonstrator präsentiert.

Darüber hinaus wurde für das Mobile Code Offloading eine prototypische Umsetzung erstellt.

17.1.3.1 Cockpit-Demonstrator VCT-B

Ziel des Cockpit-Demonstrators war die Nutzung einer eingebetteten Multicore-Plattform für Kombiinstrument und Headunit, unter Verwendung von Virtualisierung. Hierbei wurden sowohl Sicherheit und Isolation beim Zugriff auf gemeinsame Hardwareressourcen als auch flexible Nutzung der Displayflächen umgesetzt. Der Demonstrator besteht aus einem Automotive Cockpit mit zwei 12-Zoll Displays. Die Lenkradtasten und der Dreh-Drücksteller wurden per CAN angebunden und dienen der Steuerung der Anwendungen. Als eingebettete Plattform kam die Freescale i.MX6 SABRE Automotive Plattform (CPU Board und Base Board) mit vier Kernen á 800 MHz und integrierten 3D und 2D GPUs zum Einsatz. Verwendet wurde die Virtualisierungslösung PikeOS von Sysgo mit 3 virtuellen Linux-Partitionen, eine Headunit-Partition, eine Kombiinstrument-Partition und eine Virtualisierungsmanager-Partition. Die Virtualisierungsmanager-Partition hat hierbei exklusiven Zugriff auf die GPUs und Eingabegeräte. Die beiden anderen Partitionen senden Befehle über eine Shared-Memory-basierte Kommunikationsschicht zum Virtualisierungs-manager, welcher sie verarbeitet.

Zur Demonstration wurden mehrere Automotive-typische Anwendungen erstellt, beispielsweise Drehzahlmesser, Tachometer, Bordcomputer, Navigation, Video-Playback, Kontaktdaten, Anzeige der Rückfahrkamera und ein Spiel. Der Zustand des Systems und verschiedene Use-Cases können über ein Webinterface gesteuert werden. In Abbildung 33 ist in der rechten Bildhälfte der Cockpit-Demonstrator VCT-B dargestellt (Präsentation auf dem Projektabschlusstreffen in Hamburg).

Abbildung 33 Cockpit-Demonstrator und GPU-Scheduling-Demonstrator

17.1.3.2 GPU-Scheduling Demonstrator

Da durch die Virtualisierung die GPU als gemeinsame Hardwareressource von Anwendungen verschiedener Kritikalität genutzt wird, ist effektives Scheduling von GPU-Befehlen notwendig, um die geforderte Isolation zu gewährleisten. Da laufende GPU-Befehle nicht unterbrechbar sind, wurde ein Konzept zur Vorhersage der Laufzeit erstellt, sowie ein GPU-Scheduler, der abhängig von den vorhergesagten Laufzeiten und der jeweiligen Priorität die GPU-Befehle in entsprechender Reihenfolge zur Ausführung bringt. Der Scheduler hat hierbei Kenntnis über die Zielframerate jeder Anwendung und kann dadurch die GPU effizient ausnutzen, da auch Anwendungen niedrigerer Priorität größtmögliche Rechenleistung erhalten, wenn dadurch Anwendungen höherer Priorität nicht beeinträchtigt werden.

Der GPU-Scheduling-Demonstrator nutzt dieselbe Hardware-Plattform wie der Cockpit-Demonstrator VCT-B. In Abbildung 33 ist in der linken Bildhälfte der GPU-Scheduling-Demonstrator dargestellt (Präsentation auf dem Projektabschlusstreffen in Hamburg). Dieser zeigt, dass mit diesem Konzept mehrere Anwendungen unterschiedlicher Priorität auf derselben GPU ausgeführt werden können.

17.1.3.3 Code Offloading Demonstrator

Für das Mobile Code Offloading wurde eine prototypische Implementierung für die verteilte Ausführung von Anwendungskomponenten entwickelt, wobei sowohl die Live-Migration als auch die Integration des stochastischen Vorhersagemodells umgesetzt wurden. Mit Hilfe des Prototyps wurden in verschiedenen Szenarien die Reduzierung des Energieverbrauches sowie die Erhöhung der Ausfallsicherheit für verschiedene (mobile) Anwendungen auf unterschiedlichen mobilen Geräten evaluiert. Der Versuchsaufbau bestand aus einem ressourcen-beschränkten mobilen Gerät, wie zum Beispiel einem Netbook, sowie einem Server für die entfernte Ausführung der Anwendungskomponenten. Der Kommunikations-link zwischen den beiden Geräten wurde als nicht-zuverlässig betrachtet, um Verbindungsabbrüche während des Offloadings zu simulieren. Die durchgeführten Energie- sowie Zeitmessungen ergaben, dass sowohl der Energieverbrauch eines mobilen Gerätes sowie die Ausführungsdauer einer mobilen Anwendung reduziert werden können. Ferner konnte nachgewiesen werden, dass das Safe-pointing-Konzept Mobile Code Offloading robuster gegenüber Verbindungsabbrüchen macht.

17.1.4 Gegenüberstellung der erreichten Ziele mit den Zielen der Vorhabensbeschreibung

Im Rahmen von TP 1 sollten geeignete Szenarien erstellt werden. Diese Szenarien wurden verwendet, um die Konzepte in AP 4.4 zu erstellen und wurden im Cockpit-Demonstrator für repräsentative Szenarien nachvollzogen.

In TP 2 sollten die für Automotive HMI relevanten Anforderungen ermitteln werden. Die wesentlichen Ergebnisse hierzu wurden in der wissenschaftlichen Publikation „Towards Virtualization Concepts for Novel Automotive HMI Systems" veröffentlicht.

In AP 4.4 sollten Softwarekonzepte zur Virtualisierung erstellt werden. Hierzu sollte in Task 4.4.1 die Architektur aus TP 2 bewertet werden und eine State oft the Art Analyse durchgeführt werden. Hierzu sind viele Beiträge in das Ergebnisdokument E 4.4.1.1 eingeflossen. In den Tasks 4.4.2 und 4.4.3 sollten Isolations- und Echtzeitkonzepte entwickelt werden. Hierfür wurden die in Kapitel 17.1.1 beschriebenen Konzepte erstellt, evaluiert und veröffentlicht. In Task 4.4.4 sollten optimierte Virtualisierungskonzepte erstellt werden. Hierfür wurden die in Kapitel 17.1.2 beschriebenen Konzepte für Code Offloading erstellt.

In TP 6 sollten geeignete Demonstratoren aufgebaut und implementiert werden, welche die Konzepte von TP 4 prototypisch in Demonstratoren umsetzen. Die Demonstratoren präsentieren die Funktionalität der Szenarien aus TP 1. Dies wurde in den in Kapitel 17.1.3 beschriebenen Demonstratoren unter Verwendung des erstellten „Drehbuch Businesstrip" umgesetzt.

17.2 Notwendigkeit und Angemessenheit der Arbeiten

Die durch dieses Projekt erarbeiteten Ergebnisse erforderten eine wesentliche Weiterentwicklung des Stands der Forschung und Technik durch neue Konzepte und Methoden (siehe Abschnitt 17.1). Für diese Aufgaben waren wissenschaftliche Mitarbeiter (Doktoranden) mit der entsprechenden Qualifikation erforderlich. Der Umfang der gesteckten Ziele, bestehend aus zwei unterschiedlichen Schwerpunkten (siehe Abschnitt 17.1), begründet den Einsatz mehrerer Doktoranden.

Die Tragfähigkeit der entworfenen Konzepte wurde durch umfangreiche Evaluierungen und Proof-of-Concept-Implementierungen einschließlich Demonstratoren nachgewiesen. Der Nachweis von Eigenschaften wie die Einhaltung von Zeitanforderungen oder die Effizienz des Systems erforderten die praktische Umsetzung und Bewertung anhand realer Implementierungen. Hieraus begründet sich der Einsatz von

wissenschaftlichen Hilfskräften zur Unterstützung der Doktoranden bei Implementierungsaufgaben und der Durchführung von Experimenten.

17.3 Fortschritte auf dem Gebiet des Vorhabens

Während der Projektlaufzeit wurden Produkte von Hardwareherstellern entwickelt bzw. angekündigt, welche die Relevanz der im Rahmen von ARAMiS durchgeführten Arbeiten unterstreichen. Beispielsweise bietet nVidia mit seiner Modellreihe „GRID“ eine Lösung an, um eine leistungsfähige GPU in virtuellen Maschinen nutzbar zu machen. Somit sind zwischenzeitlich virtualisierbare GPUs auf dem Markt verfügbar. Allerdings bieten diese GPUs keine Präemption und auch keine deterministische zeitliche Isolation. Diese essentiellen Anforderungen, welche in den im Rahmen von ARAMiS erstellten Konzepten adressiert wurden, sind daher weiterhin von hoher Relevanz.

17.4 Veröffentlichung der Ergebnisse

[1] Towards Virtualization Concepts for Novel Automotive HMI Systems, Simon Gansel, Stephan Schnitzer, Frank Dürr, Kurt Rothermel, Christian Maihöfer. IESS 2013, Springer.

[2] Concepts for Execution Time Prediction of 3D GPU Rendering, Stephan Schnitzer, Simon Gansel, Frank Dürr and Kurt Rothermel. 2014 9th IEEE International Symposium on Industrial Embedded Systems (SIES).

