



CHROMOSOME Middleware

Produktinformationsblatt

Lag der Schwerpunkt bei der Entwicklung von eingebetteten Systemen bisher vor allem auf der Umsetzung von klar abgegrenzten Systemen mit definierten Schnittstellen zur Umwelt, ist in letzter Zeit ein klarer 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 kommuniziert drahtlos mit anderen Fahrzeugen und der Infrastruktur. Systeme müssen in Zukunft so entwickelt werden, dass sie einfach in einen größeren Kontext eingebunden werden können, ohne ihre Funktion in Bezug auf Safety, Security und Echtzeiteigenschaften zu gefährden. Dazu wird eine leistungsfähige und domänenübergreifende Middleware benötigt, die sich flexibel an das jeweilige Einsatzszenario anpasst. CHROMOSOME bietet diese Funktionalität.

Basismerkmale:

- Open Source-Lizenz
- Universelle Kommunikations-Middleware
- Modellgetriebene Entwicklungsmethodik
- Datenzentrisches Paradigma
- Modulare Dienste-Architektur
- Plattformunabhängigkeit vom Mikrocontroller bis zum Industrie-PC
- Automatische Erkennung von Teilnehmern
- Eignung für Klein- und Kleinstsysteme, volle Flexibilität für mächtigere Systeme
- Dienstgüte und Echtzeitfähigkeit
- Selbstüberwachung mit Fehlerkorrekturmechanismen
- Unterstützung industrieller Standards
- Einfache Erweiterung
- Große Dienste-Bibliothek

Anwendungsbeispiele:

- Integration heterogener Komponenten bei der Industrie- und Gebäudeautomatisierung
- Erfassung von Daten in drahtlosen Sensornetzwerken
- Kommunikation zwischen Automatisierungssystem und übergeordneter Steuerung

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 und Echtzeit integriert werden müssen (Stichwort *Mixed Criticality*). Hierfür wurden bisher domänenspezifische Middleware-Architekturen in den Bereichen der Webtechnologien und Eingebetteten Systeme verwendet. Um den vorherrschenden Trend zu unterstützen, werden aber Lösungen aus beiden Bereichen benötigt.

Modularität und Wiederverwendbarkeit

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 kommt. CHROMOSOME beinhaltet eine Bibliothek von unterstützten Plattformen und Diensten, die für eigene Anwendungen verwendet und beliebig erweitert werden kann.

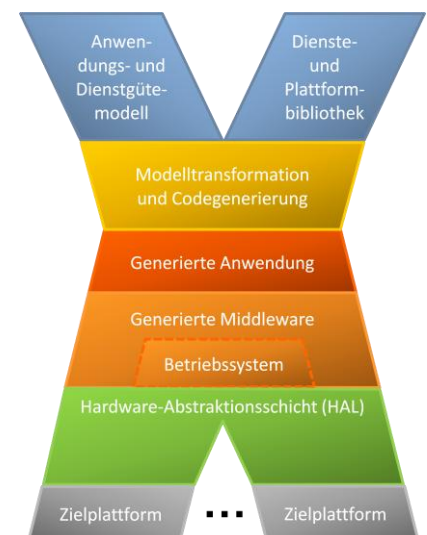


Abb. 1: CHROMOSOME-Architektur

CHROMOSOME als Lösung

CHROMOSOME schafft eine modulare, domänenübergreifende Entwicklungsmethodik, die über die heutigen Lösungen im Bereich Middleware und Betriebssysteme hinausgeht. Aktuelle Trends zur Integration heterogener Systeme, die deren Programmierung und Konfiguration auf ein höheres Abstraktionsniveau verlagern, werden unterstützt. Müssen sich beispielsweise die Entwickler bisher auf die unterschiedlichen Werkzeuge einstellen, ermöglicht CHROMOSOME es, die heterogenen Systeme in einem einheitlichen, domänenübergreifenden Modellierungsansatz abzubilden.



