

# Schlussbericht Verbundvorhaben ARTEMIS Cross-Domain Architecture (ACROSS)

---

**Projekttitel:** ARTEMIS Cross-Domain Architecture  
**Projekt-Akronym:** ACROSS  
**Laufzeit des Vorhabens:** 01.04.2010 – 31.7.2013  
**Berichtszeitraum:** 01.04.2010 – 31.7.2013  
**Zuwendungsempfänger** EADS Defence Electronics (EADS-DE, FKZ: 01IS10003D)  
EADS Innovation Works (EADS-IW, FKZ: 01IS10003C)  
fortiss GmbH (fortiss, FKZ: 01IS10003A)  
Lauterbach GmbH (Lauterbach, FKZ: 01IS10003G)  
SYSGO AG (SYSGO, FKZ: 01IS10003E)

## Autoren:

Dr. Bernd Koppenhöfer  
EADS Deutschland GmbH - CASSIDIAN (ehemals  
EADS Defence Electronics)  
Wörthstr. 85  
89075 Ulm  
E-Mail: [bernd.koppenhoefer@cassidian.com](mailto:bernd.koppenhoefer@cassidian.com)

Dr. Dietmar Geiger  
EADS Deutschland GmbH - CASSIDIAN (ehemals  
EADS Defence Electronics)  
Wörthstr. 85  
89075 Ulm  
E-Mail: [dietmar.geiger@cassidian.com](mailto:dietmar.geiger@cassidian.com)

Dr. Michael Paulitsch  
EADS Deutschland GmbH - Innovation Works  
81663 München  
E-Mail: [michael.paulitsch@eads.net](mailto:michael.paulitsch@eads.net)

Simon Barner  
fortiss GmbH  
(Projektkoordinator für Deutschland)  
Guerickestraße 25  
80805 München  
E-Mail: [barner@fortiss.org](mailto:barner@fortiss.org)

Ingo Rohloff  
Lauterbach GmbH  
Altlaufstr. 38  
85635 Höhenkirchen-Siegertsbrunn  
E-Mail: [ingo.rohloff@lauterbach.com](mailto:ingo.rohloff@lauterbach.com)

Dr.-Ing. Henrik Theiling  
SYSGO AG  
Am Pfaffenstein 14  
55270 Klein-Winternheim  
E-Mail: [hth@sysgo.com](mailto:hth@sysgo.com)

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS10003A, C-E, G gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

## Inhalt

Inhalt.....	2
1 Kurze Darstellung .....	4
1.1 Aufgabenstellung.....	4
1.2 Voraussetzungen .....	5
1.3 Planung und Ablauf des Vorhabens .....	6
1.3.1 Arbeitspakete .....	6
1.3.2 Projektablauf .....	9
1.3.3 Meilensteine.....	10
1.3.4 Begutachtungen .....	11
1.4 Ausgangslange.....	11
1.4.1 Wissenschaftlicher und technischer Stand .....	11
1.4.2 Verwendete Informations- und Dokumentationsdienste .....	12
1.4.3 Referenzen .....	12
1.5 Zusammenarbeit mit anderen Stellen.....	14
1.5.1 Zusammenarbeit im ACROSS Projekt .....	14
1.5.2 Zusammenarbeit mit weiteren Forschungsprojekten .....	16
1.5.3 Zertifizierungsbehörden .....	16
2 Eingehende Darstellung .....	17
2.1 Verwendung der Zuwendung und Ergebnisse .....	17
2.1.1 WP1 (ACROSS MPSoC).....	17
2.1.2 WP2 (Middleware and System Components) .....	18
2.1.3 WP3 (Development Methodology) .....	21
2.1.4 WP4 (Automotive).....	27
2.1.5 WP5 (Aerospace).....	30
2.1.6 WP6 (Industrial Control).....	39
2.1.7 WP7 (Dissemination & Exploitation) .....	46
2.1.8 WP8 (Project management).....	48
2.2 Übersicht über zahlenmäßigen Nachweis.....	49
2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit.....	49
2.4 Nutzen und Verwertbarkeit der Ergebnisse .....	50
2.5 Fortschritt auf dem Gebiet des Vorhabens an anderen Stellen.....	52
2.6 Veröffentlichungen des Ergebnisses .....	53

2.7	Fortschreibung des Verwertungsplanes.....	54
3	Partnerbeiträge .....	58
3.1	EADS-DE.....	58
3.2	EADS-IW.....	58
3.3	fortiss.....	58
3.4	Lauterbach.....	59
3.5	SYSGO .....	59
	Berichtsblatt .....	60
	Document Control Sheet.....	62

## 1 Kurze Darstellung

### 1.1 Aufgabenstellung

Eingebettete Systeme realisieren als Steuergeräte in unterschiedlichen Anwendungsdomänen oftmals sicherheitskritische Funktionen (z.B. Automobil, Luftfahrt, Industrieautomatisierung). Trotz prinzipiell ähnlicher Anforderungen gestalten sich der Austausch und die Wiederverwendung der verwendeten Technologie zwischen den verschiedenen Anwendungsdomänen derzeit schwierig. Dies betrifft sowohl die Hardware (insbesondere die in den Steuergeräten verbauten Prozessoren) als auch dadurch bedingt die zugehörige Software-Landschaft (Systemsoftware sowie Entwicklungswerkzeuge). Die Problematik wird durch unterschiedliche Stückzahlen (und damit Kostenstrukturen), Produktlebenszyklen und domänen-spezifische Anforderungen (z.B. Sicherheitsbestimmungen) hervorgerufen.

Das ACROSS-Projekt zielt darauf ab, die im Rahmen des GENESYS EU-Projektes definierte Referenz-Architektur sowie das zugehörige Ökosystem zu verfeinern und zu implementieren. Hierzu zählen neben einer FPGA-Implementierung der Architektur die entsprechenden Softwarekomponenten (Middleware, Betriebssystem) sowie die zugehörigen Entwicklungswerkzeuge. Durch den gewählten Ansatz wird die Wiederverwendung von Hardware und Software für die die Umsetzung sicherheitskritischer Funktionen über Domänengrenzen hinweg ermöglicht. Der modulare Ansatz ermöglicht somit eine Steigerung der Stückzahlen der verwendeten generischen Komponenten und erhöht damit deren Robustheit bei gleichzeitiger Senkung der Kosten. Die Skalierbarkeit des Ansatzes trägt zusammen mit dem werkzeuggestützten Entwicklungsansatz zur Senkung der Systemkomplexität und der Entwicklungszeit bei.

Der Ansatz wird im Projekt durch die Umsetzung typischer Anwendungen aus den oben genannten Anwendungsdomänen demonstriert und validiert.

ACROSS wurde von den nationalen Fördergebern Bundesministerium für Bildung und Forschung (Deutschland), Ministère de l'Économie, de l'industrie et de l'emploi (Frankreich), Ministero dell'Istruzione, dell'Università e della Ricerca (Italien), dem Bundesministerium für Verkehr, Innovation und Technologie (Österreich) sowie dem ARTEMIS Joint Undertaking gefördert.

Das ACROSS-Projekt hatte eine Laufzeit von 3 Jahren und 4 Monaten, und einen Gesamtumfang von 16 Mio. €. Wie Tabelle 1 zu entnehmen ist, waren insgesamt 18 europäische Partnerfirmen und Institute an ACROSS aus vier Mitgliedsstaaten der Europäischen Union (Deutschland, Frankreich, Italien, Österreich) am Vorhaben beteiligt.

Die Beiträge des in Deutschland durchgeführten Verbundprojektes zum Gesamtprojekt ACROSS fokussierten dabei die Definition und Umsetzung der Software-Landschaft (insbesondere Betriebssystem sowie Entwicklungswerkzeuge) sowie die Entwicklung und Auswertung der Demonstratoren in den Domänen Luftfahrt, Automobil und Industrieautomatisierung.

Nr.	Name des Partners	Kürzel	Land	Projektbeginn	Projektende
1	Technische Universität Wien (Koordinator)	TU Wien	AT	M1	M40
2	AVL List GmbH	AVL	AT	M1	M40
<b>3</b>	<b>EADS Deutschland GmbH - Innovation Works</b>	<b>EADS-IW</b>	<b>DE</b>	<b>M1</b>	<b>M40</b>
4	TTTech Computertechnik AG	TTT	AT	M1	M40
5	Thales SA	TRTFR	FR	M1	M40
6	SELEX Sistemi Integrati S.P.A	SELEX-SI	IT	M1	M40
7	Danube Mobile Communications Engineering GmbH & Co KG	DMCE	AT	M1	M40
8	Siemens Aktiengesellschaft Österreich	SAGÖ	AT	M1	M40
9	European Aeronautic Defense & Space Company EADS France	EADS-FR	FR	M1	M40
<b>10</b>	<b>fortiss GmbH</b>	<b>fortiss</b>	<b>DE</b>	<b>M1</b>	<b>M40</b>
11	Université Joseph Fourier Grenoble 1	UJF/Verimag	FR	M1	M40
12	Universität Augsburg	UAU	DE	M1	M18
<b>13</b>	<b>EADS Deutschland GmbH Defence Electronics</b>	<b>EADS-DE</b>	<b>DE</b>	<b>M1</b>	<b>M40</b>
<b>14</b>	<b>SYSGO AG</b>	<b>SYSGO</b>	<b>DE</b>	<b>M1</b>	<b>M40</b>
<b>15</b>	<b>Lauterbach GmbH</b>	<b>Lauterbach</b>	<b>DE</b>	<b>M1</b>	<b>M40</b>
16	Thales Communications S.A.	TCF	FR	M1	M40
17	PrismTech France SARL	PrismTech		M1	M40
18	CoFluent Design	CoFluent Design		M1	M19

Tabelle 1: Das ACROSS-Konsortium. Fettdruck: Deutsche Partner, deren Teilvorhaben in diesem Bericht beschrieben sind.

## 1.2 Voraussetzungen

Der schrumpfenden Strukturgrößen in der Halbleiter-Technik und der damit einhergehende Trend zu Multicore-Systemen (Vermeidung thermischer Problem; Nutzung der zur Verfügung stehenden Chip-Fläche; ...), ist für Entwickler aller Systemklassen eine anspruchsvolle Aufgabe. Allerdings stellt diese Entwicklung für Systeme, die zur Realisierung sicherheitskritischer Funktionen zum Einsatz kommen, eine besondere Herausforderung dar.

Die derzeit auf dem Markt befindlichen Multicore-Prozessoren sind auf die Leistung in nicht-sicherheitskritischen Anwendungen optimiert und weisen somit Eigenschaften auf, die deren Einsatz in sicherheitskritischen Anwendungen erschweren. Dabei erschweren die genutzten Techniken insbesondere die Vorhersagbarkeit des Laufzeitverhaltens der auf dem Prozessor ausgeführten Anwendungen, und damit auch die für die sichere und zertifizierbare Umsetzung kritischer Algorithmen (z.B. Regelungen) benötigte Garantie von maximalen Ausführungszeiten (Einhaltung von „Deadlines“).

Durch die zunehmende Verbreitung von Multicore-Prozessoren in absatzstärkeren Märkten wie etwa der Unterhaltungselektronik zeichnet sich bereits jetzt ab, dass es langfristig zu einer Ablösung der herkömmlichen Singlecore-Prozessoren kommen wird. Das Dilemma wird durch die z.T. sehr langen

Produktlebenszyklen verschärft, die in einzelnen Anwendungsdomänen wie etwa der Luftfahrt, aber auch in Bereichen der Automatisierungstechnik, mehrere Jahrzehnte betragen können, und somit eine langfristige Verfügbarkeit der verwendeten Komponenten voraussetzt.

Bereits hieraus ergibt sich direkter Handlungsbedarf, um Multicore-Prozessoren für sicherheitskritische Systeme einsetzbar zu machen. Darüber hinaus bieten diese Systeme aufgrund ihrer Leistungsfähigkeit trotz der oben skizzierten Probleme die Chance, das in Abschnitt 1.1 beschriebenen Ziel einer skalierbaren, domänen-übergreifend einsetzbaren Architektur zur verwirklichen, mit der sich die Komplexität heutiger und zukünftiger Systeme bewältigen lässt.

ACROSS setzt dabei am Problem der fehlenden zeitlichen Vorhersagbarkeit an und führt hierzu ein neuartiges zeitgesteuertes On-Chip-Verbindungsnetzwerk (time-triggered network-on-chip – TTNoC) ein. Durch die Vermeidung von geteilten Speichern gewährleistet der Ansatz zudem die räumliche Trennung von Anwendungen, und ist somit für die Integration mehrerer Anwendungen mit unterschiedlichen Sicherheitsanforderungen auf einen MPSoC-Prozessor geeignet, was effizientere Steuerungen ermöglichen wird (z.B. Reduktion von Energiebedarf, Gewicht und Bauraum).

Das deutsche Konsortium brachte zur Umsetzung dieses Ansatzes die folgenden Technologien ein:

- EADS-IW und EADS-DE brachten die in der Luft- und Raumfahrt üblichen Methoden zur Sicherstellung der Unabhängigkeit einzelner Funktionen ein. Diese Methoden basieren auf den Standards ARP 4754A, ARP4761 sowie DO-178 und DO-254. Das TTNoC wurde gemäß diesen Methoden entworfen.
- fortiss stellte generische Verfahren zur Modell-zu-Modell Transformation und Code-Generierung, Verfahren zur Synthese von Fehlertoleranzmechanismen sowie ein Modellierungswerkzeug für die Domäne Industrieautomation zur Verfügung. Auf dieser Basis wurde eine modellgetriebene Werkzeugkette zur Konfigurations- und Anwendungsentwicklung für die ACROSS-Plattform erstellt.
- Lauterbach stellte seine TRACE32 Produkte zum Debuggen und Tracen eines Systems zur Verfügung. Im Rahmen des ACROSS Projekts wurden die TRACE32 Produkte erweitert, um die von Lauterbach für ACROSS spezifizierten Debug- und Trace-Schnittstellen zu unterstützen.
- Die SYSGO AG stellte im Rahmen des Projektes ihr virtualisierendes Echtzeitbetriebssystem PikeOS den Partnern zur Verfügung. Zu diesem Zweck musste das Betriebssystem auf die im ACROSS-Projekt verwendete NIOS-2-Hardware portiert werden. Neben der Anpassung an die Architektur sollte PikeOS auf die Besonderheiten des Time-Triggered-Netzwerkes eingehen. Für die einzelnen Domains wurde PikeOS speziell angepasst, so dass für den NIOS-2 sowohl APEX für die Avionik, POSIX für die Automatisierung und AUTOSAR für Automotive verfügbar wurden.

## 1.3 Planung und Ablauf des Vorhabens

### 1.3.1 Arbeitspakete

Die Planung und der Ablauf des deutschen Teilvorhabens orientieren sich direkt an der Beteiligung der deutschen Partner am Gesamtprojekt. Das Projekt ist in sechs technische Arbeitspakete (engl. *work package*, im Folgenden WP) unterteilt, die von einem Verbreitungs- und Verwertungsarbeitspaket (WP7, siehe Abschnitt 2.1.7) sowie einem Management-Arbeitspaket (WP8, siehe Abschnitt 2.1.8) unterstützt werden.

Die Beiträge der deutschen Teilvorhaben zu den einzelnen Arbeitspaketen sind in Abschnitt 2.1 detailliert ausgeführt. Hierbei lieferte WP1 das ACROSS MPSoC (siehe Abschnitt 2.1.1) als Basis für die Arbeiten des deutschen Konsortiums, dessen technische Arbeiten in den Technologie-Arbeitspaketen WP2 (*Middleware and System Components*, siehe Abschnitt 2.1.2) und WP3 (*Development Methodology*, siehe Abschnitt 2.1.3) sowie den Demonstrations-Arbeitspaketen WP4 (Automotive, siehe Abschnitt 2.1.4), WP5 (Aerospace, siehe Abschnitt 2.1.5) und WP6 (*Industrial Control*, siehe Abschnitt 2.1.6) erfolgten. Dabei wurden für jedes Arbeitspaket konkrete Teilziele definiert, die zusammen die Gesamtziele von ACROSS ausmachen. Die Ausgangslage für das ACROSS-Projekt wurde im Projekt GENESYS [A-30] erarbeitet. Für die deutschen Teilvorhaben sind der dort erfolgte Entwurf der Systemarchitektur für die Arbeiten an der Middleware und Systemsoftware in WP2 relevant sowie die dort beschriebene modellgetriebene Entwicklungsmethodik für die Arbeiten in WP3 relevant.

Die ACROSS-Systemarchitektur ist von einer stabilen Schnittstelle von generischen Basisdiensten dominiert, die von verschiedenen Implementierungen realisiert werden kann (siehe Abbildung 1). Diese Basisdienste, die im Projekt von einer FPGA-Implementierung des ACROSS-MPSoC zur Verfügung gestellt werden, werden zunächst durch generische (d.h. domänen-übergreifend einsetzbare) Dienste aus WP2 abstrahiert und erweitert (z.B. Betriebssystem). Darüber hinaus werden letztere von domänen-spezifischen (wiederverwendbaren) Dienste ergänzt (in WP4, WP5, WP6), auf deren Basis die jeweiligen Applikationen aufbauen. Als Vorbild für diese Architektur kann das erfolgreiche IP-Protokoll angesehen werden, das über verschiedenste physikalische Transportprotokolle verwendet werden kann und darüber hinaus sowohl generische (TCP und UDP) als auch domänen- und applikationsspezifische Abstraktionsschichten anbietet.

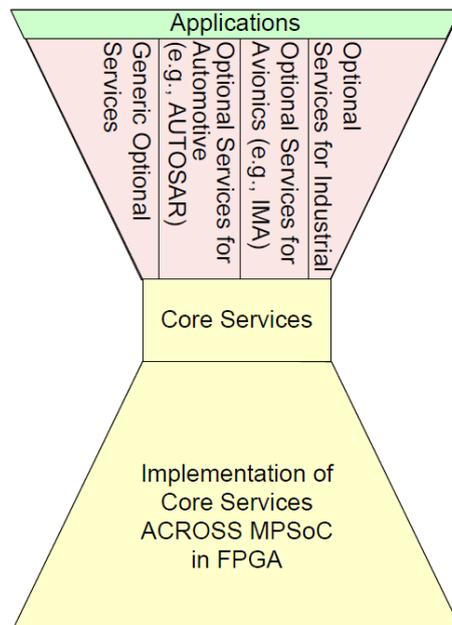


Abbildung 1: Die ACROSS „Wespentailen-Architektur“

Die technischen Beiträge der von diesem Bericht abgedeckten deutschen Teilvorhaben sind dabei in den Arbeitspaketen WP2, WP3, WP4, WP5 und WP6 angesiedelt:

- Die wesentlichen Resultate von **WP2** im deutschen Verbundvorhaben sind Komponenten der generischen optionalen Dienste aus der ACROSS-Architektur. Hierzu zählen einerseits eine

auf das ACROSS MPSoC zugeschnittene Version des PikeOS-Betriebssystems von SYSGO (Portierung auf Altera NIOS-II Architektur; Unterstützung für spezifische Fähigkeiten des ACROSS MPSoCs). Andererseits wurden Software-Komponenten zur Sicherstellung von Fehlertoleranz und Robustheit sowie Diagnosewerkzeuge und Diagnosemodule erstellt und getestet. Zur Anbindung der Diagnosekomponenten an bestehende Debugging-Werkzeuge wurden eine Debug-Schnittstelle und eine Trace-Schnittstelle definiert. Das detaillierte Vorgehen und die alle Resultate sind in Abschnitt 2.1.2 beschrieben.

- Die Resultate von **WP3** sind die Definition einer modellgetriebene Entwicklungsmethodik sowie die Implementierung entsprechender Entwicklungswerkzeuge. Diese kommen bei der Entwicklung, Analyse und Verifikation von ACROSS-basierten Anwendungen zum Einsatz, und decken daher auch die Konfiguration der HW/SW-Dienste ab, die die ACROSS-Architektur ausmachen (vgl. Abbildung 1). Für die Koordination dieses Arbeitspaketes war fortiss als federführender Partner zuständig. Das detaillierte Vorgehen und alle Resultate sind in Abschnitt 2.1.3 beschrieben.
- In den Anwendungs-Arbeitspaketen **WP4**, **WP5** und **WP6** haben die deutschen Verbundpartner zum einen domänen-spezifische Dienste in Form von Softwarekomponenten (z.B. AUTOSAR, IMA) und Entwicklungswerkzeuge (z.B. für IEC 61131-3) beigesteuert. Zum anderen war das deutsche Konsortium direkt an der Umsetzung von Demonstrator-Anwendungen in den Domänen Luftfahrt (WP5; Federführung beim Degraded Vision Landing Aid Demonstrator (Demonstrator 5-B)) sowie dem Industrial Control Demonstrator (WP6; Entwicklungswerkzeuge und Anwendungsentwicklung) beteiligt. Auf der Basis dieser Demonstratoren wurden auch in den entsprechenden Teilvorhaben die Auswirkungen der Verfügbarkeit der ACROSS-Technologie auf die jeweiligen Domänen untersucht (*impact analysis* in WP5 und WP6). In WP4 wurde eng an der Umsetzung des Automotive-Demonstrators mitgearbeitet, u.a. durch eine effiziente AUTOSAR-OS-Anbindung. Das detaillierte Vorgehen und alle Resultate sind in den Abschnitten 2.1.4, 2.1.5 und 2.1.6 beschrieben

Abbildung 2 fasst den Arbeitsfluss zusammen und verdeutlicht den Informationsaustausch zwischen den entsprechenden Arbeitspaketen.

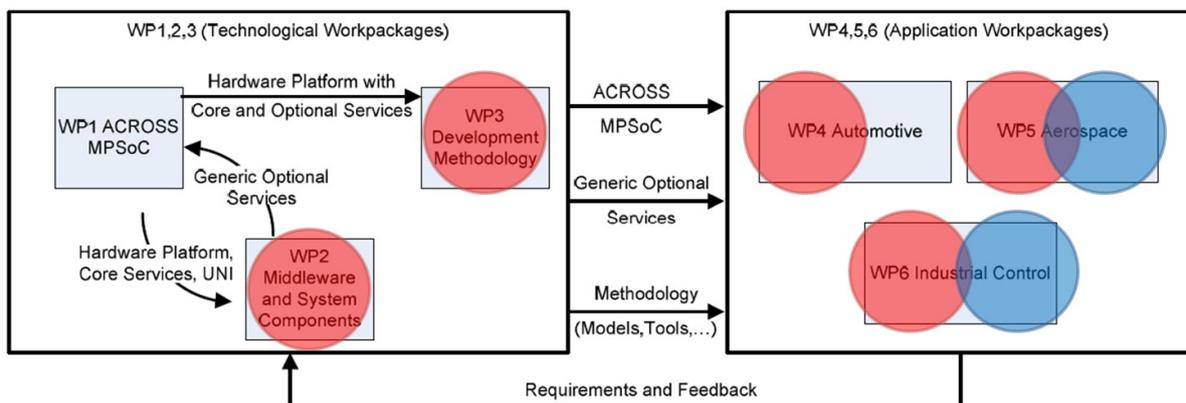


Abbildung 2: Beiträge des deutschen Verbundprojektes zu den technischen Zielen von ACROSS: Design und Implementierung der Technologie (rot) sowie Entwicklung und Auswertung der Demonstrator-Szenarien (blau).

### 1.3.2 Projektablauf

Aus der Zusammenfassung des zeitlichen Ablaufs des Projektes in Abbildung 3 geht hervor, dass das Projekt in drei Phasen unterteilt war. In der ersten Phase (Projektmonate M1 = April 2010 bis M6) wurden zunächst eine Anforderungsanalyse (M1-M2) in den Technologie-Arbeitspaketen durchgeführt. Auf dieser Basis wurde eine Spezifikation der zu entwickelnden Komponenten und Demonstratoren erstellt (M3-M6). In der zweiten Phase (M7-M23) wurde die Spezifikation zu einem Design verfeinert, das schließlich in entsprechende Hard- und Softwareprototypen umgesetzt wurde. In Projektmonat M24 startete mit Phase 3 der Demonstrator-Aufbau, so dass durch den Überlapp mit Phase 2 direkte Rückmeldung zu den jeweiligen Implementierungen gegeben werden konnte (Fehlerberichte, etc.). Schließlich endete Phase 3 mit der Auswertung der Demonstratoren (M40 = Juli 2013). Eine detaillierte Aufschlüsselung der Aktivitäten innerhalb der einzelnen Arbeitspakete erfolgt in Abschnitt 2.1.

Die oben ausgeführte Zusammenfassung des Projektablaufes berücksichtigt dabei bereits eine im Verlauf des ACROSS-Projektes (Ende 2012) beantragte kostenneutrale Verlängerung der Laufzeit um 4 Monate (ursprünglich geplante Laufzeit: 36 Monate). Der Hauptgrund für die Beantragung einer Laufzeitverlängerung für das Projekt ACROSS lag darin, dass die Komplexität der Entwicklung einer stabilen Version des ACROSS MPSoC (WP1) signifikant höher als erwartet war. Zudem erwies sich die Fehlerbehebung ebenfalls als sehr komplex. Gemeinsam mit ALTERA mussten Fehler in ALTERA Produkten (FPGA Entwicklungs-Werkzeugkette) behoben werden, um bei der Implementierung des MPSoCs den FPGA bis an seine Grenze auslasten zu können. Hierbei sei insbesondere darauf hingewiesen, dass ALTERA kein Mitglied des ACROSS-Konsortiums ist.

Da das MPSoC der zentrale Gegenstand des Projektes ist (vgl. Abbildung 2), wurden die Arbeiten in etlichen anderen Arbeitspaketen blockiert, an denen auch das deutsche Konsortium beteiligt ist. Insbesondere sind hier die zugehörige Low-Level-Software (z.B. Treiber), die Entwicklung von VHDL Modulen für Diagnoseservices, sowie das verwendete Betriebssystem anzuführen (WP2). Diese Verzögerungen bei diesen zentralen Komponenten wirkten sich folglich entsprechend auf die Arbeiten an der Entwicklungsmethodik (WP3) sowie der Demonstrator-Anwendungen (WP4, 5, 6) aus.

An dieser Stelle sei auch auf den Schlussbericht der Universität Augsburg verwiesen, die ihre Mitarbeit am Projekt nach Projektmonat M18 beendete. Der Grund dafür war, dass der erfolgreich an das ACROSS MPSoC angebundene CarCore-Prozessor entgegen der ursprünglichen Planung von keinem der Demonstratoren verwendet wurde. Somit führte Universität Augsburg nur die bis zu diesem Zeitpunkt geplanten Arbeiten durch.

Der französische Partner CoFluent Design schied aus dem Projekt aus, nachdem er durch Intel übernommen worden war. Daher konnte die in WP3 geplante Anbindung des ESL-Werkzeuges (ESL = Electronic System Level) nur skizziert werden. Stattdessen koordinierte fortiss als federführender Partner in WP3 die von SAGÖ durchgeführte Studie mit alternativen ESL-Werkzeugen.

