Component Based Design for Embedded Systems

Report on the US-EU Workshop

July 7-8th, 2005 in Paris

http://www.artist-embedded.org/FP6/ARTIST2Events/PastEvents/IST-NSF/

Table of Contents

Component Based Design for Embedded Systems

Report on the US-EU Workshop, Paris, France July 7- 8th 2005

http://www.artist-embedded.org/FP6/ARTIST2Events/PastEvents/IST-NSF/

Joseph Sifakis[1], Janos Sztipanovits[2], Gabor Karsai[2], Shankar Sastry[3],
Claire Tomlin[4], Bruno Bouyssounouse[1]

1. Executive Summary

The meeting was organized over two days. The first day featured presentations from industry participants from both the US and Europe - including Airbus, Boeing, European Aerospace and Defence Systems (EADS), Ericsson, European Space Agency, Honeywell, Israel Aircraft Industries, Raytheon, and Thalès - as well as brief position statements from European and US academic partners. The day ended with a panel discussion on conclusions for future work directions.

These conclusions were refined and integrated during the second day, which began with reviews of several active EU-US collaborative research efforts (funded by EU-IST and US-NSF), including HIPEAC in collaboration with Princeton/Rutgers, DECOS in collaboration with UC Irvine, RUNES with UC Berkeley + Caltech , ARTIST/ARTIST2 with UC Berkeley + Vanderbilt. This was followed by three presentations on Challenging Topics: “Security of Embedded Systems” by Catherine Meadows (US Naval Research Laboratory), “Component-based Design” by Joseph Sifakis (VERIMAG), and “Network Embedded Systems” by Margaret Martonosi (Princeton University). The conclusions of the workshop were developed in a session co-chaired by Drs. Alkis Konstantellos of the EU-IST and Helen Gill of the US-NSF, CISE.

The day continued with a discussion about the current EU-IST/US-NSF funding mechanism and models of collaboration for joint projects. The current “pilot” process consists of identifying EU-IST and US-NSF partner projects after their selection, and supplementing their budgets with additional resources for joint work. Projects have a joint evaluation annually to assess progress of the joint work.

The overall conclusions are:

1. Embedded Systems and Software are the basic engine of innovation for a broad range of industrial sectors. This is the technology that transforms products, creates new markets and disrupts the status-quo.
2. The existing EU-US “pilot” collaborative programs have proven to be an extremely valuable source for developing new approaches, tools and system designs for problems strongly motivated by urgent industrial priorities both in Europe and the USA, and can serve as a springboard for further expansion.
3. Joint tool development will be a key strategic precursor for an emerging research and development infrastructure encompassing both research organizations and industry. This is important because existing tool vendors are not economically motivated by a unified embedded systems design tools market.
4. The current pilot model of collaboration shows tremendous promise. The model should be finalised, and strengthened through dedicated long-term support, and increased funding for researchers, students, and faculty.

[1] Verimag Laboratory and ARTIST2 Network of Excellence
[2] Vanderbilt University and ISIS
[3] UC Berkeley and CITRIS
[4] Stanford University and CITRIS

Recommendations include:

a. Longer period of performance (3-5 years), consistent with the length of the parent projects, as well as the duration of student Ph.D. topics.

b. Increased support for student stipends (possibly with co-advising arrangements).

c. On the US side, the mechanism should not be restricted to specific programs such as the NSF-ITR.

5. Embedded Systems and Software is an area of paramount industrial and economic interest. This fact has been recognized in setting up ARTEMIS as one of the pillars of the 7th EU-IST Framework Programme. A corresponding initiative on the US side would be tremendously beneficial for balanced collaboration with the EU-IST.

   Here is a summary of the research priorities.

   a. Adaptive Dynamic Architectures for High Confidence Embedded Real Time Systems, including a focus on services such as security and privacy.

   b. Composable Tool Environments and experimental platforms.

   c. Networked Embedded Systems: beyond sensor networks.

6. Embedded Systems and Software is a key component for ensuring safety and efficiency in Critical Infrastructures, such as the production and distribution of electrical power, health care, transportation, water. It is also vital to the development of scientific experimental infrastructures for areas such as systems biology, nano-science and technology. International Collaboration is vital for addressing issues of scale, joint standards, reliability, fault tolerance, and security. It is essential that we invest in enabling technologies for embedded systems design, for tomorrow's Critical Infrastructures.

7. Dissemination mechanisms are an essential part of International Collaboration. The goal is to maximize the impact of publicly-funded research investment, using mechanisms such as restricted open source, non-exclusive royalty-free licenses, and peer-reviewed public domain repositories for tools and software.

8. The role of industry in EU-US collaboration includes:

   a. <u>Participation</u>
   The EU-IST has a mechanism for providing direct funding for industrial research participants. There is no such mechanism on the US side, which results in an imbalance, and lessens the impact of NSF research.

   b. <u>Strategic Advice</u>
   Creation of a joint EU-US Senior Industrial Advisory Board is recommended.

   c. <u>Uptake of Results</u>
   Creation of open experimental (vendor neutral) test-beds with industrial participation is highly recommended.

2. Research Priorities

2.1. Adaptive Architectures for High Confidence Embedded Real-time Systems

This priority is composed of the following interdependent topics, in the following order:

2.2.1. Architecture

The aim is to develop a formal framework for architectures of embedded real-time systems.

Architectures allow construction of complex systems by composing simpler components. The framework should consider architectures as first-class entities, having their own properties that can be studied independently of the components' behavior. Such a framework is characterized by the following:

a) It encompasses the composition of heterogeneous components, taking into account the 3 fundamental sources of heterogeneity: abstraction, execution, interaction.

