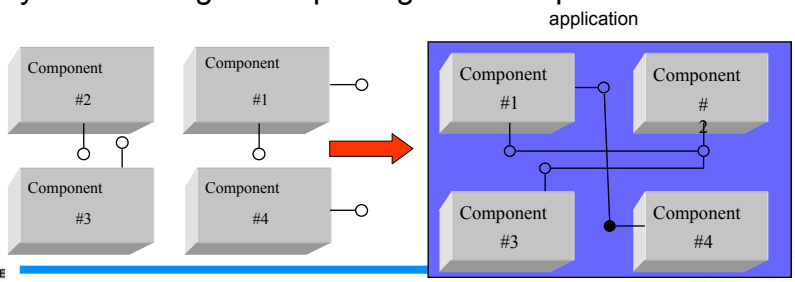
Component-based approach for embedded systems

Ivica Crnkovic
Mälardalen University
Department of Computer Science and Engineering,
Mälardalen Real-time Research Centre
Sweden
http://www.idt.mdh.se/~icc

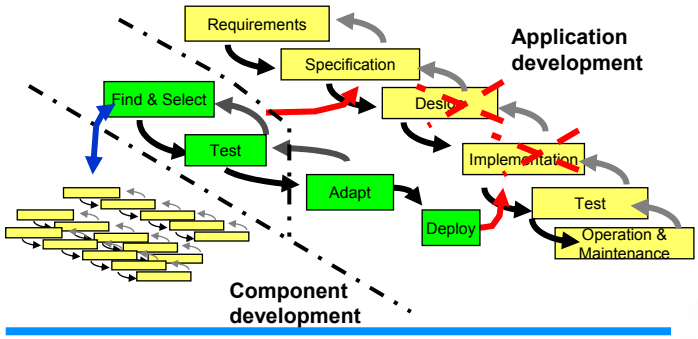
Component-based approach

- Building systems from (existing) components
  - Providing support for the development of systems as assemblies of components
  - Supporting the development of components as reusable units
  - Facilitating the maintenance and evolution of systems by customizing and replacing their components



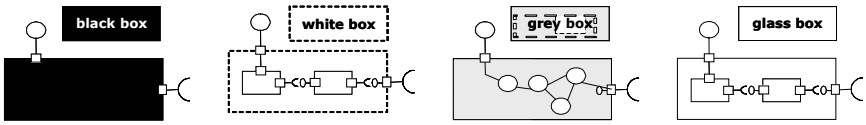
Implications

- Component development is separated from system development process
  - Less programming efforts to build systems
  - System verification and validation more difficult and more important
  - Different requirements management



Describing a Component

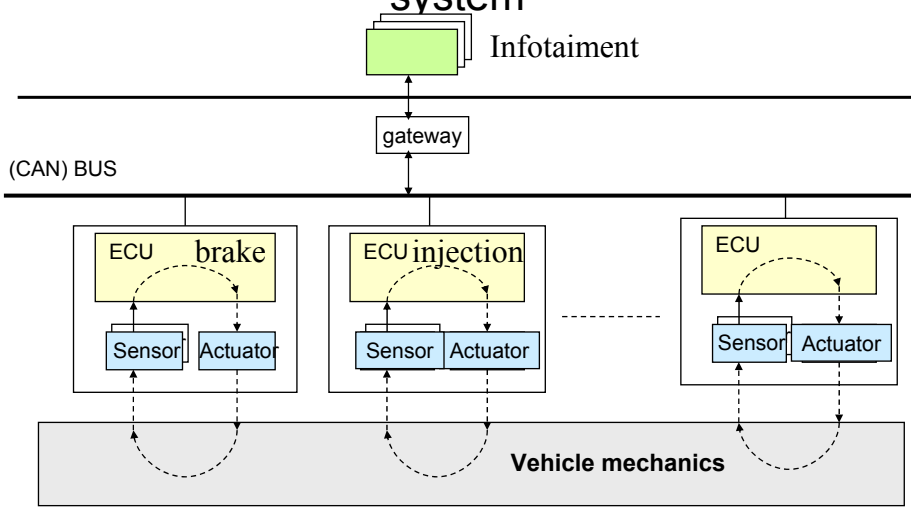
- To be able to describe a component completely the component should consist of the following elements:
  - A set of interfaces provided to, or required from the environment.
  - An executable code, which can be coupled to the code of other components via interfaces.
- How to specify a component?



