

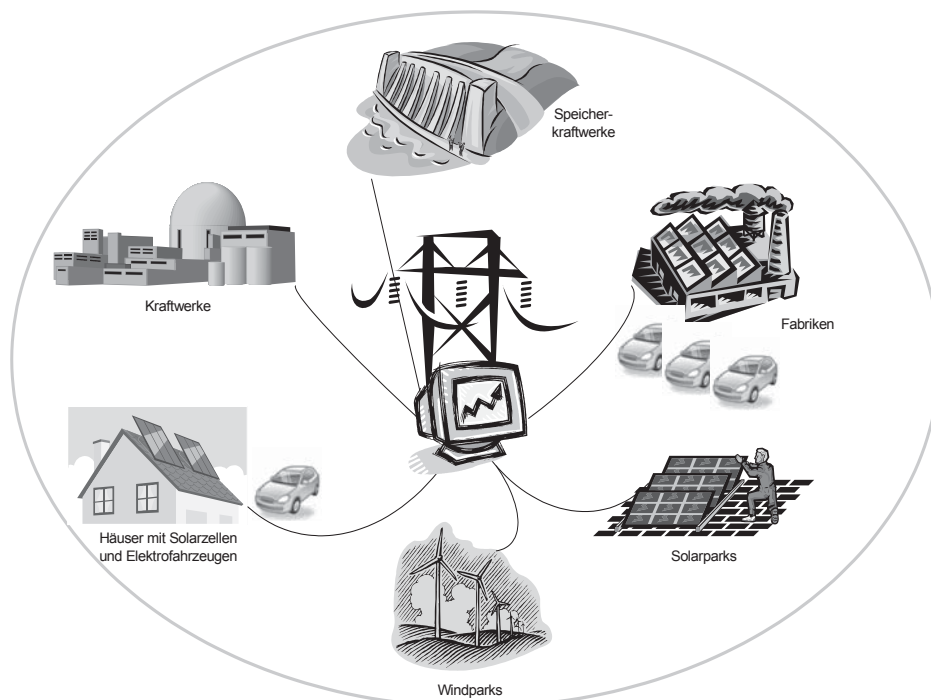
## Middleware-Architekturen zur Integrationen von Systemen in Systems-of-Systems

Dr. Christian Buckl, Michael Geisinger, ForTISS GmbH, München

**Eingebettete Systeme werden zunehmend in größere Systeme eingebunden, es entstehen Systems-of-Systems. Die Entwickler stehen nun vor der Herausforderung sowohl Anforderungen der Einzelsysteme / eingebetteten Systeme wie Echtzeit oder Ressourceneffizienz, als auch Anforderungen wie Flexibilität / Adaptivität, die bisher eher der Domäne Web zuzuordnen waren, zu erfüllen. Dieser Beitrag thematisiert dieses Spannungsfeld und diskutiert Lösungswege.**

Lag bisher der Schwerpunkt bei der Entwicklung von eingebetteten Systemen vor allem auf der Umsetzung von klar abgegrenzten Systemen mit definierten Schnittstellen zur Umwelt, ist in letzter Zeit der klare Trend hin zur Integration in größere Systeme zu beobachten: die Produktionsanlage wird zunehmend mit der Logistik und der Warenhaltung verknüpft, das intelligente Fahrzeug der Zukunft bewegt sich in einer intelligenten Umgebung und kommuniziert mit anderen Fahrzeugen und der Infrastruktur. Es entstehen sogenannte Cyber-Physical Systems / Systems-of-Systems. Systeme müssen also in Zukunft so entwickelt werden, dass sie einfach in einen größeren Kontext eingebunden werden können, ohne die Funktion des Systems in Bezug auf Safety, Security und Echtzeiteigenschaften zu gefährden. Middlewarearchitekturen wurden bisher für die Integration von Funktionen innerhalb eines Systems benutzt. Dieser Beitrag diskutiert, welche Randbedingungen in Zukunft beachtet werden müssen und wie Middlewarearchitekturen zur Lösung der Herausforderung eingesetzt werden können.

