Foundations of Hybrid and Embedded Software and Systems: Project Overview

Edited and presented by
S. Shankar Sastry, PI
UC Berkeley

Chess Review
November 18, 2004
Berkeley, CA

NSF-ITR Investigators

Ruzena Bajcsy, Ras Bodik, Bella Bollobas, Gautam Biswas, Tom Henzinger, Kenneth Frampton, Gabor Karsai, Kurt Keutzer, John Koo, Edward Lee, George Necula, Alberto Sangiovanni Vincentelli, Shankar Sastry, Janos Sztipanovits, Pravin Varaiya.

ITR-Center Mission

- The goal of the ITR is to provide an environment for graduate research on the design issues necessary for supporting next-generation embedded software systems.
  - The research focus is on developing model-based and tool-supported design methodologies for real-time fault-tolerant software on heterogeneous distributed platforms.
- The Center maintains a close interaction between academic research and industrial experience.
  - A main objective is to facilitate the creation and transfer of modern, "new economy" software technology methods and tools to "old economy" market sectors in which embedded software plays an increasingly central role, such as aerospace, automotive, and consumer electronics.

Mission of Chess

To provide an environment for graduate research on the design issues necessary for supporting next-generation embedded software systems.

- Model-based design
- Tool-supported methodologies

For

- Real-time
- Fault-tolerant
- Robust
- Secure
- Heterogeneous
- Distributed Software

We are on the line to create a "new systems science" that is at once computational and physical.

Hybrid and Embedded Software: Problem for Whom?

- *DoD (from avionics to micro-robots)*
  - *Essential source of superiority*
  - *Largest, most complex systems*
- *Automotive (drive-by-wire)*
  - *Key competitive element in the future*
  - *Increasing interest but low risk taking*
- *Consumer Electronics (from mobile phones to TVs to sensor webs)*
  - *Problem is generally simpler*
  - *US industry is strongly challenged*
- *Plant Automation Systems*
  - *Conservative solutions to date*
  - *Emerging importance of SCADA/DCS in Critical Infrastructure Protection*

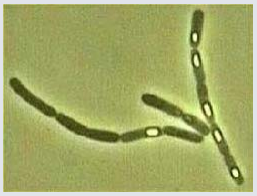
Some Applications Addressed

Automotive

Avionics: UAVs

Networked Embedded Systems

Systems Biology



Key Properties of Hybrid & Embedded Software Systems

- Computational systems
  - but not first-and-foremost a computer
- Integral with physical processes
  - sensors, actuators
- Reactive
  - at the speed of the environment
- Heterogeneous
  - hardware/software, mixed architectures
- Networked
  - adaptive software, shared data, resource discovery
  - Ubiquitous and pervasive computing devices

Project Approach

- Model-Based Design (the view from above)
  - principled frameworks for design
  - specification, modeling, and design
  - manipulable (mathematical) models
  - enabling analysis and verification
  - enabling effective synthesis of implementations
- Platform-Based Design (the view from below)
  - exposing key resource limitations
  - hiding inessential implementation details
- Tools
  - concrete realizations of design methods

Foundational Research

- The science of computation has systematically abstracted away the physical world. The science of physical systems has systematically ignored computational limitations. Embedded software systems, however, engage the physical world in a computational manner.
- We believe that it is time to construct an Integrated Systems Science (ISS) that is simultaneously computational and physical. Time, concurrency, robustness, continuums, and resource management must be remarried to computation.
- Mathematical foundation: Hybrid Systems Theory: Integrated Systems Science.

… and Embedded Software Research

- Models and Tools:
  - Model-based design (platforms, interfaces, meta-models, virtual machines, abstract syntax and semantics, etc.)
  - Tool-supported design (simulation, verification, code generation, inter-operability, etc.)
- Applications:
  - Flight control systems
  - Automotive electronics
  - National experimental embedded software platform
- From resource-driven to requirements-driven embedded software development.

