Transatlantic Collaboration on Model-Integrated Computing for Dependable Embedded Components and Systems

Wilfried Elmenreich
Vienna University of Technology
Institute of Computer Engineering
Vienna, Austria
wil@vmars.tuwien.ac.at

Gabor Karsai
Vanderbilt University
Institute for Software Integrated Systems
Nashville, TN, USA
gabor.karsai@vanderbilt.edu

Abstract

This document summarizes early results and experiences of the collaborations between the European project "Dependable Embedded Components and Systems" (DECOS) (IST-511764) and the NSF ITR Project titled "Foundations of Hybrid and Embedded Software and Systems".

The collaboration has started with a visit of a researcher from Vienna University of Technology at Vanderbilt University in summer 2005, a visit from a graduate student in Vienna and a faculty visit from Vanderbilt researchers at the Budapest University of Technology and Economics in Hungary.

1 Description of Projects

1.1 Project on Foundations of Hybrid and Embedded Software and Systems

The NSF ITR Project titled "Foundations of Hybrid and Embedded Software and Systems" aims at building the foundations of Modern Systems Science (MSS). It is a sustained effort to create a new generation of engineers that are comfortable with the juncture of computation and physical phenomena. The research is carried out at UC Berkeley's Center for Hybrid and Embedded Software Systems, the Vanderbilt University's Institute for Software-Integrated Systems, and the University of Memphis. Educational outreach programs include the California community college system, which feeds many of the engineering students to UCB and other State Universities, and HBCUs and universities with high minority populations in the South.

1.2 DECOS Project

The EU IST project "Dependabe Embedded Components and Systems" (DECOS) that will develop the basic technology to move from a federated distributed architecture to an integrated distributed architecture in order to reduce development, production, and maintenance cost. Furthermore, the objective is to increase the dependability of embedded applications in many application domains.

The DECOS project is an Integrated Project (IP) and it is coordinated by ARC Seibersdorf Research, Austria, with the following objectives:

- The development of a distributed execution platform that allows tight integration of software modules with different criticality levels from different sources (vendors). Such integration is realized by effective partitioning – not only at architecture level, but even at the level of a single hardware unit (ECU).
- The provision of methods, tools, and hardware components (silicon) to support the concept of a partitioned distributed execution platform, which will be based on a time-triggered core architecture. Partitioning will take place by encapsulation of communication and task-execution.
- A coherent and integrated diagnostic concept that records and analyses out-of-norm behavior in order to effectively support the maintenance process.

- The development of a test-bench to support the validation and certification of dependable embedded systems.
- The development of concepts and methodologies to visualize and navigate complex real-world systems during the design phase as well as during system validation in order to support design comprehension.
- The proof-of-concept demonstration that shows the applicability of the developed technologies in three application domains (automotive, aerospace, control).
- A coherent application development support by the provision of basic software building blocks easing the generation of safe SW modules.

Under the coordination of ARC Seibersdorf research the DECOS project includes over 20 partners from academic and industrial side, among them the Institute of Computer Engineering at Vienna University of Technology (TU Wien) and the Budapest University of Technology and Economics (BUTE), Hungary.

2 Collaboration Details

2.1 Involved Participants

At Vanderbilt University (VU), the Institute for Software Integrated Systems (ISIS) is the leading proponent of model-integrated computing, a paradigm that is central to MSS. ISIS does research in the fields of Model-Integrated Computing [2] (model transformation, model analysis, and code generation tools), Middleware for Distributed Real-time & Embedded Systems (Adaptive & reflective middleware, model-based integration technology above component models, Real-time CORBA (ACE/TAO), Model Driven Architectures, secure middleware), Model-Based Systems Applications (Fault management, distributed control, automotive, avionics), and Secure Networked Embedded Systems (Shooter location, security in system context).