Abbildung 1: Smart Grid als ein Beispiel für Cyber-Physical Systems:



## Integration als Herausforderung

Die Integration verteilter Systeme entwickelt sich zu einer der größten Herausforderungen bei der Entwicklung verteilter Anwendungen. Entwickler stehen vor dem Problem, dass Systeme mit sehr unterschiedlichen Anforderungen zum Beispiel in Bezug auf Sicherheit (Stichwort *Mixed Criticality*) und Echtzeit integriert werden müssen. Neben den Eigenschaften von eingebetteten Systemen wie Umgang mit ressourcenarmer Hardware, Echtzeitfähigkeit, Zuverlässigkeit und Sicherheit, kommen neue Anforderungen hinzu, die bisher vor allem im Bereich von Webtechnologien zu finden waren. Dies ist insbesondere die Unterstützung von dynamischem Verhalten. Es müssen also auch Komponenten integriert werden, die zur Entwicklungszeit noch gar nicht bekannt sind. Durch die immer stärkere Vernetzung der Systeme müssen dabei auch Funktionen mit sehr unterschiedlichen Anforderungen und Eigenschaften direkt interagieren. Bei der Interaktion innerhalb der Domänen Eingebettete Systeme und Webtechnologie wurden bisher zur Integration Middlewaretechnologien sehr unterschiedlichen Zuschnittes eingesetzt, die zwar Interaktion erlaubten, nicht aber nahtlose Integration beider Welten. In Zukunft müssen nun Middlewaretechnologien entwickelt werden, die den Anforderungen beider Domänen entsprechen.

Dabei ist eine domänenübergreifende Standardisierung notwendig, um den domänenübergreifenden Charakter von Cyber-Physical Systems (CPS) widerzuspiegeln. Obwohl die wesentlichen benötigten Funktionalitäten bereits jetzt in verschiedensten Anwendungen umgesetzt sind, steht die Forschung und Entwicklung vor diversen Herausforderungen. Die Studie „Agenda Cyber-Physical Systems“ [1] identifiziert folgende Anforderungen an zukünftige Plattformen:

1. **Domänenübergreifende Standardisierung:** Sind heute viele Dienste bereits für verschiedene Domänen umgesetzt, müssen diese Dienste in Zukunft generisch, also anwendbar für verschiedenste Domänen entwickelt werden. Hierzu muss festgelegt werden, welcher Teil der Funktionalität nur für die entsprechende Domäne relevant ist und welcher Teil generisch realisiert werden kann.
2. **Modularisierung der Plattformen & Gewährleistung der Interoperabilität:** Heutige Plattformen sind zum großen Teil noch monolithisch aufgebaut. Da CPS sich vor allem durch Heterogenität in Bezug auf die Anforderungen, aber auch in Bezug auf die eingesetzte Hardware vom kleinsten Microcontroller bis zum Cloud Computing auszeichnet, ist eine modulare Plattform notwendig. Abhängig von der Anwendung können dann die notwendigen Module ausgewählt und zu einer Ausführungsplattform verbunden werden. Um dies realisieren zu können, ist die Definition von passenden Schnittstellen von wesentlicher Bedeutung.
3. **Methoden zur sicheren Integration von Modulen in Plattformen:** Der Trend zu einer modularisierten Plattform bedeutet aber auch eine enorme Vielfalt unterschiedlicher Instanzen dieser Plattform. Es müssen entsprechende Methoden entwickelt werden, um die Korrektheit der verschiedenen Varianten insbesondere in Hinblick auf nicht-funktionale Eigenschaften sicherstellen zu können. Dies stellt insbesondere dadurch eine Herausforderung dar, dass auch die Plattform selbst dynamisch angepasst werden muss, um auf wechselnde Anforderungen der Anwendungen zu reagieren. Zusätzlich sind Mechanismen notwendig, die sicherstellen, dass kein unerlaubter Zugriff möglich ist, um die Plattform zu kompromittieren und Daten auszuspähen.