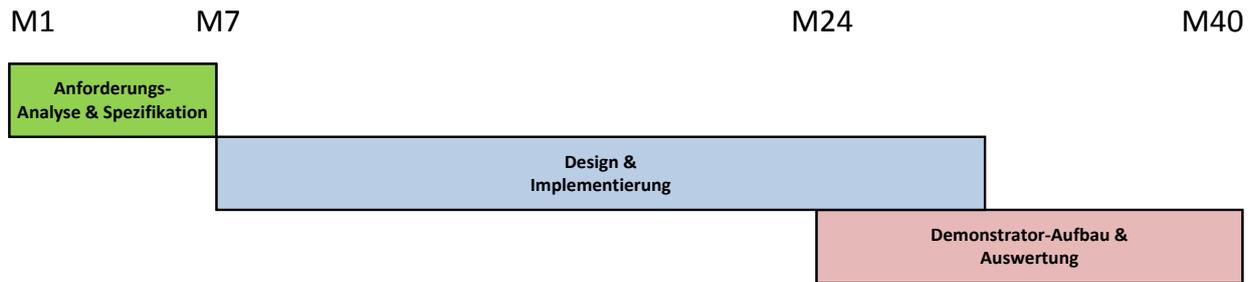


Abbildung 3: Projektphasen

### 1.3.3 Meilensteine

Tabelle 2 führt die zur Kontrolle des Projektfortschrittes definierte Meilensteine auf. Hieraus geht hervor, dass mit Ausnahme von Meilenstein m6 (Projektmonat M23) das deutsche Verbundvorhaben für die Erfüllung aller Meilensteine mit verantwortlich war. Wie in Abschnitt 1.3.2 erläutert haben technische Schwierigkeiten in WP1 dazu geführt, dass der Zeitpunkt für die Erfüllung von m6 nicht gehalten werden konnte. In der dadurch bedingten Verlängerung der Projektlaufzeit wurden daher die Fristen der nachfolgenden Meilensteine entsprechend angepasst (Fristen der ursprünglichen Planung jeweils in Klammern).

Nr.	Meilenstein-Name	Involvierte Arbeitspakete	Fertigstellung	Erfolgskontrolle
<b>m1</b>	Konsolidierte Anforderungen	WP1, 2, 3, 4, 5, 6	M2	Analyse: Vollständigkeit, Widersprüche
<b>m2</b>	Spezifikation der ACROSS-Dienste	WP1, 2, 4, 5, 6	M6	Analyse: Erfüllung der Anforderungen
<b>m3</b>	Spezifikation des Entwicklungsprozesses	WP3	M6	Analyse: Erfüllung der Anforderungen
<b>m4</b>	Erster Entwurf der Implementierung des ACROSS MPSoC und der generischen Middleware	WP1, WP2	M17	Erster Labor-Prototyp
<b>m5</b>	Generische Middleware	WP2	M32 (M29)	Voll funktionstüchtiger Labor-Prototyp
<b>m6</b>	Finale Version des MPSoCs	WP1	M23	Voll funktionstüchtiger Labor-Prototyp
<b>m7</b>	Demonstratoren auf finaler Plattform implementiert	WP4,5,6	M40 (M36)	Demonstratoren vollständig aufgebaut und funktionstüchtig
<b>m8</b>	Resultate der Auswirkungsanalyse	WP4,5,6	M34 (M32)	Analyse: Verfügbarkeit der Informationen, die zur Verwertung von ACROSS benötigt sind
<b>m9</b>	Resultate des Demonstrator-Auswertung	WP4,5,6	M40 (M36)	Demonstrator-Experimente
<b>m10</b>	Verfügbarkeit der domänen-spezifischen Dienste	WP4,5,6	M36 (M30)	Verfügbarkeit der Software und Test durch Anwender

Tabelle 2: Meilensteine: Die Fristen beziehen sich auf die Projektmonate gemäß der verlängerten Projektlaufzeit (ursprünglich geplante Fristen jeweils in Klammern).

### 1.3.4 Begutachtungen

Zur Überwachung des Projektfortschrittes wurden drei Begutachtungen mit unabhängigen Gutachtern durchgeführt (siehe Tabelle 3).

Nr.	Datum	Ort	Berichtszeitraum	Gutachter
1	26. Mai 2011	Brüssel, Belgien	M1-M12	Prof. Miroslaw Malek, Humboldt-Universität zu Berlin Prof. Yury Sheynin, St. Petersburg State University of Aerospace Instrumentation
2	22. Mai 2012	Brüssel, Belgien	M13-M24	Prof. Yury Sheynin, St. Petersburg State University of Aerospace Instrumentation  Mr. Zlatko Petrov, Honeywell
3	25. Juli 2013	Brüssel, Belgien	M25-M40	Prof. Yury Sheynin, St. Petersburg State University of Aerospace Instrumentation  Mr. Zlatko Petrov, Honeywell

Tabelle 3: Begutachtungen des ACROSS-Projektes

Wie den Berichten zu den einzelnen technischen Begutachtungen zu entnehmen ist, wurden alle zu erbringenden Deliverables (Berichte, Software, Demonstratoren) durch die Gutachter akzeptiert. Insbesondere legt der Bericht zur abschließenden Begutachtung dar, dass die Projektziele bis auf kleinere Ausnahmen erreicht werden konnten, betont die Qualität der Berichte, die Effizienz der Demonstratoren sowie die sehr gute Zusammenarbeit innerhalb und außerhalb des Projektes.

## 1.4 Ausgangslage

### 1.4.1 Wissenschaftlicher und technischer Stand

Bislang werden Multicore-Systeme aus den in Abschnitt 1.2 genannten Gründen von wenigen Ausnahmen abgesehen nicht in sicherheitskritischen Echtzeitsystemen eingesetzt. Derzeit finden in der Industrie entsprechende Vorfeldforschungen statt (z.B. Verwendung des zwei-kernigen Infineon TriCore-Prozessoren im Automobil-Bereich). Im akademischen Umfeld werden daher in verschiedenen Projekten neuartige echtzeitfähige Multicore-Prozessor-Architekturen (z.B. MERASA, parMERASA (beide FP7)) bzw. Techniken zur Anwendbarkeit herkömmlicher Architekturen wie etwa Virtualisierungs-Techniken (z.B. MultiPARTES (FP7), ARAMiS (BMBF)) erforscht.

Die ACROSS-Plattform baut auf einer Komponenten-basierten Architektur auf, in der über ein zeitgesteuertes On-Chip-Verbindungsnetzwerk<sup>1</sup> kommuniziert wird, wodurch die inhärenten Probleme von Multicore-Systemen in sicherheitskritischen Anwendungen vermieden werden. Die Grundlage für diesen Ansatz, der die Skalierbarkeit und domänen-übergreifende Einsetzbarkeit sicherstellt, wurde dabei im Projekt GENESYS(FP7) gelegt [A-30], der im INDEXYS-Projekt (ARTEMIS) verfeinert wurde. Das ACROSS-Projekt geht über die genannten Ansätze hinaus, indem es die spezifizierte Architektur bis zu einer Referenzimplementierung verfeinert (FPGA-Implementierung des MPSoC, Systemsoftware).

Auch für die Entwicklungsmethodik wurde der konzeptionelle Grundstein im GENESYS-Projekt gelegt [A-30]. Darüber hinaus flossen Entwicklungen zur modellgetriebenen Entwicklung fehlertoleranter

<sup>1</sup> Time-triggered Network-On-Chip (TTNoC)

Systeme (BMBF-Projekt Zerberus) sowie zu Werkzeugen für die Industrieautomation (BMBF-Projekt EasyKit) ein [A-36]. Wie eine Reihe von weiteren Forschungsprojekten und Ansätzen zur modellgetriebenen Entwicklung eingebetteter Systeme (z.B. ARTEMIS-Projekt CESAR), berücksichtigten die oben angeführten Ansätze nicht die spezifischen Eigenschaften von Multicore-Plattformen. Für die werkzeuggestützte Entwicklungsmethodik für die ACROSS-Plattform wurde diese Grundlage entsprechend erweitert. So konnte eine skalierbare Werkzeugkette realisiert werden, die generische und domänen-spezifische Arbeitsflüsse auf verschiedenen Abstraktionsgraden unterstützt, und die durch geeignete Plattformmodelle sowie Code- und Konfigurationsgeneratoren dazu beiträgt, die Komplexität des ACROSS-MPSoC für den Entwickler beherrschbar zu machen.

Für einige Erweiterungen am Betriebssystemkern von PikeOS gab es Vorarbeiten aus dem INTERESTED-Projekt (FP7), in dem zusammen mit TTEch ein externer Netzwerk-Bus mit PikeOS synchronisiert wurde. Dies waren die ersten Erfahrungen von SYSGO mit Time-Triggered-Bussen. Aufbauend auf den Erkenntnissen wurden der sehr anders arbeitende TTNOC im ACROSS-Projekt untersucht und neue Lösungen für das Betriebssystem erarbeitet.

#### 1.4.2 Verwendete Informations- und Dokumentationsdienste

Bei der Planung und Durchführung des Verbundvorhabens wurden die üblichen Informations- und Dokumentationsdienste verwendet, um auf die relevante Fachliteratur zuzugreifen. Im Folgenden eine Auswahl:

- Universitätsbibliotheken (TU München, ...) inkl. elektronischer Recherche-Dienste
- IEEE Xplore Digital Library (<http://ieeexplore.ieee.org/>)
- ACM Digital Library (<http://dl.acm.org/>)
- SpringerLink (<http://link.springer.com/>)
- Open Access Publikationen (z.B. <http://arXiv.org/>)
- Google Scholar (<http://scholar.google.de/>)
- Google Books (<http://books.google.de/>)
- Microsoft Academic Search (<http://academic.research.microsoft.com/>)
- Portal der International Electrotechnical Commission (<http://www.iec.ch/>)
- The American Institute of Aeronautics and Astronautics (AIAA) (<http://www.aiaa.org/>)
- The Society of Automotive Engineers (<http://www.sae.org/>)
- Radio Technical Commission for Aeronautics, Inc (RTCA) (<http://www.rtca.org/>)
- Europäische Agentur für Flugsicherheit (EASA) (<http://www.easa.europa.eu/>)
- Federal Aviation Administration (FAA) (<http://www.faa.gov/>)

#### 1.4.3 Referenzen

Zur Erstellung der ACROSS im Rahmen des ARTEMIS-Berichtswesens wurde auf die folgende Fachliteratur zurückgegriffen. Darüber hinaus enthalten die im Rahmen des Projektes entstandenen Publikationen (siehe Abschnitt 2.1.7.2) jeweils eigene Referenzlisten.

Als zentrale Quellen für den im Projekt gewählten Ansatz sind einerseits die Ergebnisse des GENESYS-Projektes [A-30] sowie die Dissertation „The Time-Triggered System-on-Chip Architecture“ von Christian Paukovits an der Technischen Universität Wien [A-8] zu nennen. Darüber hinaus spielten domänen-spezifische Standards und Regelwerke wie etwa ARINC 653 [A-3, A-5, A-6] eine wichtige Rolle beim Entwurf und der Validierung des ACROSS-Ansatzes.

- [A-1] AbsInt. *ait WCET Analyzer*. <http://www.absint.com/>.
- [A-2] Aeronautical Radio Inc.. *ARINC 429 Mark 33 Digital Information Transfer System (DITS) Part 1–17*. Mai 2004.
- [A-3] Aeronautical Radio Inc.. *ARINC 653 Avionics Application Software Standard Interface Part 1 - Required Services*, 1. Dez. 2005.
- [A-4] Aeronautical Radio Inc.. *ARINC 653: Avionics application software standard interface*, 2551 Riva Road, Annapolis, MD, USA. Okt. 2003.
- [A-5] Aeronautical Radio Inc.. *ARINC Specification 653-2*, 1. Dez., 2005.
- Altera. *Avalon Interface Specifications*, Mai 2011.
- [A-6] C. Buckl, I. Gaponova, M. Geisinger, A. Knoll and E.A. Lee. *Model-based specification of timing requirements*. Proceedings of the tenth ACM international conference on Embedded software, ACM, pp. 239-248. 2010.
- [A-7] C. Buckl, *Model-Based Development of Fault-Tolerant Real-Time Systems*, Dissertation. Technische Universität München, Mai 2008.
- [A-8] C. Paukovits. *The Time-Triggered System-on-Chip Architecture*. Dissertation, Technische Universität Wien, Fakultät für Informatik, Dez. 2008.
- [A-9] C. Rochange and P. Sainrat. *Multicores and critical systems: Challenges for temporal analysability*, SAE AeroTech Congress & Exhibition, pp. 1-15, 2011.
- [A-10] CoFluent Design. *CoFluent Studio*. <http://www.cofluentdesign.com/>.
- Cypress. *Dual port memory datasheet for CY7C132/CY7C136 and CY7C142/ CY7C146 (2Kx8 Dual- Port Static RAM)*.
- [A-11] D. Steinberg, F. Budinsky, M. Paternostro and E. Merks. *EMF: Eclipse Modeling Framework*, 2nd edition, Addison-Wesley Professional, 2009.
- [A-12] Eclipse Foundation. *Eclipse Modeling Framework*. <http://www.eclipse.org/modeling/emf/>.
- [A-13] European Aviation Safety Agency. *Certification memorandum - development assurance of airborne electronic hardware (chapter 9), Software & Complex Electronic Hardware section*, EASA, CM
- [A-14] EASA CM - SWCEH-001 Issue 01, 1.. Aug. 2011.
- European Aviation Safety Agency. *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25 Amendment 11.*, 4.Juli 2011.
- [A-15] European Aviation Safety Agency. *EASA Certification Memo: Data buses*, MEMO-SWCEH-011, Issue 1, Rev. 2., 16. Mai 2008.
- [A-16] European Telecommunications Standards Institute. *ETSI Tech Report 003 Network Aspects (NA); General aspects of Quality of Services (QoS) and Network Performance (NP)*.
- [A-17] Festo. *MPS Station Sortieren*: <http://www.festo.com/>.
- Festo. *Schaltungsunterlagen Station Sortieren Profibus-DP*. Drw-No: 195761 el., 5. Apr. 2006.
- [A-18] Gartner Group. *Cool Vendors in Platform and Integration Middleware*, ID Number: G00155600, März 2008.
- [A-19] H. Kopetz. *Real-Time Systems. Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, Norwell, Messachusetts (USA), 1st edition, ISBN 0-7923-9894-7. April 1997.
- [A-20] HMS. *Fieldbus Appendix Anybus-M Profibus DP-V Rev. 1.31 (85-0344-ABM-DP-V1\_1\_31.pdf)*.
- [A-21] HMS. *Parallel Interface Design Guide Anybus-S Slave & Master Doc.Id. SCM-1200-015 Rev. 2.06 (Anybus-M-2737-ABS\_M\_Parallel\_DG\_2\_06\_SCM-1200-015.pdf)*.
- [A-22] International Organization for Standardization. *ISO 1151-1. Flight dynamics; Concepts, quantities and symbols; Part 1 : Aircraft motion relative to the air*, Issue 4, 15. Apr. 1988.
- [A-23] International Telecommunication Union. *ITU-T Recomm. E.800 Terms and definitions related to the Quality of Service and Network Performance including Dependability*.
- [A-24] International Telecommunication Union. *ITU-T Recomm. I.350 General Aspects of Quality of Service and Network Performance in Digital Networks, including ISDNs*.
- [A-25] K. Driscoll and K. Hoyme, *The Airplane Information Management System: an integrated real-time flight-deck control system*. Proceedings of the IEEE Real-Time Systems Symposium (RTSS), pp. 267-270. 1992.
- [A-26] L.L. Pullum. *Software Fault-Tolerance - Techniques and Implementation*. Artech House Boston.
- [A-27] Object Management Group. *Systems Modeling Language (SysML)*. <http://www.omg-systemml.org/>

- [A-28] Object Management Group. *Unified Modeling Language (UML)*. <http://www.uml.org/>.
- [A-29] Object Management Group. *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems*. <http://www.omgarte.org/>. 2001.
- [A-30] R. Obermaisser and H. Kopetz (Eds.), *GENESYS: A Candidate for an ARTEMIS Cross-Domain Reference Architecture for Embedded Systems*. Südwestdeutscher Verlag für Hochschulschriften, 2009.
- [A-31] R. Passerone, I.B. Hafaiedh, S. Graf, A. Benveniste, D. Cancila, A. Cuccuru, S. Gerard, F. Terrier, W. Damm, A. Ferrari, L. Mangeruca, B. Josko, T. Peikenkamp, and A. Sangiovanni-Vincentelli. *Metamodels in Europe: Languages, Tools, and Applications*. IEEE Design & Test of Computers, IEEE Computer Society Press, 26(3), pp. 38-53, 2009.
- [A-32] R.S. Boyer and J.S. Moore. In R.S. Boyer and W. Pase (Eds.). *MJRTY - A Fast Majority Vote Algorithm Automated Reasoning*, Springer Netherlands, 1, pp. 105-117, 1991.
- [A-33] Rapita Systems. *Rapita Verification Suite (RVS)*. <http://www.rapitasystems.com/>. RTCA Inc.. *DO-178B Software Considerations in Airborne Systems and Equipment Certification*, 1. Dez. 1992.
- [A-34] RTCA, Inc.. *Design Assurance Guidance for Airborne Electronic Hardware*, DO-254, 19. Apr. 2000.
- [A-35] RTCA, Inc.. *Software Considerations in Airborne Systems and Equipment Certification*, DO-178C, 13. Dez. 2011.
- [A-36] S. Barner, M. Geisinger, C. Buckl, and A. Knoll. *EasyLab: Model-based development of software for mechatronic systems*. *Mechatronic and Embedded Systems and Applications*, IEEE/ASME, Okt. 2008.
- [A-37] SAE International. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, Aerospace Recommended Practice SAE ARP4761, Dez. 1996. SAE International. *Guidelines for Development of Civil Aircraft and Systems*, Aerospace Recommended Practice SAE ARP4754, Rev. A, Revised Okt. 2012.
- [A-38] Sysgo AG. *PikeOS Personality Manual: POSIX*. 2011.
- [A-39] UJF/Verimag. *BIP Toolset*. <http://www-verimag.imag.fr/BIP,196.html?lang=en>.
- [A-40] Martin Schoeberl, *A Time-Triggered Network-on-Chip*. International Conference on Field-Programmable Logic and its Applications (FPL 2007) , pp. 377-382 IEEE, 2007.

## 1.5 Zusammenarbeit mit anderen Stellen

### 1.5.1 Zusammenarbeit im ACROSS Projekt

Die Zusammenarbeit zwischen den deutschen Partnern, deren Teilvorhaben in diese Abschlussbericht beschrieben sind, ist in Abschnitt 2.1 ausführlich beschrieben. Darüber hinaus hat auch eine intensive Zusammenarbeit den übrigen Projektpartnern stattgefunden, die in diesem Abschnitt zusammengefasst wird.

Generell gab es eine enge Zusammenarbeit innerhalb der einzelnen Arbeitspakete, sowie an den in Abbildung 2 illustrierten Schnittstellen zwischen den Arbeitspaketen. Im Folgenden werden einzelne zentrale Kooperationen innerhalb des Projektes beschrieben. Die Ziele der jeweiligen Arbeitspakete sind in Abschnitt 2.1 beschrieben. Darüber hinaus sind die verwendeten Partnerkürzel Tabelle 1 zu entnehmen.

#### 1.5.1.1 Technische Universität Wien

Alle Partner des deutschen Konsortiums arbeiteten intensiv mit dem Projektkoordinator TU Wien zusammen. Dabei standen neben dem Projektmanagement und Controlling insbesondere die initiale Anforderungsanalyse, die strategische Entwicklung des Arbeitsplanes, sowie die Bewertung der Projektergebnisse im Vordergrund.

Darüber hinaus fand eine enge Abstimmung bzgl. der von WP2 zur Verfügung gestellten generischen optionalen Dienste statt, die den domänen-übergreifenden, in Software implementierten Anteil der ACROSS-Plattform darstellen (TU Wien war der Koordinator von WP2). Im Zuge dessen stimmte fortiss die Schnittstelle zwischen WP2 und WP3 eng mit der TU Wien ab, um die Integration der von WP2 implementierten generischen optionalen Dienste in die Werkzeugkette aus WP3 sicherzustellen.

Die TU Wien leistete unverzichtbare Hilfe bei der Erstellung des MPSoC FPGA Designs. Diese Unterstützung war die Voraussetzung für die Entwicklung zusätzlicher VHDL Module für das MPSoC FPGA Design. Ebenso hat die TU Wien hier im Rahmen eines Hardware-Software-Codesigns Rückmeldungen zur Software-Architektur auf dem MPSoC im Betriebssystem PikeOS berücksichtigt, um eine effiziente Koppelung zu erreichen.

Desweiteren hat die TU Wien bei der Integrationsarbeit für die Demonstratoren in den Automotive- und Avionik-Bereichen mit dem Betriebssystem und der Middleware eng mit SYSGO und den entsprechenden Demonstrator-Partnern kooperiert.

#### **1.5.1.2 TTTech Computertechnik AG**

Alle deutschen Partner arbeiteten intensiv mit TTT zusammen, um die bedarfsgerechte Realisierung des zentralen ACROSS MPSoC durch TTT sicherzustellen (TTT ist Leiter des zugehörigen Arbeitspaketes WP1).

Darüber hinaus arbeitete fortiss intensiv mit TTT in WP3 zusammen, um die Integration der für die Konfiguration des ACROSS MPSoCs benötigten spezifischen Werkzeuge (TTNoC Scheduler, MPSoC und I/O Konfigurator) in die ACROSS Werkzeugkette zu sicherzustellen. Darüber hinaus steuerte fortiss Anforderungen aus dem Kontext des Industrial Control Application Demonstrators in WP6 an das ACROSS MPSoC bei.

#### **1.5.1.3 Université Joseph Fourier Grenoble 1**

Der Fokus von UJF/Verimag lag auf der Entwicklung und Anwendung formaler Methoden zur Analyse und Verifikation ACROSS-basierter Anwendung auf Basis des BIP-Paketes vgl. (siehe Abschnitt 2.1.3). Als federführender Partner in WP3 definierte fortiss in Zusammenarbeit mit UJF/Verimag den ACROSS-Entwicklungsprozess und kooperierte bei der Definition, Umsetzung und Bewertung der Werkzeugkette.

#### **1.5.1.4 Siemens AG Österreich**

Unter der Leitung des federführenden Partners SAGÖ hat fortiss an der Umsetzung des Industrial Control Application Demonstrators in WP6 mitgewirkt. Hierbei war SAGÖ einerseits als industrieller Repräsentant der Domäne Industrieautomatisierung für die Definition der Anforderungen und die Erstellung des Gesamtkonzeptes hauptverantwortlich. Andererseits übernahm SAGÖ die domänen-spezifische E/A-Anbindung des ACROSS MPSoCs an den Demonstrator (Festo MPS Sortierstation). Der Fokus von fortiss beruhte auf der Umsetzung der gewählten Anwendung (Sortieren von Werkstücken). Hierzu stimmte fortiss sowohl die Schnittstelle zwischen der Anwendung und der E/A-Komponente als auch die Eigenschaften der verwendeten Werkzeugkette (domänen-spezifischen Programmierumgebung aus WP6, generische Werkzeugkette aus WP3) mit SAGÖ ab.

#### **1.5.1.5 CoFluent Design**

Aus der Zusammenarbeit resultierte bis zum Ausscheiden von CoFluent Design aus dem Projekt eine grobe Skizze der Einbindung des CoFluent Studio Werkzeuges in WP3. Im weiteren Projektverlauf

koordinierte fortiss als federführender Partner in WP3 die vom österreichischen Partner SAGÖ ersatzweise durchgeführte Studie mit alternativen ESL-Werkzeugen.

#### **1.5.1.6 SELEX Sistemi Integrati**

Im Rahmen des Supports für Demonstratoren der Avionik fand eine enge Kooperation mit SELEX Sistemi Integrati statt, um die Integration der ACROSS-Plattform in den Demonstrator zu erreichen. Hier ist auch noch einmal die gute Zusammenarbeit mit der TU Wien zu unterstreichen, die ebenfalls unterstützend mitgearbeitet hat und die Integration von Demonstrator, Hardware, und Betriebssystem zusammen mit SYSGO vorangetrieben hat.

#### **1.5.2 Zusammenarbeit mit weiteren Forschungsprojekten**

In ACROSS fanden die folgenden Kooperationen mit weiteren nationalen und internationalen Forschungsprojekten statt:

- Alle Partner: Wissenschaftlicher Austausch mit den Forschungsprojekten RECOMP (BMBF u.a., ARTEMIS JU), MultiPARTES (FP7), CERTAINTY (FP7) und ARAMIS (BMBF) im Rahmen eines durch das ACROSS Projekt organisierten eintägigen Workshops „Integration of mixed-criticality subsystems on multi-core processors“ (im Rahmen der HiPEAC 2013 Konferenz).
- fortiss: Teilweise Anbindung an die im Rahmen von RECOMP entwickelte Methodik durch die Implementierung von Modell-zu-Modell-Transformationen zwischen der ACROSS-Werkzeugkette und dem in RECOMP verwendeten Werkzeug AutoFocus 3.

#### **1.5.3 Zertifizierungsbehörden**

EADS-IW und EADS-DE ist seit 2010 Mitglied in der Multicore For Avionics (MCFA) Arbeitsgruppe<sup>2</sup>. Hier werden unter Avionics OEM und Tier 1-Lieferanten und Chip-Herstellern regelmäßige Treffen abgehalten, die die notwendigen Eigenschaften und Dokumente zur Zertifizierung von Multicore-Prozessoren diskutieren und dokumentieren. Des Weiteren arbeitet die Gruppe mit den Zertifizierungsbehörden European Aviation Safety Agency (EASA), Federal Aviation Administration (FAA) sowie Transport Canada Civil Aviation (TCCA) zusammen. Hier wurden verschiedene Ansätze diskutiert und vorgestellt. Der Fokus ist allerdings auf sogenannten Commercial Off The Shelf (COTS) Prozessoren. Die Arbeiten von ACROSS haben aber einen Ansatz vorgestellt, der eine Zertifizierung für die Luftfahrt einfacher erscheinen lässt als andere komplexe Network-on-Chip Architekturen.

EADS-IW und EADS-DE haben auch zu Certification Memoranda (CM) der EASA Software & Complex Electronic Hardware section (im Speziellen EASA CM – SWCEH – 001 und EASA CM – SWCEH – 001) Stellung genommen und Feedback gegeben. Bei diesen zwei Memoranda steht der Entwicklungsprozess von Hardware und Software (peripher auch für Multicore-Architekturen) im Fokus und es werden Empfehlungen für Prozesse erarbeitet.

---

<sup>2</sup> <http://media.freescale.com/phoenix.zhtml?c=196520&p=irol-newsArticle&ID=1606741&highlight>

## 2 Eingehende Darstellung

### 2.1 Verwendung der Zuwendung und Ergebnisse

Im Folgenden werden die Verwendung der Zuwendung und die erzielten Ergebnisse im Einzelnen beschrieben (Granularität: Unterarbeitspakete (Tasks)). Dabei werden die Ergebnisse den im Projektplan definierten Zielen gegenübergestellt.

#### 2.1.1 WP1 (ACROSS MPSoC)

An WP1 waren die folgenden deutschen Partner beteiligt: EADS-DE, EADS-IW (, Universität Augsburg)

##### 2.1.1.1 Ziele

WP1 hat das Ziel, das FPGA-basierte ACROSS MPSoC zu entwickeln. Die Arbeiten wurden unter der Federführung von TTTech durchgeführt.