CBSE process issues

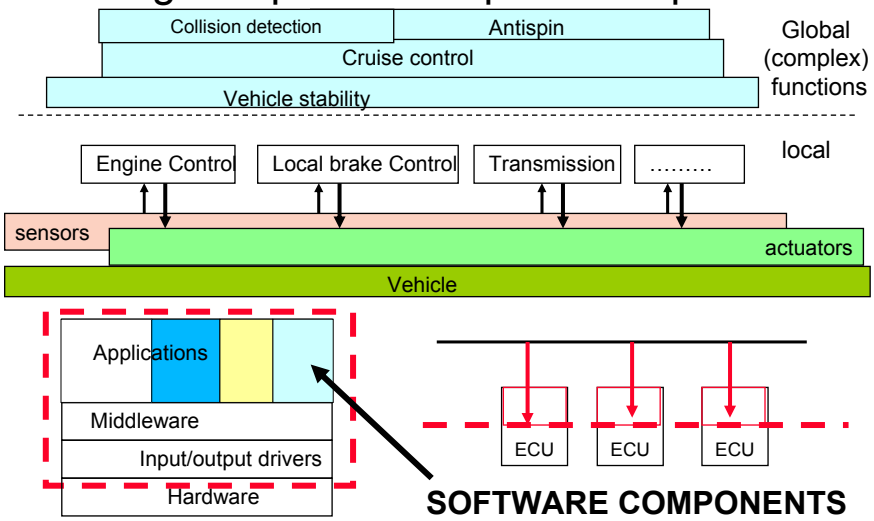
- Different types of approaches
  - Using components as well defined design units
  - Product-line approach
    - Building many variants and versions of a products based on the same architecture and varying components. Components are developed in-house
  - COTS (Commercial off the shelf) components
    - Building systems from existing components

Example: The architecture of a car control system



ECU – Electronic Control Unit

Challenge – open and dependable platform



How much is CBSE attractive for different domains?

- Advantages from a business point of view:
  - Shorter time-to-market, lower development and maintenance costs
- Advantages from technical and engineering point of view
  - Increased understability of (complex) systems
  - Increased the usability, interoperability, flexibility, adaptability, dependability…
- Advantages from strategic point of view of a society
  - Increasing software market, generation of new companies
- CB-approach has been successful in many application domains:
  - Web- and internet-based applications
  - Desktop and office applications, Graphical tools, GUI-based applications
  - In certain segments of telecommunication, consumer electronics…

Do existing component technologies meet the requirements of embedded systems?

- Widely-used component models (Microsoft COM/DCOM and .NET, Sun EJB,J2SEE, OMG Corba CM,…)
  - Focus on functionality, flexibility, run-time adaptability, simpler development and maintenance
  - Do not consider non-functional requirements
    - Timing properties (performance), resource consumptions
    - Reliability, availability, quality of services…

Important questions for component-based development feasibility:

- Which are the primary requirements in different domains?
- Can component-based development provide solutions that meet these requirements?

Specific requirements of embedded systems

- Real-time requirements
- Resource consumption
  - CPU, Memory, Power, Physical space
- Dependability
  - Safety, reliability, availability
- Life-cycle properties (long life systems)
  - Maintainability, expandability
  - Portability
- Increasing interoperability

Basic concepts for Component-based Embedded Systems

- Main concern
  - Predictability of different properties (on account of flexibility)

- Difference between small and large embedded systems

Component models for small embedded systems vs. General-purpose component models

General-purpose components

- Run-time composition (late binding)
- Set of interfaces, “bugs” of services
- Contractual-based interfaces
- Not resource-usage aware
- Coarse-grained
- Portability - Binary independence (Black-box reuse)

Small Embedded Systems

- Composition time separated form run-time
- Optimization at configuration time (direct references, unneeded functionality removed)
- Explicit context dependencies
  - Run-time environment (CPU, RTOS, resource constraints)
- Fine-grained
- Reuse & portability
  - white, glass, or grey box components
  - source code portability

There is a need for component models dedicated to (small) embedded systems

Examples of component models for embedded systems

- SaveCCM
  - Experimental (research) component model
  - Simple enough to be analyzable and predictable
- Koala – Philips
  - Consumer electronics
  - Treats software components as hardware components
- PECOS – ABB
  - Small fieldbus devices
    - Managing power consumption and real-time properties

Component-based approach for LARGE embedded systems

- the resource constraints are not the primary concerns.
- The complexity and interoperability important
- Minimizing the development costs

- *For this reason general-purpose component technologies are of more interest than in a case for small systems.*