## Modularität und Wiederverwendbarkeit durch Datenzentrierung

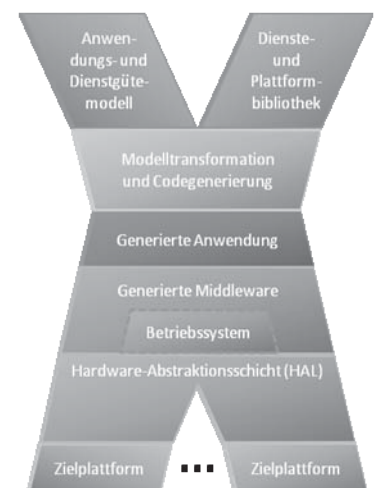
Im Folgenden geht dieser Beitrag insbesondere auf die zweite Anforderung ein. Um sich an die vielfältigen Einsatzbereiche anpassen zu können, muss eine geeignete Lösung hochgradig modular aufgebaut sein. So kann auch den Ressourcenanforderungen von eingebetteten Systemen entsprochen werden, während dieselbe Technologie ohne Kompromisse auf einem Industrie-PC zum Einsatz kommen kann. Durch geeignete Schnittstellen kann eine Wiederverwendung von Komponenten erreicht werden. Eine Möglichkeit, um diese Modularisierung technologisch umzusetzen, ist es Konzepte aus dem Bereich der Microkernel-Betriebssysteme zu verwenden. QNX [2] verwendet beispielsweise ein Message-Passing-System um die meisten Komponenten eines klassischen Betriebssystems als eigenständige Prozesse auszuführen. Dadurch kann der eigentliche Kernel stark minimiert und das Betriebssystem stark modularisiert werden.

Um Komponenten der benötigten Middleware auch nachträglich integrieren zu können, ist ein allgemeineres Konzept als Nachrichten notwendig. Hier hat sich das Konzept der Datenzentrierung, wie zum Beispiel im OMG-Standard DDS [3] angewendet, als sinnvoll herausgestellt. Komponenten können durch die Veröffentlichung und das Abonnement von Daten miteinander kommunizieren und so ein dynamisches Gesamtsystem bilden. Wesentlicher Vorteil ist, dass das gleiche Konzept sowohl für die Modularisierung der Middleware als auch der Anwendungsfunktionalität verwendet werden kann.

Erweitert man das Konzept der Datenzentrierung um Metadaten zur Beschreibung nicht-funktionaler Anforderungen (wie Zeit- oder Zuverlässigkeitsanforderungen) so können auch die Anforderungen aus dem Bereich eingebetteter Systeme von der Ausführungsplattform berücksichtigt werden.

## CHROMOSOME als Experimentalplattform

Am Forschungs- und Transferinstitut fortiss GmbH wird derzeit an einer prototypischen Umsetzung einer solchen Middlewareplattform inklusive entsprechender Entwicklungsmethodik gearbeitet, um verschiedene Lösungskonzepte aufzuzeigen und die prinzipielle Machbarkeit eines solchen Ansatzes aufzuzeigen. Nebenstehende Abbildung zeigt die grundsätzliche Architektur dieser Middleware CHROMOSOME [4]. CHROMOSOME wird als Open Source Middleware angeboten, um den Einsatz in möglichst vielen Anwendungen zu fördern. Eine erste Veröffentlichung ist für 2012 geplant.



Durch den modularen Aufbau von CHROMOSOME wird auch eine inkrementelle Entwicklung der Middleware selbst unterstützt. Die Entwicklung erfolgt dabei projektgetrieben auf Basis sehr unterschiedlicher Projekte, um eine möglichst breite Anwendbarkeit zu erreichen. Im Folgenden werden die unterschiedlichen Projekte kurz vorgestellt, um die unterschiedlichen Anforderungen darzustellen.