[3] Increasing the Efficiency and Responsiveness of Mobile Applications with Preemptable Code Offloading. Florian Berg, Frank Dürr, Kurt Rothermel. IEEE 3rd International Conference on Mobile Services (MS 2014), June 27 - July 2, 2014, Alaska, USA

[4] Optimal Predictive Code Offloading. Florian Berg, Frank Dürr, Kurt Rothermel. 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 2014), December 2 - 5, 2014, London, UK

[5] An Access Control Concept for Novel Automotive HMI Systems, Simon Gansel, Stephan Schnitzer, Ahmad Gilbeau-Hammoud, Viktor Friesen, Frank Dürr, Kurt Rothermel, Christian Maihöfer. ACM SACMAT 2014, London, Ontario, 2014.

[6] Efficient Compositing Strategies for Automotive HMI Systems, Simon Gansel, Stephan Schnitzer, Riccardo Cecolin, Frank Dürr, Kurt Rothermel and Christian Maihöfer. 2015 10th IEEE International Symposium on Industrial Embedded Systems (SIES).

[7] Context-aware Access Control in Novel Automotive HMI Systems, Simon Gansel, Stephan Schnitzer, Ahmad Gilbeau-Hammoud, Viktor Friesen, Frank Dürr, Kurt Rothermel, Christian Maihöfer and Ulrich Krämer, 11th International Conference on Information Systems Security (ICISS 2015), Kolkata, India, 2015.

Geplante zukünftige Veröffentlichungen:

[8] Real-time Scheduling of 3D GPU Rendering Commands (Kooperation Universität Stuttgart und Daimler AG).

18 Universität Stuttgart (ISTE)

18.1 Wissenschaftlich-technische Ergebnisse

Gemäß Vorhabensbeschreibung war vor Beginn des Projektes Stand der Wissenschaft:

„Die Verwendung von Multicore-Prozessoren für Realzeitanwendungen in Embedded Systems stellt Industrie und Forschung vor große Herausforderungen. Offene Fragestellungen sind u.a. die Ermittlung der WCET aufgrund von Wettstreitigkeiten bei Zugriffen auf geteilte Ressourcen in solch einem System, die sichere Abstraktion/Abschottung der Plattform gegenüber nicht autorisierten Zugriffen und generell die effiziente Ausschöpfung der Potentiale von Nebenläufigkeit. Die in der Literatur genannten Konzepte und Methoden nehmen meist Vereinfachungen bei der Systemarchitektur an oder schalten für die Performance wichtige Features aus. […] Um die hohe parallele Rechenleistung von Multicore-Plattformen in Anwendungen nutzbar zu machen, müssen bestehende Methoden des Software-Engineering für die Programmierung paralleler Systeme erweitert werden. Dies beginnt bei der Auswahl geeigneter Modellierungs- und Programmierparadigmen für parallele Echtzeitsysteme und setzt sich über den Softwareentwicklungsprozess bis in die Qualitätskontrolle und die Analyse geforderter Eigenschaften wie Safety, Performance, Energie-Effizienz und Produktkosten fort.“

Universität Stuttgart, ISTE, stellte im Projekt ARAMiS daher Expertenwissen zu statischen Programmanalysen und allgemeines Know-how im Bereich der Entwicklung und Wartung sicherheitskritischer Systeme bereit.

Bei den Analysen verfolgten wir zwei wesentliche Szenarien:

1. Analyse der Korrektheit von Programmen im Multi-Core-Einsatz, speziell die Erkennung von Data-Races (Safety-Aspekte)
2. Analysen zur statischen Zuordnung von nebenläufig ausführbaren Einheiten zu Kernen der Zielplattform, so dass neben vorgegebenen Einschränkungen die Kommunikation zwischen Kernen minimiert wird (Performance, Produktionskosten Aspekte).

Dabei sind unsere Arbeiten primär auf die Migration vorhandener Single-Core-Systeme für den Einsatz auf Multi-Cores ausgerichtet, um die Wiederverwendung vorhandener Komponenten zu ermöglichen. Bei dieser Migration besteht eine hohe Gefahr neuer Data-Races; ferner ist eine Verteilung nebenläufiger Tasks auf mehrere CPUs selten bereits angedacht und muss daher erst

etabliert werden, so dass unsere beiden Analysen bei der Migration helfen können.

Sie sind aber auch nützlich für neu erstellte Systeme, da mit ihrer Hilfe die Abwesenheit von Data-Races nachgewiesen bzw. die Einhaltung einer geplanten Kommunikationsarchitektur geprüft werden kann.

Für diese Aufgaben erstellte und erweiterte Universität Stuttgart, ISTE, seine Analysewerkzeuge.

Wir haben in den Teilprojekten 1, 2, 4 und 5 mitgearbeitet, wobei ein besonderer Fokus auf Teilprojekt 5 lag.

18.1.1 TP1 Anforderungen und Szenarien

Gemäß Vorhabensbeschreibung hat sich TP1 zum Ziel gesetzt, „relevante Szenarien für den zukünftigen Einsatz von Multicore-Technologie“ zu definieren.

Universität Stuttgart, ISTE, hat sich hauptsächlich an den Migrationsszenarien für vorhandene, für Single-Core-Rechner ausgelegte Softwarekomponenten beteiligt. Sich daraus ergebende globale Projektanforderungen wurden in den jeweiligen Ergebnisdokumenten beschrieben. Dabei haben wir, zusammen mit Daimler, das Szenario „Kontinuität“ für die Domäne Automotive entwickelt.

18.1.2 TP2 Systementwurf

In TP2 wurde gemäß Vorhabensbeschreibung am Systementwurf für Multicore-Systeme gearbeitet. Besonders wurden dabei Anforderungen aus dem Bereich Safety/Zertifizierbarkeit berücksichtigt.

Im TP2 war Universität Stuttgart, ISTE, maßgeblich an der Arbeit an Anforderungen beteiligt. Hauptbeitrag waren Anforderungen an Migrations- und Entwicklungsprozesse, die von Multi-Core-Systemen eingehalten werden müssen, um Funktion und Sicherheit zu gewährleisten.

Hier erarbeitete Anforderungen flossen in die Ergebnisdokumente ein und bildeten die Grundlage unserer im TP4 durchgeführten Konzipierungen sowie der in TP5 durchgeführten Implementierungsarbeiten.

18.1.3 TP4 Software

Gemäß Vorhabensbeschreibung wurden in TP4 anwendungsspezifische Fragestellungen bezüglich der „Weiterverwendung von

Legacy Code“ und Safety-Aspekten der Realisierungskonzepte bearbeitet.

Im TP4 haben wir daher an mehreren konzeptionellen Projekten gearbeitet.

Das Konzept der Partitionierung anhand der von Analysewerkzeugen tatsächlich in der Software gefundenen Kommunikationsmuster wurde entwickelt. Dieses ermöglicht einen teilweise automatisierten Migrationsprozess von Single-Core- zu Multi-Core-Systemen. Ein entsprechendes Werkzeug wurde im TP 5 von uns realisiert.

Kommunikationsmuster in eingebetteter Software der Automotive-Industrie wurden analysiert. Insbesondere wurde der AUTOSAR-Standard untersucht. Dabei wurden Aspekte der Analysierbarkeit und Eignung der Kommunikation und Synchronisation nebenläufiger Softwarekomponenten für Multi-Core berücksichtigt. Basierend auf den Erfahrungen aus früheren Projekten im Automotive-Umfeld wurden auch Anti-Muster identifiziert und dokumentiert, die in neu zu entwickelnder Software ausgeschlossen werden sollten. Insgesamt wurden somit Vorschläge gemacht, wie sich AUTOSAR weiterentwickeln kann, um besser von Multicore zu profitieren.

Das Konzept der bereits vor Projekt ARAMiS vorhandenen Data-Race-Analyse wurde in Zusammenarbeit mit den Projektpartnern analysiert, überdacht und in Feldversuchen an realen Systemen der Automobilindustrie getestet. Daraus ergaben sich die in TP5 entwickelten Erweiterungen in den Bereichen Ergebnisvisualisierung, Analyseautomatisierung und Präzisionsverbesserung. Der Austausch mit den Projektpartnern ermöglichte eine zielgerichtete Priorisierung der möglichen Erweiterungen unserer Werkzeugkette.

Bei der Konzipierung der Werkzeuge behielten wir stets die im TP5 geplante Gestaltung und Realisierung einer übergreifenden Werkzeug- und Methoden-Plattform im Blick. Die Gestaltung von Schnittstellen zum Informationsaustausch unter den Werkzeugen war wesentlicher Teil dieser Bemühungen. Einige dieser Schnittstellen wurden in TP5 auch realisiert.

18.1.4 TP5 Durchgängige Entwicklungsmethodik und Anbindung an RTP

Gemäß Vorhabensbeschreibung arbeitet TP5 an durchgängigen Entwicklungsmethoden und Werkzeugen. Es beschäftigt sich mit der Anpassung von Werkzeugen und Methoden an neue Gegebenheiten von Multicore-Architekturen und deren Zusammenschluss zu verbundenen, wo sinnvoll auch Herstellerübergreifenden, Werkzeug- bzw. Methodenketten.

Im Rahmen von TP5 brachte Universität Stuttgart, ISTE, daher sowohl eine Erweiterung eines existierenden Werkzeugs zur Data-Race-Analyse als auch ein neues Werkzeug zur Partitionierung von Software in das Projekt ein.

Die Arbeit an zu den Werkzeugen gehörenden Methoden war ebenfalls Teil unseres Beitrages.