Die Arbeiten gliederten sich in die folgenden Schritte:

1. Konsolidierung bereits verfügbarer Anforderungen
  - GENESYS-Projekt
  - ARTEMIS Strategic Research Agenda (SRA)
2. Implementierung einer für die drei Anwendungs-Domänen geeigneten FPGA-basierenden ACROSS MPSoC Architektur.
  - Erstellung der FPGA-Firmware
  - Definition und Herstellung einer zusätzlichen Interface-Platine (Evaluation and Test Board).

Das deutsche Konsortium hatte dabei die Aufgabe, die luftfahrtspezifischen Anforderungen einzubringen sowie eine Zertifizierbarkeit bezüglich des Einsatzes in der Luftfahrt sicherzustellen.

##### 2.1.1.2 Ergebnisse

Zusammenfassend lässt sich feststellend, dass alle in Abschnitt 2.1.1.1 aufgeführten Ziele erreicht wurden: Anhand der Demonstratoren für die drei Domäne konnte die Eignung des ACROSS MPSoC für sicherheits-kritische Anwendungen bewiesen werden. Die vom deutschen Konsortium erbrachten Nachweise sind im Abschnitt 2.1.5.2 beschrieben.

Das deutsche Konsortium hat seine Ergebnisse in den folgenden Berichten dokumentiert:

- D1.1 „Architectural Study and ACROSS MPSoC Requirements Capture Document“
- D1.2 „Specification of ACROSS Hardware Coded Core Services and Modular Building-Block Components“
- D1.3 „Design Document for the MPSoC VHDL Code Library“
- D1.4 „Validation, Verification and Test Report of the ACROSS MPSoC FPGA“
- D1.5 „ACROSS MPSoC Evaluation and Test Board specification, design description and test report“

### 2.1.1.3 Vorgehen

Im Folgenden wird das Vorgehen in Arbeitspaket WP1 beschrieben.

- **T 1.1 „Architectural Study and MPSoC Requirements Capture“**
  - Untersuchung verschiedener möglicher ACROSS MPSoC Architekturen sowie Auswahl der endgültigen Architektur
  - Erstellung und Konsolidierung von Anforderungen
- **T 1.2 „Specification of ACROSS Hardware Coded Core Services and Modular Building-Block Components“**
  - Definition der von WP4, 5 und 6 benötigten Core-Services
  - Erstellung einer Design-Spezifikation für die Core-Services
- **T 1.3 „Design and Development of ACROSS Modular Building-Block Components“**
  - VHDL Codierung des ACROSS MPSoC
  - Überprüfung des Designs hinsichtlich Zertifizierbarkeit in der Domäne Luftfahrt
- **T 1.4 „FPGA Integration of Building Blocks into Selected Architectures (resulting in ACROSS MPSoC FPGA)“**
  - Implementierung der von WP4, 5 und 6 benötigten MPSoC-Varianten
- **T 1.5 „Validation, Verification and Test of the ACROSS MPSoC FGAs“**
  - Validierung und Verifikation der von WP4, 5 und 6 benötigten MPSoC-Varianten
- **T 1.6 „ACROSS MPSoC Evaluation and Test Board“**
  - Spezifizierung, Design und Test des von WP4, 5 und 6 benötigten „Evaluation and Test“ Boards. Dieses Board stellt spezifische Interfaces bereit, die von den Demonstratoren in WP4, 5 und 6 benötigt werden, wie z.B. CAN oder ARINC 429.
- **T 1.7 „Support for use in MPSoC in the Demonstrators“**
  - Umfasst alle Support-Aktivitäten von TTTech für den Einsatz des ACROSS MPSoCs bei den übrigen Projektpartnern.

### 2.1.2 WP2 (Middleware and System Components)

An WP2 waren die folgenden deutschen Partner beteiligt: fortiss, EADS-DE, Lauterbach, SYSGO

In WP2 des ACROSS-Projektes wurden die Middleware und Systemkomponenten der ACROSS-Plattform entwickelt. Diese Arbeiten umfassten also die unteren Ebenen der Software-Anpassung an die Hardware, sowie an die aus WP1 gelieferten Kernkomponenten in Software (z.B. an Treiber usw.). WP2 verlief wie üblich in mehreren Phasen.

Zunächst wurden die Anforderungen erhoben und ein System aus „Generischen Optionalen Diensten“ erarbeitet, die zur Umsetzung des Projektplans benötigt werden. Aus dieser Anforderungsphase ergaben sich die unten aufgeführten Generischen Optionalen Dienste (Details finden sich in Deliverable D2.1 und D2.2). Diese Dienste umfassen alle Software-Dienste, die Anwendungsprogramme auf der ACROSS-Hardware mitunter benutzen wollen, die aber dennoch optional sind in dem Sinne, dass die ACROSS-Plattform auch ohne diese Dienste funktionsfähig bleibt. Dies ist die Abgrenzungslinie zu WP1, in dem alle Kerndienste, die essentiell notwendig für das Betreiben der ACROSS-Plattform sind, behandelt werden.

Eine weitere Abgrenzung ist zu den Demonstrator WPs zu stecken. Diese Abgrenzung basiert auf dem „generischen“ Teil der in WP2 erarbeiteten Dienste. In WP4, WP5, WP6 werden ebenfalls Middleware und System-Dienste erarbeitet, diese sind dann aber nicht generisch, sondern spezifisch für die

entsprechende Anwendungsdomäne. So ist etwa der AUTOSAR-Layer ebenfalls ein Optionaler Dienst, aber eben spezifisch für die Automotive- Domäne. In WP2 hingegen werden nur generische Dienste behandelt, die also potentiell von allen Domänen nutzbar sind.

Die in WP2 ausgemachten Generischen Optionalen Dienste sind in der folgenden Liste zusammengefasst.

- OPT1: Secure Global Time Service (TU Wien, TTTech)
- OPT2: Transparent Gateway Service (TTTech)
- OPT3: Secure Group Communication Service (TU Wien)
- OPT4: Secure Boot Service (TU Wien)
- OPT5: Voting Service (**fortiss**)
- OPT6: State Externalization Service (TU Wien)
- OPT7: State Restoration Service (TU Wien)
- OPT8: Betriebssystem (**SYSGO**)
- OPT9: Bee Services (Middleware) (ED)
- OPT10: Real-Time Tracing Service (**Lauterbach**)
- OPT11: Debugging Service (**Lauterbach**)
- OPT12: Health Monitoring Service, LDU (**Lauterbach**)
- OPT13: Health Monitoring Service, Receiver (**Lauterbach**)
- OPT14: Virtual CAN Service (TU Wien)
- OPT15: I/O Service (TTTech)

Wie der Liste zu entnehmen ist, hat die TU Wien die meisten dieser Dienste verantwortet. Die deutsche Beteiligung ist in der Liste hervorgehoben und wird in Kapitel 3 genauer beleuchtet, in dem die entsprechenden Beiträge der Partner darstellt sind. Eine im späteren Verlauf etwas grob anmutende Klassifizierung ist OPT8, welches das gesamte Betriebssystem mit all seinen Teilmodulen in einem Optionalen Dienst zusammenfasst.

Nach der Anforderungserhebungsphase wurde die Design-Phase von WP2 eingeleitet, in der diese Generischen Optionalen Dienste genauer konzipiert wurden. APIs und andere Design-Ergebnisse sind in D2.3 dokumentiert.

Nach der Planungsphase wurde in WP2 mit der prototypischen Umsetzung begonnen.

#### **2.1.2.1 Ziele**

Im WP2 wurden mit den Generischen Optionalen Diensten die Grundlagen für die folgenden Arbeitspakete gelegt. Diese konnten auf den Optionalen Diensten aufsetzen, um die höheren Software-schichten, die domänen-spezifischen Dienste und die Anwendungen, umzusetzen.

Schon aus der Liste der Dienste im vorangehenden Kapitel ist ersichtlich, dass die Dienste zwar streng genommen optional sind, aber oft auch sehr grundlegend, so dass WP2 hier einen integralen, essentiellen Teil der Gesamtarbeit im ACROSS-Projekt inne hatte.

Das wesentliche Ziel aus WP2 war also, den Anwendungen die Benutzung der ACROSS-Plattform über wohldefinierte Schnittstellen nutzbar zu machen.

### 2.1.2.2 Ergebnisse

fortiss hat mit dem Voting-Service (OPT5) im Rahmen von WP2 einen domänen-übergreifenden Robustheitsdienst entwickelt, mit dessen Hilfe Verfahren zur aktiven Fehlertoleranz (z.B. durch Replikation von Tasks) implementiert werden können. Der Voting-Service ist ein konfigurierbares und erweiterbares Softwarepaket, das eine Reihe etablierter Voting-Verfahren (z.B. median, mean, majority) und Ein-/Ausgabe-Mechanismen zur Verfügung stellt (z.B. Speicherbereich, PikeOS-Port zur Inter-Partitions- bzw. Inter-Core-Kommunikation). Der Dienst ist an das Verfahren zur automatischen Exploration des Entwurfsraums der WP3-Werkzeugkette angebunden (siehe Abschnitt 2.1.3).

Die SYSGO AG hat im Rahmen von WP2 das Betriebssystem PikeOS auf die ACROSS-NIOS-2-Plattform portiert (OPT8). Die Arbeiten waren sehr umfassend, da die NIOS-2-Architektur in PikeOS noch nicht unterstützt wurde. Entsprechend mussten alle Software-Ebenen des PikeOS-Systems erweitert werden. Auf unterster Ebene kam eine Unterstützung des PikeOS-Kernels für die NIOS-2-CPU in der verwendeten Version hinzu, das „Architecture Support Package (ASP)“ von PikeOS. Im Weiteren ein „Platform Support Package (PSP)“ zur Unterstützung der grundlegenden vom Betriebssystem benutzten Geräte und Dienste wie Uhrentreiber, Interrupts und Boot-Vorgang. Die PikeOS-Systemzeit wurde von dem entsprechenden Kerndienst des TTNoC abgeleitet, so dass alle Kerne mit PikeOS die gleiche synchronisierte Systemzeit hatten. Weiterhin wurde die „PikeOS System Software (PSSW)“ angepasst, die die grundlegenden Kommunikations- und Rechtedienste in PikeOS anbietet. Zu guter Letzt wurden Erweiterungen und Treiber nötig, um den speziellen Netzwerkbus TTNoC unter PikeOS anzusteuern und nutzen zu können. Hier ist eines der wichtigsten Ergebnisse für SYSGO entstanden: eine Kernel-Erweiterung zur verbesserten Anbindung von Time-Triggered-Bussen, über die das PikeOS-Time-Partitioning mit dem Netzwerkbus synchronisiert wurde. Alle Arbeiten sind für PikeOS als Produkt relevant.

Die Ergebnisse aus WP2 sind in den folgenden Deliverables zusammengefasst:

- D2.1 Anforderungen an Generische Optionale Dienste
- D2.2 Funktionale Spezifikation der Generischen Optionalen Dienste
- D2.3 Design von Middleware und Systemkomponenten
- D2.4 Implementierung der Optionalen Dienste

Im Rahmen von WP2 wurde im Oktober 2011 auch ein Workshop zur Einführung in PikeOS in Frankfurt angeboten, an dem 20 Mitarbeiter aus dem Projekt teilnahmen.

### 2.1.2.3 Vorgehen

Die einzelnen Arbeiten der deutsche Partner aus WP2 werden in diesem Abschnitt noch einmal stichwortartig zusammengefasst.

- **T2.1 „Requirements Consolidation“**
  - Sammeln der Anforderungen von den Partnern durch Eingaben aus den anderen WPs
  - Auswerten, Zusammenstellen der Anforderungen
  - Planung einer Menge von Generischen Optionalen Diensten
  - Zusammenfassung der Anforderungen in D2.1
- **T2.2 „Functional Specification of Middleware and System Components“**
  - Ausbau der Anforderungen zu einer funktionalen Beschreibung
  - Zuordnung von Generischen Optionalen Diensten zu Partnern

- Pro Partner: Ausarbeitung einer Beschreibung der Funktionalität des Dienstes
- **T2.3 „Design of Middleware and System Components“**
  - Ausarbeitung eines technischen Konzepts zur Umsetzung der Generischen Optionalen Dienste in der ACROSS-Plattform
  - Definition von Schnittstellen
  - Prüfung der Konsistenz mit den Arbeiten aus WP1 und WP2
- **T2.4 „Implementation and Testing of Generic Optional Services“**
  - Umsetzung der technischen Konzepte aus T2.3 in einer Implementierung in Software
- **T2.5 „Support of Using Middleware and System Components in Demonstrators“**
  - Zusammenarbeit mit den benutzenden Partnern zur Sicherstellung der erfolgreichen Nutzung der in WP2 erarbeiteten Konzepte und Software-Schichten

### 2.1.3 WP3 (Development Methodology)

An WP3 waren die folgenden deutschen Partner beteiligt: fortiss, Lauterbach.

#### 2.1.3.1 Ziele

Ziel von Arbeitspaket WP3 war es, eine modellgetriebene Entwicklungsmethodik zu definieren, die die Modellierung, Analyse, Verifikation sowie die Konfiguration von ACROSS-basierten Anwendungen unterstützt. Dabei gliedert sich die Entwicklungsmethodik in generische sowie domänen-spezifische Teile. Letztere berücksichtigen dabei die Anforderungen aus den Arbeitspaketen WP4, 5, 6.

Die Ziele des Arbeitspaketes, das unter der Federführung von fortiss durchgeführt wurde, lassen sich wie folgt zusammenfassen:

1. Konsolidierung der Anforderungen
  - In anderen Projekten gesammelte Anforderungen (GENESYS, ARTEMIS SRA)
  - Anforderungen der Anwendungsarbeitspakete (WP4, 5, 6)
2. Definition des Entwicklungsprozesses
3. Definition geeigneter generischer und domänen-spezifischer Modelle
4. Werkzeugunterstützung
  - Modell-zu-Modelltransformationen
  - Code- und Konfigurationsgenerierung
  - Diagnose- und Debugging-Werkzeuge, die die in ACROSS spezifizierten Debug- und Trace-Schnittstellen unterstützen.
5. Integration von Analyse- und Verifikationswerkzeugen

#### 2.1.3.2 Ergebnisse

Zusammenfassend lässt sich feststellen, dass die deutschen Verbundpartner ihre geplanten Ergebnisse zu den in Abschnitt 2.1.3.1 aufgeführten Zeile erreicht haben. Die Infrastruktur zum Tracen und Debuggen des TTNoC wurden vollständig entwickelt und getestet; dies beinhaltet die Entwicklung und Integration der benötigten VHDL Komponenten und die Erweiterung von Lauterbachs TRACE32 Produkten, um die entsprechenden Schnittstellen zu unterstützen. Bedingt durch die Verzögerungen im Projektablauf, konnte diese Infrastruktur in den finalen Demonstratoren allerdings nicht mehr genutzt werden.

Zunächst wurde auf Basis der konsolidierten Anforderungen der Entwicklungsprozess definiert (siehe Abbildung 4).

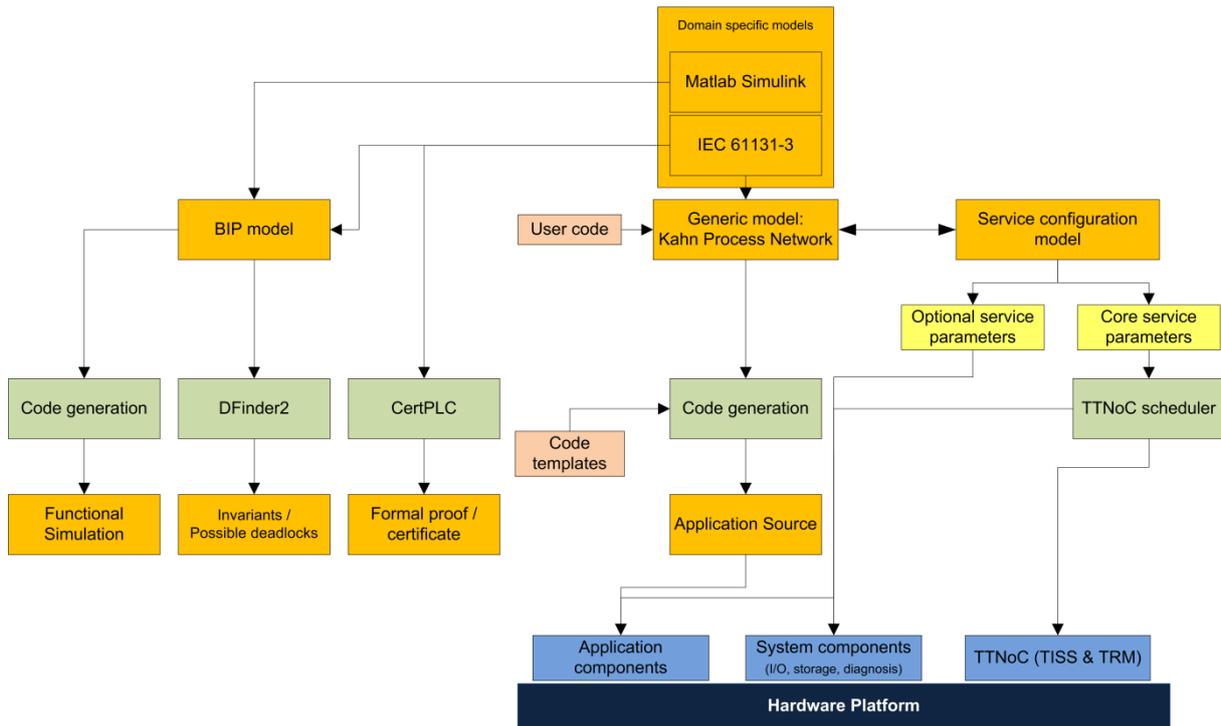


Abbildung 4: Dokumentenbasierter ACROSS-Entwicklungsprozess

Die ACROSS-Entwicklungsmethodik ist in Prozesse unterteilt, die auf verschiedenen Abstraktionsstufen angesiedelt sind und auf einander aufbauen. Dieser modulare Ansatz ermöglicht es, die ACROSS-Entwicklungsmethodik in etablierte Entwicklungsprozesse zu integrieren. Darüber hinaus macht sich der Ansatz die Komponentenarchitektur der ARCOSS-Plattform zunutze, in dem er eine Rollen-basierte Trennung des Entwicklungsprozesses ermöglicht. Diese Separierung erlaubt es, dass sich Domänenexperten auf die Entwicklung der Applikation und Plattformexperten auf die Integration verschiedener Anwendungen konzentrieren. Die Verwendung von Modellen auf allen Ebenen legt die Basis für die werkzeuggestützte Analyse und Verifikation von Designs in frühen Projektphasen. Im Einzelnen wurden unter der Leitung von fortiss die folgenden Prozesse definiert:

- **Werkzeuggestützte Konfiguration von ACROSS-Diensten:** Dieser Prozess stellt den minimalen Funktionsumfang zur Verfügung, der für die Entwicklung ACROSS-basierter Anwendungen benötigt wird. Die modellgetriebenen Werkzeuge werden vom Systemintegrator zur Konfiguration der verwendeten Dienste sowie zum Deployment verwendet.
- **Modellgetriebene Anwendungsentwicklung:** Hier verwendet auch der Applikationsentwickler (generische oder domänen-spezifische) Modelle, um die Funktionalität und die nicht-funktionalen Anforderungen seiner Anwendung zu spezifizieren. Mit Hilfe der entwickelten Modell-zu-Modell-Transformationen werden alle Anwendungsmodelle in das generische Format übersetzt, aus dem mit Hilfe von Codegeneratoren der Zielcode erzeugt wird. Zusätzlich wird die benötigte Plattformkonfiguration erzeugt (siehe oben). Hierbei kommen die entwickelten Methoden zur automatischen Entwurfsraumexploration und Zuverlässigkeitsanalyse zum Einsatz, die die optimale Abbildung von Applikationen auf die ACROSS-Plattform unter verschiedenen Nebenbedingungen ermöglicht.

- Simulation und Verifikation: Das deutsche Konsortium leistete hier Beiträge zur Anbindung der BIP-Werkzeugkette (Simulation, Analyse von Deadlock-Bedingungen). Darüber hinaus wurden bestehende Debug-Werkzeuge erweitert, um mit den ACROSS spezifischen Trace- und Debug-Schnittstellen zu kommunizieren.

Die deutschen Verbundpartner haben die folgenden Module der ACROSS-Werkzeugkette erstellt:

ACROSS-Workbench	<ul style="list-style-type: none"> <li>• Meta-Modelle und Editoren zur Beschreibung von ACROSS-basierten Anwendungen</li> <li>• Methoden zur automatischen Entwurfsraumexploration und Zuverlässigkeitsanalyse</li> </ul>
Anbindung an MPSoC-Konfigurator / TTNoc Scheduler und den I/O-Konfigurator	<ul style="list-style-type: none"> <li>• Transformation der in der ACROSS-Workbench erstellten Modelle zu den jeweiligen plattformspezifischen Formaten</li> </ul>
Anbindung an PikeOS-Konfigurationswerkzeug (CODEO)	<ul style="list-style-type: none"> <li>• Transformation der in der ACROSS-Workbench erstellten Modelle in das VMIT-Format</li> </ul>
Anbindung an den ACROSS MPSoC Build-Server	<ul style="list-style-type: none"> <li>• ACROSS-Workbench Modul zur Benutzung des ACROSS MPSoC Build Servers (von TU Wien bereitgestellt) <ul style="list-style-type: none"> <li>○ Generierung von lauffähigen MPSoC-Images mit applikationsspezifischer Konfiguration</li> <li>○ Generierung von Boot-Images</li> </ul> </li> </ul>
Anbindung domänen-spezifischer Entwicklungswerkzeuge	<ul style="list-style-type: none"> <li>• Import von in EasyLab erstellten IEC 61131-3 Modellen in die ACROSS-Workbench</li> </ul>
Anbindung an die BIP-basierten Verifikationsmethoden	<ul style="list-style-type: none"> <li>• Transformation der IEC 61131-3 Modelle nach BIP</li> </ul>
CertPLC	<ul style="list-style-type: none"> <li>• Werkzeug zur Verifikation von SPS-Anwendungen auf Basis des Coq-Theorembeweislers</li> </ul>
Debugger	<ul style="list-style-type: none"> <li>• Erweiterung der TRACE32 Produkte, um mit den ACROSS spezifischen Trace- und Debug-Schnittstellen zu kommunizieren.</li> </ul>

Tabelle 4: Tool-Map

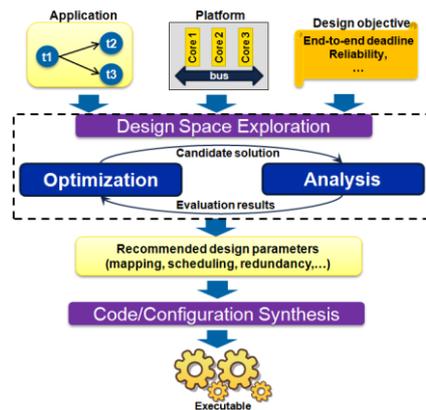


Abbildung 5: Ansatz zur Entwurfsraum-Exploration

Im Folgenden wird näher auf einzelne Aspekte der ACROSS-Werkzeugkette eingegangen. Abbildung 5 zeigt den Ansatz zur Entwurfsraum-Exploration. Hierzu werden Modelle der Anwendung und ACROSS-Plattform zusammen mit Zielvorgaben (nicht-funktionale Anforderungen) an die Optimierung- und Analyseschleife übergeben. Diese bestimmt mit Hilfe von genetischen Algorithmen eine bzgl. der vorgegebenen Design-Ziele optimale Abbildung einer Anwendung. Zur Bewertung der jeweiligen Lösungskandidaten werden Scheduling-Verfahren und Analysen (z.B. Zuverlässigkeit) ein-

gesetzt. Das Ergebnis der Anwendung dieser Verfahren auf den Industrial Control Application Demonstrator aus WP6 ist in Abbildung 6 abgebildet (siehe Abschnitt 2.1.6. für eine Beschreibung der Demonstratoranwendung).

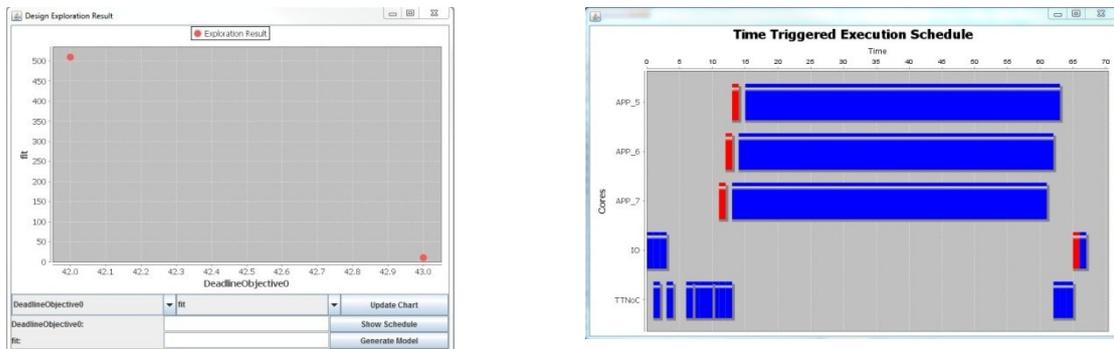


Abbildung 6: Pareto-Front des Optimierungslaufes (links) und fehlertolerante Schedule einer möglichen Lösung

Basierend auf den Ergebnissen dieser Entwurfsraumexploration kann das Werkzeug dazu genutzt werden, per Modell-zu-Modelltransformation eine fehlertolerante Version eines Eingabemodells zu erzeugen. Abbildung 7 zeigt die für den Industrial Control Application Demonstrator berechnete TMR-Konfiguration. Schließlich zeigt Abbildung 8 die Integration der plattformspezifischen Konfigurationswerkzeuge (hier ACROSS MPSoC und PikeOS). Ausgehend von den im plattformunabhängigen Anwendungsmodell spezifizierten Anforderungen (Periodenlänge, Nachrichtenpräzedenzen, Nachrichtengröße, ...) werden die entsprechenden Parameter des Schedules berechnet (z.B. Nachrichtenphase). Auf dieser Basis werden spezifische Konfigurationen für das Verbindungsnetzwerk TTNoC (unten) und die verwendeten PikeOS-Instanzen berechnet (oben). Mit Hilfe von Modell-zu-Modell-Transformationen wird sichergestellt, dass die die Parametrisierung der Kommunikationskanäle in den beiden Konfigurationen konsistent ist (z.B. IDs, Nachrichtenart (periodisch, sporadisch), Verwendung von Zeitstempeln, ...).