Architektur

- Hardware-Abstraktionsschicht (HAL) abstrahiert konkrete Ausführungsplattform (vgl. Abb. 1)
- Primitive Komponenten kapseln Hardware-Funktionalität (z.B. Ansteuerung der Peripherie)
- CHROMOSOME-Kerndienste stellen datenzentrische Kommunikation und Betriebssystem-Funktionalität bereit
- Höhere Komponenten implementieren Anwendungslogik

Plattformunterstützung und Entwicklungswerkzeuge

- x86: Windows, Linux (Visual Studio, MinGW, GCC)
- ARM: FreeRTOS (Eclipse, ARM-GCC, proprietäre Compiler)
- AVR: Contiki (AVRStudio, WinAVR)

Datenzentrische Kommunikation

- Basiert auf den Prinzipien von *Publish & Subscribe* und *Request/Response* (Client/Server)
- Zunächst Unterstützung von Ethernet (IP), WLAN (IP), serieller Kommunikation (nativ IP) sowie IEEE 802.15.4 (IP) (vgl. Abb. 3)
- Steuerung der Kommunikationsbeziehungen über Filterung von Datenströmen mittels Metadaten (vgl. Abb. 4)

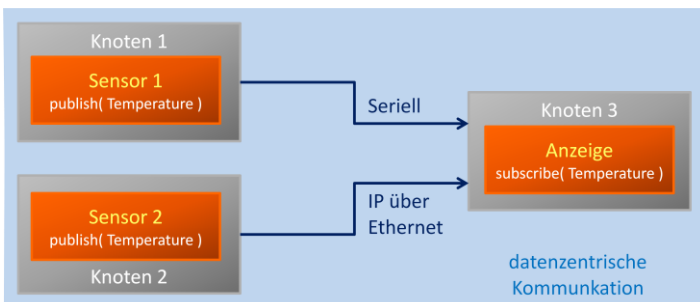


Abb. 3: Abstraktion des Kommunikationsmediums

Entwicklungsprozess

- Werkzeugunterstützte Spezifikation und Implementierung anwendungsspezifischer Softwarekomponenten
- Modellbasierte Entwicklung der verteilten Anwendung
- Auswahl der geeigneten Zielplattform(en)
- Generierung und automatisiertes Deployment der Anwendung
- Verteiltes Debugging
- Optional Entwicklung einer speziellen domänenspezifischen Sprache für die Anwendung

Inbetriebnahme

- Laufzeitkonfiguration und Optimierung des Systems
- Spezifikation von Dienstgütemerkmalen
- Integration von Fehlertoleranz-Mechanismen
- Optional statische Generierung des Systems

Selbstorganisation und Selbstkonfiguration

- Software-Komponenten spezifizieren ihre Ein- und Ausgangsdaten
- *Patterns* definieren die Struktur der Anwendung
- Kommunikationspfade werden dynamisch erstellt und bei Knoten- oder Link-Ausfall neu berechnet

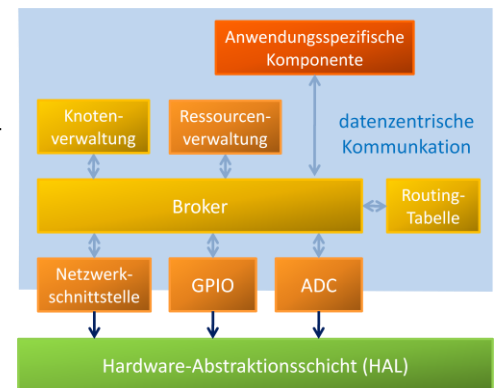


Abb. 2: Beispielhafte Konfiguration von Softwarekomponenten auf einem Sensorknoten

CHROMOSOME ist Open Source!

CHROMOSOME wird der Öffentlichkeit als Open Source unter der Apache-Lizenz Version 2.0 zur Verfügung gestellt, die die Nutzung kaum einschränkt. Eine erste Veröffentlichung ist für das 1. Quartal 2012 geplant. Danach werden schrittweise neue Funktionen folgen, wie die Unterstützung weiterer Plattformen sowie grafische Modellierungswerkzeuge.

Testen Sie CHROMOSOME!

Sie wollen mehr über CHROMOSOME erfahren oder das System schon vorab ausprobieren? Nehmen Sie mit uns Kontakt auf! Gerne beraten wir Sie, wie Sie CHROMOSOME in Ihrer Anwendung optimal einsetzen können! Nennen Sie uns Ihre Anforderungen!

Mehr unter chromosome@fortiss.org und <http://chromosome.fortiss.org/>.

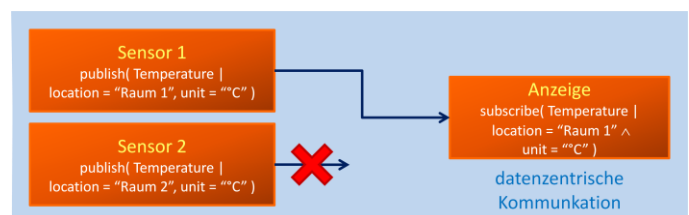


Abb. 4: Filterung von Datenströmen mittels Metadaten