18.1.4.1 Verbesserung der Data-Race-Analysen

Automatisierte Data-Race-Analysen sind bekannt dafür, den Benutzer über sehr viele Race-Conditions zu informieren, teils, weil die statische Analyse zwangsläufig zu False Positives führt, teils, weil die Race-Conditions gutartig und teilweise sogar beabsichtigt sind; beides ist jedoch von einem Werkzeug nicht autonom beurteilbar.

Verbesserungen der Data-Race-Analysen können daher einerseits die Genauigkeit der Analysen zur Vermeidung von False Positives schärfen, andererseits heuristisch die Fehlersuche durch bessere Informationsdarstellung unterstützen. Im Projekt ARAMiS haben wir beide Wege verfolgt.

Im Bereich der Data-Race-Analyse konnte die geplante Erweiterung zur Erkennung von Referenzparametern realisiert werden. Durch diese Erweiterung werden im C-Quelltext des analysierten Programms diejenigen formalen Parameter identifiziert, die eine Übergabe eines bestimmten Objekts per Referenz bewirken (und nicht der Gewinnung eines beliebig weiterreichbaren Zeigerwerts dienen). Eine Schwächung der Analyseergebnisse kann so vermieden oder reduziert werden. Durch die Nutzung dieser Information wurde das Resultat der Data-Race-Analyse verbessert, so dass die Datenfluss-Analyse einer bestimmten Funktion im Kontext ihres Aufrufs genauer und damit auch die Präzision der Data-Race-Analyse gesteigert wird. Ebenfalls hilfreich für die Qualität der Ergebnisse ist das Einbeziehen von Informationen aus dem Aufrufkontext einer Methode. Dadurch wird zwar keine vollständige Kontext-Sensitivität erreicht, es ist jedoch auch sichergestellt, dass die Laufzeit in beherrschbaren Größenordnungen bleibt.

Eine Verbesserung der Analyse-Qualität durch k-Kontexte zur Verwendung kontextsensitiver Informationen an Unterprogrammgrenzen wurde in Betracht gezogen und evaluiert. Zugunsten anderer Projektarbeiten wurde dieser speicher- und laufzeitintensive Ansatz jedoch nicht weiterverfolgt.

Die Berücksichtigung von Pfadprädikaten bei den Analysen durch Einsatz von Constraint-Solvern wurde prototypisch implementiert und an kleinen Beispielen getestet. Mit Hilfe des Constraint-Solving können nicht realisierbare Pfade erkannt und ignoriert werden. Ferner kann gelegentlich die Gleichzeitigkeit mehrerer

konkurrierender Datenzugriffe ausgeschlossen werden, weil ihre Pfadprädikate in den jeweiligen Threads nicht gleichzeitig erfüllt sein können. Dieser Ausschluss erweist sich als schwieriges Thema, auch weil für nicht-triviale Fälle die Pfadprädikate selbst möglicherweise Race-Conditions ausgesetzt sind, deren Gutartigkeit in Bezug auf den intendierten Ausschluss erst nachzuweisen ist.

In TP 4 wurde Bedarf für weitere Automatisierung der Analysen erkannt. Hierzu wurde ein Werkzeug konzipiert und erstellt, das die Data-Race-Analyse zum Teil automatisiert und auch eine einfache Visualisierung enthält.

Dazu wurde eine Menge „verdächtiger“ Programmmuster identifiziert und zur heuristischen Bewertung herangezogen. Die Erkennung der Muster wurde implementiert. Die im User-Interface angezeigten heuristischen Bewertungen sollen es dem Nutzer erlauben, seine Untersuchungen richtig zu priorisieren – also sich beim Auftreten einer Vielzahl von Warnungen auf diejenigen zu konzentrieren, die mit höchster Wahrscheinlichkeit problematisch sind.

Nach einer ersten Version, in der mit einer fixen Zahl von Mustern nach den wahrscheinlichen Fehlern gesucht wurde, ist die neueste Version dazu übergegangen, die Benutzer in Grenzen die Formulierung zusätzlicher Filter interaktiv beisteuern zu lassen. Wir gehen davon aus, dass die „richtigen“ Filter sich von System zu System unterscheiden.

Die Verbesserungen des Data-Race-Werkzeuges wurden mehrfach beim Projektpartner Daimler an realen Industriesystemen erprobt. Es wurden dabei Verbesserungen gegenüber dem jeweils vorherigen Stand nachgewiesen und wertvolle Anregungen für weitere Verbesserungen gewonnen.

Bei diesen Praxistests wurden wiederholt zwei Systeme analysiert – eines mit etwa 41 Tausend SLoCs, eines mit etwa 55 Tausend SLoCs. Unsere Analysen kamen mit der Größe der Systeme zurecht.

Bei diesen Praxistests wurde insbesondere auch Bedarf an weiteren Fähigkeiten der Ergebnisvisualisierung und der Navigation im Quellcode zur Inspektion der jeweiligen Data-Race-Warnung festgestellt. Daraufhin wurde eine Visualisierungskomponente von Grund auf neu entworfen und implementiert.

18.1.4.2 Partitionierung von nebenläufigen Einheiten

Im Verlauf des Projektes sind dazu zwei Implementierungen des Partitionierungskonzeptes entstanden. Eine Proof-Of-Concept- und eine weitere Implementierung, die ein vollständiges graphisches Benutzerinterface enthält, das Zugang zu den

ermittelten Kommunikationsbeziehungen erlaubt und damit auch Architekturerhaltsprüfungen ermöglicht. Das Partitionierungswerkzeug wurde an verschiedenen Software-Systemen erprobt und evaluiert. Als Testsysteme wurden zwar reale Industrieprojekte verwendet, jedoch war in den verwendeten Projekten kein Multi-Core-Einsatz geplant. Eine Bewertung des Ergebnisses war somit nur begrenzt möglich.

Die Werkzeuge erlauben die Verteilung nebenläufiger Komponenten auf die Kerne eines Multi-Core-Rechners, wobei einerseits harte Constraints an die Zuweisung erfüllt werden, die sich etwa aus der Hardware-Ausstattung oder aus der Sicherheitseinstufung der Komponente ergeben, und andererseits die teure Kommunikation über Kerne hinweg minimiert wird, soweit diese Kommunikation in statischen Analysen erkennbar und abschätzbar ist. Durch Betrachtung der entstehenden Kommunikationsgraphen eines bereits verteilten Systems kann auch geprüft werden, ob ursprüngliche Intentionen über das Kommunikationsverhalten auch in der Evolution des Systems bewahrt wurden.

Die sowohl der Data-Race-Analyse als auch der Partitionierung zugrundeliegende Basisanalyse setzt auch Zeigeranalysen ein, um die durch Indirektion hergestellte Kommunikation zu entdecken. Diese Basisinformation wird zunächst in einen Kommunikationsgraphen verdichtet, indem das Maß der Kommunikation zwischen nebenläufigen Einheiten etabliert wird. Dieser Graph wird dann mit bekannten Methoden des „Simulating Annealing“ so in Untergruppen partitioniert, dass die Kommunikation zwischen den Gruppen minimiert wird. Andere Verfahren zur Approximierung bzw. Ermittlung der besten Partitionierung wurden ebenfalls evaluiert.

Als wissenschaftliches Ergebnis steht hier der Nachweis, dass die an sich für die Data-Race-Analysen entwickelte Algorithmik zur Erkennung des gemeinsamen Zugriffs auf globale Variablen auch dazu eingesetzt werden kann, ein Maß für die Kommunikation zu entwickeln. Weiterhin wurde untersucht, wie sich anhand dieses Maßes Kommunikation über Core-Grenzen hinweg minimieren lässt.

Die Methodik wurde 2012 bereits publiziert und das prinzipielle Funktionieren auch für industrielle Systeme nachgewiesen. Allerdings war bei den realen Systemen die Anzahl verfügbarer Cores zu klein, als dass die Werkzeuge ihre Vorteile hätten voll entfalten können oder die Partitionierung den Einsatz von Werkzeugen notwendig gemacht hätte. Für größere Architekturen mit mehr Cores wurde das Funktionieren anhand synthetischer Tests nachgewiesen. Unsere Arbeit stieß insbesondere auch bei anderen Projektpartnern auf großes Interesse. Ergebnisdokumentbeiträge und Publikationen von Partnern beziehen sich auf unsere Arbeit.

18.1.4.3 Kooperation gemeinsame Werkzeugplattform

Als Beitrag zur gemeinsamen Werkzeugplattform wurden unsere Werkzeuge und die dazu gehörenden Methoden zunächst in Steckbriefen dokumentiert. Später wurden diese Dokumentationen formalisiert und umgewandelt, so dass sie dem Meta-Model einer gemeinsamen Werkzeugplattform hinzugefügt werden konnten. Im Meta-Model haben viele Werkzeughersteller ihre Werkzeuge dokumentiert und somit Vernetzungsoptionen aufgezeigt.

18.1.4.4 Zusammenarbeit mit anderen Werkzeugherstellern

Zur Validierung der Ideen und Konzepte der gemeinsamen Werkzeugplattform wurde ein Datenaustauschprojekt realisiert. Hierzu wurden Werkzeughersteller-übergreifend Daten ausgetauscht. Dadurch konnte eine Präzisionsverbesserung und eine weitere Automatisierung der Data-Race-Analyse erreicht werden. Ein Ziel dieses Projektes war es auch, den Wert von Informationsaustauschmöglichkeiten, wie sie im Meta-Model definiert sind, in der Praxis zu zeigen.