1. Belle II: Entwicklung eines Strom- und Spannungsüberwachungsmoduls auf Basis von CHROMOSOME zur Verwendung in dem derzeit im Aufbau befindlichen japanischen Teilchenbeschleuniger Belle II [5]. Das System muss dabei Module zur Fehlerüberwachung, Ansteuerung diverser I/O-Module und zur Kommunikation mit der übergeordneten Leitebene bereitstellen.
2. AutoPnP [6]: Entwicklung einer wandelbaren Produktionsanlage zur Reduktion von Rüstzeiten. Der Schwerpunkt in diesem System wird auf den Aspekten von Plug&Play bei gleichzeitiger Sicherstellung der zeitgerechten Steuerung der Produktionsanlage liegen.
3. OSCar: Entwicklung einer neuartigen IKT-Architektur für Elektrofahrzeuge auf Basis der Ergebnisse der Studie „Mehr Software [im] Wagen“ [7]. Geplant ist eine zentralisierte Rechenarchitektur mit smarten Sensoren / Aktoren in der Peripherie. Das System soll einerseits die zeitgerechte und sichere Ausführung sicherheitskritischer Funktionen erlauben, andererseits auch die Integration von neuen Funktionen, aber auch Hardware zu erlauben.

### **Zusammenfassung: Mehrwert durch Integration**

Stand bisher bei der Entwicklung von eingebetteten Systemen das individuelle System im Fokus, so wird sich in Zukunft die Wertschöpfung vor allem Richtung Integration von Systemen und Anbieten von systemübergreifenden Diensten verlagern. Um die steigende Komplexität, bedingt durch die Integration verschiedener Systeme, beherrschen zu können, müssen die Einzelsysteme standardisiert und in eine gemeinsame Plattform eingebracht werden. Middlewarearchitekturen sowohl im Bereich eingebetteter Systeme als auch im Bereich der Webtechnologien waren hier in der Vergangenheit bei der Integration auf Komponentenebene, also innerhalb eines Einzelsystems, erfolgreich. Für die Zukunft müssen die Konzepte aus diesen beiden Bereichen integriert und kompatibel gemacht werden. Erste Forschungsprojekte gehen diese Herausforderung bereits an.

[1] Integrierte Forschungsagenda Cyber-Physical System, <http://www.acatech.de/?id=1405>

[2] QNX Neutrino, <http://www.qnx.com/products/neutrino-rtos/index.html>

[3] DDS, [http://www.omg.org/technology/documents/dds\\_spec\\_catalog.htm](http://www.omg.org/technology/documents/dds_spec_catalog.htm)

[4] CHROMOSOME, <http://chromosome.fortiss.org/>

[5] Belle II Teilchenbeschleuniger, <http://belle2.kek.jp/>

[6] AutoPnP, <http://www.fortiss.org/de/forschung/projekte/autopnp.html>

[7] „Mehr Software (im) Wagen“, Studie im Auftrag des Bundesministeriums für Wirtschaft und Technologie, März 2011, <http://www.fortiss.org/ikt2030>

### **Autorenprofil:**

Dr. Christian Buckl leitet den Bereich Cyber-Physical Systems am gemeinnützigen Forschungs- und Transferinstitut fortiss. Seine Forschungsinteressen umfassen die modellgetriebene Entwicklung eingebetteter Systeme mit einem Schwerpunkt auf nicht-funktionalen Eigenschaften, sowie auf der Entwicklung entsprechender Laufzeitsysteme zur Ausführung dieser Systeme.



Michael Geisinger forscht in der Abteilung Cyber-Physical Systems des Forschungs- und Transferinstituts fortiss im Bereich modellgetriebene Entwicklung für eingebettete Systeme, drahtlose Sensornetzwerke, Automatisierungstechnik und Middleware-Architekturen.

