Foundations of Hybrid and Embedded Software and Systems: Project Overview

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

Chess Review
November 21, 2005
Berkeley, CA
Ruzena Bajcsy, Ras Bodik, Bela Bollobas, Gautam Biswas, Tom Henzinger, Kenneth Frampton, Gabor Karsai, Kurt Keutzer, John Koo, Edward Lee, George Necula, Alberto Sangiovanni Vincentelli, Shankar Sastry, Janos Sztipanovits, Claire Tomlin, 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.

The fate of computers lacking interaction with physical processes.

"ITR Project Overview", S. Sastry
Hybrid and Embedded Software: Problem for Whom and What have we done

- **DoD** (from avionics to micro-robots)
  - Essential source of functionality/superiority
  - UAV flight control, F-22/F-35 avionics, UAR
- **Automotive** (drive-by-wire(less)?)
  - Key competitive element:
  - Studies for Ford, GM, Toyota, Siemens
- **Ubiquitous Computing Devices** (from mobile phones to TVs to sensor webs)
  - Networked Embedded Systems
  - Several generations of Sensor Webs/Motes
- **Plant Automation Systems**
  - SCADA/DCS in Critical Infrastructure Protection
  - Closing the loop around sensor webs
Some Applications Addressed

Automotive

Avionics: UAVs

Systems Biology

Networked Embedded Systems

"ITR Project Overview", S. Sastry
More Applications

Shooter Localization using Vanderbilt Algorithms

Conflict Detection and Resolution for Manned and Unmanned Aircraft

"ITR Project Overview", S. Sastry
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
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
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.

... 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
- Approximate Solutions to H-J equations and controller synthesis
- Autonomous rotorcraft
- Automotive systems design
- Networked Embedded Systems
- Systems Biology
Tool Development Efforts

- GME
- GReAT
- DESERT
- Fresco
- Giotto/Massaccio
- Ptolemy
- HyVisual
- Metropolis
- Hyper
- MESCAL
NSF ITR Organization

- **PI:** Shankar Sastry
- **coPIs:** Tom Henzinger, Edward Lee, Alberto Sangiovanni-Vincentelli, Janos Sztipanovits
- **Participating Institutions:** UCB, Vanderbilt, Memphis
- **Five Thrusts:**
  - Hybrid Systems Theory (Tomlin/Henzinger)
  - Model-Based Design (Sztipanovits)
  - Advanced Tool Architectures (Lee)
  - Applications: automotive (ASV), aerospace (Tomlin/Sastry), biology (Tomlin)
  - Education and Outreach (Karsai, Lee, Varaiya)
  - Weekly seminar series
  - Ptolemy workshop May 9th, 2003, April 27th, 2004,
  - NEST + CHESS Workshop May 9th, 2003
  - BEARS Open House, February 27th, 2004, February 25th, 2005
Thrust 1 Hybrid Systems

• **Deep Compositionality**
  - Assume Guarantee Reasoning for Hybrid Systems
  - Practical Hybrid System Modeling Language
  - Interface Theory for hybrid components (Chakrabarty)

• **Robust Hybrid Systems**
  - Bundle Properties for hybrid systems
  - Topologies for hybrid systems (Ames)
  - Stochastic hybrid systems (Abate, Amin)

• **Computational hybrid systems**
  - Approximation techniques for H-J equations (Mitchell, Bayen)
  - Synthesis of safe and live controllers for hybrid systems

• **Phase Transitions and Network Embedded Systems**
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
  - Semantic Anchoring
  - 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
Software Releases

VisualSense
Visual editor and simulator for wireless sensor network systems

Ptolemy II

HyVisual - Hybrid System Visual Modeler

Viptos - Visual interface between Ptolemy and TinyOS

Metropolis: Design Environment for Heterogeneous Systems

"ITR Project Overview", S. Sastry
The Hyper toolbox (in development)

• Inspired by hybrid systems domain
• Consider Interchange Format Philosophy:
  - For all models which could be built in Tool₁ or Tool₂ (i.e., as defined by $A_1$) there must exist a translator to/from an Interchange Format
• Alternative philosophy:
  - For a model, $m$, built in Tool₁ or Tool₂, this model may be translated to the other tool if the semantics used by $m$ are an intersecting subset of the semantics $S_1 \cap S_2$.

$\text{Tool}_1 = \langle C_1, A_1, S_1, M_{s1}, M_{c1} \rangle$

$C = \text{Concrete Syntax}$, $A = \text{Abstract Syntax}$, $S = \text{Semantics}$
$M_s = \text{Semantic Mapping}$, $M_c = \text{Concrete Syntax Mapping}$
The Hyper toolbox (in development)

• Examine semantics used by a model to determine compatibility
• This provides several potential uses
  - Produce Tool\(_{1\cap2}\) after user request for models compatible across Tool\(_1\), Tool\(_2\)
  - Check to see if model \(m_3\), produced in Tool\(_{1\cap3}\) is compatible with Tool\(_2\)
  - Produce Tool\(_{\text{simulate}\cap\text{verify}}\) when capability is more important than specific semantics

• Implementation strategy
  - Strong typing, metamodeling of type structures
  - Previous Chess work in operational semantics and Interchange Formats
Thrust IV: Applications