Mit Symtavision wurde ein Austauschformat entworfen und realisiert, in dem die Eigenschaften von Parallelität in analysierter Software festgehalten werden. Symtavision produziert Werkzeuge, die sich z. B. die Antwortzeiten der Software zum Thema nehmen und dafür Information zur echten Gleichzeitigkeit von Programmkomponenten durch statische und dynamische Analysen ermitteln. Unsere Werkzeuge haben ebenfalls Bedarf an dieser Information, denn nur an solchen Stellen können Data-Races entstehen. Statt nun die Information selbst herzuleiten, wozu man die Eigenschaften der jeweiligen Kernelumgebung in die Analysen einbeziehen müsste, oder sievom Benutzer abzufragen, wie das in Bauhaus bislang geschah, haben wir über diese Schnittstelle mit den Symtavision-Werkzeugen erfolgreich kommuniziert und damit einen weiteren arbeitsaufwändigen Schritt in der Anwendung unserer Werkzeuge eliminiert sowie eine Präzisionsverbesserung erreicht.

Auf beiden Seiten wurden die notwendigen Implementierungsarbeiten geleistet, um den Austausch praktisch zu erproben. Die Erprobung fand mit einem Industriesystem statt und wurde in den Ergebnisdokumenten dokumentiert.

Das Austauschformat wurde auch anderen Werkzeugherstellern überlassen, die ähnliche Informationen bereitstellen können. Ein vollständiger Test dieses weiteren Austausches war vor Ende des Projektes jedoch nicht mehr möglich.

Im Rahmen der Kooperation mit der Universität Kiel wurde ein Konzept zur Generierung von SAT-Constraints aus der Bauhaus-Programmzwischendarstellung IML entworfen und prototypisch implementiert, damit die Constraints dem in Kiel entwickelten SAT-

Solver zur Lösung vorgelegt werden können. Später wurde dieses Austauschformat mehrfach angepasst und verbessert.

18.2 Notwendigkeit und Angemessenheit der Arbeiten

Die zur Projektausschreibung angegebenen Gründe für die Notwendigkeit der Förderung haben nach wie vor Gültigkeit.

„Die in diesem Projekt zu leistende Hersteller- und Mobilitäts-domänenübergreifend abgestimmte Forschungs- und Entwicklungsarbeit schafft die Voraussetzung für mittel- und langfristige Wettbewerbsvorteile von in Deutschland hergestellten Produkten mit Multicore-Technologie […]. Der für die Abstimmung und Durchführung des Projekts erforderliche erhebliche Ressourceneinsatz ist jedoch mit Risiken verbunden und übersteigt die Möglichkeiten der Projektpartner und wird daher zur Förderung eingereicht."

Unsere Arbeiten haben dabei kritische ungelöste Probleme der Industrie einer Lösung nähergebracht.

Der von uns in Anspruch genommene Förderumfang ist angemessen. Mit weniger Personaleinsatz wären die Entwicklungs- und Forschungsarbeiten nicht zu leisten gewesen, und dem Projektergebnis hätten wesentliche Inhalte gefehlt.

Die zur Verfügung gestellten Mittel waren auch ausreichend. Dies zeigt sich darin, dass wir alle grundsätzlichen Aufgaben im Projekt ARAMiS bearbeitet haben und unsere Arbeiten auch erfolgreich abgeschlossen sind.

Mit weiteren Mitteln hätten zwar weitere Forschungsfragen bearbeitet werden können, dies kann jedoch auch zukünftig Ziel von Förderung sein.

18.3 Fortschritte auf dem Gebiet des Vorhabens

Durch das Projekt ARAMiS und parallel dazu laufende Forschung wurden wichtige Forschungserkenntnisse und technische Fortschritte im Einsatz statischer Programmanalysen für höhere Sicherheit und Effizienz automotiver Software erzielt. Unsere Beiträge dazu sind den Ergebnisdokumenten und insbesondere auch dem Abschlussbericht zu entnehmen. Unsere Forschung macht wesentliche Teile des Kapitels 13 aus. Unsere Beiträge finden sich ab Seite 403 und 428.

Die Forschung im Bereich Partitionierung hat wichtige Problemfelder erfasst und ermöglicht es so, durch entsprechende Umsetzungsmuster guten Nutzen aus zur Verfügung stehender paralleler Rechenleistung zu ziehen. Der Blick über aktuelle off-

the-shelf-verfügbare Architekturen hinaus erlaubt es auch, Methoden auf zukünftige Architekturen anzupassen.

Die Fortschritte, die im Bereich Data-Race-Analyse erzielt wurden, bringen derartige Analysen nahe an die wirtschaftliche Anwendbarkeit im Regelbetrieb. In Projekten, in denen die Fehlerfreiheit von besonderer Bedeutung ist, kann sie heute bereits eingesetzt werden.

18.4 Veröffentlichung der Ergebnisse

Während der Projektlaufzeit einschließlich kostenneutraler Verlängerung wurden von Universität Stuttgart, ISTE, folgende Publikationen erstellt:

[1] Martin Wittiger, Steffen Keul
Extraktion von Interthread-Kommunikation in eingebetteten Systemen
Automotive 2012
Karlsruhe, Deutschland, November 2012

[2] Martin Wittiger, Timm Felden
Recognition of Real-World State Based Synchronization
17. Workshop Software Reengineering & Evolution
Bad Honnef, Deutschland, Mai 2015

[3] Nikolaos Koutsopoulos, Mandy Northover, Timm Felden, Martin Wittiger
Advancing Data Race investigation and Classification through Visualization
3rd IEEE Working Conference on Software Visualization (VISSOFT 2015)
Bremen, Germany, September 27-28, 2015

[4] Sandro Degiorgi, Martin Wittiger
Ergebnisbewertung konservativer statischer Data-Race-Analysen
15. Workshop Software-Reengineering Bad Honnef, Deutschland, Mai 2013

Im Projekt ARAMiS ausgeführte Arbeiten sind Grundlage unserer Forschung. Daraus werden sich auch in Zukunft Publikationsmöglichkeiten ergeben.

Schon vor Beginn des ARAMiS-Projektes und auch parallel dazu, aber nicht im inhaltlichen oder finanziellen Kontext von ARAMiS, wurden von uns folgende Publikationen erstellt, die den wissenschaftlichen Stand beschreiben.

[1] Görg, Torsten; Performance Tuning of PDG-based Code Clone Detection (In: Proceedings of the 17. Workshop Software-Reengineering & -Evolution (WSRE), 04.-06. Mai 2015, Bad Honnef)

[2] Felden, Timm; Efficient and Change-Tolerant Serialization for Program Analysis (In: Proceedings of the 16. Workshop Software-Reengineering & -Evolution (WSRE), 28.-30. April 2014, Bad Honnef)

[3] Görg, Torsten; Northover, Mandy; A Canonical Form of Arithmetic and Conditional Expression (In: Proceedings of the 16. Workshop Software-Reengineering & -Evolution (WSRE), 28.-30. April 2014, Bad Honnef)

[4] Felden, Timm; Görg, Torsten; Werkzeugunterstützte Eliminierung von Data-Races in Eclipse; (In: 15. Workshop Software-Reengineering (WSR 2013) Softwaretechnik-Trends, Band 33, Heft 2. GI, Mai 2013)

[5] Keul, Steffen; Tuning Static Data Race Analysis for Automotive Control Software. (In 11th IEEE International Working Conference on Source Code Analysis and Manipulation, pages 45-54, IEEE, Sept 2011)

[6] Plödereder, Erhard; Programming Languages Meet Multicore. (In: Reliable Software Technologies, Ada-Europe 2011, LNCS(6652), Juni 2011, S.189-192)

[7] Prokharau, Mikhail; Gerlach, Daniel; Keul, Steffen; Static Analysis of Predicate-based Synchronisation. (In: 13. Workshop Software-Reengineering (WSR 2011), Softwaretechnik-Trends, Band 31, Heft 2, S.16-17. GI, Mai 2011)

[8] Raza, Aoun; Franke, Stefan; Plödereder, Erhard; Detecting High-Level Synchronization Errors in Parallel Programs. (In: Reliable Software Technologies, Ada-Europe 2011, LNCS(6652), Juni 2011, S.17-30)

[9] Keul, Steffen; Static Versioning of Global State for Race Condition Detection (In: Ada-Europe 2010, volume 6106 of LNCS, pages 111-124. Springer, 2010)

[10] Keul, Steffen; Prokharau, Mikhail; Gerlach, Daniel; Jenke, Carola; Raza, Aoun; RaceVis: Ein Werkzeug zur Visualisierung von Data Races. (In: 12. Workshop Software-Reengineering, Softwaretechnik-Trends, Band 30, Heft 2. S. 82-83. GI, Mai 2010)

19 Universität Paderborn

19.1 Wissenschaftlich-technische Ergebnisse

19.1.1 Einleitung

Die Arbeiten der Universität Paderborn umfassten Entwicklungen bei der ARAMIS Referenzarchitektur (innerhalb von Teilprojekt 2) und neue Ansätze zur Vollvirtualisierung und Laufzeitmigration von virtuellen Maschinen (innerhalb von Teilprojekt 4).