The Institute of Computer Engineering at TU Wien consists of three working groups, the Real-Time Systems Group, the Embedded Computing Systems Group, and the System-on-a-Chip Group. The DECOS project work is done by the Real-Time Systems Group. The research by the Real-Time Systems Group is focused on distributed, fault-tolerant real-time systems and the time-triggered paradigm [1].

The DECOS partner at BUTE is the Department of Measurement and Information Systems. Education, research, and development at the department is related to embedded information systems, intelligent system design, and dependable computing technology.

2.2 Forms of Collaboration

The actual collaboration has taken different forms:

- Researcher visits: Dr. Wilfried Elmenreich from TU Wien has visited VU/ISIS. During his visit he learned about the VU/ISIS metaprogrammable toolsuite and started using it for building a DSML and modeling tool for the TTA.
- Faculty visit: Profs. G. Karsai and J. Sztipanovits from VU/ISIS have visited BUTE in Hungary and discussed collaboration plans with Prof. G. Peceli.
- Graduate student visit: Mr. Tivadar Szemethy (VU/ISIS graduate student) has attended the Summer School on Dependable Embedded Systems in Vienna, Austria, in Sep, 2005, which is organized and supported by the DECOS project.

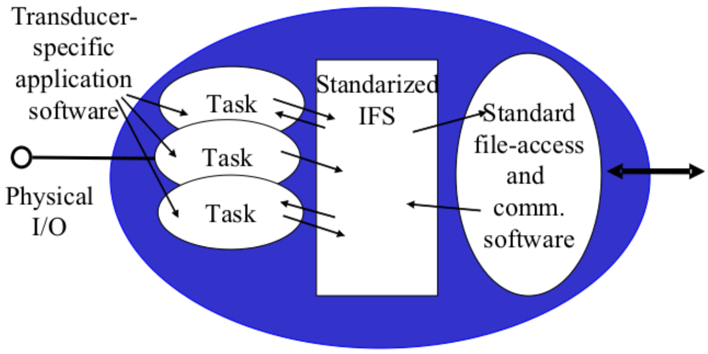

Figure 1: Architecture of a Smart Transducer Node

3 Technical Results

For the research within the DECOS project, the model-integrated computing approach from VU is of special interest.

VU/ISIS has expertise and research background in domain-specific modeling tools and environments, while TU (Technical University) Wien, Austria (a leading DECOS participant) has expertise in fault-tolerant embedded system platforms, like the TTA. The two projects could mutually benefit from each other and work has started on using the VU/ISIS tools in a domain-specific modeling environment for the DECOS platform.

The ISIS tool suite (GME, GREaT, DESERT) supports the modeling of various hierarchic systems. By designing an appropriate meta model the tools support concise and understandable modeling of the target domain. In contrast, by using a pure UML approach, the modeling possibilities are still theoretically unlimited, but the ability to comprehend a model might be limited, since UML does not adopt the native concepts of the target system.

For example, in the DECOS architecture, we use the concepts of a Distributed Application Subsystem, a job and a task as native concepts, thus these concepts should also be reflected in the modeling language [3].

In an early experiment, a modeling tool for a particular "Sensor DAS" using the tool Generic Modeling Environment (GME) [4] has been implemented.

GME is an open source tool that is used for the design of the meta model and the domain-specific model as well. It can be extended by so-called interpreters in order to support automatic software synthesization and generation of documentation or input for analysis tools. GME also integrates well with the Graph Rewrite And Transformation (GReAT) tool suite, which is a metamodel based graph transformation language useful for the specification and implementation of model-to-model transformations.

The meta model of our implementation describes the sensors, actuators, control algorithms and timing requirements of a transducer system. It supports a design approach that provides a separation of local and system issues in order to handle complexity and support software reuse. Thus, we implemented two views consisting of a platform-independent model and a platform-specific model. As target system we assume a time-triggered fieldbus system according to the smart transducer interface standard of the OMG [5]. This system will integrate well with the timing of the time-triggered messages within the DECOS architecture.

