DECOS-TADE and DECOS-ISIS Collaborations

www.decos.at

Roman Obermaisser
Overview

- Project Aims and Goals
  - Dependable Embedded Components and Systems (DECOS)
  - Timeliness-Assured Design Environment for Distributed Object-Based Embedded Computing (TADE)
  - Research at Institute for Software-Integrated Systems (ISIS)
- DECOS-TADE Collaboration
  - Technical Topics
  - Respective US-EU Roles and Contributions in the Project
  - Experience and Outlook
- DECOS-ISIS Collaboration
  - Technical Topics
  - Contributions
  - Preliminary Results
Project Aims and Goals of DECOS
Integrated Project

- Enabling technology to move from a federated distributed architecture to an integrated distributed architecture
- A system architecture that combines the complexity management advantages of the federated approach, but also realizes the functional integration and hardware benefits of an integrated system (Hammett, 2003)
- Reduce development, production and maintenance cost
- Increased dependability of embedded applications
- Technology invariant software interfaces
- Encapsulated communication and computational resources
Project Aims and Goals of TADE

• Strengthen the technical foundation for enabling the system engineers to produce guarantees for timely service capabilities of various subsystems of distributed real-time embedded computing systems, especially on the basis of the recently established distributed real-time object-oriented programming methods and tools.

• Develop an integrated tool-set named the Timeliness-Assured Design Environment (TADE), aimed for enabling major reduction in the system engineers' efforts in producing DREC systems with service time guarantees.
Research at Institute for Software-Integrated Systems (ISIS) at Vanderbilt University

- Model-Integrated Computing
  - Specify integrated, multiple-view models
  - Model interpreters translate information in models to input languages of analysis tools
  - Automatic software synthesization
  - UML-based meta programming

- Model-Driven Architecture for Embedded Software
Technical Topics of DECOS/TADE Collaboration

- Formalization of the structure of the real-time distributed component named TMO and the linking interface specification of DECOS jobs
- Collaborative work on the integration of event-triggered and time-triggered communication paradigms
- Comparison of commercial off-the-shelf platforms (common hardware and operating systems) as used in TADE with the introduced integrated DECOS platform
- Comparative studies of the TADE testbed and the DECOS test-bench, along with cooperative experiments
  - Performance measurements
  - Fault-injection experiments for evaluation of encapsulation of communication and computational resources
  - Validation of development tools
Contribution of TADE: Time-Triggered Message-Triggered Object (TMO) Model

• Powerful extension of conventional distributed object computing
  – Globally referenced time base
  – Distributed object computing
  – Spontaneous methods
  – Basic concurrency constraint
  – Guaranteed completion time and deadline for result arrival
• Tools and methods for composition of real-time distributed object programs
• Execution engines on several major platforms (e.g., Windows, Linux)
Contribution of DECOS

- Design methodology with tool support based on the Model Driven Architecture (MDA)
  - Meta-models constrain the development process
  - Formal specification of linking interfaces (value, time, dependability)
  - Tools facilitate refinement of models
- Generic architectural services as a validated stable base line for applications
- Encapsulated of communication and computational resources
Virtual Networks in DECOS

Job 1   Job 1   Job 2   Job 2
DECOS Integrated Component Model

- Horizontal and vertical partitioning
- Safety-critical and non safety-critical subsystem

Application Computer
(Processor Core)
Collaboration Results: TMO on top of a Time-Triggered Core Architecture (1)

- TMO execution engine for a time-triggered core architecture
- Implementation of a Kernel Abstraction Layer (KAL) that maps TMO middleware onto the DECOS core architectural services (C1-C4)
- Integrated DECOS architecture is enriched with support for application subsystems based on distributed object computing
  - Support for realization of DECOS jobs as TMOs
  - Exploitation of encapsulated event-triggered and time-triggered virtual networks for method invocations and multicast communication channels
Collaboration Results: TMO on top of a Time-Triggered Core Architecture (2)

• TMO with improved temporal performance
  – Global time base with a precision of 5 µs
  – Minimization of communication jitter by devising conflict-free time-triggered schedules at design time for both processing activities (e.g., spontaneous methods) and communication activities (e.g., messages for remote method invocations and those exchanged via multicast channels)

• TMO with improved dependability
  – Improved reliability of application TMOs through the fault-tolerance mechanisms of the underlying services (i.e., fault-tolerant communication service, fault-tolerant clock synchronization, fault isolation between components).
  – Basis for building fault-tolerant TMO applications with active redundancy at relative ease (e.g., Triple Modular Redundancy)
Outlook of TADE/DECOS Collaboration

- Measurements of the temporal performance of the TMO execution engine developed within the TADE/DECOS collaboration
- Realization of a multimedia application as a demonstrator based on the newly developed TMO execution engine
- Enhancement of the linking interface specification framework based on the component formalization efforts of TADE
- Comparative studies of the TADE testbed and the DECOS test-bench
- TMO based on a Time-Triggered Ethernet Platform
Technical Topics of DECOS-Vanderbilt Collaboration

- Model Integrated Computing for DECOS
- Case study: Meta-modeling of sensor DAS
- Interpreters for exporting information to other tools (compiler, scheduling tools, etc.)
- Model transformation tools support export to other model-based tools (e.g. MATLAB)
Contributions

• Contribution of DECOS
  – Generic architectural services
  – Virtual networks with predefined temporal properties
  – Design methodology for development process
  – Implementation platform
• Contribution of ISIS
  – Metamodeling methodology
  – Tool integration platform and methodology
  – Generic Modeling Environment (GME)
  – Model Transformation Tool (GReAT)
  – Design Space Exploration Tool (DESERT)
Collaboration Results

- Generic modeling of a sensor DAS using GME from ISIS
- Modeling of a particular sensor DAS using GME
- Approach allows for a separation of concerns between system designer and component implementer in practice
Outlook on DECOS-ISIS Cooperation

- Implementation of model interpreter
- Modeling of case study that exists in real hardware
- Integration of model transformation tool
- Integration of design space exploration tool