Zum einen entwickelte die Universität Paderborn ein Rahmenwerk für eine integrierte logische und technische ARAMIS Referenzarchitektur. Die Architekturen wurden durch UML-Meta-Modelle definiert und sind kompatibel zu bestehenden Standards aus verschiedenen technischen Bereichen (ARINC, AUTOSAR, DMTF CIM), um die Migration von bzw. auf diese Normen zu erleichtern. Generell bieten die Referenzarchitekturen einen hersteller- und technologieneutralen Rahmen für die Integration von Hard- und Software-Komponenten für Multi-Core basierte Architekturen, die alle relevanten Aspekte der Anwendungs-software, Middleware, Betriebssystem, Virtualisierung, Firmware und Hardware umfassen.

Zum anderen wurden Arbeiten zur Virtualisierung von Hardware-Ressourcen durchgeführt. Hier wurden projektspezifische Kriterien für die Bewertung von Systemsoftware definiert, partner-übergreifend die Evaluation von System-Software koordiniert und das Linux-Modul KVM ausgewertet. Ein inhaltlicher Schwerpunkt waren Arbeiten zur dynamischen Zuweisung von Hardware-Ressourcen zu virtuellen Maschinen mit Echtzeitanforderungen. Die Universität Paderborn entwickelte Ansätze zur Laufzeit-Migration virtueller Maschinen und zum adaptiven Scheduling von virtuellen Maschinen auf Mehrkern-Steuergeräten.

19.1.2 Erstellung einer logischen und technischen Rechnerarchitektur

Zur Entwicklung eines Metamodels zur Definition einer Multicore-basierten logischen und technischen Rechnerarchitektur analysierte und bewertete die Universität Paderborn existierende Standardinformationsmodelle. Zu den untersuchten Standards zählten ARINC 653, AUTOSAR 4.0, IP-XACT IEEE 1685-2009 und DMTF CIM. Eine Übersicht der Auswertung findet sich in Abbildung 34. Während die Standards ARINC und AUTOSAR den ARAMiS-Mobilitätsdomänen Automobil und Luftfahrt entstammen, adressieren IP-XACT und DMTF CIM teilweise vertikale und

komplementäre Domänen. IP-XACT beschreibt elektronische Komponenten und deren Designs für den Systems-on-Chip Entwurf. DMTF CIM ist ein sehr umfassender Standard, der alle Bereiche (Software, Middleware, Hardware) in allen Details umfasst und bereits zum Management von verteilten Server- und IT-Systemen im praktischen Einsatz bei der Konfiguration von größeren IT-Infrastrukturen wie Servern und Rechenzentren ist, was einen hohen Grad an Praxistauglichkeit nachweist.

Zur Evaluierung wurden zunächst aussagekräftige Vergleichskriterien erarbeitet, die sich mittels folgender globaler Aspekte zusammenfassen lassen: Systemarchitektur, Multicore, Virtualisierung, Safety, Security, Zeitverhalten und Energiemanagement.

Zum Vergleich wurden die in ARAMiS definierten Granularitätsebenen, nämlich „Computer“, „Board“ „Chip“, „IP“, sowie die logische, technische und geometrische Modellierungsperspektive berücksichtigt und ausgewertet.

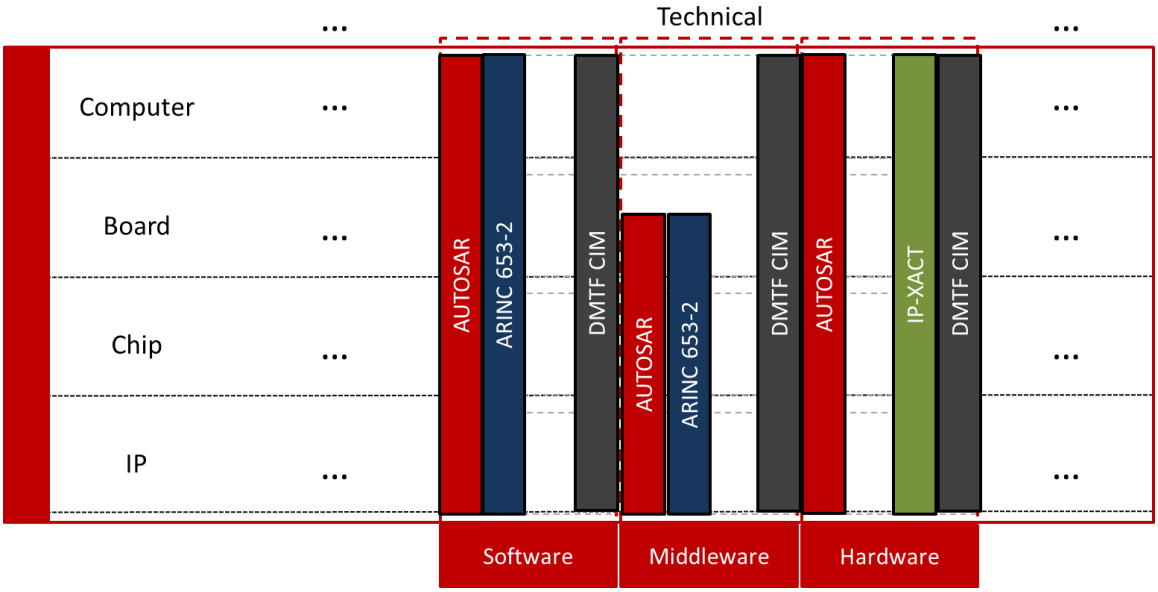

Abbildung 34 Vergleich von domänenspezifischen Standards hinsichtlich ihrer Eigenschaften zur Definition der ARAMiS Referenzarchitekturen

Die Evaluierung aller Standards ergab die eindeutige Überlegenheit von CIM (Common Information Model) der DMTF (Distributed Management Task Force), da das Modell alle Ebenen (Software, Middleware, Hardware) umfasst, schon verschiedene Unterklassen und Anwendungsbereiche (Core, Device, Network, Physical, Security, Energy/Power Mgmt., ...) unterscheidet und bereits im praktischen Einsatz bei der Konfiguration von Servern ist, was eine gewisse Vollständigkeit gewährleistet. Dieser Vollständigkeit stand aber die enorme Komplexität des

Metamodells mit über 1400 Klassen gegenüber. Zusätzlich konnten die existierenden Klassen nicht direkt übernommen werden, da die Modelle in einem DMTF-spezifischen Format (Meta Object Format – MOF) definiert sind.

Die detaillierte Analyse findet sich im ARAMIS-Bericht E2.1.1.3.

Die weiteren Arbeiten bestanden deshalb darin, die für ARAMIS relevanten Teile der logischen und technischen Rechner-architektur von CIM aus dem Standard zu identifizieren, zu extrahieren und für die Partitionierung in logische und technische Architektur inklusive der Definition von Abbildungsregeln für ARAMIS aufzuarbeiten. Wir wählten hierfür einen Ansatz mit generischen Schnittstellen zu anderen Teilen des CIM Standards, so dass die ARAMIS-Modelle bei Bedarf um die entsprechenden Anwendungsgebiete im CIM-Standard, wie z.B. Security, erweitert werden können.

19.1.2.1 Definition der logischen Rechnerarchitektur und Funktionen der Laufzeitumgebung

Zur Definition der logischen Rechnerarchitektur und Funktionen der Laufzeitumgebung erarbeitete die Universität Paderborn ein Metamodel als Rahmenwerk für den modellbasierten Entwurf logischer und technischer Rechnerarchitekturen. Der grund-legende Aufbau, wie in Abbildung 35 gezeigt, orientiert sich an den bekannten Modellierungsebenen der Object Management Group (OMG) und basiert auf der OMG-Standardmodellierungssprache UML2. Entsprechend der Vorgehensweise bei UML2-basierten Standards, wie z.B. AUTOSAR, wurden die jeweiligen Metamodelle mit Hilfe von UML2-Klassen, sowie deren Attributen und Assoziationen definiert. Für die abgeleiteten Domänen- und Demonstrator-Modelle kam SysML 1.2 zum Einsatz. SysML wurde gewählt, da sie in manchen Bereichen die ingenieur-orientierte Sichtweise besser als die UML2 unterstützt. Da SysML eine kompatible Variante (UML2-Profil) der UML2 ist, ist somit auch eine Kompatibilität der SysML-Modelle gewährleistet. Wegen der Zweiteilung in logische und technische Rechnerarchitektur wurden zur Abbildung von Elementen der logischen Rechnerarchitektur auf Elemente der technischen Rechnerarchitektur eindeutige Regeln definiert.

Hierzu wurde zunächst ein UML-basiertes Metamodell zur Definition von logischen Rechnerarchitekturen entwickelt. Als Grundlage dienten die vorher erwähnten Evaluierungsergebnisse, sowie eine Identifikation der funktionalen Komponenten der ARAMIS-Demonstratoren. Abbildung 36 zeigt beispielhaft einen Ausschnitt des Metamodells zur Beschreibung von logischen Komponenten.