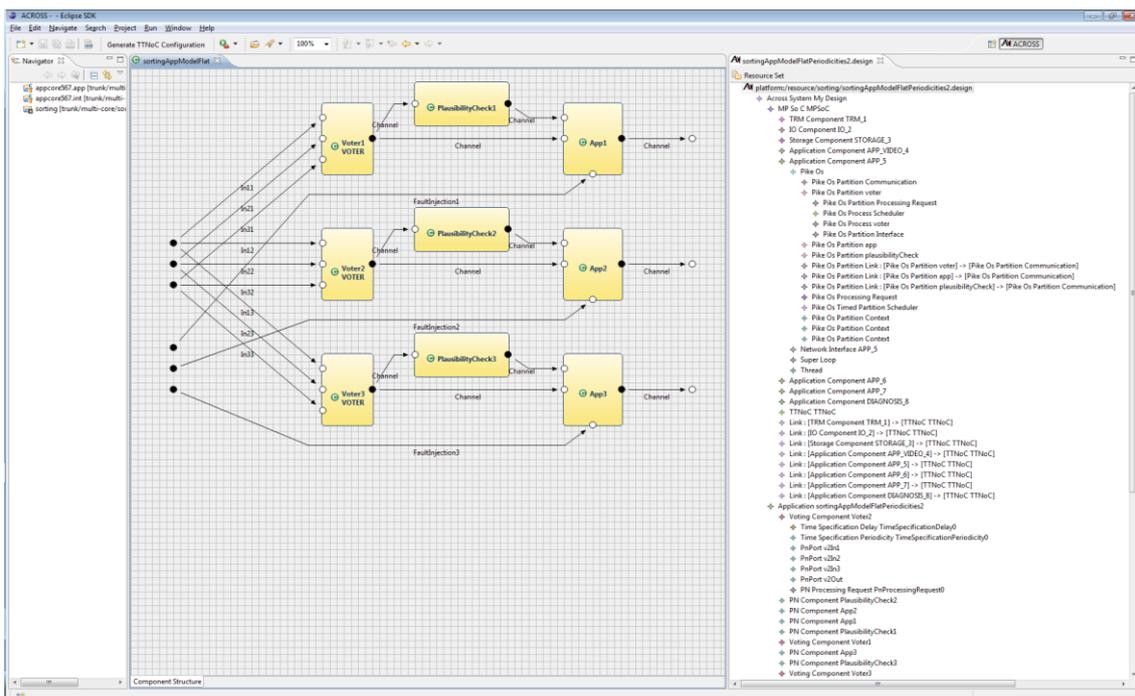


Abbildung 7: ACROSS-Workbench. Das Modell zeigt eine fehlertolerante Version der Sortieranwendung aus WP6.

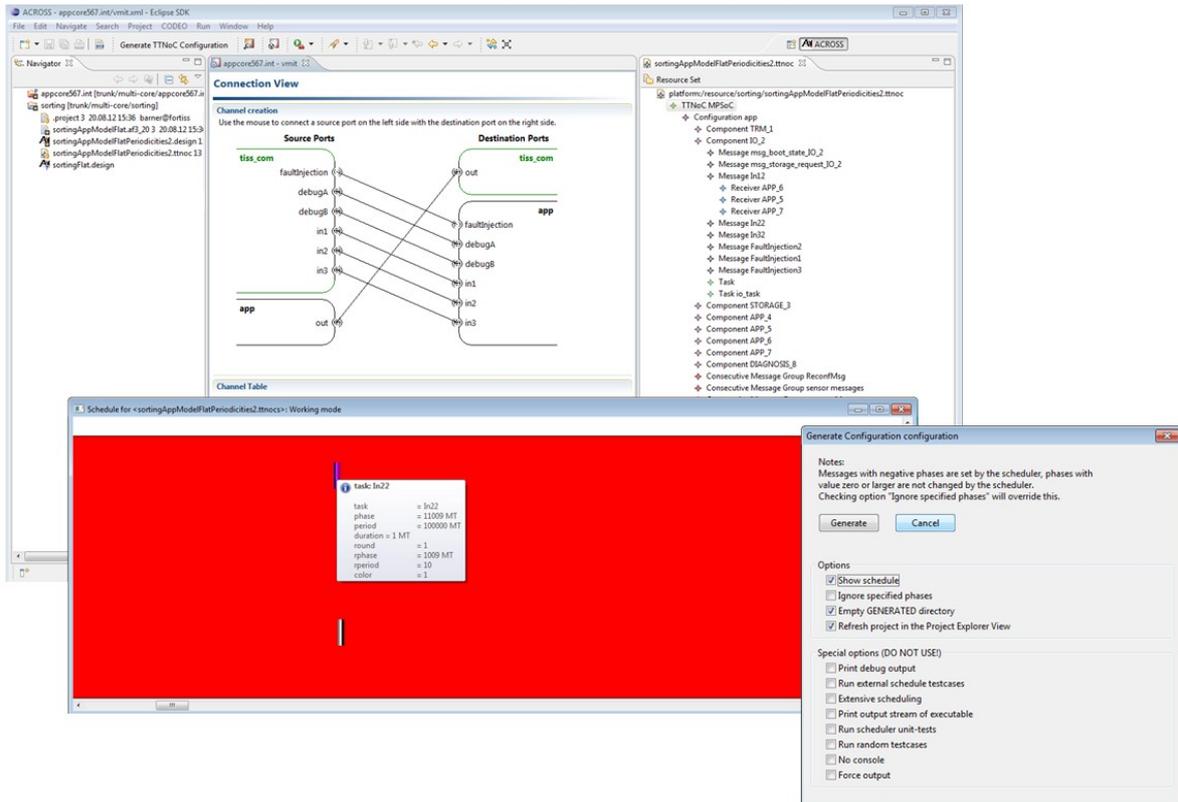


Abbildung 8: Integration plattformspezifischer Konfigurationswerkzeuge in die ACROSS-Workbench: PikeOS CODEO (oben) sowie der TTNoc-Scheduler (unten).

Das deutsche Verbundprojekt hat zu den folgenden Berichten in WP3 beigetragen:

- D 3.1 „Requirements for models and tools“
- D 3.2 „Specification Development process and model views“
- D 3.3 „Initial draft for generic models and domain-specific model extensions“
- D 3.4 „Initial prototypes of verification, model-2-model transformation, and code generation tools“
- D 3.5 „Model specification and tool prototypes“
- D 3.5 „Validation Report on Models and Tools“

Neben den in Tabelle 4 aufgeführten Komponenten der ACROSS-Werkzeugkette hat fortiss als federführender Partner die folgenden Beiträge erbracht:

- Eclipse-Update-Site zur Verteilung der Werkzeugkette innerhalb des Konsortiums
- Bug-Tracker
- Materialien (Videos / Folien) für Toolchain-Tutorial

### 2.1.3.3 Vorgehen

Im Folgenden wird Vorgehen in Arbeitspaket WP3 beschrieben:

- **Task T 3.1 „Requirements consolidation“:** Konsolidierung der Anforderungen an die Entwicklungsmethodik aus
  - ARTEMIS Strategic Research Agenda (SRA)
  - GENESYS Projekt
  - Anforderungen der Technologie- und Anwendungspartner

- **Task T 3.2 „Specification of the overall development process“**
  - Skalierbarer Entwicklungsprozess, der unterschiedliche Ausbaustufen und Automatisierungsgrade vorsieht.
  - Koordination der Definition der entsprechender Werkzeugkette und Erstellung einer Tool-Map:
    - ACROSS MPSoC Konfigurations- und Scheduling-Tool (Werkzeug bereitgestellt von TTTech)
    - ACROSS I/O Konfiguration Tool (Werkzeug bereitgestellt von TTTech)
    - ESL Modelle (bereitgestellt von SAGÖ) und Konzepte zur Anbindung von CoFluent Studio (Werkzeug bereitgestellt von CoFluent Design)
    - Integration der BIP-Werkzeugkette (bereitgestellt von UJF/Verimag)
    - ACROSS Workbench (Integrationsplattform, bereitgestellt von fortiss)
    - PikeOS-Konfigurationswerkzeuge (bereitgestellt von SYSGO in WP2)
    - IEC 61131-Werkzeug (bereitgestellt von fortiss in WP6)
- **Task T 3.3 „Specification of Models“**
  - Domänen-übergreifende Meta-Modelle zur plattformunabhängigen Beschreibung von Applikationen
  - Berücksichtigung domänen-spezifischer Meta-Modelle (insbesondere IEC 61131-3 aus WP6)
  - Meta-Modelle zur Beschreibung der folgenden nicht-funktionalen Eigenschaften
    - Zeitanforderungen (z.B. Periodizitäten, Ende-zu-Ende Latenzen, ...)
    - Zuverlässigkeit (Ausfallwahrscheinlichkeit des Gesamtsystems, Fehlermodell, ...)
  - Meta-Modelle zur Beschreibung der ACROSS Plattform
    - Abstraktion des MPSoC (Prozessorkerne, TTNoC-Verbindungsnetzwerk)
    - PikeOS-Betriebssystem (Partitionen, Kommunikation, ...)
  - Meta-Modelle, die die Abbildung der plattformunabhängigen Applikationsmodelle auf die ACROSS-Plattform beschreiben.
- **Task T 3.4 „Integration / Implementation of Model-to-Model Transformation and Configuration Tools“**
  - Modell-zu-Modell-Transformationen
    - Transformation von IEC-61131 Modellen nach BIP
    - Transformation der plattformunabhängigen Applikationsmodelle von und nach AutoFocus 3, das im RECOMP-Projekt verwendet wird.
    - Transformation der plattformunabhängigen Applikationsmodelle in plattformspezifische Modelle
      - Konfigurationsmodelle für ACROSS MPSoC, die insbesondere die Kommunikation zwischen Prozessorkernen beschreiben.
      - Konfigurationsmodelle für PikeOS, die OS-Partitionen, die Kommunikation zwischen diesen sowie die Anbindung an die Core-2-Core-Kommunikation über das TTNoC beschreiben.
  - Modell-zu-Text-Transformation (Code-Generierung)
    - Generierung von Applikationscode aus generischen und domänen-spezifischen Modellen
    - Optional: Generierung gegen Betriebssystem API
  - Konfigurationsgenerierung

- **Task T 3.5 „Verification and Analysis Tools“**
  - CertPLC: Werkzeug zur Verifikation von SPS-Anwendungen
  - Methoden zur automatischen Nutzung von Fehlertoleranzmechanismen
    - Zuverlässigkeitsanalyse, die unterschiedliche Fehlererkennungsmechanismen unterstützt.
    - Fehlertolerantes Multiprozessor-Scheduling
- **Task T 3.6 „Support for use of development methodology in the application domain“:**

Unterstützung der Applikations-Arbeitspakete WP4, 5, 6 bei der Umsetzung der Demonstratoren durch die folgenden Aktivitäten:

  - Video-gestütztes Tele-Tutorial zur Installation und Nutzung der ACROSS-Werkzeugkette
  - Individueller Support auf Anfrage (E-Mail, Telefon, ...)
  - Teilnahme am Demonstrator Integration Workshop:
    - Vorführung praktischer Beispiele
    - Fehlerbehebung
    - Hilfe bei der Erstellung initialer Modelle
  - Pflege einer Eclipse-Update-Site, zur komfortablen Verteilung der Werkzeugkette im Konsortium
  - Bereitstellung eines Bug-Tracking-Systems
  - Bewertung und Validierung der Werkzeugkette
    - Validierung der Erfüllung der Anforderungen aus dem GENESYS-Projekt sowie der ACROSS-spezifischen Anforderungen
    - Auswertung der Erfahrungen in den Applikations-Arbeitspaketen (basierend auf den jeweiligen finalen Berichten)

#### 2.1.4 WP4 (Automotive)

An WP4 waren die folgenden deutschen Partner beteiligt: SYSGO

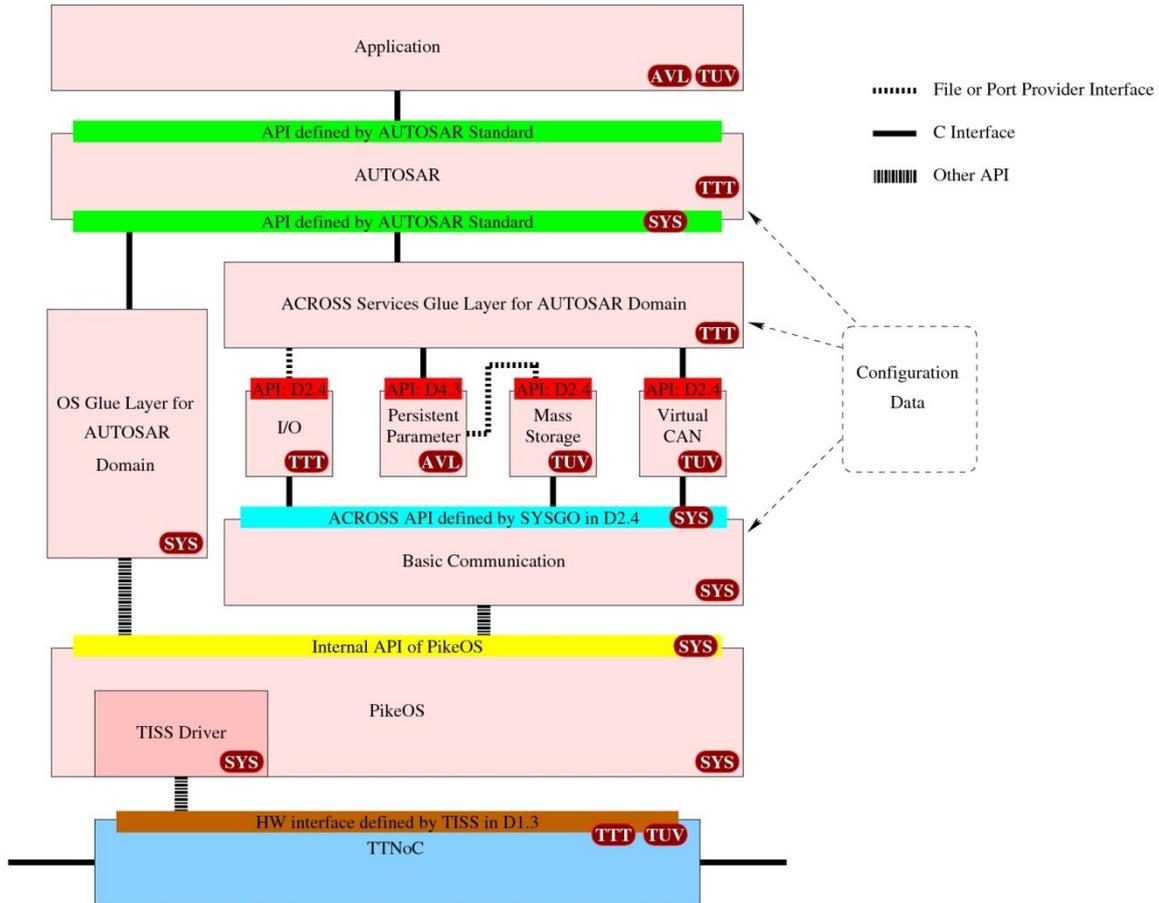
WP4 beschäftigte sich mit der Automotive-Domäne. In diesem Bereich wurde von AVL ein Demonstrator erarbeitet und vorgestellt. Der Ablauf zur Erstellung der Dienste-Landschaft in WP4 wurde an den von WP2 angelehnt und entsprechend in die Phasen Anforderungserhebung, Funktionale Planung, Design und Implementierung aufgeteilt. Zusätzlich gab es Tasks für den Demonstrator selbst.

Neben den von AVL durchgeführten Arbeiten, die direkt den Demonstrator umsetzen, d.h. die Arbeiten, die die Anwendungsschicht betreffen, wurden in WP4 die Spezifischen Optionalen Dienste für die Automotive-Domäne entwickelt. An diesem Bereich gab es Beteiligungen von deutschen Partnern, vornehmlich der SYSGO AG, die in Zusammenarbeit mit dem Subunternehmer OpenSynergy die Anbindung des PikeOS-Betriebssystems an die AUTOSAR-OS-Schicht durchführte.

Die folgenden Domänen-spezifischen Optionalen Dienste wurden erarbeitet:

- DSS1\_AU/AUTO1: AUTOSAR Service, AUTOSAR Stack (TTTech)
- DSS1\_AU/AUTO2: AUTOSAR Service, OS Glue Layer (**SYSGO**)
- DSS2\_AU/AUTO4: Persistent Parameter Service (TU Wien)
- DSS3\_AU/AUTO3: Automotive-domain specific Interface Service (TTTech)

Der Service DSS1\_AU wurde zwischen den Deliverables D4.2 und D4.3 aufgespalten, um eine bessere Zuordnung der Verantwortlichkeit zu den Projektpartnern zu ermöglichen. Dies hing auch damit zusammen, dass sich das Verständnis der komplexen AUTOSAR-Materie im Verlauf des Projektes verbessert hatte. Um eine klare Situation zu schaffen, wurden die Dienste auch mit einer neuen Bezeichnung versehen (AUTOx statt DSS\_AUy).



**Abbildung 9: AUTOSAR-Stack mit verschiedenen Software-Schichten und verantwortlichen Partnern. Die mit ‚SYS‘ markierten Teile wurden durch SYSGO in teilweiser Zusammenarbeit mit dem Subunternehmer OpenSynergy bearbeitet. Die übrigen Arbeiten sind auf die europäischen Partner Technische Universität Wien (TUV), TTTech (TTT) und AVL verteilt.**

Die Arbeiten an der Anbindung von AUTOSAR an das PikeOS-Betriebssystem, dem AUTOSAR-OS, wurden vom Subunternehmer OpenSynergy für SYSGO AG durchgeführt. Im Rahmen dieser Arbeiten wurde eine direkte, verbesserte Anbindung an das Betriebssystem erreicht, um die nötige Performance auf der NIOS-2-Architektur zu erreichen. Zu Beginn des ACROSS-Projekts gab es nur eine prototypische Implementierung, die auf der POSIX-Personality aufbaute und die im Rahmen dieser Zusammenarbeit gegen eine native, effiziente Lösung ersetzt wurde.

In Abbildung 9 sieht man den Aufbau der Architektur in WP4. Im unteren Teil ist die Hardware, insbesondere das TTNoC-Netzwerkmodul zu sehen. Die niedrigste Software-Schicht bildet das PikeOS-Betriebssystem, in dessen Rahmen auch der Treiber für den Netzwerkzugriff (der „TISS-Treiber“) entwickelt wurde. Der AUTOSAR-Stack besteht aus vielen Software-Modulen, die von unterschiedlichen Projektpartnern beigesteuert wurden. Dank der starken Standardisierung von AUTOSAR war ein Baukastensystem relativ einfach umzusetzen. Bei gemeinsamen Integrationsworkshops in Wien wurden

die letzten Hürden, die teilweise durch unterschiedliche AUTOSAR-Standards verursacht wurden, erfolgreich überwunden.

AUTOSAR sieht vor, dass hier Teile des C-Codes aus Spezifikationen generiert werden, welches von OpenSynergy durchgeführt wurde. Die Anbindungsschicht zur Applikation wurde z.T. von SYSGO definiert, so wie es sich im PikeOS-Betriebssystem darstellt. Desweiteren wurden von SYSGO die grundlegenden Dienste zur Kommunikation über das TTNoc bereitgestellt, über die die CAN-Nachrichten durch das NoC zum Knoten 1, dem I/O-Knoten getunnelt wurden. Diese Konstruktion ist nötig, weil im ACROSS-System nur der I/O-Knoten äußere Netzwerk- und Busanbindung hat. Die Kommunikationsdienste sind generische Dienste, die in D2.4 im Rahmen von WP2 beschrieben wurden.

Desweiteren wurde in WP4 nach der Implementierung Support bei der Benutzung der Automotive-spezifischen Middleware und Systemkomponenten geleistet. In diesem Rahmen fanden in Wien zwei Treffen zwischen der AVL, TU Wien, TTTech, SYSGO und OpenSynergy statt, bei dem die Integration des Demonstrators erfolgreich vorangetrieben wurde.

#### **2.1.4.1 Ziele**

Das übergeordnete Ziel in WP4 war die Erstellung eines Demonstrators für die Automotive-Domäne auf der ACROSS-Plattform. Weitere Ziele waren die Entwicklung einer Middleware-Schicht sowie der im vorangehenden Abschnitt aufgeführten spezifischen Systemkomponenten.

#### **2.1.4.2 Ergebnisse**

Die Ergebnisse aus WP4 wurden in den folgenden Deliverables zusammengefasst:

- D4.1: Anforderungserhebung für die Automotive-Domäne
- D4.2: Funktionale Spezifikation der Automotive-Dienste
- D4.3: Design der Automotive-Dienste
- D4.4: Implementierung der Kontrollalgorithmen auf dem MPSoC
- D4.5: Testen der Kontrollalgorithmen
- D4.6: Spezifikation des Demonstrators
- D4.7: Design des Demonstrators
- D4.8: Implementierung des Demonstrators auf dem ersten MPSoC-Prototyp
- D4.9: Testen des Demonstrators
- D4.10: Bewertung und Erarbeitung von Metriken zum Testen der Applikation
- D4.11: Bedeutungsanalyse der Applikation auf dem MPSoC

Bei diesen Deliverables war als einziger deutscher Partner SYSGO beteiligt an den nicht-Demonstrator-spezifischen Deliverables. Die Deliverables zum Demonstrator wurden von anderen europäischen Projektpartnern erarbeitet.

Daneben gibt es die folgenden Ergebnisse:

- Software: NIOS-2-Port von AUTOSAR-OS für PikeOS. Dies ist eine effiziente Neuimplementierung basierend auf der Native-Personality von PikeOS, die SYSGO und OpenSynergy auch auf andere Architekturen portieren wollen, um sie kommerziell zu nutzen.
- Software: Anbindung der ACROSS-Netzwerk-Dienste an AUTOSAR (z.B. CAN über TTNoc).

### 2.1.4.3 Vorgehen

Das Arbeitspaket 4 war in mehrere Tasks unterteilt. Die nicht-Demonstrator-spezifischen Tasks wurden auch unter deutscher Beteiligung bearbeitet (T4.1 – T4.4). Die folgende Liste gibt eine Übersicht über die Arbeiten in WP4.

- **T4.1 „Requirements of automotive-domain services“**
  - Erarbeiten von Anforderungen an die automotive-domänen-spezifischen Dienste anhand von Eingaben aus WP4.
  - Auswerten, Zusammenstellen der Anforderungen
  - Planung einer Menge von Generischen Optionalen Diensten
  - Zusammenfassung der Anforderungen in D4.1
- **T4.2 „Specification of automotive-domain services“**
  - Ausbau der Anforderungen zu einer funktionalen Beschreibung
  - Zuordnung von Generischen Optionalen Diensten zu Partnern
  - Pro Partner: Ausarbeitung einer Beschreibung der Funktionalität des Dienstes
- **T4.3 „Design of automotive-domain services“**
  - Ausarbeitung eines technischen Konzepts zur Umsetzung der Generischen Optionalen Dienste in der ACROSS-Plattform
  - Definition von Schnittstellen
- **T4.4 „Implementation of control algorithms on MPSoC“**
  - Umsetzung der technischen Konzepte aus T4.3 in einer Implementierung in Software
- **T4.5 „Testing of control algorithms on MPSoC“**
- **T4.6 „Specification of demonstrator“**
- **T4.7 „Design of demonstrator“**
- **T4.8 „Implementation of demonstrator applications using the first prototype of MPSoC“**
- **T4.9 „Testing of demonstrator applications using the final MPSoC“**
- **T4.10 „Assessment, identification of the metrics for benchmarking according to application“**
- **T4.11 „Impact analysis of the applications on MPSoC“**

### 2.1.5 WP5 (Aerospace)

An WP5 waren die folgenden deutschen Partner beteiligt: EADS-DE, EADS-IW, SYSGO

#### 2.1.5.1 Ziele

Das Ziel von WP5 war es, die ACROSS Referenzplattform, die aus dem ACROSS MPSoC, der generischen Middleware sowie der werkzeuggestützten Entwicklungsmethodik besteht, auf die Domäne der zivilen Luftfahrt anzuwenden. Hierzu wird die generische Plattform durch geeignete auf den Bereich der Luftfahrt zugeschnittene HW-Schnittstellen und Middleware-Dienste erweitert. Um die Anwendbarkeit und die Leistungsfähigkeit des Ansatzes zu bewerten, werden domänen-spezifische Demonstratoren auf Basis der ACROSS-Referenzplattform aufgebaut und untersucht.

Die Ziele des Arbeitspaketes, das unter der Federführung von EADS-DE durchgeführt wurde, lassen sich wie folgt zusammenfassen:

1. Konsolidierung bereits verfügbarer Anforderungen
  - GENESYS-Projekt
  - ARTEMIS Strategic Research Agenda (SRA)
2. Implementierung domänen-spezifischer Middleware-Dienste
3. Aufbau von zwei Aerospace Application im Cooperative UAV Demonstrator (Demonstrator 5-A) sowie dem Degraded Vision Landing Aid Demonstrator (Demonstrator 5-B):
  - Das deutsche Konsortium war für den Demonstrator 5-B zuständig. Hierbei lag der Fokus auf dem Applikationsteil einer typischen Luftfahrtanwendung.
  - Evaluierung von Plattform und Entwicklungsmethodik.
4. Evaluation der Eignung der ACROSS MPSoC Architektur für sicherheits-kritischen Anwendungen in der Domäne Luftfahrt

### 2.1.5.2 Ergebnisse

Zusammenfassend lässt sich feststellend, dass alle der in Abschnitt 2.1.5.1 aufgeführten Ziele erreicht wurden: Ergänzend zu einer konsolidierten Anforderungsliste an den domänen-übergreifenden Teil der ACROSS-Methodik wurden domänen-spezifische Middleware-Dienste in den ACROSS-Ansatz integriert. Auf dieser Basis wurde unter Leitung des Projektpartners EADS-DE erfolgreich der Demonstrator 5-B aufgebaut.

Als Beispielanwendung für Demonstrator 5-B wurde das von EADS-DE entwickelte DeViLA System verwendet. DeViLA steht für **Degraded Vision Landing Aid**. Es handelt sich um ein Assistenzsystem für Hubschrauber Piloten zur Unterstützung von Landungen unter reduzierter Sicht, z.B. infolge von aufgewirbeltem Staub. DeViLA fasste verschiedene Sensordaten zusammen, z.B. Flughöhe, Position, Abstand zu Landestelle, Geschwindigkeit und Drift. Es verarbeitet diese Informationen und erzeugt ein für Piloten leicht verständliches Videobild, siehe Abbildung 10.



Abbildung 10: Beispiel für Anzeigen des DeViLA Systems

Der prinzipielle Aufbau von Demonstrator 5-B ist in Abbildung 11 dargestellt. Er dient der interaktiven Demonstration der Fähigkeiten der neuen, auf dem ACROSS MPSoC basierenden DeViLA Recheneinheit. Ein kommerzieller Flugsimulator (Microsoft FXS) stellt eine interaktive, Joystick basierende Benutzerschnittstelle zur Verfügung. Der Flugsimulator erzeugt Echtzeit-Sensordaten eines Hubschraubers und gibt sie über ein Standard-Avionic-Interface (ARINC 429) an die DeViLA Recheneinheit weiter. Die Recheneinheit verarbeitet die Daten, erzeugt das Videobild und gibt es über eine HDMI Schnittstelle an einen handelsüblichen TFT-Monitor aus.