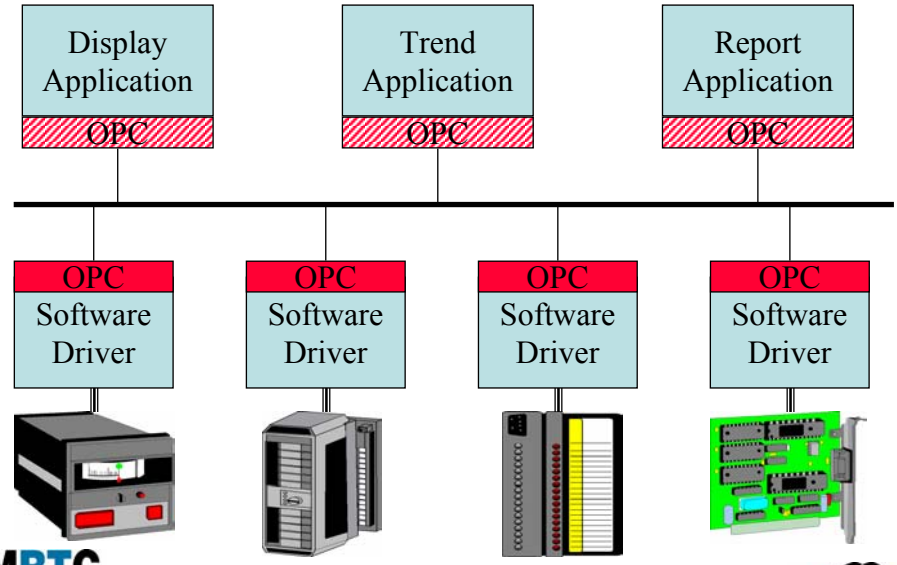
Widely-used component models and embedded systems

- Direct use of component models
  - CORBA (telecommunication)
  - COM/DCOM, .NET – process industry
- Improved component-models (with added functionalities)
  - OPC (OLE process control Foundation)
- Restricted (use of) component-models to achieve predictability
  - Using only specification (IDL) , no multiple interface, etc.

OPC (OLE for Process Control)

- industry standard for communication between components in field devices, automation equipment and business applications (generic driver)
- data model
- set of interfaces for COM
  - server interfaces
  - callback interfaces
- managed by OPC Foundation (www.opcfoundation.org)

OPC: How does OPC Solve the Problem?



Example - Adoption of a general-purpose component model

ABB Controller - Componentization of a system

- Requirements
  - Improve the development process
    - Distributed development
    - Shorter time-to-market Redefined architecture
- Conditions:
  - Keep the main requirements (real-time) unchanged
- Solution
  - Componentize the software archit
  - Use of a subset of a component r

Use of the Component Object Model (COM)

- COM is a public specification (by Microsoft) which emphasizes separation of interfaces and implementation
- A well-defined subset of the COM specification has been adopted
- COM compliant interfaces can be used on platforms without COM run-time support

- On platforms with COM run-time support:
  - Protocol handlers implemented as COM classes

- On platforms without COM run-time support:
  - Protocol handlers implemented as C++ classes, which are statically linked with the framework

Conclusion: What are the needs within Embedded Systems?

- Need for component models and frameworks for embedded systems.
  - the run-time platform must provide certain services, which however must use only limited resources.
- Obtaining extra-functional properties of components in particular timing and performance properties.
- Component certification
- Platform and vendor independence
- Component noninterference applications, (in terms of memory protection, resource usage, etc).
- Tool support: The adoption of component technology depends on the development of tool support.

References

SAVE "book"

Component-Based Development of Safety-Critical Vehicular Systems

1. Problem definition and scope
2. Vehicular Embedded control systems
3. (Modeling of vehicular embedded control systems)
4. System and software life-cycle processes
5. Component-based software development
6. Safety-critical and dependability aspects
7. Real-time systems
8. Verification and testing methods
9. (Case studies)

ARTIST book – Roadmap and research for design of embedded systems

Advanced real-time
Component-based approach
Flexible real-time

References

- Conferences and Workshops
  - Component-based Software engineering Symposium:
    - http://www.sei.cmu.edu/pacc/CBSE7/
    - http://www.sei.cmu.edu/pacc/CBSE8/
  - Euromicro conference, CBSE track
    - http://www.idt.mdh.se/ecbse/2004/
    - http://www.idt.mdh.se/ecbse/2005/
  - WCOP International Workshop on Component-Oriented Programming
    - http://research.microsoft.com/~cszypers/events/wcop2004
  - International Working Conference on Component Deployment
    - http://cd04.cs.ucl.ac.uk/
  - Workshop on Component Composition