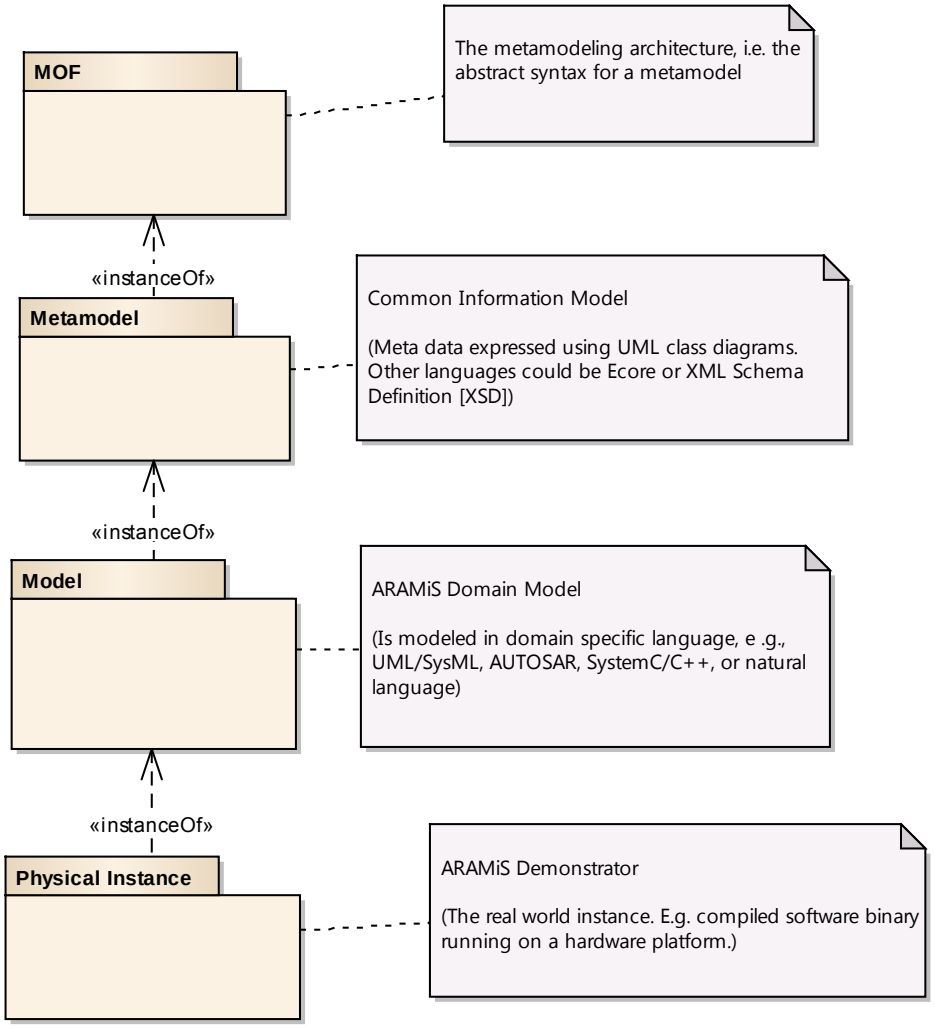

Abbildung 35 Ein modellbasiertes Entwurfsrahmenwerk für logische und technische ARAMiS-Rechnerarchitekturen

Auf der Basis dieses Metamodells wurde exemplarisch ein Modell der logischen Systemarchitektur des LSSI-Demonstrators (Fahrzeugplattform D) entwickelt. Diese Arbeit fand in Kooperation mit der AUDI AG und der EFS GmbH statt. Die sich dadurch neu ergebenen Anforderungen wurden wiederum in das Metamodell eingepflegt, um eine fortlaufende Anwendung auf die Demonstrator-Modelle zu gewährleisten.

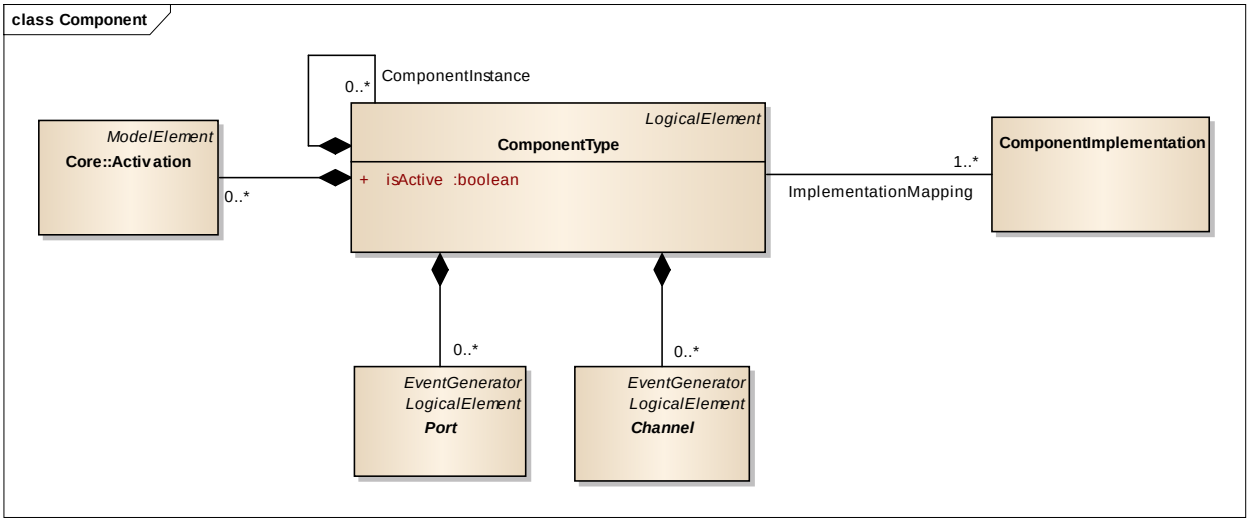

Abbildung 36 Ausschnitt des UML-Metamodells zur Beschreibung von logischen Komponenten

19.1.2.2 Technische Rechnerarchitektur und Architektur der Laufzeitumgebung

Komplementär zur logischen Rechnerarchitektur definierte die Universität Paderborn ein UML-basiertes Metamodell zur formalen Beschreibung technischer Rechnerarchitekturen. Der Fokus der Arbeiten lag dabei sowohl auf der formalen Definition von implementierungsnahen Softwarekonzepten, wie z.B. Speicherpartitionen, OS-Applikationen, oder schedule-fähigen Einheiten, sowie von mehrkern-spezifischen Charakteristika der Laufzeitumgebung, d.h. der ausführenden Hardwareplattform.

Als Anwendungsfall diente wiederum der physikalische LSSI-Demonstrator der AUDI AG. In Kooperation mit den Projektpartnern wurde ein SysML-Modell mittels des Modellierungswerkzeugs Enterprise Architect (Sparxsystems) erstellt.

Bei der Softwaremodellierung lag das Hauptaugenmerk auf der Spezifikation und Analysefähigkeit des sog. Deployments, wie in einem Anwendungsbeispiel für einen Mehrkern-AURIX™-Prozessor in Abbildung 37 gezeigt. Hierfür wurde die Softwarearchitektur in OS-Applikationen unterteilt und mit Hilfe von spezifischen Mapping-Klassen auf die einzelnen Rechenkerne im Hardwaremodell verteilt. Das Hardwaremodell selbst berücksichtigt die ARAMiS Granularitäts-Ebenen „Computer“, „Chip“ und „IP“.

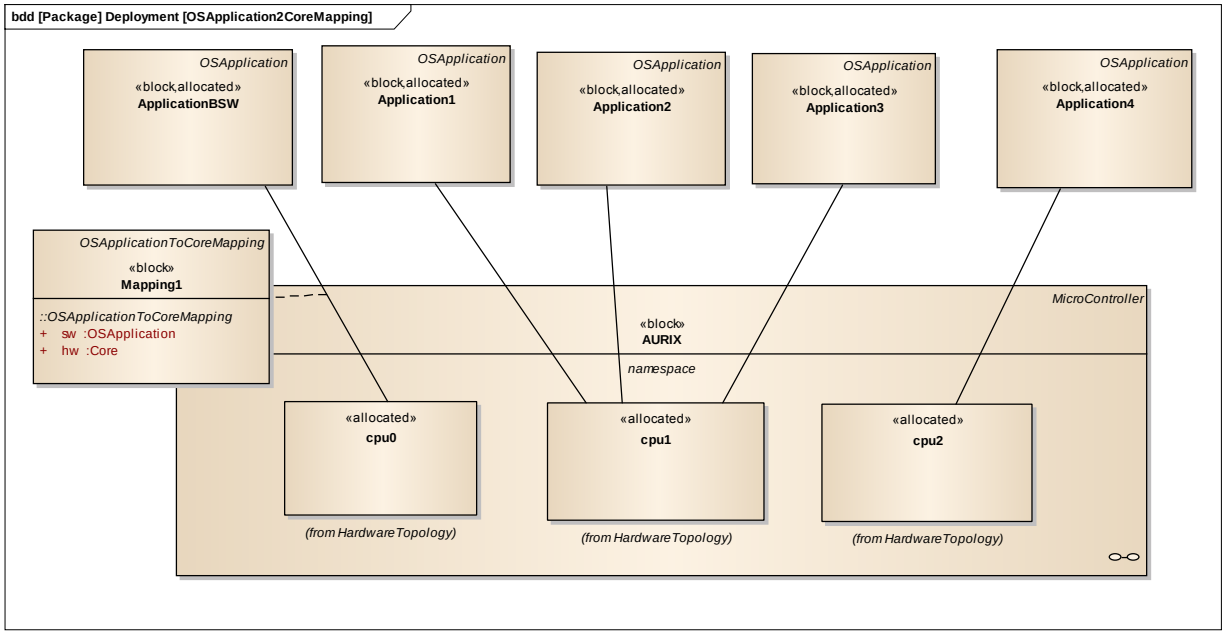

Abbildung 37 Zuordnung von OS-Anwendungen zu ausführbaren Rechenkernen (Systemaspekt "Deployment")

Optimierung der Systemarchitektur

Im Zuge eines Design-Reviews wurde anhand der ARAMIS-Demonstratoren die Vollständigkeit und Anwendbarkeit der logischen und technischen Metamodelle überprüft. Die abschließenden Versionen der Metamodelle finden sich in den entsprechenden EA-Dateien und deren Dokumentation in den ARAMIS-Berichten E2.1.3.9 und E2.1.4.6.

