CHROMOSOME: A Plug & Play-Capable Run-Time Environment for Embedded Real-Time Systems

Presented by Rousomanis Georgios (10703)

Buckl et al., 2014

fortiss GmbH & Technische Universität München

May 2025

Motivation: Why Adaptive Embedded Systems?

- Traditional embedded systems rely on static configuration – no updates post-deployment.
- New-generation systems must support:
  - Dynamic integration of new functionality (e.g., OTA updates).
  - Adaptation to topology changes (e.g., nodes joining or leaving a network).
- Existing run-time systems (e.g., RTOS):
  - *Pro*: guaranteed satisfaction of requirements (e.g., real-time behavior, safety)
  - *Con*: rely on a static configuration to satisfy requirements
- Web-service-style middleware (e.g., cloud, enterprise systems)
  - *Pro*: satisfy adaptability requirements – very flexible
  - *Con*: do not guarantee fulfillment of resource constraints
- Challenge: Combine plug & play capabilities with hard real-time guarantees.

CHROMOSOME (XME): What Is It?

- A modular run-time environment designed for embedded real-time systems.
- Built to support plug & play at both software and network levels.
- Cross-platform and open-source – usable across industrial and automotive domains.
- Resource-aware: Uses lookup tables to ensure that added components fit system limits.
- Components are dynamically loaded only if sufficient memory, CPU time, and bandwidth exist.
- Compatible with both time-triggered and event-driven execution models.

Core Concepts of XME

- **Requirements-centric design**: Application developers specify constraints like WCET, latency, safety level.
- **Data-centric communication**: No hard-coded connections; communication defined by topic types and attributes.
- **Plug & play orchestration**: New nodes/components can register and integrate safely at run-time.
- **Shadow configuration**: Ensures system consistency during reconfiguration — existing tasks are not interrupted.
- **Topic dictionaries**: Promote interoperability across independently developed modules.
- **Modularity**: Different runtime components (e.g., schedulers) can be selected per platform.

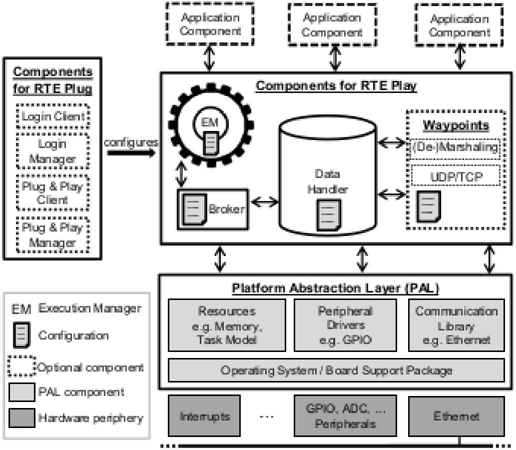
Communication via Topics and Dictionaries

- XME replaces traditional service calls with publish/subscribe over semantic "topics."
- Each topic includes:
  - Data type (e.g., sensor_reading, vehicle_position).
  - Qualifiers: min/max range, units, criticality level, precision.
- Topic matching ensures syntactic and semantic compatibility between components.
- Enables components developed independently to interoperate safely.

XME Architecture Overview

- The XME ecosystem is a federation of runtime nodes.
- Each node includes core services: Execution Manager, Broker, Data Handler, PAL.
- Nodes communicate via directed, type-safe publish-subscribe channels.
- The architecture separates application logic from low-level hardware using PAL.
- Reconfiguration is handled by plug-phase components: Login and Plug & Play Managers.
- Adaptivity features are only enabled if the node includes plug-phase services.

XME Architecture Overview



Play-Phase Components

- **Execution Manager (EM)**:
  - Invokes application components according to their priority and WCET.
  - Monitors for timing violations; reports if a task exceeds limits.
- **Broker**:
  - Watches data subscriptions and enables components only when inputs are ready.
- **Data Handler**:
  - Provides a consistent data interface to components.
  - Ensures safe, bounded-latency access to shared data.
- These ensure deterministic behavior during the "play" (runtime) phase.

Waypoints and Platform Abstraction Layer (PAL)

Waypoints:

- Lightweight modules for data transformation and verification.
- Can include marshaling (endianness), redundancy, CRCs, encryption, etc.
- Configured dynamically per data route based on system requirements.

Platform Abstraction Layer (PAL):

- Provides uniform access to hardware (GPIO, timers, interrupts).
- Makes applications portable across OSes like PikeOS, Linux, or even bare metal.

Plug-Phase: Dynamic Integration in XME

- **Login Phase:**
  - New node's *Login Client* contacts a central *Login Manager*.
  - Enables access to XME services.
- **Component Discovery:**
  - *Plug & Play Client* sends *manifests* to the *Plug & Play Manager*.
  - Describes topics, resource needs, and constraints.
- **Validation:**
  - *Logical Route Manager* computes routes.
  - Configurators check network and node resource availability.
- **Decision:**
  - If checks pass: update lookup tables and apply new config.
  - Else: reject change to maintain safety and real-time guarantees.

CHROMOSOME Modeling Tool (XMT)

Purpose: Support the design, validation, and configuration of distributed embedded systems.

- **Model-driven design tool**, built on Eclipse.
- Defines an iterative workflow:
  - Specification → Implementation → Evolution
- Allows developers to:
  - Declare requirements (timing, safety, communication patterns).
  - Model the XME ecosystem structure and topic dictionaries.
  - Validate the system early to catch design issues.
  - Auto-generate RTE configurations (adaptive or static).
- **Static mode:** Generates an offline configuration with zero run-time overhead.

Use Case 1: RACE (Automotive)

- Objective: enable adaptive automotive architecture with hard real-time support.
- Deployed using PikeOS partitions – each app runs in a secure container.
- XME runs in a master partition; manages partition communication and coordination.
- Real-time scheduling configured based on WCET data and component criticality.
- Allows runtime replacement of software modules without rebooting the ECU.

Use Case 2: AutoPnP (Industrial Automation)

- Targeted for factories needing reconfiguration without halting production.
- Hardware modules are detected at runtime; software activated accordingly.
- Event-triggered model handles inputs like sensor events or operator changes.
- Uses ROS for interoperability; XME adds plug & play and timing control.
- Demonstrates flexibility in industrial robotics and assembly systems.

Comparison with Other Frameworks

System	Real-Time Support	Adaptivity
ROS	No	High
DDS	Partial (QoS-based)	Medium
FRESCOR	Contract-based	High
CHROMOSOME	**WCET + Static Sched.**	**High + Modular**

XME uniquely blends static real-time safety with dynamic reconfigurability.

Future Directions

- Full end-to-end timing analysis across multi-node ecosystems.
- Binary deployment of application components, not just nodes.
- Security-aware middleware: role- and topic-based access control.
- Health monitoring and self-healing orchestration.

Conclusion

- XME enables real-time embedded systems to adapt safely and modularly.
- Combines model-driven design, static analysis, and runtime orchestration.
- Plug & play becomes practical for cyber-physical systems (CPS).
- Publicly available – suitable for research and industrial applications.

Thank You!

Questions?