Some Current Research Focus Areas

- Software architectures for actor-oriented design
- Interface theories for component-based design
- Virtual machines for embedded software
- Semantic models for time and concurrency
- Design transformation technology (code generation)
- Visual syntaxes for design
- Model checking hybrid systems
- Autonomous helicopters
- Automotive systems design
- Networked Embedded Systems
- Systems Biology

- GME
- GReAT
- DESERT
- Fresco
- Ptolemy
- HyVisual
- Metropolis
- BEAR
- MESCAL

Center Organization

- Funding Sources
  - Large NSF ITR
  - Other federal (NSF, DARPA, MURI, etc.)
  - Industrial (Participating Member Companies): IT and applications (automotive, aerospace, consumer electronics, systems biology)
- Outreach
  - Curriculum development
  - Community colleges+ San Jose State University (EECS 20)
  - SUPERB-IT program (Sum. Prog. for Engg. Res. In IT @Berkeley)
  - SIPHER program (Sum. Intern. Prog. For Hybrid and Embedded Res @Vanderbilt)
- National Experimental Platform for Hybrid and Embedded Systems and Software NEPHEST → Embedded Software Consortium for Hybrid and Embedded Systems (ESCHER)

NSF ITR Organization

- PI: Shankar Sastry
- coPIs: Tom Henzinger, Edward Lee, Alberto Sangiovanni-Vincentelli, Janos Sztipanovits
- Participating Institutions: UCB, Vanderbilt, Memphis State
- Five Thrusts:
  - Hybrid Systems Theory (Henzinger)
  - Model-Based Design (Sztipanovits)
  - Advanced Tool Architectures (Lee)
  - Applications: automotive (ASV), aerospace (Sastry)
  - Education and Outreach (Karsai, Lee, Varaiya)
- Five year project: kick-off meeting November 14th , 2002. First Review May 8th, 2003, Second Review Dec 3rd, 2003, Third Review May 10th, 2004.
  - Weekly seminar series
  - Ptolemy workshop May 9th, 2003, April 27th 2004
  - NEST + CHESS Workshop May 9th, 2003
  - BEARS Open House, February 27th 2004

Thrust 1 Hybrid Systems

- Deep Compositionality
  - Assume Guarantee Reasoning for Hybrid Systems
  - Practical Hybrid System Modeling Language
  - Interface Theory for hybrid components
- Robust Hybrid Systems
  - Bundle Properties for hybrid systems
  - Topologies for hybrid systems
  - Stochastic hybrid systems
- Computational hybrid systems
  - Approximation techniques for H-J equations
  - Synthesis of safe and live controllers for hybrid systems
- Phase Transitions

Thrust II: Model Based Design

- Composition of Domain Specific Modeling Languages
  - Meta Modeling
  - Components to manipulate meta-models
  - Integration of meta-modeling with hybrid systems
- Model Synthesis Using Design Patterns
  - Pattern Based Modal Synthesis
  - Models of Computation
  - Design Constraints and Patterns for MMOC
- Model Transformation
  - Meta Generators
  - Scalable Models
  - Construction of Embeddable Generators

Thrust III:Advanced Tool Architectures

- Syntax and Synthesis
  - Semantic Composition
  - Visual Concrete Syntaxes
  - Modal Models
- Interface Theories
- Virtual Machine Architectures
- Components for Embedded Systems

Thrust IV: Applications

- Embedded Control Systems
  - Avionics
  - Veitronics
  - Wireless Embedded Systems
- Embedded Systems for National/Homeland Security
  - Air Traffic Control
  - UAVs/UGVs
- Networks of Distributed Sensors
- Stochastic Hybrid Systems in Systems Biology
- Hybrid Models in Structural Engineering
  - Active Noise Control
  - Vibration damping of complex structures

Thrust V: Education and Outreach

- Curriculum Development for MSS
  - Lower Division
  - Upper Division
  - Graduate Courses
- Undergrad Course Insertion and Transfer
  - Goals and ABET requirement
  - New courses for partner institutions (workshop held March 1st 2003, Summer 2004)
  - Introduction of new courses (will be replacing control course at upper division level by embedded control course jt with San Jose State)
  - CHESS-SUPERB/ Summer Program in Embedded Software Research SIPHER program (6 + 4 students in Summer 03, 3 + 5 in Summer 04)
- Graduate Courses
  - EECS 249 Design of Embedded Systems: Models, Validation, and Synthesis
  - EECS 290N Concurrent Models of Computation for Embedded Software
  - EECS 291E/ME 290S Hybrid Systems