19.1.3 Entwurf neuer Virtualisierungskonzepte

Die Arbeiten unterteilten sich in grundlegende Evaluierungen in den Bereichen Betriebssysteme und Middleware und in den eigentlichen Entwurf der Virtualisierungskonzepte.

19.1.3.1 Evaluierungskriterien und KVM

Die Universität Paderborn definierte in diesem Zusammenhang projektspezifische Kriterien zur Evaluation von Echtzeitsystem-software. Diese fokussieren sich auf die Unterstützung von Multicore-Hardware und die Zertifizierbarkeit der System- und Anwendungssoftware. Der Kriterienkatalog schließt quantitative Leistungskriterien (z.B. Bootzeit, Interruptlatenzen, Kontext-wechselzeiten, Speicherbedarf) und qualitative Kriterien (z.B. Robustheit, unterstützte Prozessoren, Wartbarkeit, Dokumen-tation, Werkzeugunterstützung, Zertifizierung) ein. Die Kriterien bestehen aus domänenübergreifenden- sowie Automobil- und

luftfahrt-spezifischen Teilen. Sie dienten schließlich als Grundlage für die Software-Evaluationen innerhalb von ARAMIS.

Die Universität Paderborn untersuchte außerdem KVM (Kernel-based Virtual Machine) als frei verfügbare Open-Source-Virtualisierungslösung auf Eignung für den Einsatz im Echtzeitbereich. KVM ist als Kernelmodul im Linux Betriebssystem enthalten. Es erweitert Linux um Systemvirtualisierungs-funktionalität zu einem sogenannten *hosted Hypervisor* (Type-2 Hypervisor), wie in Abbildung 38 dargestellt. So wird es möglich mehrere Gastsysteme bestehend aus Betriebssystemkernel und Applikationen durch den Linux-Kernel ausführen zu lassen. Hierbei nutzt KVM für das Management der Gastsysteme die von Linux zur Verfügung gestellten Funktionalitäten, wie Treiber, Scheduling-verfahren und Speicherverwaltung.

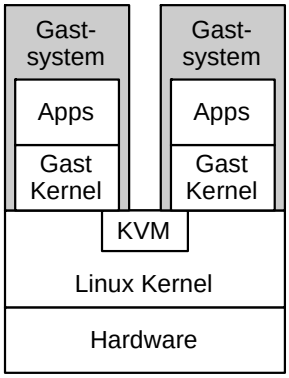

Abbildung 38 System-Virtualisierung mit KVM

Die Untersuchungen ergaben, dass der Einsatz von KVM für eingebettete Echtzeitsysteme nur sehr eingeschränkt möglich ist. Als Type-2-Hypervisor ist ein Linux-Kernel zwingend Teil des Hypervisors. Dies resultiert in einen signifikant höheren Overhead bezüglich Ausführungszeiten, Latenzen und Speicherbedarf im Vergleich zu einem Hypervisor, der nicht auf ein Betriebssystem aufsetzt. Zudem werden nur wenige für eingebettete Systeme geeignete Prozessoren und Betriebssysteme unterstützt. Im Rahmen von ARAMiS sind insbesondere auch die fehlende Zertifizierung von funktionaler Sicherheit, die eingeschränkte Unterstützung zur Partitionierung und die fehlende Konformität mit AUTOSAR und ARINC653 problematisch.

Eine abschließende Dokumentation findet sich in den ARAMIS-Berichten E4.1.2.1, E4.1.2.2 und E4.1.2.3.

19.1.3.2 Entwurf optimierter Virtualisierungskonzepte

Die Universität Paderborn erarbeitete des Weiteren ein Konzept zur Laufzeit-Migration von virtuellen Maschinen zwischen

Steuergeräten unter Einhaltung von Echtzeitbedingungen. Voraussetzung hierfür sind über ein Netzwerk verbundene Steuergeräte, auf denen jeweils eine Hypervisor-Instanz ausgeführt wird (siehe Abbildung 39). Das Konzept erlaubt auch die Fortsetzung der Ausführung eines Gastsystems im Falle bestimmter Hardwarefehler, beispielsweise beim Ausfall von Co-Prozessoren oder von Eingabe-Ausgabe-Geräten mit der Einschränkung, dass die Kommunikation zum Zielsteuergerät noch möglich ist. Insbesondere stellt Migration jedoch eine Technik zur proaktiven Fehlertoleranz dar, mit der man bei Hardware mit integrierten Testschaltungen auf Fehlfunktionen möglichst vor Eintritt des Ausfalls reagieren kann.

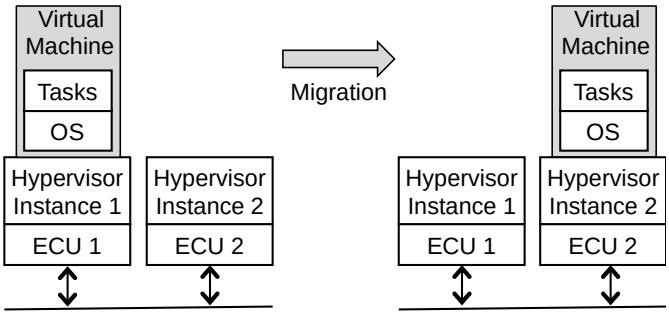

Abbildung 39 Laufzeit-Migration virtueller Maschinen

Das Konzept adressiert die zwei großen Herausforderungen der Laufzeitmigration von Echtzeitsystemen: potentielle Verletzung der Echtzeitanforderungen durch den Transfer zum Zielsteuergerät und Eingliederung in das Scheduling auf dem Zielsteuergerät. Der Entwurf adressiert ferner das Co-Design von Hypervisor und Gastbetriebssystem. Der Ansatz wurde exemplarisch in einen an der Universität Paderborn entwickelten Hypervisor integriert und mit einem ebenso an der Universität Paderborn entwickelten Echtzeitbetriebssystem mit für eingebettete Systeme typischer Hardware evaluiert (PowerPC 405).

In diesem Zusammenhang wurde ferner ein Konzept zum adaptiven Scheduling virtueller Maschinen entwickelt. Die Architektur kombiniert serverbasiertes hierarchisches Echtzeit-Multicore-Scheduling (siehe Abbildung 7) mit einem adaptiven Umschalten zwischen verschiedenen Ausführungsmodi, die durch unterschiedliche Ressourcenzuteilungen charakterisiert sind. Die Lösung garantiert, dass eine definierte Minimalzuteilung nicht unterschritten wird, auch wenn sich mehrere Gastsysteme einen Prozessorkern teilen, was die Zertifizierung des Systems ermöglicht. Als partitioniertes Mehrkern-Scheduling besteht das Konzept aus zwei Teilen:

1. Partitionierung, d.h. Aufteilung der Menge der Gastsysteme in Teilmengen, von denen jede auf einem Prozessorkern ausgeführt wird;
2. Laufzeitscheduling der Gastsysteme, die demselben Prozessorkern zugewiesen wurden.

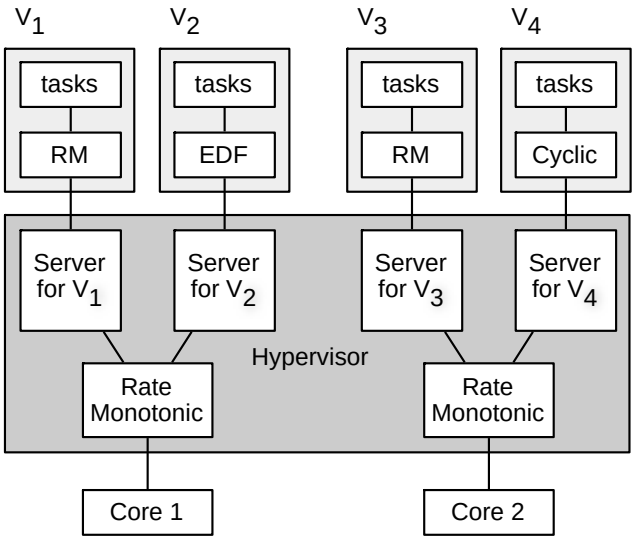

Abbildung 40 Scheduling-Architektur

Ein Prototyp belegt die Machbarkeit des Konzepts mit einem geringen Ausführungszeit- und Speicherbedarfsoverhead. Ferner wurde durch eine Schedulingsimulation die Effektivität des adaptiven Schedulings durch Vergleich mit einer statischen Lösung untersucht. Die Simulationsergebnisse belegen, dass das vorgestellte Konzept dem Bedarf der Gastsysteme deutlich besser gerecht wird und die Prozessorauslastung verbessert.

Zusätzlich wurde in Kooperation mit der Audi AG und der Audi Electronics Venture GmbH der Entwurf und die Implementierung eines Hypervisors für den im Projekt eingesetzten Prozessor Infineon TriCore™ durchgeführt und das Schedulingkonzept anschließend im Hinblick auf von Bosch gelieferten Anforderungen untersucht und erfolgreich validiert.