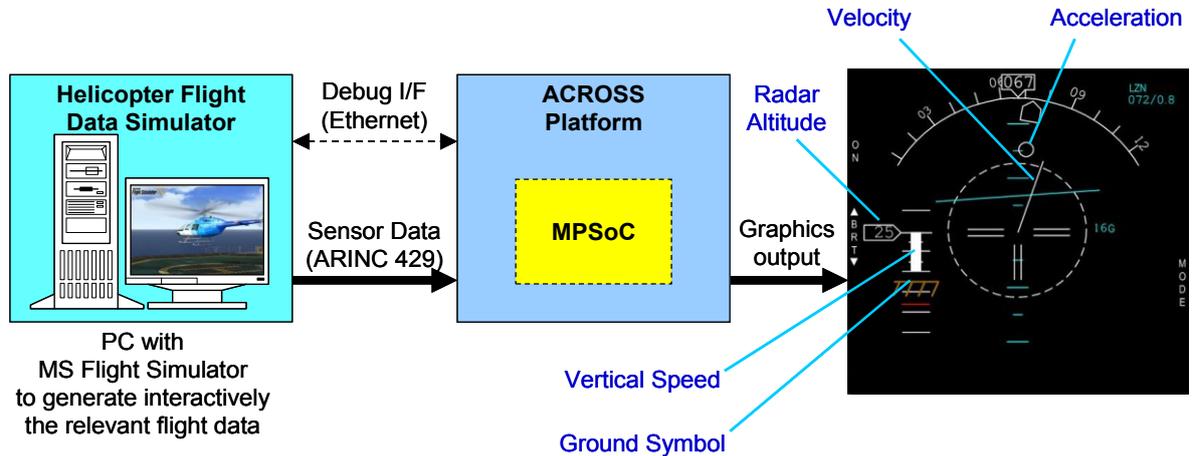


Abbildung 11: Prinzipieller Aufbau von Demonstrator 5-B

Der detaillierte Aufbau von Demonstrator 5-B ist in Abbildung 12 dargestellt. Die interaktiven Flugdaten des im Microsoft Flugsimulator X integrierten Helikopter-Modells werden mittels der Flight Simulator Universal Inter Process Communication (FSUIPC) Schnittstelle an die Aircraft Data Bridge übertragen. Sie extrahiert die von der DeViLA Recheneinheit benötigten Informationen (z.B. Flughöhe über Boden, Geschwindigkeit, Flugrichtung, Rollwinkel) und emuliert bei diesen Daten das Zeitverhalten eines realen Helikopters. Anschließend wandelt sie die Daten in das ARINC 429 Format um und leitet sie an den ARINC 429 Converter weiter. Der ARINC 429 Converter führt den Datentransfer zur ACROSS Plattform gemäß ARINC 429 Protokoll durch.

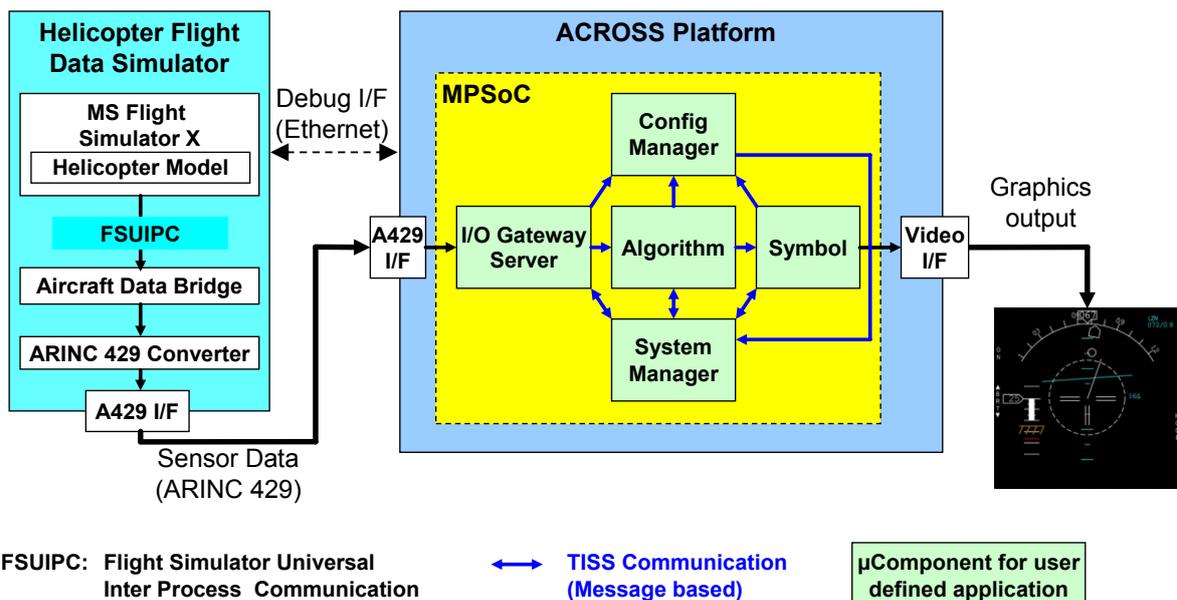


Abbildung 12: Detaillierter Aufbau von Demonstrator 5-B

Im Rahmen von WP5 wurde das DeVILA System auf das ACROSS MPSoC portiert, siehe Abbildung 13. Die Systemfunktionen wurden dabei auf vier Rechnerknoten des MPSoCs verteilt. Diese Knoten kommunizieren ausschließlich über das TTNoC miteinander. Die Knoten besitzen folgende Funktionen:

- **I/O Gateway Server:**  
Wickelt die komplette externe Kommunikation des DeVILA Systems ab.
- **Algorithm:**  
Führt folgende Funktionen des DeVILA Systems aus
  - Verarbeitung der Sensordaten
  - Plausibilitätscheck der Sensordaten
  - Erzeugung der digitalen DeVILA Symbol Anweisungen
- **Symbol:**  
Verarbeitet die von dem Knoten „Algorithm“ erzeugten digitalen Symbol Anweisungen. Diese Anweisungen werden in Grafik-Elemente zerlegt und in einen lokalen Bildspeicher geschrieben. Der Inhalt dieses Bildspeichers wird dann in ein digitales HDMI Video Signal umgewandelt und an dem TFT-Monitor ausgegeben, siehe Abbildung 14.
- **Config Manager und System Manager:**  
Werden auf einem gemeinsamen Rechnerknoten ausgeführt. Sie haben folgende Aufgaben
  - Booten des Systems inkl. SW-Konsistenzprüfung
  - Überprüfung der eingebauten HW Monitore (z.B.: lokale Spannungsüberwachung)
  - Fehlerbehandlung, Fehlerspeicherung und Fehlerreporting

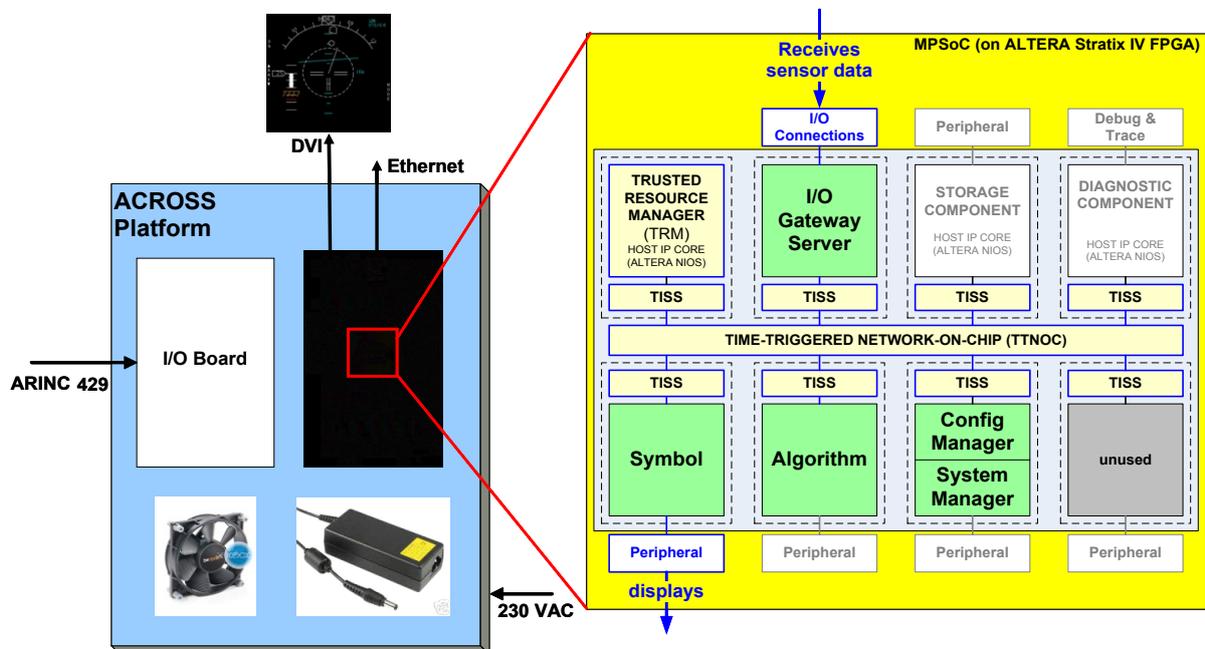


Abbildung 13: Übersicht Demonstrator 5-B Hardware inkl. Aufteilung der MPSoC Rechnerknoten

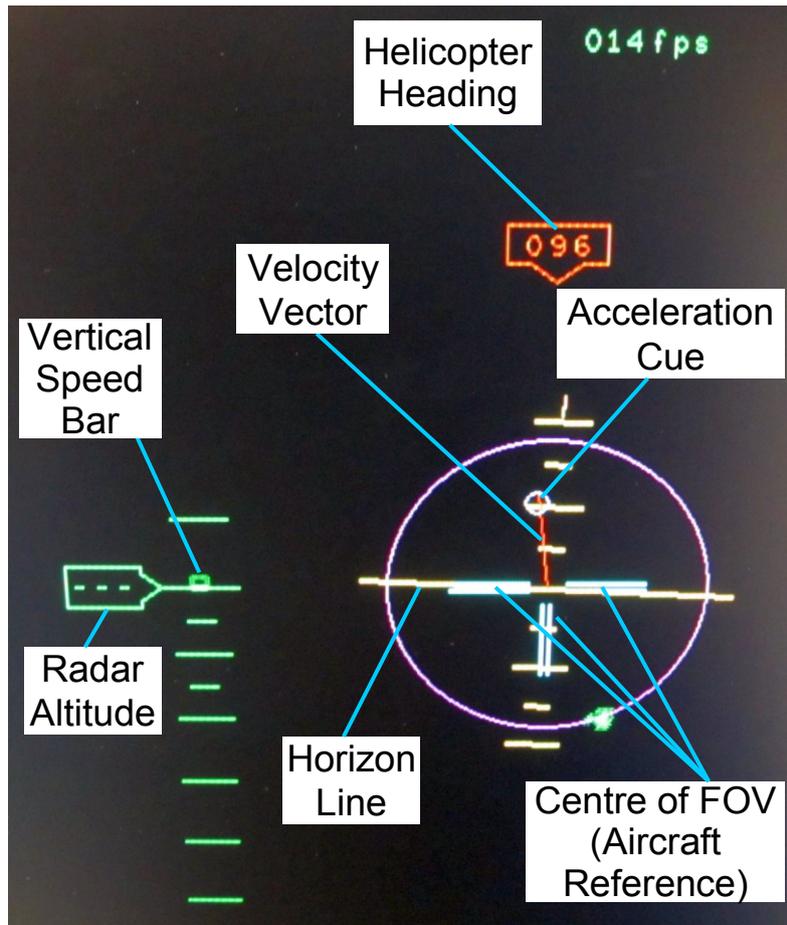


Abbildung 14: Von Demonstrator 5-B erzeugtes Videobild

Die SW Architektur von Demonstrator 5-B ist in Abbildung 15 dargestellt.

Als Ergebnis der Evaluation von Demonstrator 5-B lässt sich festhalten, dass der ACROSS Ansatz sehr gut für den Einsatz in der Luftfahrt-Domäne geeignet ist. Die Analyse der MPSoC Architektur zeigte, dass wesentliche für den Einsatz in der Luftfahrt benötigten Anforderungen erfüllt werden, wie z.B. die funktionale Unabhängigkeit und die Garantie von Echtzeiteigenschaften. Insbesondere die inhärente Segregierung und die klare Struktur des ACROSS MPSoCs erleichtern den Nachweis einer korrekten Funktionalität von Anwendungen mit unterschiedlicher Kritikalität auf einer gemeinsamen Hardwareplattform. Für den industriellen Einsatz sind jedoch noch folgende Verbesserungen notwendig:

- Ersatz der NIOS-II-Kerne durch performantere Prozessorkerne
- Werkzeugkette:
  - Automatisierung aller Transformationsschritte
  - Zertifizierung

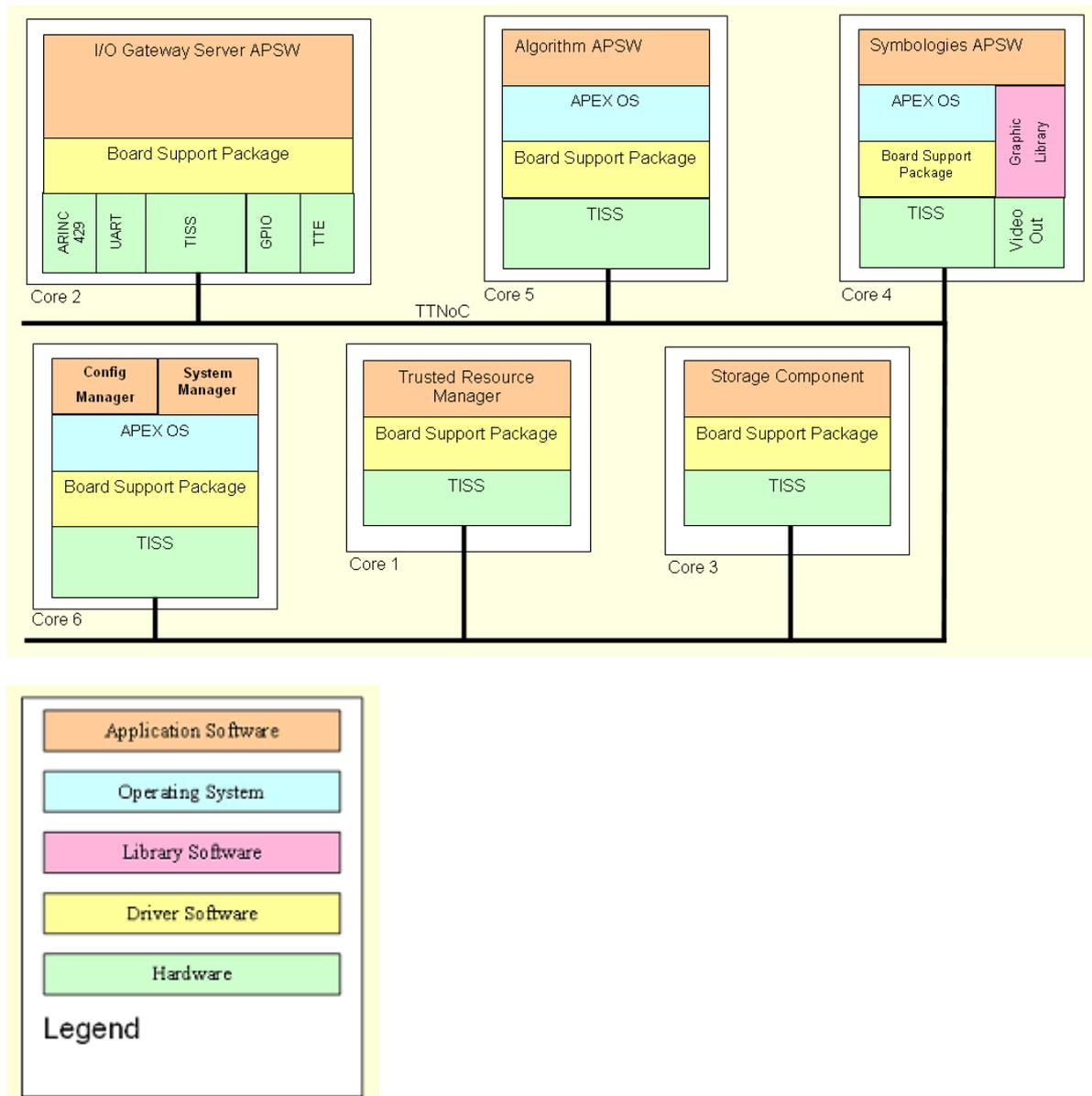


Abbildung 15: Demonstrator 5-B SW Architektur

Die Implementierung von Demonstrator 5-B erfolgte in folgenden Schritten:

- Erstellung von Modellen ausgewählter Services zur Validierung in frühen Projektphasen (Software / Modelle)
- Implementierung der Luftfahrt-spezifischen Dienste (Software)
- Implementierung der zugehörigen Test-Suites (Software)
- Vorläufige Implementierung der Applikationssoftware des Aerospace Application Demonstrators B auf Emulations-HW zur Validierung und Optimierung der APEX-Schnittstelle.
- Implementierung des Aerospace Application Demonstrators B auf der endgültigen MPSoC Plattform unter Benutzung von 1 Core und PikeOS Software mit 3 Partitionen
- Endgültige Implementierung des Aerospace Application Demonstrators B auf der endgültigen MPSoC Plattform mit 3 Cores

Das deutsche Konsortium hat zu den folgenden Berichten zur Beschreibung des Aerospace Application Demonstrator B beigetragen:

- D5.1 Requirements document
- D5.2 Specification of aerospace domain specific services
- D5.3 Design specification of aerospace-specific services
- D5.4 Report in implementation of aerospace-specific services
- D5.5 Test suite for aerospace-specific services
- D5.6 Test report for aerospace-specific services
- D5.7 Specification of aerospace application demonstrator
- D5.8 Design specification of aerospace application demonstrator
- D5.9 Preliminary implementation of the aerospace application demonstrator
- D5.10 Final implementation of the aerospace application demonstrator
- D5.11 Report on impact analysis of MPSoC
- D5.12 Report on assessment using aerospace application demonstration

### 2.1.5.3 Vorgehen

In WP5 wurden folgende Demonstratoren mit unterschiedlichen Anwendungen aus dem Bereich Luftfahrt erstellt, um die Eigenschaften der ACROSS-Architektur zu untersuchen und zu bewerten.

- Aerospace Application Demonstrator A (Demonstrator 5-A) entstand unter der Leitung von SELEX-SI.
- Aerospace Application Demonstrator B (Demonstrator 5-B) entstand unter der Leitung von EADS-DE.

Die Anwendungen der Demonstratoren sind unabhängig voneinander. Sie ergänzen sich dahingehend, dass Demonstrator 5-A auf die SW-Aspekte bei der Zusammenführung einer großen Anzahl von sicherheits-kritischen Kommunikationssystem fokussiert ist, wohingegen der Demonstrator 5-B die Vorteile der ACROSS Architektur bezüglich der HW-Implementierung einer sicherheitskritischen Luftfahrtanwendung untersucht.

Das deutsche Konsortium war hierbei unter Beteiligung der Partner EADS-DE, EADS-IW und SYSGO an der Umsetzung von Demonstrator 5-B beteiligt (siehe Abschnitt 2.1.5.2).

Hierzu wurden die folgenden Schritte vorgenommen:

- **T 5.1 „Requirements consolidation“:**  
Analyse der Anforderungen für
  - ACROSS MPSoC Plattform (in Hardware implementierte Basisdienste)
  - Generische optionale Services (Software)
  - Entwicklungsmethodik
  - Domänen-spezifische Dienste
    - Programmierschnittstelle ARINC 653 (APEX)
    - Real-time Data Distribution Services for ARINC 653 over ACROSS MPSoC (RTDDS over APEX)
- **T 5.2 „Specification of domain-specific services“:**  
Spezifikation von domänen-spezifischen Diensten, welche die in T 5.1 erarbeiteten Anforderungen erfüllen. T 5.2 gliedert sich dabei in folgende Sub-Tasks:
  - **T 5.2a Specification of domain-specific service : RT-DDS Middleware:**  
Spezifikation einer RT\_DDS Middleware unter Verantwortung von SELEX-SI

- **T 5.2b Specification of domain-specific service : Operating system:**  
Spezifikation eines Luftfahrt-spezifischen ARINC 653 Interface (APEX) unter Verantwortung des deutschen Konsortiums
- **T 5.2c Specification of domain-specific services : Reliability analysis:**  
Spezifikation der Anforderung an ein Fault-Injection Tool unter Verantwortung von EADS-FR und TRTFR
- **T 5.3 „Design of domain specific services“:**  
Verfeinerung des Designs der in T 5.2 spezifizierten domänen-spezifischen Dienste.  
T 5.3 gliedert sich dabei in folgende Sub-Tasks:
  - T 5.3a Design of domain specific services : RT-DDS Middleware  
in Verantwortung von SELEX-SI
  - T 5.3b Design of domain specific services : Operating System  
in Verantwortung von EADS-DE
  - T 5.3c Design of domain specific services : Reliability analysis methodology  
in Verantwortung von EADS-FR und TRTFR
- **T 5.4 „Implementation of domain specific services“:**  
Jeweils Implementierung einer vorläufigen Version für die Zwischenintegration des Demonstrators (T 5.8) und der endgültigen Version (T 5.9) der domänen-spezifischen Dienste.
  - **T 5.4a Implementation of domain specific service : RT-DDS middleware**  
in Verantwortung von SELEX-SI
  - **T 5.4b Implementation of domain specific service : operating system**  
in Verantwortung von EADS-DE
  - **T 5.4c Implementation of domain specific services : reliability analysis methodology**  
in Verantwortung von EADS-FR und TRTFR
- **T 5.5 „Testing of domain specific services“:**  
Koordination der Erstellung einer Unit-Test-Suite für die domänen-spezifischen Dienste.
  - **T 5.5a Testing of domain specific services needed for demonstrator-A**  
in Verantwortung von SELEX-SI
  - **T 5.5b Testing of domain specific services needed for demonstrator-B**  
in Verantwortung von EADS-DE
- **T 5.6 „Specification of demonstrator application“**
  - **T 5.6a Specification of demonstrator application-A**  
in Verantwortung von SELEX-SI
  - **T 5.6b Specification of demonstrator application-B**  
in Verantwortung von EADS-DE.  
Spezifizierung von
    - Übergeordneter Systemfunktion: Helicopter Landing Aid System
    - Segmentierung in Stimulation, Processing und Display Unit
    - HW- und SW Architektur der einzelnen Segmente
- **T 5.7 „Design of demonstrator applications“**
  - **T 5.7a Design of demonstrator application-A**  
in Verantwortung von SELEX-SI
  - **T 5.7b Design of demonstrator application-B**  
in Verantwortung von EADS-DE.  
Verfeinerung der in T5.6b definierten Elemente. Festlegung von
    - HW-Umgebung

- SW-Umgebung
  - detaillierte Funktion der drei Applikationen „I/O Server“, „Algorithm“ und „Symbol“
  - Boot-Sequenz
  - Datenfluss
  - Health Monitoring
- **T 5.8 „Preliminary Implementation of demonstrator applications“**
    - **T 5.8a Preliminary Implementation of demonstrator application-A**  
in Verantwortung von SELEX-SI
    - **T 5.8b Preliminary Implementation of demonstrator application-B**  
in Verantwortung von EADS-DE.  
Implementierung und Integration der drei Applikationen auf einer vorläufigen HW-Plattform zur Reduktion des Integrationsrisikos auf der Zielplattform. Hierzu emuliert die vorläufige Plattform die APEX Schnittstelle der endgültige ACROSS-MPSoC Plattform.
  - **T 5.9 „Implementation of demonstrator applications on final platform“**
    - **T 5.9a Implementation of demonstrator application-A on final platform**  
in Verantwortung von SELEX-SI
    - **T 5.9b Implementation of demonstrator application-B on final platform**  
in Verantwortung von EADS-DE.  
Implementierung und Integration der drei Applikationen auf der endgültige ACROSS-MPSoC Plattform.
  - **T 5.10 „Assessment using demonstrator applications“:**  
Auswertung der Demonstratoren 5-A und 5-B. Das deutsche Konsortium bewertete Demonstrator 5-B hinsichtlich
    - Determinismus (Garantie von Hard Real Time Scheduling durch das ACROSS MPSoC)
    - Funktionale Sicherheit (Fehlertoleranz und Segregierung unterschiedlicher Applikationen)
    - Hardware/Software compliance testing
    - Zuverlässigkeit
    - Verarbeitungsgeschwindigkeit
    - Wartbarkeit
    - Testbarkeit
  - **T 5.11 „Impact analysis of MPSoC“:**  
Analyse der Demonstratoren 5-A und 5-B. Das deutsche Konsortium bewertete anhand von Demonstrator 5-B, ob das im Projekt implementierte ACROSS MPSoC den typischen Anforderungen entspricht, die zur Zertifizierung für zivile Luftfahrtanwendungen erfüllt werden müssen.

Im Rahmen von WP5 wurden die beiden folgenden Workshops durchgeführt:

- **Workshop (15. – 17.11.2011):**
  - **„Safety Analysis of MPSoC Design according to civil aircraft standard ARP4754A“:**  
Validierung des MPSoC Design (TTNoC und zugehörige Middleware Services) bezüglich Zertifizierbarkeit gemäß ARP4754A.

- **„Avionic Development Processes according to DO-178B and DO-254“:**  
Analyse der beim MPSoC Design angewandten HW und SW Entwicklungsprozesse bezüglich Übereinstimmung mit den Anforderungen aus DO-254 und DO-178B.
- **Workshop „Assessment of MPSoC Arch w.r.t. DO-254“ (22.11.2012):**  
Bewertung der MPSoC HW Architektur hinsichtlich Übereinstimmung mit den Anforderungen aus DO-254.

### 2.1.6 WP6 (Industrial Control)

An WP6 waren die folgenden deutschen Partner beteiligt: fortiss, SYSGO

#### 2.1.6.1 Ziele

Das Ziel von WP6 war es, die ACROSS Referenzplattform, die aus dem ACROSS MPSoC, der generischen Middleware sowie der werkzeuggestützten Entwicklungsmethodik besteht, auf die Domäne der Industrieautomatisierung anzuwenden. Hierzu sollte die generische, domänen-übergreifende Plattform durch geeignete auf den Bereich der Industrieautomatisierung zugeschnittene Middleware-Dienste erweitert werden. Dementsprechend sollte auch die generische Entwicklungsmethodik durch eine Erweiterung auf die Bedürfnisse der Domäne angepasst werden. Um die Anwendbarkeit und die Leistungsfähigkeit des Ansatzes zu bewerten, wurden domänen-spezifische Demonstratoren auf Basis der ACROSS-Referenzplattform aufgebaut und untersucht.

Zusammenfassung der Ziele:

1. Konsolidierung bereits verfügbarer Anforderungen
  - GENESYS-Projekt
  - ARTEMIS Strategic Research Agenda (SRA)
2. Implementierung domänen-spezifischer Dienste
  - Middleware-Dienste auf Basis der ACROSS HW/SW-Plattform
  - Domänen-spezifische Standardsprachen / -Modelle (insbesondere IEC 61131-3)
3. Aufbau von Industrieautomatisierungs-Demonstratoren
  - deutsches Konsortium: Fokus auf Applikationsteil des Industrial Control Application Demonstrators
4. Evaluierung von Plattform und Entwicklungsmethodik

#### 2.1.6.2 Ergebnisse

Zusammenfassend lässt sich feststellend, dass alle der in Abschnitt 2.1.6.1 aufgeführten Ziele erreicht wurden: Ergänzend zu einer konsolidierten Anforderungsliste an den domänen-übergreifenden Teil der ACROSS-Methodik wurden domänen-spezifische Middleware-Dienste und ein IEC 61131-3-kompatibles Entwicklungswerkzeug in den ACROSS-Ansatz integriert. Auf dieser Basis wurde unter Leitung des Projektpartners SAGÖ erfolgreich der Industrial Control Application Demonstrator aufgebaut.