- Heterogeneity of abstraction results from the need at design time for dealing with languages, models and implementations representing a system and its components at different levels of abstraction, such as requirements, functional specifications, application software, and the physical implementation. An important abstraction is the one relating application software to its implementation on a given platform.
- Heterogeneity in the interactions between components. Interactions can be atomic or non atomic, and may involve strong or weak synchronization.
- Heterogeneity of execution, including both synchronous and asynchronous execution.

b) It is equipped with theory, methods and tools for correctness-by-construction. These allow the inference of overall system properties from architectural properties. Given the inherent difficulty of this problem, we intend to focus on correctness-by-construction for simple and generic properties such as deadlock-freedom and component composability.

2.2.2. Adaptivity

Adaptivity is the capacity of a system to meet given requirements including safety, security, and performance, in the presence of uncertainty in its external or execution environment. This capacity includes adaptation of the system's structure and parameters determining its dynamic behaviour.

Adaptivity is a means for enforcing predictability of behavior in the presence of uncertainty, which is characterised as the difference between average and worst-case behavior in a system's environment.

Component-based design is essential for adaptive systems, to allow seamless modifications to the architecture (e.g., adding or deleting components).

The aim is to develop holistic adaptive component-based design techniques allowing the satisfaction of both critical and resource-optimization properties, and thus bridge the current gap between critical and best-effort engineering. Standard practice for critical systems is based on worst-case behaviour analysis and the static allocation of resources (eg: worst-case execution times, static redundancy). This leads to a physical separation between critical and non-critical parts of a system running on dedicated physical units, and implies higher costs and reduced hardware reliability (e.g., increasing numbers of ECUs in automotive systems). Current

technological trends from federated to integrated architectures (e.g., Integrated Modular Avionics) lead us to act urgently in this direction.

Work on adaptive systems should lead the way in integrating approaches and results from control theory, hybrid systems, planning and learning theory, as well as ad hoc adaptive techniques for networks and multi-media systems.

2.2.3. High-Confidence

An important trend in embedded systems is the shift from traditional post-design techniques (e.g.: massive redundancy) to techniques where high-confidence is a guiding concern from the very start of system design.

For such techniques, architecture is a means for ensuring high-confidence properties such safety, security, and privacy. A first objective is to study architectures and structuring principles for embedded systems, allowing given high-confidence requirements to be met.

Adaptivity is the last rampart for ensuring high confidence in complex systems. It is inevitable that these systems have various kinds of defects - due to design errors, faults, failures. Furthermore, their environments are increasingly complex, non-deterministic, and possibly hostile. A second objective is to study adaptive techniques that enhance a system's resistance to these phenomena.

2.2. Composable Tool Environments and Experimental Platforms

There is a need to radically change the state-of-the-art in high-confidence software development, by delivering advanced tool chain prototypes that are suitable for the model-integrated development of embedded software. Existing development environments are typically programming language-centered and IDE-based, follow a vendor-specific model, and often completely ignore the verification and testing aspects. Embedded systems developers need *composable* domain-specific tool chains that are comparable to tool chains built by the EDA industry, which include: (1) modeling tools; (2) model, system and code verification tools; (3) code synthesis and generation tools; and (4) run-time verification tools.

2.2.1. Composable Tool Chains

There are significant technical challenges in building composable, domain-specific tool chains. Reusability of high-valued, generic tools in strongly different domain specific environment is a hard problem that must be solved to make the approach viable for the tool industry. New research results in meta-programmable tools and tool architectures show strong promise that effective solutions can be found.

The composition of tools requires solutions for the explicit specification of semantics in modeling languages, because semantic ambiguities may produce conflicting results across different tools. This is unacceptable in most embedded software applications, and particularly so in safety-critical applications. *Semantic anchoring* of domain-specific modeling languages is an emerging research direction that addresses the core issues in semantic integration of heterogeneous tool chains.

We believe that it is highly improbable that a composable tool chain for high-confidence embedded systems design will emerge from the EDA industry. Business pressures make companies uninterested in disclosing the semantics used in their tools, and in supporting composability of tool chains. Therefore, standard-based, open tool integration frameworks for embedded systems need to be established with the involvement of large end-users and tool vendors.

2.2.2. Experimental Platforms

International research efforts require that research results can be compared and evaluated on reference problems defined on open experimental platforms. Development of innovative, inexpensive experimental platforms that can be easily reproduced at research sites and that offer a rich problem space is an important goal, which can significantly accelerate progress.

2.3. Networked Embedded Systems: Beyond Sensor Networks

Recent technology developments have made it possible and practical to deploy secure, distributed, and networked embedded systems in societal-scale infrastructure systems. Most of these networked embedded systems are in fact sensor networks: they gather data *in situ* and provide advanced embedded information processing and fusion capabilities to monitor and understand the physical environment. Sensor network research has addressed important issues and provided solutions for ad-hoc networking, power-aware and resource-limited operating systems, distributed system protocols and new programming models for application development, such as programming the ensemble and not individual nodes.

However, it becomes vastly more valuable if this new technology is extended for *closing the loop.* This capability is synergistic with most embedded computer-based systems that are usually part of a sensor *and actuator* network. Closing the loop imposes new requirements on the technology infrastructure. They must coordinate embedded components, fuse data streams into coherent views, diagnose events, and control their own physical infrastructure by driving actuators. Moreover, a network of embedded computing components may need to adapt *their own behavior* as a function of what they detect, so as to be able to generate feedback response under timing and reliability constraints.

Whereas engineering disciplines offer analytical and technical toolkits for the design of the classical aspects of such systems (e.g. structural dynamics, feedback control theory, thermodynamics and hydrology), there is little help for the design of fine-grained networked embedded systems, which couple myriad details of hardware and software platforms to the “semantics” of an external, and physical, world.