Figure 1 depicts the architecture of a smart transducer node. One node may host several tasks. In this programming paradigm, each task constitutes a job, which is the unit of distribution.

One node may host multiple jobs belonging to different application subsystems. A job is the unit of distribution, thus an application subsystem typically consists of jobs running on several nodes due to the availability of external and internal node resources, reliability and legacy node issues.

In order to work, a node requires the implementation of its jobs (e. g., polling a sensor, instrumenting an actuator or calculating a set point value from a given input), the binding of the jobs input and output to the communication interface and the correct set-up of the nodes communication and execution schedule. By following a two-level design approach, the job implementation is done by a programmer who is an expert regarding the local hardware, while the set-up of the nodes communication and execution

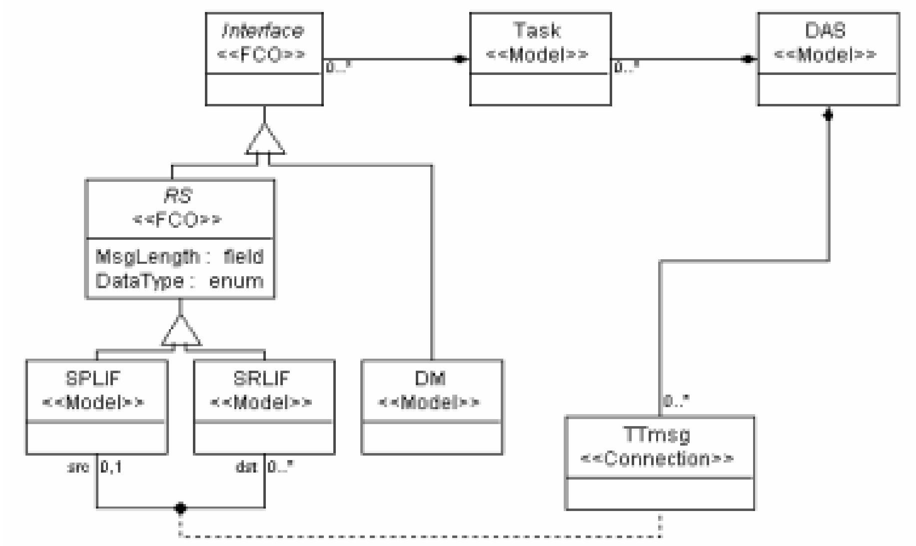

Figure 2: UML Meta model of a time-triggered smart transducer application

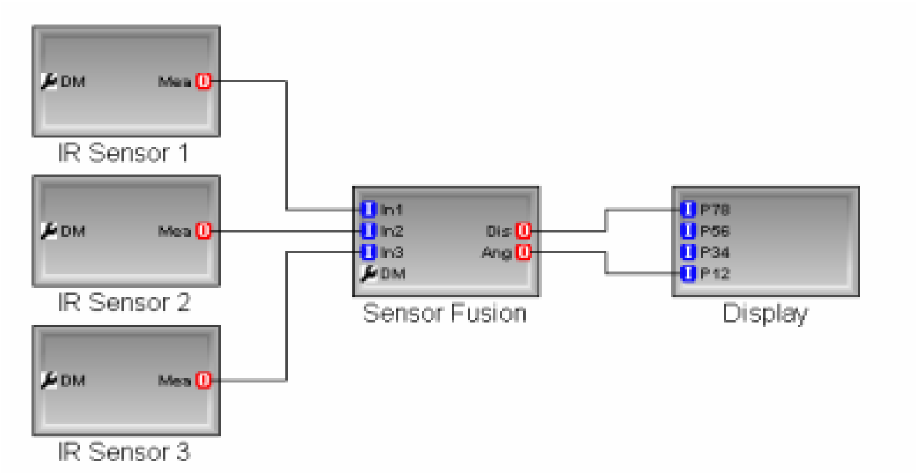

Figure 3: Example of a simple smart transducer system modeled in GME

schedule is done by a system integrator using appropriate tools. The binding of the jobs input and output to the communication interface is, however, difficult to separate, since in the current implementation of the smart transducer interface, this task requires information from both domains.