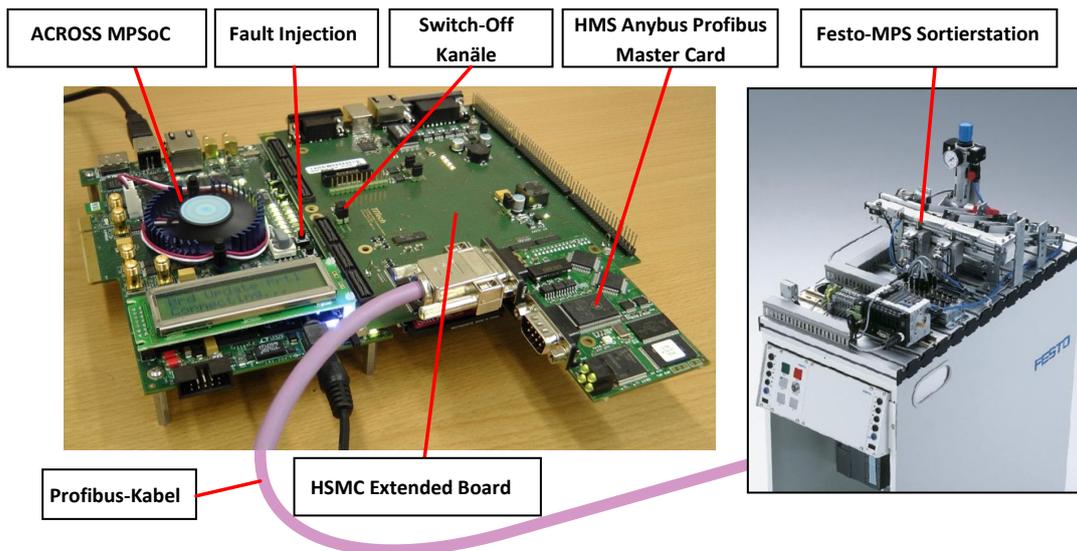


Abbildung 16: Aufbau des Industrial Control Demonstrators: Das ACROSS-MPSoC ist über Profibus an die Festo MPS Sortierstation angebunden

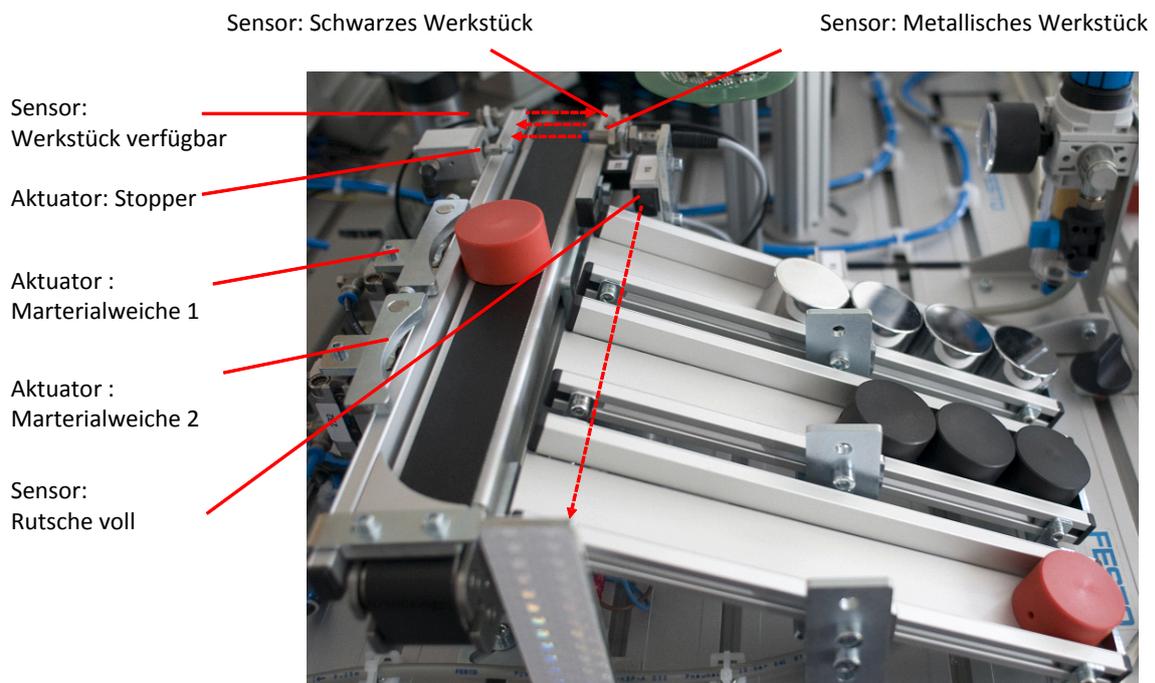


Abbildung 17: Festo MPS Sortierstation

Für den Demonstrator wurde die herkömmliche SPS-basierte Steuerung für eine Sortierstation aus der Festo MPS-Reihe durch das ACROSS-MPSoC ersetzt (siehe Abbildung 16). Hierzu wurde der FPGA-basierte Prototyp über das von WP1 entwickelte HSMC-Erweiterungsboard, das die elektrische Aufbereitung der FPGA-Signale übernahm, und eine kommerzielle Profibus-Karte der Firma HMS an die Station angebunden. Die Funktionalität des Demonstrators besteht darin, Werkstücke nach ihrer Materialeigenschaft (rot, schwarz, metallisch) in entsprechende Materialrutschen zu sortieren (siehe Abbildung 17). Dabei wurde dieser Vertreter einer typischen Aufgabe aus der Industrie-Automatisierung dazu genutzt, die Eignung des ACROSS-Ansatzes bezüglich Performanz (erreichbare Zykluszeit), Echtzeitfähigkeit, Zuverlässigkeit und funktionaler Sicherheit zu untersuchen. Darüber hinaus diente der Demonstrator dazu, die domänen-gerechte Entwicklungsmethodik, eines der Kernthemen des

deutschen Verbundprojektes, zu evaluieren. Abbildung 18 ist ein Überblick über den verwendeten Workflow zu entnehmen:

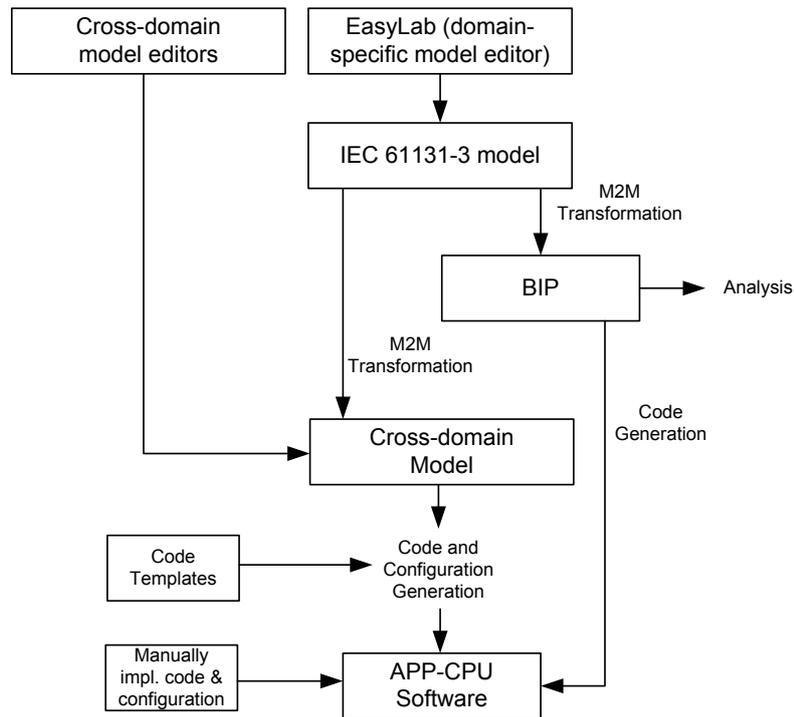


Abbildung 18: Workflow für den Industrial Control Demonstrator

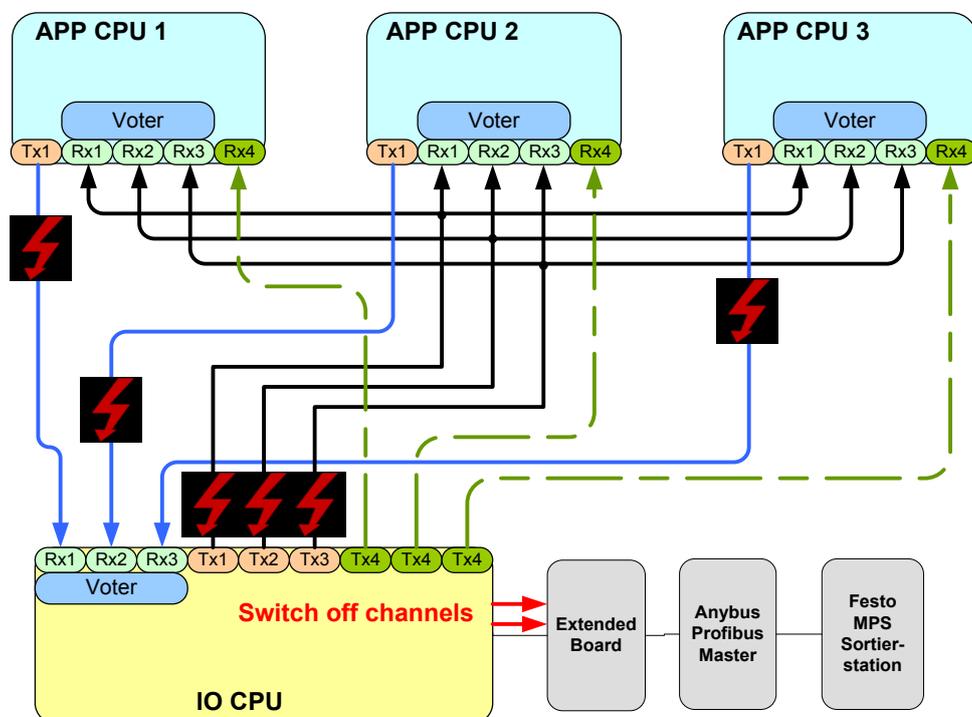


Abbildung 19: Fehlertolerante Systemarchitektur

Hieraus geht hervor, dass generische und domänen-spezifische Modelle als Eingabe dienen. Ein Ausschnitt der domänen-spezifischen Modelle, die zur Spezifikation der Anwendungsfunktionalität ver-

wendet werden, ist in Abbildung 20 zu sehen. Diese wurden mit Hilfe des von fortiss im Rahmen von WP6 zur Verfügung gestellten Entwicklungswerkzeuges EasyLab erstellt (domänen-spezifischen Dienst DSS4\_IC).

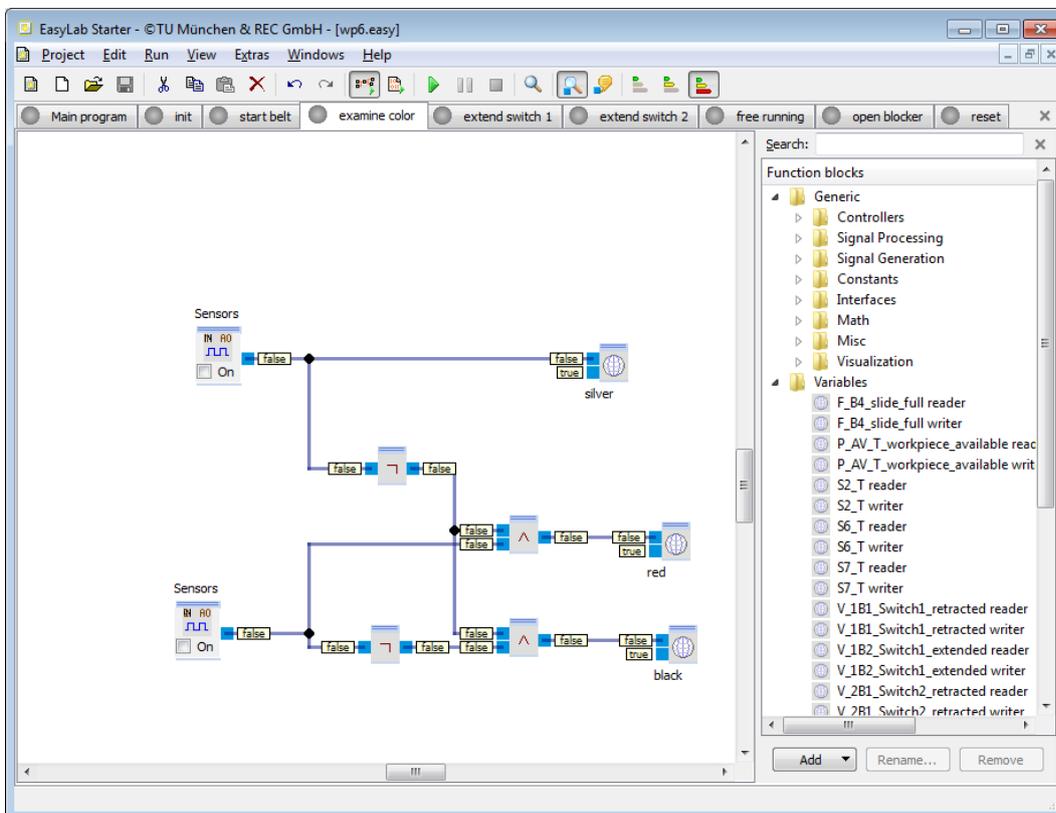
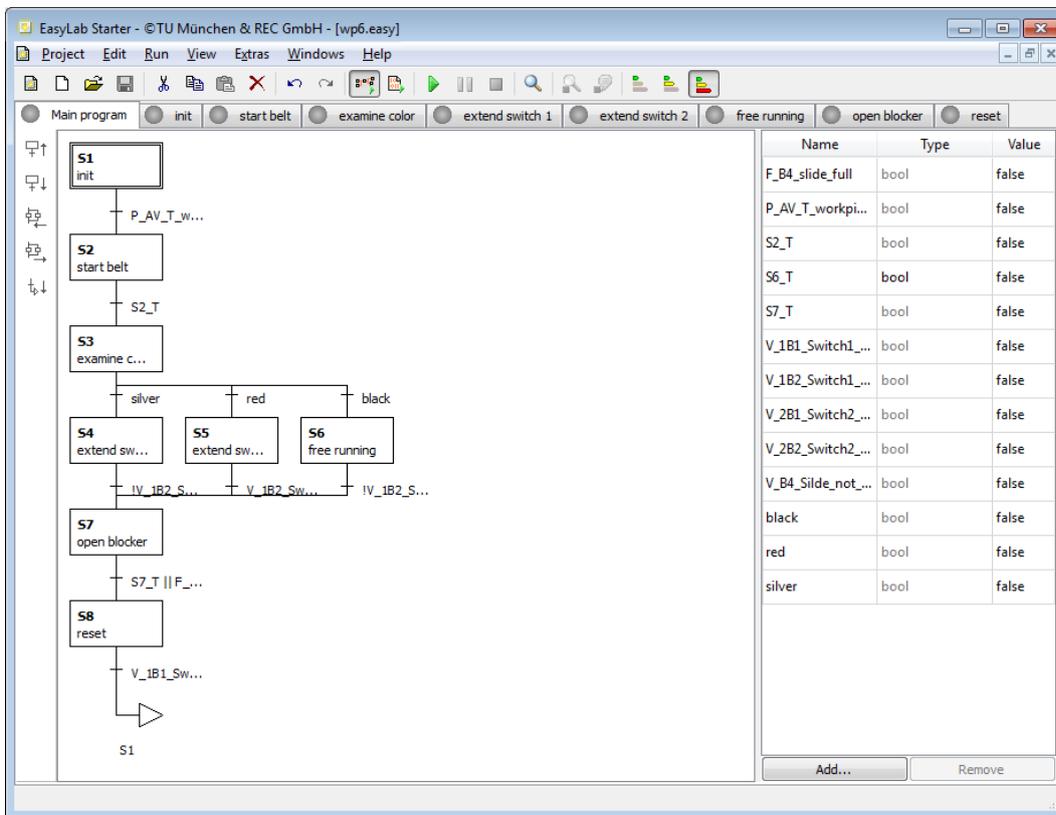


Abbildung 20: Domänen-spezifischen Modell des Funktionalität des Industrial Control Application Demonstrators: Sequence-Flow-Chart, das das Sortierverfahren umsetzt (oben), und Function-Block-Diagramm zur Bestimmung der Farbe eines Werkstückes (unten).

Die generischen Modelle wurden dazu verwendet, um die Architektur der verteilten Anwendung sowie nicht-funktionale Anforderungen wie etwa Echtzeitfähigkeit (Vorgabe der Ende-zu-Ende Latenz zwischen Sensor und Aktuator) und Zuverlässigkeit (fail-safe oder fail-operational Verhalten beim Auftreten einer Kommunikationsstörung) zu modellieren. Das im Zuge der Beschreibung der Ergebnisse von WP3 abgebildete Modell in Abbildung 7 zeigt eine fehlertolerante Version der Sortieranwendung. Andererseits dienen die Modelle als Eingabe für die von UJF/Verimag entwickelten Analyse- und Simulationsverfahren. Nach der Transformation in das domänen-übergreifende Modell aus WP3 werden diese Eingaben dazu verwendet, die folgenden Implementierungsartefakte zu generieren:

- Fehlertolerante Version der Applikation. Hierzu leitet ein automatisches Verfahren zur Entwurfsraumexploration ein Triple-Modular-Redundancy Schema her (unter Berücksichtigung von Zuverlässigkeitsanforderungen und Ausfallwahrscheinlichkeiten).
- Funktionaler Code für Sortieranwendung, der auf der API der von PikeOS zur Verfügung gestellten POSIX-Personality aufsetzt.
- Auf Basis der Eingabemodelle wurde die Kommunikationskonfiguration für das ACROSS MPSoC (durch Transformation in die von WP1 bereitgestellten Modelle) und PikeOS (durch Transformation in das dort verwendete XML-Format) generiert (Zykluszeit: 10 ms).

Die I/O-CPU-Software inklusive der Profibus-Anbindung der Sortierstation wurde von SAGÖ zur Verfügung gestellt.

Als Ergebnis der Evaluation des Demonstrators lässt sich festhalten, dass der ACROSS Ansatz aus den folgenden Gründen gut für die Domäne der Industrieautomation geeignet ist:

- Garantie von Echtzeiteigenschaften
- Unterstützung von Fehlertoleranz-Mechanismen: Das Resultat der Entwurfsraumexploration ist in Abbildung 19 zu sehen, bei der die eingehenden Sensorsignale durch redundant ausgelegte Sensoren an Replika der Applikationslogik gesendet werden (APP CPU 1, 2, 3). Die hier berechneten, redundanten Aktuator-Signale werden schließlich auf der IO CPU analysiert und im Erfolgsfall an die Station gesendet.

Zur Validierung der Lösung können über die in Abbildung 16 mit „Fault Injection“ beschrifteten DIP-Schalter Kommunikationsfehler provoziert werden. Falls ein Fehler erkannt wird, kann dieser durch das TMR-Schema korrigiert werden – bei zwei Fehlern wird ein Notaus-Signal an die Station gesendet.

Allerdings sind für den industriellen Einsatz weitere Verbesserungen notwendig:

- Ersatz der NIOS-II-Kerne durch performantere Prozessorkerne
- Werkzeugkette: Beide Punkte konnten der Natur eines Forschungsprojektes geschuldet im Rahmen des ACROSS-Projektes nicht umgesetzt werden:
  - Automatisierung aller Transformationsschritte
  - Zertifizierung

Das deutsche Verbundprojekt hat zu den folgenden Berichten zur Beschreibung des Industrial Control Application Demonstrators beigetragen:

- D 6.1 Requirement document WP6“
- D 6.2 Specification of Industrial Control domain specific services
- D 6.3 Design specification of Industrial Control domain specific services
- D 6.4 Source code of Industrial Control domain specific middleware
- D 6.5 Test suite for Industrial Control specific services
- D 6.6 Specification of Industrial Control specific application demonstrator
- D 6.7 Design specification of Industrial Control specific application demonstrator
- D 6.8 Preliminary implementation of Industrial Control specific application demonstrator
- D 6.9 Final implementation of Industrial Control specific application demonstrator
- D 6.10 Report on impact analysis of MPSoC
- D 6.11 Report on assessment using Industrial Control specific application demonstrator

Die Implementierung des Demonstrators gliedert sich in die folgenden Module:

- Finale Implementierung der Industrieautomations-spezifischen Dienste (Software)
- Finale Implementierung der zugehörigen Test-Suites (Software)
- Finale Implementierung des Industrial Control Application Demonstrators
- Portierung der POSIX-Personality auf die NIOS-II Version von PikeOS (Software)
- Modelle ausgewählter Services zur Validierung in frühen Projektphasen (Software / Modelle)

### 2.1.6.3 Vorgehen

In WP6 wurden unter Leitung von SAGÖ zwei Demonstratoren zur Validierung der ACROSS-Architektur und der Entwicklungsmethodik erstellt:

1. Industrial Control Application Demonstrator
2. Wireless Communication Application Demonstrator

Das deutsche Konsortium war hierbei unter Beteiligung der Partner fortiss und SYSGO hauptsächlich an der Umsetzung von Demonstrator 1 beteiligt (siehe Abschnitt 2.1.6.2).

Hierzu wurden die folgenden Schritte vorgenommen:

- **Task T 6.1 „Requirements“:** Analyse der Anforderungen an:
  - ACROSS MPSoC Plattform (in Hardware implementierte Basisdienste)
  - Generische optionale Services (Software)
  - Entwicklungsmethodik
  - Domänen-spezifische Dienste
    - Unterstützung für Standardmodelle und Sprachen (IEC 61131-3)
    - Programmierschnittstelle (POSIX)
    - Modell-zu-Modell-Transformationen zwischen domänen-spezifischen Sprachen und generischen Modellen.
    - Fault-Injection-Mechanismen zur Demonstration von Fehlertoleranzkonzepten
    - Safety-Dienste (Plausibilitäts-Prüfung, Not-Abschaltung)

- **Task T 6.2 „Specification of domain specific services“:** Spezifikation von domänen-spezifischen Diensten, die die o.g. Anforderungen erfüllen. Das deutsche Verbundprojekt war dabei für die folgenden Dienste verantwortlich:
  - DSS1\_IC: Plausibility Safety Service
  - DSS4\_IC: Domain Specific Language Service
- **Task T 6.3 „Design of domain specific services“**
  - Verfeinerung des Designs der in T 6.1 spezifizierten Services
    - DSS1\_IC: Plausibility Safety Service
    - DSS4\_IC: Domain Specific Language Service
  - Modellierung weiterer ausgewählter domänen-spezifischer Services unter Verwendung der Modelle und Methoden aus WP3
- **Task T 6.4 „Implementation of domain specific services“:** Jeweils Implementierung einer vorläufigen Version für die Zwischenintegration des Demonstrators (T 6.8) und der endgültigen Version (T 6.9)
  - DSS1\_IC: Plausibility Safety Service
  - DSS4\_IC: Domain Specific Language Service
- **Task T 6.5 „Testing of domain specific services“:** Koordination der Erstellung einer Unit-Test-Suite für die spezifizierten domänen-spezifischen Dienste:
  - Erstellung von Tests für die folgenden Dienste:
    - DSS1\_IC: Plausibility Safety Service
    - DSS4\_IC: Domain Specific Language Service
  - Unterstützung bei den Tests für die folgenden Dienste (vgl. T 6.8)
    - DSS75\_IC: App2IO Communication Service
    - DSS3\_IC: Switch Off Channel Service (Anteil, der auf Applikations-CPU läuft, s.o.)
- **Task 6.6 „Specification of demonstrator application“:** Spezifikation des Industrial Control Application Demonstrators
  - Analyse der zur Verfügung stehenden Konzepte
  - Mechanischer und elektrischer Aufbau der gewählten Demonstrator-Variante
  - Beschreibung der benötigten Software
- **Task “T 6.7 Design of demonstrator application“:** Verfeinerung der Spezifikation des Industrial Control Application Demonstrators aus T 6.6
  - Beitrag zur Software-Architektur für Gesamtdemonstrator
  - Design der Software für die Applikations-CPU's (I/O-SW durch SAGÖ)
  - Design der Fehlertoleranzmechanismen (Voting) und des Fault-Injection-Frameworks zu dessen Validierung
- **Task T 6.8 „Implementation of demonstrator application“:**
  - Im Rahmen dieses Tasks: Inbetriebnahme des Industrial Control Application Demonstrators bei fortiss
    - Mechanischer und elektrischer Aufbau der Festo MPS Sortierstation
    - Konfiguration der Profibus-Verbindung zwischen ACROSS MPSoC und Sortierstation (mit Unterstützung von SAGÖ).
  - Integration der von der ACROSS Plattform zur Verfügung gestellten Basisdienste und optionalen Dienste und der unten genannten domänen-spezifischen Dienste gegen die Interims-Plattform.
  - Implementierung der Demonstrator-spezifischen Applikation.

- Bei der Interims-Plattform handelt es sich um einen Single-Core Prototyp des ACROSS MPSoC, für den eine erste PikeOS-Version zur Verfügung gestellt wurde, die die NIOS-II-Architektur unterstützt.
- In diesem Task wurde durch SYSGO die Unterstützung bei der Anbindung des Demonstrators an die POSIX-Personality von PikeOS geleistet, die die Grundlage für die Applikationssoftware bildete.
- Hierbei war sah die Arbeitsteilung vor, dass das deutsche Konsortium (fortiss) für die Software auf den Applikations-CPU's verantwortlich war.
  - DSS75\_IC: App2IO Communication Service: Implementierung durch SAGÖ, Integrationstest durch fortiss
  - DSS3\_IC: Switch Off Channel Service (Anteil, der auf Applikations-CPU läuft, s.o.)
  - DSS1\_IC: Plausibility Safety Service
  - DSS4\_IC: Domain Specific Language Service: Generierung der Applikationslogik
- **Task T 6.9 „Implementation of demonstrator application on final platform“**
  - Dienstintegration und Implementierung der Demonstrator-spezifischen Applikation auf der finalen Version des ACROSS MPSoC
  - Bei der finalen Plattform handelt es sich um das ACROSS MPSoC, für den eine zugeschnittene PikeOS-Version zur Verfügung gestellt wurde. Im Rahmen dieses Tasks wurde auch die Unterstützung der POSIX-Personality an das TTNoC geleistet.
  - Weitere Arbeitsteilung analog zu T 6.8.
- **Task T 6.10 „Assessment using demonstrator application“**
  - Auswertung des Industrial Control Application Demonstrators
    - Messung von Zykluszeiten
    - Auswertung der Fehlertoleranzkonzepte
- **Task T 6.11 „Impact analysis of MPSoC“**
  - Analyse, ob das im Projekt implementierte ACROSS MPSoC typischen Anforderungen aus dem Bereich der Industrieautomatisierung entspricht (z.B. Echtzeit, funktionale Sicherheit, Zuverlässigkeit)
  - Analyse und Bewertung der domänen-spezifischen Entwicklungsmethodik und der entsprechenden Werkzeuge

### 2.1.7 WP7 (Dissemination & Exploitation)

An WP7 waren die folgenden deutschen Partner beteiligt: fortiss, EADS-DE, EADS-IW, Lauterbach, SYSGO.