Die Ergebnisse wurden abschließend in den ARAMIS-Berichten E4.4.4.1 und E4.4.4.2 dokumentiert.

19.2 Notwendigkeit und Angemessenheit der Arbeiten

Das Projektvorhaben ARAMiS formte eine domänenübergreifende Forschungsallianz aus den Mobilitätsdomänen Automotive, Avionik und Bahn mit dem Ziel Mehrkern-Technologie und Virtualisierung für diese Domänen nutzbar, d.h. in zukünftigen Produkten einsetzbar zu machen.

Eine der Hauptziele des Projektes war die Entwicklung eine domänenübergreifenden Reference Technology Plattform (RTP) auf Basis der domänenübergreifenden Definition einer logischen und technischen Rechnerarchitektur. Aus vielzähligen Standards ist bekannt, dass solche Definitionen als Metamodelle in Form von XML-Datenschemata oder UML2-Definitionen festgelegt werden, um sie effizient in Anwendungen der industriellen Praxis zu überführen. Da bei Projektstart nur domänenspezifische Standards (ARINC, AUTOSAR, IP-XACT, CIM) zur Verfügung standen, musste der getätigte Aufwand investiert werden, um die Gemeinsamkeiten der im Einzelnen doch recht komplexen Standards herauszuarbeiten und zu einem gemeinsamen Metamodellkern zusammenzufassen. Da hierfür als Ausgangspunkt der DMTF-Standard CIM zur Verfügung stand, konnte die Tätigkeit in dem geplanten Aufwand mit Reduktion auf die domänenübergreifenden Kernkonzepte durchgeführt werden.

Eine der potentiellen Kerntechnologien bei der domänenüber-greifenden Einführung von Mehrkernprozessoren bildet der Einsatz von zertifizierten virtuellen Maschinen. Hier mussten im Umfeld von Echtzeitsystemen Arbeiten (a) zur Evaluierung verfügbarer Virtualisierungslösungen und deren potentieller Erweiterbarkeit und (b) zur Entwicklung neuer Konzepte, die die momentanen Einschränkungen aufheben, durchgeführt werden. Von fundamentaler Bedeutung ist eine geeignete Ressourcenverwaltung, die die Hardwareressourcen effizient nutzt. Da diese Arbeiten auch für den industriellen Einsatz nutzbar sein sollten, waren hierzu Evaluierungen im industrienahen Umfeld notwendig. Der durchgeführte Zeitrahmen von 3 Jahren entspricht für diese Anwendung dem üblichen Rahmen einer Vorentwicklung.

19.3 Fortschritte auf dem Gebiet des Vorhabens

Die Arbeiten der Universität im ARAMIS-Projekt umfassten die Gebiete der Definition einer domänen-übergreifenden Rechnerarchitektur und der Entwicklung von neuen Virtualisierungskonzepten für mehrkernbasierte Echtzeitsysteme.

Im Bereich der Definition von Architekturen standen in den verwandten Gebieten folgende domänenspezifische Standards zur Verfügung: ARINC 653, AUTOSAR 4.0, IP-XACT IEEE 1685-2009 und DMTF CIM. In ARAMIS wurden die Gemeinsamkeiten der im Einzelnen doch recht komplexen Standards herausgearbeitet und zu einem gemeinsamen Metamodelkern zusammengefasst, der zum einen bzgl. der einzelnen Standardkonzepte kompatibel aber auch offen erweiterbar bzgl. einzelner domänenspezifischen Eigenschaften ist. Zurzeit sind keine Arbeiten bekannt, die die Zusammenführung dieser Standards in dieser Komplexität mit dem Ziel der industriellen Anwendung betrachten. Die Nachhaltigkeit der Arbeiten wurde dadurch erreicht, dass der entwickelte Ansatz

offen für Erweiterungen ist, so dass er je nach Bedarf bzgl. verschiedener anderer CIM-Domänen wie Safety nahtlos ergänzt werden kann. Die Berücksichtigung von Ergänzungen hätte jedoch den Projektrahmen innerhalb von ARAMIS gesprengt.

Des Weiteren entwickelte die Universität Paderborn neue Konzepte zur Laufzeitmigration und zum adaptiven Scheduling von virtuellen Maschinen für Mehrkernarchitekturen. Diese Konzepte wurden auf renommierten internationalen Tagungen (Peer-Review) publiziert, was die Originalität der erzielten Ergebnisse belegt. In den beiden Gebieten sind bis heute lediglich die folgenden konkurrierenden Arbeiten bekannt.

Im Bereich der Migration virtueller Maschinen untersuchten Roy et al. ein Migrationskonzept, welches Service-Level-Agreement-Grenzwerte berücksichtigt, nicht jedoch Echtzeitbedingungen [Roy, A., Ganesan, R., Dash, D., and Sarkar, S.: Reducing Service Failures by Failure and Workload aware Load Balancing in SaaS Clouds. In Proc. of the IEEE/IFIP Dependable Systems and Networks Workshop, 2013].

Im Bereich des Echtzeitschedulings virtueller Maschinen präsentierten Lee et al. ein adaptives Scheduling-Framework, welches wie die von der Universität Paderborn entwickelte Lösung auf periodischen Servern und der Weitergabe von Idle-Zeit basiert. Im Gegensatz berücksichtigen sie jedoch keine unterschiedlichen Kritikalitäten und/oder Fairness bei der Weitergabe der Idle-Zeit. Außerdem verteilt der Ansatz der Universität Paderborn die Schedulingressourcen auch im Falle von Modus-Wechseln [Lee, J., Xi, S., Chen, S., Phan, L. T. X., Gill, C., Lee, I., Lu, C., and Sokolsky, O.: Realizing Compositional Scheduling Through Virtualization. In Proc. of the Real-Time and Embedded Technology and Applications Symposium, 2011].

19.4 Veröffentlichung der Ergebnisse

Im Rahmen des Projektes wurden neben den ARAMIS-Berichten die folgenden Buch- und Konferenz/Workshop-Beiträge mit Bezug auf ARAMIS unter Beteiligung der Universität Paderborn erstellt.

[1] S. Groesbrink, L. Almeida: A Criticality-Aware Mapping of Real-time Virtual Machines to Multi-core Processors. Proceedings 19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2014.

[2] S. Groesbrink, S. Oberthür, D. Baldin: Towards Adaptive Resource Management for Virtualized Real-Time Systems. In: 4th Workshop on Adaptive and Reconfigurable Embedded Systems (CPSWeek 2012), Apr. 2012.

[3] S. Groesbrink, L. Almeida, M. de Sousa, S.M. Petters: Fair Bandwidth Sharing among Virtual Machines in a Multi-criticality Scope. In: 5th Workshop on Adaptive and Reconfigurable Embedded Systems (CPSWeek 2013), Apr. 2013.

[4] S. Groesbrink: On the Homogeneous Multiprocessor Virtual Machine Partitioning Problem. In: International Embedded Systems Symposium, Jun. 2013.

[5] S. Groesbrink, L. Almeida, M. de Sousa, S.M. Petters.: Towards Certifiable Adaptive Reservations for Hypervisor-based Virtualization. Proceedings 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Apr. 2014.

[6] S. Groesbrink: Virtual Machine Migration as a Fault Tolerance Technique for Embedded Real-Time Systems. 8th International Conference on Software Security and Reliability (SERE), Jun. 2014.

[7] W. Mueller, B. Defo, F. Mischkalla: Engineering Standards beyond AUTOSAR – Domain Specific Approaches or Complementary Solutions?. In: 6th AUTOSAR Open Conference, Nov. 2013.

[8] F. Mischkalla, W. Mueller: Efficient Power Intent Validation Using Loosely-Timed Simulation Models: A Non-Invasive Approach, In: 23International Workshop on Power And Timing Modeling, Optimization and Simulation (PATMOS), Oct. 2013.

[9] F. Mischkalla, W. Mueller: Architectural Low-Power Design Using Transaction-Based System Simulation. In: 14th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), Jul. 2014.

[10] F. Mischkalla: Advanced SoC Virtual Prototyping for System-Level Power Planning and Validation. In: 24th International Workshop on Power and Timing Modeling Optimization and Simulation (PATMOS), Sep. 2014.

[11] B. Koppelmann, M. Becker, W. Mueller: Portierung der TriCore-Architektur auf QEMU. In: Workshop Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV), Mar. 2014.

[12] M. Becker, C. Kuznik, W. Mueller: Fault Effect Modeling in a Heterogeneous SystemC Virtual Platform Framework for Cyber-Physical Systems. In: International Conference on Cyber-Physical Systems (ICCPS), Apr. 2014.

[13] M. Becker, C. Kuznik, W. Mueller: Virtual Platforms for Model-Based Design of Dependable Cyber-Physical System Software. In: 17th Euromicro Conference on Digital Systems Design (DSD), Aug. 2014.

[14] C. Kuznik, B. Defo, W. Mueller. Semi-automatische Generierung von Überdeckungsmetriken mittels methodischer Verifikationsplan Verarbeitung. In: Workshop Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV), Mar. 2014.

[15] C. Kuznik, W. Mueller: Verific-MM: Systematized Verification Generation with UCIS for Improved Automation on Verification Closure. In: Conference on Design, Automation and Test in Europe (DATE), Mar. 2014.