The case study of a GME meta model of a sensor application attacks this problem by giving the system integrator a tool that relieves him from dealing with low-level issues. Therefore, a generic time-triggered smart transducer application had to be modelled in UML (see Figure 2) and registered in GME as a modelling paradigm.

Using this paradigm, GME can be used as a domain-specific modelling tool. Figure 3 depicts a simple smart transducer system consisting of three sensors, one calculation job and one display job modelled in the domain-specific GME meta model.

As next step, the collaboration will continue by building a model interpreter that translates the models into an analysis model, and by integrating the modeling environment with the design space exploration tool (another VU/ISIS tool).

4 Experiences in the Collaboration Process

The experiences in the DECOS Vanderbilt collaboration have been very positive, especially during the 6 week research visit of Dr. Elmenreich at Vanderbilt University.

The researchers at the guest institute had been very cooperative, so that the visitor could optimize the time in order to exchange knowledge. Particularly, this has been done by giving a scientific talk presenting the DECOS goals and architecture, followed by personal discussions with the respective researchers.

Being physically at the place had been very helpful, since even though the existing publications and project pages of the guest institute had been very informative, some approaches, algorithms, and tools had not yet been publicly available, only available on request, or could only be downloaded and installed under guidance, respectively.

For DECOS, this visit has resulted in a deeper understanding of Model-Integrated Computing and the tool chain from ISIS. A key benefit of the visit at the partner universities had been the availability of in-person explanations, which would have been difficult to perform by email or phone.

There had been no problems during the visit in Nashville, however, the process of applying a scholar Visa at the US embassy appeared to be a necessary, but sumptuous task. Due to the large amounts of paperwork and involved parties (applicant, home university, guest university, partner institute, US embassy), the time to complete the Visa application process was difficult to estimate and complicated the planning (e.g. flight dates) of the visit.

5 Conclusion and Outlook

The giving example has shown the great potential of the model-integrated computing approach from ISIS for the research goals of DECOS.

An important point is the dissemination and use of the acquired knowledge within DECOS. Therefore, we will study the possibility to use GME as a modeling tool in other aspects of the DECOS architecture. A possible application could be to use GME to model the hardware elements of a cluster within the DECOS architecture according to the hardware specification model elaborated in DECOS work package 1.2.

In the future, we hope to continue and extend the collaboration between DECOS and ISIS, since the first experiences have shown a great potential for cooperative work in the field of embedded systems. A further cooperation between ISIS and DECOS generally will be very fruitful for both groups.

Acknowledgments

This work was supported in part by the European IST project DECOS under contract No. IST-511764. I would like to give special thanks to Janos Sztipanovits, Akos Ledeczi, Sandeep Neema and all other researches at ISIS for their great support in all aspects during my stay at Vanderbilt University.

References

[1] W. Elmenreich, G. Bauer, and H. Kopetz. The time-triggered paradigm. In *Proceedings of the Workshop on Time-Triggered and Real-Time Communication*, Manno, Switzerland, December 2003.

[2] G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty. Model-integrated development of embedded software. *Proceedings of the IEEE*, 91(1):145–164, January 2003.

[3] H. Kopetz, R. Obermaisser, P. Peti, and N. Suri. From a federated to an integrated architecture for dependable embedded systems. Research Report 22/2004, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, 2004.

[4] A. Ledeczi, A. Bakay, M. Maroti, P. Volgyesi, G. Nordstrom, J. Sprinkle, and G. Karsai. Composing domain-specific design environments. *Computer*, pages 44–51, November 2001.

[5] Object Management Group (OMG). *Smart Transducers Interface V1.0*, January 2003. Specification available at http://doc.omg.org/formal/2003-01-01 as document ptc/2002-10-02.