#### 2.1.7.1 Ziele

In WP7 wurden unter der Leitung des österreichischen Projektpartners TTTech die folgenden Ziele verfolgt:

1. Bekanntmachung des Projektes und seiner Vision
2. Verbreitung der Projektergebnisse
3. Verfolgung von Verwertungsaktivitäten

### 2.1.7.2 Ergebnisse

Die unter Beteiligung der deutschen Verbundpartner veröffentlichten wissenschaftlichen Publikationen sind in Abschnitt 2.6 aufgeführt.

Darüber hinaus wurden im Rahmen des Projektes die folgenden Berichte unter der Beteiligung der deutschen Projektpartner erstellt:

- D7.1 Project homepage (Lieferung von Inhalten)
- D7.3 Use and Dissemination Plan (M13)
- D7.4 Use and Dissemination Plan (M25)
- D7.5 Use and Dissemination Plan (M41)

### 2.1.7.3 Vorgehen

Im Folgenden wird beschrieben, welchen Beitrag die über die diesem Schlussbericht berichteten Teilprojekte zu den o.g. WP7-Aktivitäten geleistet haben.

Punkt 1.) wurde in **Task T 7.3** („Public Relations“) verfolgt. Das deutsche Konsortium leistete die folgenden Beiträge:

- Inhalte für die ACROSS Homepage [www.across-project.eu](http://www.across-project.eu)
- Beiträge zu den ACROSS-Newslettern

Ziel 2.) wurde im Rahmen der **Tasks T 7.1** („Spreading Know-How“) und **T 7.2** („Training“) verfolgt und wie folgt unterstützt:

- ARTEMIS Autumn Event & ITEA 2 Co-Summit (Ghent, Belgien, Oktober 2010). Vertreter von fortiss beantworteten zusammen mit dem Projektkoordinator TU Wien Fragen der Besucher. Vorträge vor dem ARTEMISIA Steering Board und in der Multicore-Session.
- Präsentationen von EADS-DE, EADS-IW und fortiss auf dem Workshop „A cross-domain approach for mixed-criticality integration based on heterogeneous MPSoCs“, der im Rahmen der HiPEAC Konferenz 2012 (24. Januar, 2012 in Paris, Frankreich) durch den Koordinator TU Wien organisiert wurde.
  - „A cross-domain methodology for MPSoCs“ (S. Barner, fortiss)
  - „ACROSS in the Aerospace Domain“ (B. Koppenhöfer, EADS-DE; mit F. Fedi, SELEX-SI)
  - Panel-Diskussion
- EADS-DE: Präsentation des ACROSS-Projektes bei einem Workshop der CASSIDIAN-internen Forschungs- und Entwicklungsorganisation (November 2011)
- EADS-IW und EADS-DE: Präsentation des ACROSS-Projektes auf EADS-Ebene bei Airbus (November 2011) mit Diskussion der technischen Details von ACROSS.
- ARTEMIS Autumn Event & ITEA 2 Co-Summit (Paris, Frankreich, Oktober 2012). Vertreter von fortiss beantworteten zusammen mit dem Projektkoordinator TU Wien Fragen der Besucher.
- Vorträge auf einem auf Initiative des ACROSS-Projektes organisierten Workshops „Integration of mixed-criticality subsystems on multi-core processors“ auf der HiPEAC Konferenz 2013 in Berlin:
  - „The ACROSS Development Methodology“ (S. Barner, fortiss)
  - „Industrial Application of ACROSS Technology“ (B. Koppenhöfer, EADS-DE)
  - Panel-Diskussion

- SYSGO hat ein White-Paper über die Anbindung von zeitgesteuerten Architekturen an das PikeOS-Betriebssystem geschrieben und veröffentlicht, in dem die unterschiedlichen Architekturen aus INTERESTED und ACROSS noch einmal verglichen werden.
- Vorstellung und Diskussion von ACROSS auf dem Dagstuhl-Seminar 13052 „Multicore Enablement for Embedded and Cyber Physical Systems“, das von Michael Paulitsch, EADS-IW mit organisiert wurde. 30 Personen nahmen am Seminar teil, darunter 40 % aus der Industrie, was einen beträchtlichen Anteil für ein Dagstuhl-Seminar darstellt.
- Publikationen und ggfs. zugehörige Vorträge auf Konferenzen: Siehe Abschnitt 2.1.7.2

Auf die Maßnahmen zur Ergebnisverwertung (Punkt 3.) wird in Abschnitt 2.7 eingegangen. Darüber hinaus wurden die folgenden Aktivitäten ausgeführt (im Rahmen der Tasks **T 7.4 „Standardisation** und **T 7.5 „Exploitation“**):

- EADS-DE, EADS-IW haben das ACROSS-Projekt zusammen mit EADS-FR auf der „8th Advanced Technology Workshop and 2nd Innovation Fair“ Vertretern des EADS chief technology office und führenden EADS Innovatoren in Hamburg (21./22.03.2011) vorgestellt. Hierzu wurde eine Session zu Multicore-Prozessoren und Zertifizierung organisiert, in der der ACROSS-Ansatz vorgestellt wurde.
- EADS-DE, EADS-IW haben das ACROSS-Projekt zusammen mit EADS-FR als Teil der RTG (research technology group) for electronics vorgestellt (Mai 2010). RTGs sind EADS-interne Forschungsnetzwerke an denen alle großen Geschäftsbereiche (Airbus, EADS-DE, Eurocopter, und Astrium) teilnehmen.
- EADS-DE hat einen dreitägigen Workshop zum Entwicklungsprozess und zur Zertifizierung in der Luftfahrt bei TTech, Wien, Österreich veranstaltet (November 2011). Im Rahmen des Workshops wurde der ACROSS-Ansatz begutachtet und diskutiert. Als Ergebnis des Workshops konnte festgehalten werden, dass der ACROSS-Ansatz keine Hindernisse für Zertifizierungsaspekte für die Luftfahrt enthält. An diesem Workshop beteiligten sich EADS-IW, TU Wien und TTech.

## 2.1.8 WP8 (Project management)

### 2.1.8.1 Ziele

In WP8 wurde das Management und Controlling des gesamten ACROSS-Projektes durch den Projektkoordinator TU Wien durchgeführt (unterstützt durch TTT).

### 2.1.8.2 Ergebnisse

Die folgenden Berichte wurden unter Beteiligung der deutschen Verbundpartner erstellt.

- D8.2 Periodic Report 1 (M12)
- D8.3 Periodic Report 2 (M24)
- D8.4 Periodic Report 3 (M40)
- D8.5 Project Final Report (Part A, B, C) (M40)

Darüber hinaus lieferte das deutsche Konsortium Beiträge zu den projektinternen vierteljährlichen Zwischenberichten.

### 2.1.8.3 Vorgehen

Das deutsche Verbundprojekt unterstützte den Koordinator durch seine Zuarbeiten zu den entsprechenden technischen Berichten (siehe Abschnitt 2.1.8.2), internen Statusberichten (Projektfortschritt, Aufwände), bei der Fortentwicklung des Arbeitsplans und durch Teilnahme an den Konsortialtreffen.

## 2.2 Übersicht über zahlenmäßigen Nachweis

Die entstandenen Kosten entsprechen im Wesentlichen der Projektplanung aus dem Förderantrag.

Größer Posten waren die Personalkosten zur Umsetzung der in diesem Bericht beschriebenen Arbeitspakete. Darüber hinaus wurden in begrenztem Umfang Investitionen in das für den Aufbau der Demonstratoren benötigte Equipment getätigt.

Aus dem Projektverlauf ergab sich Notwendigkeit zur folgenden Anpassungen der ursprünglichen Planung:

- Um weitere Verzögerungen durch die in Abschnitt 1.3.2 beschriebenen technischen Schwierigkeiten bei der Erstellung der finalen Version des ACROSS MPSoCs zu vermeiden, setzte fortiss weniger studentische Hilfskräfte als geplant ein und ließ die entsprechenden Arbeiten durch wissenschaftliche Mitarbeiter ausführen.
- Lauterbach hatte einen deutlich kleineren Mittelbedarf als geplant. Hierfür gibt es zwei Hauptursachen: Durch die von Lauterbach gewählte stundengenaue pauschalierte Abrechnung, spiegeln die Kosten ausschließlich die produktiv für das Projekt eingesetzte Arbeit dar. Der Umfang der notwendigen Arbeit (gemessen in produktiven Arbeitsstunden) über drei Jahre wurde überschätzt.  
Durch die Verzögerungen in der Implementierungsphase des Projekts kam es zu längeren „Wartezeiten“, die durch die stundengenaue Abrechnung zu keinen projektbezogenen Kosten führten.
- EADS-IW hatte bei den Materialkosten signifikant den Plan zu Lasten der Personalmittel überschritten, da Hardware-Simulationssoftware in ACROSS benötigt wurde, diese aber ursprünglich nicht eingeplant war.

## 2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Im ACROSS-Projekt wurde eine neuartige, auf den Einsatz in sicherheitskritischen Anwendungen zugeschnittene Multicore-Architektur entwickelt und untersucht. Das vorliegende Verbundprojekt leistete hierzu einerseits zentrale Beiträge zum Software-Ökosystem, das für die effiziente Nutzung des neuartigen Prozessors benötigt wird (insbesondere: Betriebssystem und Werkzeugkette). Andererseits steuerte das Verbundprojekt realistische Anforderungen aus industriellen Anwendungen bei (insbesondere aus den Bereichen Luftfahrt und Industrieautomatisierung) und erprobte und bewertete das System in entsprechenden Demonstrator-Anwendungen.

Aufgrund der Neuartigkeit der Prozessorarchitektur, und dem ambitionierten Ansatz, die gesamte Spannweite vom Prozessorprototyp über die Systemsoftware (Betriebssystem, Middleware) bis hin zur Werkzeugkette im Projekt abzudecken, stellte das Vorhaben ein technisches Risiko dar, das von keinem der beteiligten Konsortialpartner hätte getragen werden können.

Durch das Erlangen der Projektziele (siehe Abschnitt 2.4) profitieren die deutschen Verbundpartner sowohl durch die Schaffung von Werten als auch durch den Aufbau von Knowhow direkt von der

Beteiligung am ACROSS-Projekt. Somit war die Unterstützung durch die Förderung des BMBF und des ARTEMIS JU, durch die das mit dem Vorhaben verbundene technische Risiko gemildert werden konnte, gerechtfertigt und notwendig. Zusammenfassend lässt sich feststellen, dass der Verlauf der Arbeit weitestgehend der im Projektantrag festgelegten Planung entsprach und nahezu alle im Arbeitsplan formulierten Aufgaben erfolgreich bearbeitet wurden (siehe auch: Schlussbericht der Universität Augsburg).

Wie in Abschnitt 1.3.2 erläutert, wurde für das verbleibende deutsche Konsortium eine Verlängerung der Projektlaufzeit um 4 Monate beantragt. Diese wurde durch die projektweite Verzögerung aufgrund technischer Schwierigkeiten bei der Fertigstellung der finalen Version des ACROSS MPSoC nötig. Allerdings waren hierzu keine zusätzlichen Ressourcen für das deutsche Verbundprojekt nötig.

## 2.4 Nutzen und Verwertbarkeit der Ergebnisse

Durch die im Projekt erreichten wissenschaftlichen/technischen Ziele ist ein wichtiger Schritt in Richtung einer auf sicherheitskritische Anwendungen zugeschnittenen Multicore-Architektur genommen worden. Da im Projekt Anforderungen aus unterschiedlichen Anwendungszweigen berücksichtigt wurden, steht nunmehr ein domänen-übergreifender Ansatz zur Verfügung, der eine umfassende Erschließung des Marktes mit sicherheitskritischen Anwendungen ermöglicht.

Die deutschen Technologie-Partner haben wesentliche Beiträge zum Software-Ökosystem für die ACROSS-Plattform geleistet, die im Folgenden kurz zusammengefasst werden (der Verwertungsplan wird in Abschnitt 2.7 ausführlich dargestellt):

- fortiss hat im ACROSS-Projekt eine Werkzeugkette zur modellgetriebenen Erstellung von Anwendungen erstellt, die auf dem ACROSS MPSoC aufsetzen. Neben der direkten Nutzbarkeit zur Programmierung und Konfiguration der ACROSS-Plattform ermöglicht es die modulare Architektur der Werkzeugkette jedoch, weitere Plattformen zu unterstützen und plattformunabhängige Ergebnisse aus dem ACROSS-Projekt somit indirekt weiter zu nutzen. Fortiss hat bei seinen Arbeiten die folgenden verwertbaren Ergebnisse erzielt:
  - Generische oder domänen-spezifische Meta-Modelle zur Beschreibung von Anwendungen (Funktionalität, Architektur) sowie von nicht-funktionalen Anforderungen (zeitliches Verhalten, Zuverlässigkeit).
  - Framework zur Modellierung von Ausführungsplattformen, das als Grundlage für die Erstellung der Meta-Modelle der ACROSS-Plattform diente.
  - Modell-zu-Modell Transformationen. Teilweise können diese unabhängig von der ACROSS-Plattform genutzt werden, wie etwa die Transformation von domänen-spezifischen Modellen in generische / domänen-übergreifende Meta-Modelle. Andere Transformationen (z.B. Generierung von Plattform-spezifischen Modellen durch Mapping eines Anwendungsmodells auf ein Modell der ACROSS-Plattform) sind zumindest in Teilen von der der ACROSS-Plattform abhängig.
  - Code- und Konfigurationsgeneratoren: Auch diese Module sind in Teilen von der Ausführungsplattform abhängig. So ließe sich z.B. die Anbindung der ACROSS-Werkzeugkette an das PikeOS-Konfigurationswerkzeug (CODEO) auch für andere von PikeOS unterstützten Zielplattformen übertragen (ggfs. mit Anpassungen).
  - Die Methoden zur automatischen Entwurfsraumexploration und Zuverlässigkeitsanalyse, die die optimale Abbildung von Applikationen auf die ACROSS-Plattform unter verschiedenen Nebenbedingungen ermöglichen, basieren auf dem o.g. Framework

zur Modellierung von Ausführungsplattformen und sind somit auch auf weitere Plattformen übertragbar.

- Lauterbach hat an das ACROSS-MPSoC angepasste Debugging- und Diagnosewerkzeuge bereitgestellt. Hierzu hat Lauterbach im Rahmen des ACROSS Projekts eine Debug-Schnittstelle und eine Trace-Schnittstelle definiert und seine TRACE32 Produkte entsprechend erweitert, um diese Schnittstellen zu unterstützen.
- SYSGO hat die folgenden verwertbaren Ergebnisse erzielt:
  - SYSGO hat eine auf das ACROSS-MPSoC zugeschnittene Version des PikeOS-Betriebssystems erstellt, die kommerziell genutzt werden kann. Dies umfasst alle Software-Ebenen von PikeOS vom Mikrokern über die System-Software bis hin zu speziellen Treibern für die ACROSS-Architektur.
  - SYSGO konnte auf dem Gebiet der Time-Triggered-Busse das Standardprodukt PikeOS weiterentwickeln und hat eine Erweiterung zur leichteren Auf- und Re-Synchronisation von externen Netzwerken ins Produkt einfließen lassen. Auf dieser Basis wird die Unterstützung ähnlicher Architekturen in Zukunft einfacher. Kundenanfragen zu solchen Erweiterungen, etwa für spezielle Time-Triggered-Bus-Systeme gab es bereits.
  - SYSGO hat zusammen mit OpenSynergy eine effiziente Anbindung von AUTOSAR-OS an das PikeOS-Betriebssystem erarbeitet, welche im Produkt weiterverwendet werden kann und auch auf andere Architekturen portiert werden wird. Dies ist ein wesentlicher Fortschritt für die Unterstützung von AUTOSAR, die auch ins Produkt eingeflossen ist. Von hieraus kann in künftigen Projekten weitergearbeitet werden.

Durch die Beteiligung an den Arbeitspaketen, in denen die Demonstrator-Anwendungen konzipiert und umgesetzt wurden, konnten die beteiligten deutschen Partner die bedarfsgerechte Ausgestaltung der ACROSS-Plattform sicherstellen und sich frühzeitig das Knowhow für die Verwendung der im Projekt erarbeiteten Technologie aneignen.

- EADS-DE und EADS-IW konnten anhand der Implementierung des Aerospace Demonstrators beweisen, dass das ACROSS MPSoCs für typische Anwendungen in der Luftfahrt sehr gut geeignet ist. Wesentliche Sicherheitsanforderungen für einen Einsatz in der Luftfahrt wie z.B. funktionale Unabhängigkeit, werden erfüllt. Insbesondere die inhärente Segregierung und die klare Struktur des ACROSS MPSoCs erleichtern den Nachweis einer korrekten Funktionalität von Anwendungen mit unterschiedlicher Kritikalität auf einer gemeinsamen Hardwareplattform. Eine unmittelbare Anwendung der MPSoC Architektur ist derzeit noch nicht gegeben, da eine Implementierung auf Basis von in FPGAs implementierten Prozessor-Kernen nicht die benötigte Verarbeitungsgeschwindigkeit aufweist. Des Weiteren muss angemerkt werden, dass FPGAs in der Luftfahrt, aufgrund der Empfindlichkeit gegenüber Effekten aufgrund von Strahlung (Single Event Upsets, SEU), nur mit zusätzlichen Mitigierungen eingesetzt werden können. Eine Implementierung als ASIC hätte zwar die geforderten Leistungsdaten und Robustheit gegenüber SEU. Aufgrund der geringen Stückzahlen in der Luftfahrt ist eine spezifische ASIC-Herstellung jedoch wirtschaftlich nicht sinnvoll. Mit Hilfe der ACROSS Architektur konnte insbesondere ein neuer Trend der Chiparchitekturen hin zu Nachrichten-basierter On-Chip-Kommunikation im Kontext der Luftfahrt evaluiert werden.

## 2.5 Fortschritt auf dem Gebiet des Vorhabens an anderen Stellen

EADS-IW und EADS-DE haben gemeinsam mit SYSGO und der TU Braunschweig im Project RECOMP an einer NoC Architektur mit Separierungseigenschaften und deren Anwendung im Luftfahrtbereich gearbeitet. Allerdings haben die Arbeiten ergeben, dass zumindest im Prototypenstadium die Architektur der TU Braunschweig einiges komplexer ist und damit eine Zertifizierung nach DO-254 schwieriger als die ACROSS Architektur erscheinen lässt.

EADS-IW, EADS-DE und SYSGO haben im Rahmen von RECOMP COTS Architekturen und mögliche temporale Einflüsse auf die Partitionierung generell ausgearbeitet. Des Weiteren wurde von EADS-IW und EADS-DE die Auswertung von anderen kommerziell eingesetzten NoC vorgenommen. Hier standen die Architekturen von Freescale im Vordergrund, die im Kommunikationssektor (Switches) eingesetzt werden. In der Luftfahrt stellt Freescale (ähnlich wie Texas Instrument) viele Informationen über den Entwicklungs- und Produktionsprozess zur Verfügung, was für die Zertifizierung und Zuverlässigkeitserwägungen essentiell ist.

In Deutschland hat das Projekt ARAMiS das Thema Multicore aufgegriffen und noch einmal holistischer untersucht. Weitere Ergebnisse zu dem Thema liegen vor (z.B. hat EADS-IW ein Monitoring-Verfahren mit EADS-DE entwickelt und mit SYSGO und AbsInt umgesetzt; auch Partitionierung von I/O Architekturen sind für EADS-IW und EADS-DE im Fokus von ARAMiS). Wenngleich in ARAMiS andere technische Lösungen als in ACROSS zum Einsatz kommen, so kann ARAMiS als sinnvolle Ergänzung zu ACROSS betrachtet werden, da dort Themen untersucht werden, die von ACROSS nicht detailliert behandelt wurden. ARAMiS fokussiert sich auf die Verwendung von COTS Architekturen im Sicherheitsbereich (Safety) und behandelt auch Aspekte von Security. Hardware-nahe Aktivitäten gibt es auf der Board Ebene und im Bereich I/O Interfaces (Virtualisierung). ACROSS spezialisiert sich auf neue Architekturen für Multicore-Systeme für den Safety-Bereich.

Eine weitere Arbeit zum Thema mehr Dynamik und Systemaspekte ist das ARTEMIS Projekt EMC2, welches sich aktuell in der Verhandlungsphase befindet. Der Start von EMC2 ist für den 1. April 2014 geplant. In EMC2 versuchen EADS-DE und EADS-IW Zertifizierbarkeit und hohe Performanz zu vereinen. Studien zum Thema ACROSS und weitere Verwendung sind geplant. Allerdings benötigt man hier auch das Interesse von anderen Industrien wie Automotive und Industrial Control, um finanziell konkurrenzfähig zu sein. Ein spezieller ACROSS-Chip ist finanziell nicht akzeptabel, wenn als Kunde nur der Luftfahrtbereich interessiert ist. Die Fixkosten eines ACROSS-Chips müssen über eine große Stückzahl amortisiert werden.

Darüber hinaus sind dem Konsortium keine anderen Aktivitäten auf dem Gebiet der Nutzung einer zeitgesteuerten Network-On-Chip Architektur zur Schaffung einer domänen-übergreifenden Referenzplattform für sicherheitskritische Echtzeitsysteme bekannt. Dieses Alleinstellungsmerkmal der ACROSS-Plattform bedingt demzufolge die Einzigartigkeit der im vorliegenden Verbundprojekt durchgeführten Arbeiten. Dies umfasst insbesondere die auf die ACROSS-Plattform zugeschnittene werkzeuggestützte Entwicklungsmethodik, die Anpassung des PikeOS-Betriebssystems an die ACROSS-Architektur sowie die Umsetzung der Demonstrator-Applikationen.

## 2.6 Veröffentlichungen des Ergebnisses

Die folgenden wissenschaftlichen Publikationen wurden unter Beteiligung der deutschen Verbundpartner erstellt:

- [1] J. O. Blech, A. Hattendorf, J. Huang. Towards a Property Preserving Transformation from IEC 61131-3 to BIP. <http://arxiv.org/abs/1009.0817>, 2010.
- [2] "A Workflow for Runtime Adaptive Task Allocation on Heterogeneous MPSoCs" by J. Huang, A. Raabe, C. Buckl, A. Knoll at the Design, Automation & Test in Europe (DATE 2011), März 2011, Grenoble, France
- [3] "An Invariant Preserving Transformation for PLC Models" by J. O. Blech, A. Hattendorf, J. Huang. IEEE International Workshop on Model-Based Engineering for Real-Time Embedded Systems Design, 2011
- [4] J. O. Blech. A Tool for the Certification of PLCs based on a Coq Semantics for Sequential Function Charts. <http://arxiv.org/abs/1009.0817v1>, 2011
- [5] "Probabilistic Compositional Reasoning for Guaranteeing Fault Tolerance Properties" by J. O. Blech at 15th International Conference On Principles Of Distributed Systems (ODOPIS), Toulouse, France, Dez. 2011.
- [6] "Verification of PLC Properties Based on Formal Semantics in Coq" by J. O. Blech and S. Ould Biha at 9th International Conference on Software Engineering and Formal Methods (SEFM), Montevideo, Uruguay, Nov. 2011.
- [7] "Analysis and Optimization of Fault-Tolerant Task Scheduling on Multiprocessor Embedded Systems" by J. Huang, J. O. Blech, A. Raabe, C. Buckl and A. Knoll at International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS). Embedded Systems Week. Taipei, Taiwan, Okt. 2011.
- [8] "Reliability-Aware Design Optimization for Multiprocessor Embedded Systems" by J. Huang, J. O. Blech, A. Raabe, C. Buckl and A. Knoll at 14th Euromicro International Conference on Digital System Design (DSD), Oulu, Finland, Aug. 2011.
- [9] "A Tool for the Certification of Sequential Function Chart based System Specifications" by J. O. Blech at 6th International Workshop on Systems Software Verification. Nijmegen, The Netherlands, Aug. 2011.
- [10] "Towards Fault-Tolerant Embedded Systems with Imperfect Fault Detection", by J. Huang, K. Huang, A. Raabe, C. Buckl, A. Knoll at 49th Design Automation Conference (DAC), San Francisco, Juni 2012.
- [11] "Static Scheduling of a Time-Triggered Network-on-Chip based on SMT Solving", by J. Huang, J. O. Blech, A. Raabe, C. Buckl, A. Knoll at Design, Automation and Test in Europe (DATE), Dresden, Germany, März 2012.
- [12] Towards Fault-Tolerant Embedded Systems with Imperfect Fault Detection" by J. Huang, K. Huang, A. Raabe, C. Buckl, A. Knoll at 49th Design Automation Conference (DAC), San Francisco, USA, Juni 2012

[13] System Integration of a Degraded Vision Landing Aid Application on a Custom High-Criticality Research Platform” by M. Paulitsch, D. Geiger, B. Koppenhöfer, P. Ganal at SAE 2012 Aerospace Electronics and Avionics Systems Conference, Phoenix, Arizona, USA, Oktober 2012

[14] Design and Implementation of a Degraded Vision Landing Aid Application on a Multicore Processor Architecture for Safety-Critical Application” by H. Karray, M. Paulitsch, B. Koppenhöfer, D. Geiger at The 9th Workshop on Software Technologies for Future Embedded and Ubiquitous Systems Conference (SEUS 2013), Paderborn, Germany, Juni 2013.

[15] „Sicher steuern mit Multicore-Prozessoren“ by S. Barner, J. Huang, S. Voss. In Computer & AUTOMATION, Juni 2013

[16] “A framework for reliability-aware design exploration for MPSoC based systems” by J. Huang, A. Raabe, K. Huang, C. Buckl, A. Knoll in Design Automation for Embedded Systems (DAEM), Dezember 2013

## 2.7 Fortschreibung des Verwertungsplanes

Dieser Abschnitt enthält eine Fortschreibung des Verwertungsplanes der am Verbundprojekt beteiligten Partner:

### 2.7.1.1 Verwertungsplan EADS-DE

EADS-DE wendet die Ergebnisse aus dem ACROSS Projekt bei der Architekturdefinition von Systemen in den Bereichen „Terrain and Obstacle Collision Avoidance“, „Sense and Avoid“ sowie „Pilot Assistance“ an. Insbesondere sollen die Erkenntnisse hinsichtlich robuster Segregierung umgesetzt werden. Aufgrund der fehlenden Verfügbarkeit eines kommerziellen Chip-Herstellers, der bereit ist, die ACROSS Architektur umzusetzen, ist eine direkte Anwendung der ACROSS Ergebnisse in zukünftigen Produkten gegenwärtig noch nicht möglich. Jedoch wurden spezifische Anforderungen an eine Multicore-Prozessorarchitektur (insbesondere ein deterministisches Network-on-Chip) mit Chip-Herstellern diskutiert. Dies hat zu einem besseren Verständnis der Chip-Hersteller in Bezug auf die Anforderungen aus dem Bereich Luftfahrt geführt.

### 2.7.1.2 Verwertungsplan EADS-IW

EADS-IW hat zu dem unter Federführung von EADS-DE entwickelten ACROSS Degraded Vision Landing Aid Demonstrator beigetragen. Für diesen Demonstrator wurden die sicherheitskritische Funktionsblöcke ARINC 429, Symbol-Applikation und Algorithm-Applikation entwickelt, die – sofern sie auf einer performanten Hardwareplattform, beispielsweise einem ASIC, implementiert werden – von der Firma Cassidian unmittelbar in Luftfahrtprojekten eingesetzt werden können.