Outreach Continued

- Interaction with EU-IST programs
  - Columbus (with Cambridge, l'Aquila, Rome, Patras, INRIA)
  - Hybridge, Hycon (with Cambridge, Patras, NLR, Eurocontrol, Brescia, KTH)
  - ARTISTE, ARTIST-2: Educational Initiatives (Grenoble, INRIA, ETH-Zurich)
  - RUNES, new EU-IST program in network embedded systems (Ericsson, KTH, Aachen, Brescia, Pisa, Patras, …)
- Foundation of non-profit ESCHER
  - Interaction with F-22/JSF design review teams
  - Secure Networked Embedded Systems: SCADA systems

SCADA of the Future

- Current SCADA
  - Closed systems, limited coordination, unprotected cyber-infrastructure
  - Local, limited adaptation (parametric), manual control
  - Static, centralized structure
- Future requirements
  - Decentralized, secure open systems (peer-to-peer, mutable hierarchies of operation)
  - Direct support for coordinated control, authority restriction
  - Trusted, automated reconfiguration
    - Isolate drop-outs, limit cascading failure, manage regions under attack
    - Enable re-entry upon recovery to normal operation
    - Coordinate degraded, recovery modes
  - Diagnosis, mitigation of combined physical, cyber attack
  - Advanced SCADA for productivity, market stability, manageability

Secure Network Embedded Systems

Embedded Software prevalent in all critical infrastructures. Critical to high confidence embedded software are open source techniques for

- Automated Design, Verification and Validation
  - Verified design in a formal, mathematical sense
  - Validated design in an engineering sense
  - Certifiable design to allow for regulatory and certification input
- High Confidence Systems
  - Narrow waisted middleware
    - Trusted abstractions, limited interfaces
    - Algorithms and protocols for secure, distributed coordination and control
  - Security and composable operating systems
  - Tamper Proof Software
- Generative Programming
- Intelligent Microsystems: infrastructure of the future with security codesign with hardware and software.

Layers of Secure Network Embedded Systems

- Physical Layer
  - Attacks: jamming, tampering
  - Defenses: spread spectrum, priority messages, lower duty cycle, region mapping, mode change, tamper proofing, hiding.
- Link Layer
  - Attacks: collision, exhaustion, unfairness
  - Defenses: error correcting code, rate limitation, small frames

Layers of Secure Network Embedded Systems

- Network and Routing Layer
  - Attacks: neglect and greed, homing, misdirection, black holes
  - Defenses: redundancy, probing, encryption, egress filtering, authorization, monitoring, authorization, monitoring, redundancy
- Transport Layer
  - Attacks: flooding, desynchronization
  - Defenses: client puzzles, authentication
- Embedded System/Application Layer
  - Attacks: insider misuse, unprotected operations, resource overload attacks, distributed service disruption
  - Defenses: authority management (operator authentication, role-based control authorization), secure resource management, secure application distribution services

Adaptive Networked Infrastructure

Core partners: Berkeley (lead), Cornell, Vanderbilt
Outreach partners: San Jose State, Smith, Tennessee Tech, UC Davis, UC Merced.

Principal investigator: Edward A. Lee, Professor, EECS, UC Berkeley, eal@eecs.berkeley.edu

• *Approach*: Engineering methods for integrating computer-controlled, networked sensors and actuators in societal-scale infrastructure systems.

• *Enabling technologies*: wireless networked embedded systems with sensors and actuators

• *Resource management test beds*:
- electric power
- transportation
- water

- *Deliverables*: Engineering Methods, Models, and Toolkits for:
- design and analysis of systems with embedded computing
- computation integrated with the physical world
- analysis of control dynamics with software and network behavior
- programming the ensemble, not the computer
- computer-integrated systems oriented engineering curricula

Strategic Framework