Darüber hinaus können die beim Demonstrator gewonnenen Erfahrungen bezüglich einer leicht zertifizierbaren Segregierung von Funktionsblöcken bei der Implementierung künftiger dezentraler Rechenanwendungen im Bereich Luftfahrt eingesetzt werden. Solche dezentrale Rechenanwendungen haben stets auch generische Funktionen wie beispielsweise eine bidirektionale Umsetzung von Daten zwischen dem zentralen Avionik-Netzwerk und den verteilten Aktuatoren und Sensoren oder die Übernahme von einfachen Regelaufgaben für die verteilten Systeme. Ein einfacher Nachweis der sicheren Segregierung dieser generischen Funktionen ist unumgänglich für eine wirtschaftlich sinnvolle Implementierung solcher verteilter Systeme. Der Einsatz von ACROSS-MPSoC basierten Plattformen wird nicht für rechenintensive, dedizierte Systeme aus den Bereiche „Terrain and Obstacle Collision Avoidance“, „Sense and Avoid“ sowie „Pilot Assistance“ weiter untersucht werden, solange

keine Aussicht auf die Verfügbarkeit eines performanten ASIC besteht. Allerdings werden die Erfahrungen der modularen Programmiermöglichkeiten, der Segregation und des nachrichtenbasierten Austauschs beim Einsatz im Bereich IMA-Recheneinheiten auf anderen COTS-basierten Chips angewendet werden. EADS-IW und Cassidian wird die Pläne von TTTech hinsichtlich einer Weiterentwicklung der ACROSS Plattform mit Anforderungen an interessante performante Prozessorkerne im Projekt EMC2 unterstützen.

EADS-IW hat ausgewählte Ergebnisse mit EADS-DE und anderen Unternehmen der EADS Gruppe (Eurocopter, Airbus) geteilt und somit die Basis für eine weitere, potentielle Verwertung von ACROSS-Ergebnissen gelegt. Eine direkte Nutzung des derzeitigen Chips (FPGA) ist aber unwahrscheinlich wegen der bereits erwähnten geringen Performanz und Anfälligkeit zu Single Event Effekten (kosmische Strahlung).

### 2.7.1.3 Verwertungsplan fortiss

Der Hauptbeitrag von fortiss zu ACROSS ist die Erstellung einer Entwicklungsmethodik, die es erlaubt, Anwendungen auf Multicore System-On-Chip Plattformen zu realisieren. Hierzu hat fortiss ein modulares Entwicklungswerkzeug erstellt, das ein zentrales Element des ACROSS-Entwicklungsprozesses ist (vgl. Abschnitt 2.1.3). Die direkte Verwertung dieses Projektergebnisses als Programmier- und Konfigurationsumgebung für die ACROSS-Plattform hängt davon ab, ob sich das ACROSS-MPSoC auf dem Markt behaupten kann.

Wie in Abschnitt 2.4 erläutert, bietet das Werkzeug zwar explizit Unterstützung für die ACROSS-Plattform. Allerdings ist eine Vielzahl der Module generisch und kann auch unabhängig vom Markterfolg der ACROSS-Plattform genutzt werden. Hierzu zählen:

- Applikations-Modellierung (generisch und domänen-spezifisch)
- Entwurfsraumexploration
- Zuverlässigkeitsanalyse
- Framework zur Modellierung von Plattformen
- Plattformunabhängige Teile der
  - Modell-zu-Modell-Transformationen
  - Codegeneratoren
  - Konfigurationsgeneratoren (z.B. Anbindung an PikeOS-Konfigurationswerkzeug CODEO)

fortiss verfolgt diese sekundäre Nutzung seiner ACROSS-Ergebnisse durch die folgenden Aktivitäten:

- Aufbauend auf den Vorarbeiten des ACROSS-Projektes wird fortiss zur werkzeuggestützten Entwicklungsmethodik für das FP7 EU Projekt DREAMS<sup>3</sup> beitragen. Das DREAMS-Projekt zielt darauf ab, eine Architektur für verteilte Echtzeitsysteme zur Integration gemischt-kritischer Anwendungen zu entwickeln. Folglich werden dort Aspekte von ACROSS unter einer Erweiterung des Fokus (z.B. Betrachtung von Off-Chip Netzwerken) weiterverfolgt. Die in ACROSS erzielten Ergebnisse tragen somit wesentlich dazu bei, dass zwei Arbeitsplätze für wissenschaftliches Personal geschaffen werden konnten.

---

<sup>3</sup> <http://www.dreams-project.eu/>

- Plattformunabhängige Teile der ACROSS Werkzeugkette wurden unter der Apache Open Source Lizenz entwickelt und sind über das AutoFocus 3<sup>4</sup>-Repository verfügbar.
- Die im Rahmen von ACROSS gewonnen Erkenntnisse bezüglich Entwurf und Entwicklung einer Werkzeugkette für komplexe eingebettete Zielplattformen sind in die Vorlesung „Echtzeitsysteme“ des Lehrstuhls für Echtzeitsysteme und Robotik an der TU München eingeflossen, der mit fortiss assoziiert ist.

#### **2.7.1.4 Verwertungsplan Lauterbach**

Lauterbach hat im Rahmen des ACROSS Projekts eine Debug-Schnittstelle und eine Trace-Schnittstelle definiert. Lauterbach hat seine TRACE32 Produkte erweitert, um diese Schnittstellen zu unterstützen.

Die erfolgreiche wirtschaftliche Verwertung des Projekts hängt für Lauterbach davon ab, ob und in welchem Umfang sich die ARTEMIS Cross-Domain Architektur (ACROSS) in der Industrie durchsetzt. Die ACROSS Architektur zielt vor allem auf den Markt der sicherheitskritischen Systeme ab. Sollte die ACROSS Architektur in diesem Marktbereich Erfolg haben, würde dies Lauterbach die Möglichkeit eröffnen, sich in diesem Markt mit neuen Werkzeugen zu etablieren.

Aufgrund der traditionell eher konservativen Strukturen dieses Marktbereichs ist anzunehmen, dass die mögliche Einführung von ACROSS-basierten Systemen über einen längerfristigen Zeitraum stattfindet. Lauterbach rechnet deshalb nicht mit einer kurzfristigen Verwertung der Ergebnisse des ACROSS Projekts.

Aufgrund der Tatsache, dass sich momentan keinerlei Vorhersage darüber machen lässt ob und wann sich die ACROSS Architektur in der Industrie durchsetzt, ist es zum jetzigen Zeitpunkt nicht möglich einen detaillierten Verwertungsplan zu erstellen.

#### **2.7.1.5 Verwertungsplan SYSGO**

SYSGO konnte im ACROSS-Projekt die folgenden verwertbaren Artefakte erarbeiten.

Offensichtlich ist zunächst die Portierung des PikeOS-Betriebssystems auf die NIOS-2-Architektur. Diese Architektur wird vollständig in Produktqualität unterstützt inklusive aller nötigen Software-Schichten und Personalities (ab Version 3.2, ServiceRelease 1). Dies ist auch der breiten Domänenauslegung des ACROSS-Projektes geschuldet. SYSGO erhofft sich nun, diese Portierung kommerziell nutzen zu können. Bisher gab es leider keine Anfragen seitens des Marktes, was auf eine geringe Verbreitung der NIOS-2-Architektur schließen lässt.

Desweiteren gibt es durch die Unterstützung der TTNoC Time-Triggered-Netzwerkarchitektur im Produkt PikeOS nun Erweiterungen zur vereinfachten Synchronisation mit derartigen Netzwerken. SYSGO hat dies in einem White-Paper dokumentiert und knüpft bereits in Kunden- und Forschungsprojekten an diese neuen Möglichkeiten des PikeOS-Kerns an. Die Erweiterungen sind voll ins Produkt integriert und in PikeOS 3.4 verfügbar. Sie wurden auch kürzlich im Rahmen eines Zertifizierungsprojektes für einen Kunden mit dem Betriebssystemkern mitzertifiziert nach EN50128.

Weiterhin erwähnenswert sind Erweiterungen an der PikeOS System Software (PSSW) zur effizienten Nutzung von Treibern aus anderen Treibern heraus. Aufgrund der starken Modularisierung entstand im ACROSS-Projekt die Notwendigkeit, relativ viele Treiber, die einander benutzen, ins PikeOS-Sys-

---

<sup>4</sup><http://af3.fortiss.org/>

tem einzubinden. Aus Effizienzgründen sollte dies als System-Extension entstehen. Die nötigen Erweiterungen wurden im Rahmen von ACROSS umgesetzt und in den verschiedenen WPs von Projektpartnern verwendet (etwa vom CAN-Service in WP4). Die Erweiterungen an PikeOS sind ebenfalls in die Entwicklung von PikeOS-3.4 eingeflossen und wurden in einem Kundenprojekt nach EN50128 mitzertifiziert.

Bei den domänen-spezifischen Verwertungen ist vor allem die verbesserte AUTOSAR-Anbindung zu unterstreichen. Im Rahmen von ACROSS konnte in Zusammenarbeit mit OpenSynergy eine wesentlich effizientere Umsetzung der AUTOSAR-Personality entwickelt werden. Eine in einer vorherigen Version benutzte zusätzliche Software-Zwischenschicht, die auf der POSIX-Personality von PikeOS aufbaute, konnte entfernt werden. Die neue Version knüpft nun direkt an die PikeOS Native Personality an und bildet z.B. AUTOSAR-Threads direkt auf PikeOS-Threads ab und nutzt auch den nativen PikeOS-Kernel-Scheduler. Dadurch wurde der Speicherbedarf der Software-Schicht verringert und die Effizienz erhöht, was im ACROSS-Projekt wichtig war wegen der starken Modularisierung der Architektur. Diese AUTOSAR-Anbindung ins ebenfalls in die Weiterentwicklung des Produkts eingeflossen und ist nun kommerziell verfügbar.

Im Rahmen der Unterstützung des Avionik-Demonstrators wurde eine Erweiterung aus dem ARINC-653-Standard, Part 2 implementiert: SAPs „Source-Addressable Ports“. SAPs verhalten sich ähnlich wie Queuing-Ports, erlauben aber durch die zusätzliche Verwendung von Adressen die Anbindung an das UDP-Protokoll zur Netzwerkkommunikation. SAPs sowie die Anbindung an UDP wurden in PikeOS umgesetzt und sind vollständig verfügbar. SYSGO plant, diese auf Kundenwunsch endgültig ins Produkt aufzunehmen.

Auch wenn sich in naher Zukunft die ACROSS-Plattform selbst nicht kommerziell durchsetzen sollte, kann SYSGO die Ergebnisse aus ACROSS sehr gut im Produkt verwerten, da viele Verbesserungen am Betriebssystem und seinen umgebenden Komponenten für ähnliche Ansätze nutzbar sind und nicht auf die konkrete Plattform beschränkt sind.

## 3 Partnerbeiträge

### 3.1 EADS-DE

Die Beiträge von EADS-DE zu den technischen Arbeitspaketen erstrecken sich auf WP1, WP2 und WP5.

- Der Fokus der Arbeiten lag auf WP5, in dem EADS-DE federführender Partner bei der Entwicklung des Aerospace Application Demonstrator B war. Die MPSoC HW und Middleware wurde im Rahmen des ACROSS Projekt beigestellt. Die übrigen HW Elemente wurde von EADS-DE beschafft. Die Konsolidierung bereits verfügbarer Anforderungen erfolgte in Zusammenarbeit mit EADS-IW und SYSGO. Die in Abschnitt 2.1.5 beschriebenen Beiträge wurden von EADS-DE erbracht mit folgenden Ausnahmen: Die Entwicklung der domänen-spezifischen Middleware-Dienste erfolgte durch SYSGO. Die Evaluation der Eignung der ACROSS MPSoC Architektur für sicherheits-kritischen Anwendungen in der Domäne Luftfahrt erfolgt in Zusammenarbeit mit EADS-IW.
- WP1, 2 und 5: WP-übergreifend wurden die Projektpartner geschult für die Anforderungen an Firmware- und Software-Entwicklung nach den Luftfahrtstandards ARP4754A, ARP4761 sowie DO-178 und DO-254. Die von den Projektpartnern entwickelten Firmware- und Middleware-Elemente wurden geprüft hinsichtlich ihrer Konformität mit den o.a. Luftfahrtstandards. Die Tätigkeiten erfolgten in Zusammenarbeit mit EADS-IW.

### 3.2 EADS-IW

Die Beiträge von EADS-IW zu den technischen Arbeitspaketen erstrecken sich auf WP1, WP2 und WP5.

- Der Fokus der Arbeiten lag auf WP5.
  - In Zusammenarbeit mit EADS-DE und SYSGO wurde die Konsolidierung bereits verfügbarer Anforderungen durchgeführt.
  - In Zusammenarbeit mit EADS-DE wurde die Evaluation der Eignung der ACROSS MPSoC Architektur für sicherheits-kritische Anwendungen in der Domäne Luftfahrt durchgeführt und in Teilen umgesetzt.
- WP1, 2 und 5: WP-übergreifend wurden in Zusammenarbeit mit EADS-DE die Projektpartner geschult für die Anforderungen an Firmware- und Software- Entwicklung nach den Luftfahrtstandards ARP4754A, ARP4761 sowie DO-178 und DO-254. In WP1 wurden darüber hinaus die Anforderungen und Erfahrungen von sicherheitskritischen embedded Netzwerken (wie TTEthernet) für Network-on-Chip eingebracht. Die von den Projektpartnern entwickelten Firmware- und Middleware-Elemente wurden geprüft hinsichtlich ihrer Konformität mit den o.a. Luftfahrtstandards. Kommerziell verfügbare Plattformen wurden auf Performanz im Worst-Case analysiert.

### 3.3 fortiss

Die Beiträge von fortiss zu den technischen Arbeitspaketen erstrecken sich auf WP2, WP3 und WP6.

- Der Fokus der Arbeiten lag auf WP3, in dem fortiss der federführende Partner war. Insofern war fortiss maßgeblich für die Konzeption und Umsetzung der Entwicklungsmethodik sowie der zugehörigen Werkzeugkette verantwortlich. Mit Ausnahme der Beiträge von Lauterbach

zur Integration von Debugging-Verfahren wurden alle in Abschnitt genannten 2.1.3 Beiträge des deutschen Konsortiums von fortiss erbracht.

- WP2: fortiss steuerte den Voting-Dienst (OPT5) bei. Darüber hinaus trug fortiss zur Integration weiterer von WP2 erstellter Dienste in die Werkzeugkette aus WP3 bei (z.B. OPT8, Betriebssystem).
- WP6: Mit Ausnahme der von SYSGO zur Verfügung gestellten POSIX-Personality für PikeOS wurden alle in Abschnitt 2.1.6 genannten Beiträge des deutschen Konsortiums von fortiss erbracht.

### 3.4 Lauterbach

Lauterbach arbeitete hauptsächlich an WP2 und WP3 mit.

Für WP2 wurden VHDL Komponenten für folgende Diagnoseservices entwickelt:

- OPT10: Real-Time Tracing Service
- OPT11: Debugging Service
- OPT12: Health Monitoring Service, LDU
- OPT13: Health Monitoring Service, Receiver

Insbesondere wurden zwei Schnittstellen (eine Debug-Schnittstelle und eine Trace-Schnittstelle) zur Kommunikation mit einem externen Analysewerkzeug spezifiziert und implementiert.

Die Details der Implementierung sind in Deliverable D2.4 („Implementation of middleware and system components“) festgehalten.

Als Teil des WP3 wurden Lauterbachs TRACE32 Produkte so erweitert, dass man über die spezifizierten Schnittstellen mit den Diagnoseservices kommunizieren kann.

### 3.5 SYSGO

SYSGO hat in WP2, WP4, WP5 und WP6 mitgearbeitet.

Für WP2 wurden umfangreiche Arbeiten zur Unterstützung von OPT8 „Operating System Services“ durchgeführt. Betriebssystemkern, System-Software und Native-Personality wurden portiert, sowie Treiber zum Zugriff auf den TTNoC in PikeOS erstellt. Weiterhin wurde eine Unterstützung von „gestapelten“ Treibern, d.h. Treibern, die andere Treiber aus einer System-Extension heraus benutzen, implementiert.

In WP4 wurde die Unterstützung für ein effizientes AUTOSAR-OS, eine AUTOSAR-Personality von PikeOS, erarbeitet und die Projektpartner bei der Integration unterstützt.

In WP5 wurde ARINC653 auf den NIOS-2 portiert. Weiterhin wurden Erweiterungen zum Zugriff auf das UDP-Netzwerkprotokoll implementiert, die als Standard-ARINC-653-Part2-Modul ausgeführt wurden.

In WP6 wurde POSIX auf den NIOS-2 portiert und die Projektpartner bei der Integration unterstützt.

## Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel Schlussbericht ARTEMIS Cross-Domain Architecture (ACROSS)	
4. Autor(en) [Name(n), Vorname(n)] Koppenhöfer, Bernd Geiger, Dietmar Paulitsch, Michael Barner, Simon Rohloff, Ingo Theiling, Henrik	5. Abschlussdatum des Vorhabens Juli 2013
	6. Veröffentlichungsdatum 18.12.2013
	7. Form der Publikation Broschüre
8. Durchführende Institution(en) (Name, Adresse) EADS Deutschland GmbH – CASSIDIAN (ehemals EADS Defence Electronics) Wörthstr. 85, 89075 Ulm  EADS Deutschland GmbH - Innovation Works 81663 München  fortiss GmbH Guerickestraße 25, 80805 München  Lauterbach GmbH Altlaufstr. 38, 85635 Höhenkirchen-Siegertsbrunn  SYSGO AG Am Pfaffenstein 14, 55270 Klein-Winternheim	9. Ber. Nr. Durchführende Institution
	10. Förderkennzeichen 01IS10003A, C-E, G
	11. Seitenzahl 59
12. Fördernde Institution (Name, Adresse)  Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. Literaturangaben 56
	14. Tabellen 4
	15. Abbildungen 20
16. Zusätzliche Angaben Siehe auch „Schlussbericht zu Nr. 3.2 BNBEST-BMBF 98“, FZK 01IS10003B	
17. Vorgelegt bei (Titel, Ort, Datum)	

## 18. Kurzfassung

### Derzeitiger Stand von Wissenschaft und Technik

Die derzeit auf dem Markt befindlichen Multicore-Prozessoren sind auf die Leistung in nicht-sicherheitskritischen Anwendungen optimiert und weisen somit Eigenschaften auf, die deren Einsatz in sicherheitskritischen Anwendungen erschweren. Dabei behindern die genutzten Techniken insbesondere die Vorhersagbarkeit des Laufzeitverhaltens der auf dem Prozessor ausgeführten Anwendungen, und damit auch die für die sichere und zertifizierbare Umsetzung kritischer Algorithmen (z.B. Regelungen) benötigte Garantie von maximalen Ausführungszeiten (Einhaltung von „Deadlines“). Aus den genannten Gründen werden Multicore-Systeme, von wenigen Ausnahmen abgesehen, derzeit nicht in sicherheitskritischen Echtzeitsystemen eingesetzt. Das ACROSS-Projekt baut auf einer Reihe von Vorarbeiten in den Bereichen Prozessorarchitektur (Projekte GENESYS, INDEXYS), virtualisierende Betriebssysteme (Projekt INTERESTED) und Entwicklungswerkzeuge (Projekte GENESYS, EasyKit) auf.

### Begründung/Zielsetzung der Untersuchung

Das ACROSS-Projekt zielt darauf ab, die im Rahmen des GENESYS EU-Projektes definierte zeitgesteuerte Referenz-Architektur sowie das zugehörige Ökosystem zu verfeinern und zu implementieren. Hierzu zählen neben einer FPGA-Implementierung der Architektur die entsprechenden Softwarekomponenten (Middleware, Betriebssystem) sowie die zugehörigen Entwicklungswerkzeuge. Durch den gewählten Ansatz wird die Wiederverwendung von Hardware und Software für die Umsetzung sicherheitskritischer Funktionen über Domänengrenzen hinweg ermöglicht. Diese Modularität ermöglicht eine Steigerung der Stückzahlen der verwendeten generischen Komponenten und erhöht damit deren Robustheit bei gleichzeitiger Senkung der Kosten. Die resultierende Skalierbarkeit des Ansatzes trägt zusammen mit dem werkzeuggestützten Entwicklungsansatz zur Beherrschung der Systemkomplexität und der Entwicklungszeit bei.

### Methode

Im Projekt wird in Zusammenarbeit mit den anderen europäischen Partnern des ARTEMIS-Projektes ACROSS eine FPGA-Implementierung des MPSoCs sowie des zugehörigen Ökosystems (Middleware, Betriebssystem sowie zugehörige Entwicklungswerkzeuge) erstellt. Dabei stellt das zeitgesteuerte On-Chip Netzwerk, über das die Kerne des MPSoCs kommunizieren, den Schlüssel zur Garantie der für den Einsatz in sicherheitskritischen Anwendungen benötigten Vorhersagbarkeit dar. Die ACROSS-Plattform, bestehend aus MPSoC, Betriebssystem und Middleware-Diensten wird im Projekt anhand von Demonstratoren aus den Domänen Automotive, Luftfahrt und Industrieautomatisierung untersucht.

### Ergebnis

Das deutsche Konsortium, dessen Arbeiten dieser Schlussbericht zusammenfasst, hat die folgenden Ergebnisse erzielt:

- Werkzeugkette zur modellgetriebenen Erstellung von ACROSS-basierten Anwendungen. Die Ergebnisse umfassen Meta-Modelle, Modell-zu-Modell-Transformationen, Generatoren, ein Verfahren zur automatischen Entwurfsraumexploration und Zuverlässigkeitsanalyse sowie Debugging- und Diagnosewerkzeuge.
- Eine auf das ACROSS MPSoC zugeschnittene Version des virtualisierenden Echtzeitbetriebssystems PikeOS, das Mechanismen zur zeitlichen und räumlichen Partitionierung der verwendeten Prozessorkerne sowie eine Schnittstelle zum zeitgesteuerten On-Chip-Netzwerk zur Verfügung stellt.
- Aufbau und Auswertung von Demonstratoren aus den Bereichen Luftfahrt und Industrieautomation. Diese Versuche haben ergeben, dass die ACROSS Plattform sehr gut für typische Anwendungen in diesen Bereichen geeignet ist.

### Schlussfolgerung/Anwendungsmöglichkeiten

Einer unmittelbaren Anwendung der Ergebnisse stehen derzeit noch die zu langsame Verarbeitungsgeschwindigkeit der verwendeten Prozessorkerne, die Empfindlichkeit der FPGA-Implementierung gegenüber Strahlung (insbesondere relevant für den Bereich Luftfahrt) sowie die Unwirtschaftlichkeit einer ASIC-Implementierung zum jetzigen Zeitpunkt entgegen. Dennoch hat die Evaluierung der Demonstratoren ergeben, dass der in ACROSS untersuchte Ansatz für eine für sicherheitskritische Anwendungen geeignete MPSoC-Architektur grundsätzlich valide und in den untersuchten Anwendungsdomänen anwendbar ist. Aus dem Projekt ist auch eine Reihe von Ergebnissen hervorgegangen, die auch ohne die ACROSS Plattform indirekt verwertet werden können. Diese umfassen Erweiterungen des PikeOS-Betriebssystems sowie die plattformunabhängigen Teile der ACROSS Werkzeugkette.

## 19. Schlagwörter

MPSoC; zeitgesteuerte Architektur; Luftfahrt; Industrieautomatisierung; Echtzeit-Betriebssystem; zeitliche und räumliche Separierung; modellgetriebene Entwicklung;

## 20. Verlag

## 21. Preis

## Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Report
3. title Schlussbericht ARTEMIS Cross-Domain Architecture (ACROSS)	
4. author(s) (family name, first name(s)) Koppenhöfer, Bernd Geiger, Dietmar Paulitsch, Michael Barner, Simon Rohloff, Ingo Theiling, Henrik	5. end of project July 2013
	6. publication date Dec. 18, 2013
	7. form of publication Brochure
8. performing organization(s) (name, address)	9. originator's report no.
	10. reference no. 01IS10003A, C-E, G
	11. no. of pages 59
12. sponsoring agency (name, address)  Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. no. of references 56
	14. no. of tables 4
	15. no. of figures 20
16. supplementary notes See also „Schlussbericht zu Nr. 3.2 BNBEST-BMBF 98“, FZK 01IS10003B	
17. presented at (title, place, date)	

18. abstract

State of the art

Multi-core processors currently available on the market are optimized for performance in non safety-critical applications and thus have properties that make them difficult to use in safety-critical applications. The techniques used particularly complicate the predictability of the runtime behaviour of applications running on the processor, and thus also the safe and certifiable implementation of critical algorithms (such as control algorithms) that require maximum execution time guarantees (in order to meet deadlines). Therefore, apart from a few exceptions, multi-core systems are currently not used in safety-critical real-time systems. The ACROSS project builds on a series of previous work in the field of processor architecture (projects GENESYS, INDEXYS), virtualizing operating systems (INTERESTED project) and development tools (projects GENESYS, EasyKit).

Motivation and Goals

The ACROSS project aims at refining and implementing the time-triggered reference architecture defined in the frame of the EU project GENESYS, as well as the associated ecosystem. In addition to a FPGA implementation of the architecture, this includes both the corresponding software components (middleware, operating system) and the associated development tools. The selected approach enables the cross-domain re-use of hardware and software for the implementation of safety-critical functions. This modularity allows to increase the lot size of generic components, thus rising their robustness, while reducing their cost. At the same time, the resulting scalability in conjunction with the tool-based development approach contributes to mastering the system complexity and reducing the development time.

Method

In the project, a FPGA implementation of the MPSoC and the associated ecosystem (middleware, operating system and associated development tools) is created in cooperation with the other European partners of the ARTEMIS project ACROSS. Here, the time-triggered on-chip network that connects the cores of the MPSoC, is key to guaranteeing the predictability that is required for safety-critical applications. In the scope of the project, the ACROSS platform consisting of MPSoC, operating system and middleware services is evaluated using demonstrators from the automotive, aerospace and industrial automation domain.

Result

The German consortium, who's contributions are summarized in this report, achieved the following results:

- Tool-chain for the model-driven development of ACROSS-based applications. The results comprise meta-models, model-to-model transformations, generators, an automated design-space exploration and reliability analysis method as well as debugging and diagnosis tools
- A version of the virtualizing real-time operating system PikeOS tailored to the ACROSS MPSoC. It provides mechanisms for the spatial and temporal partitioning of processor cores, as well as an interface to the time-triggered on-chip network.
- Implementation and evaluation of demonstrators for the aerospace and industrial automation domain. These experiments have shown that the ACROSS platform fits well for typical applications in these domains.

Conclusions/Applications

The immediate application of the results currently conflicts with the slow processing speed of the selected processor cores, the sensitivity of the FPGA implementation to radiation (particularly relevant for the aerospace sector) as well as the economic inefficiency of an ASIC implementation at the current time. Nevertheless, the evaluation of the demonstrators showed that the approach for MPSoC architecture that is suitable for safety-critical applications investigated in ACROSS is fundamentally valid and applicable in the examined application domains. From the project, a series of results has emerged, which can be exploited indirectly without the ACROSS platform. These include extensions to the PikeOS operating system and the platform-independent parts of the ACROSS tool-chain.

19. keywords

MPSoC; time-triggered architecture; aerospace; industrial automation; real-time operating system; temporal and spatial separation; model-driven development

20. publisher

21. price