- **Embedded Control Systems**
  - Avionics: F-22, F-35, UAV flight control, Open Control Platform
  - Vehtronics: Engine control, Braking control, architectures
- **Embedded Systems for National/Homeland Security**
  - Air Traffic Control; Smart Walls, Sector Control
  - UAVs: flight control, autonomous navigation, landing
- **Networks of Distributed Sensors and Networked Embedded Systems**
- **Stochastic Hybrid Systems in Systems Biology**
- **Hybrid Models in Structural Engineering**
  - Active Noise Control
  - Vibration damping of complex structures
Antibiotic biosynthesis in *Bacillus subtilis*

**Diagram:**
- SpaK, SpaR, SpaT, SpaC, SpaE, SpaF, SpaI, SpaS, mature subtilin
- Signal transduction pathway
- Subtilin precursor
- Modification transport cleavage
- Immunity
- SigH
- SpaRK, SpaS

**States:**
- \( S_1 \) = discrete states (with randomness)
- \( S_2 \) = continuous states

**Symbols:**
- \( \text{spaRK} \)
- \( \text{spaS} \)

**Modeling:**
- Hybrid system

---

"ITR Project:
Project to Identify..."
Planar cell polarity in *Drosophila*

- Simulations
- Parameters estimation
- Study of mutants
Thrust V: Education and Outreach

- **Curriculum Development for MSS**
  - Lower Division
  - Upper Division
  - Graduate Courses
- **Undergrad Course Insertion and Transfer**
  - New courses for partner institutions (workshops held March 1st 2003, Summer 2004), ABET requirements
  - Introduction of new undergrad control course at upper division level by embedded control course coordinated 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, 6+4 students in Summer 05)
- **Graduate Courses**
  - EECS 249 Design of Embedded Systems: Models, Validation, and Synthesis
  - EECS 290N Concurrent Models of Computation for Embedded Software
  - Vanderbilt EECE 395 / EECS 291E/ME 290S Hybrid Systems
SUPERB: Projects Overview

Camera Networks
- Lana Carnel
- Visual Segmentation Code
- Camera and Network Modeling
- Murphy Gant

Hybrid Systems Theory and Modeling
- Derivation of Equations of Motion
- Bobby Gregg

Modeling/Analysis On-Chip Networks
- Reinaldo Romero
- Tradeoff Analysis in Design

Zeno in Communications Networks
- Shams Karimkhan

"ITR Project Overview", S. Sastry
SIPHER Student Projects

- Process Control using Model-based Tools
  - Karlston Martin
  - Shantell Hinton
- Embedded Controllers for Vibration Control
  - Alicia Vaden
- Sensor Networks - Camera Control
  - Chanel Mitchell
  - Omar Abdul-Ali
- Autonomous Robot Control
  - Lauren Mitchell
  - Sarah Francis
- Embedded Software Tools
  - Ryan Thibodeaux
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 EU-IST program in network embedded systems (Ericsson, KTH, Aachen, Brescia, Pisa, Patras, ...)
  - EU-US Embedded Systems meeting, Paris, July 2005 organized by Sztipanovits

• **Foundation of non-profit ESCHER**
  - Interaction with F-22/JSF design review teams
  - Secure Networked Embedded Systems: TinyOS, Tiny DB, etc.
  - Bio-SPICE repository
The Embedded Open Control Platform (EOCP)

OCP provides an insulation layer between software-based control algorithms and the testbed/platform/OS on which they run.

Development (Algorithm) by Technology Developer

Core OCP

Control Algorithm(s)

1 2 3 4 ... Nc

Controls API

Communications Layer

Platform Independent Testbed Configurable

Platform

1 ... Np

Testbed

1 ... Nr

Deployment Layer by OCP Developer

Core OCP

Configured for OCP by OCP Developer

Platform

Laptop Computer

Desktop Computer

PC-104 Stack

AC160 Hummingbird

F-15 Eagle

Yamaha R-Max

T-33 Trainer Jet

SMART Bat

"No existing platform of appropriate form-factor/weight"

Project Cost

$67M

$67M

$43M

$123k

$75k

$3k

"ITR Project Overview", S.
Development, Deployment, and Demystification

Objective: Separate development and deployment platforms, provide out-of-the-box self-configuration scripts for new dev/deploy platforms

Host Platform
- Host Compiler
- Binary Libraries
- Header Files
- Communications Layer

Development Platform
- Binary Libraries (Target)
- Header Files (Target)
- Abstract (CORBA) Communications Layer
- Host-Target Crosscompiler

Target Platform
- Binary Libraries
- Header Files
- Communications Layer
  - CORBA Interface Layer
  - OCP Executable Layer

Benefit: Allow OCP developers to not necessarily be Unix Developers

"ITR Project Overview", S. Sastry
Outreach Continued

• Three NITRD-HCSS studies
  - Aviation Safety and Certification: Planning Meeting Seattle Nov 9, 10\textsuperscript{th} 2005. Tomlin main study leader main meeting
  - High Confidence SCADA systems: Planning meeting, Washington, DC March 21-23, 2006

• NSF-EU workshop to be held in Helsinki, June 2006
Network Embedded Systems: A Progress Report

Edited and presented by Shankar Sastry
UC Berkeley

Chess Review
November 21, 2005
Berkeley, CA
Bell's Law -
new computer class per 10 years

- Enabled by technological opportunities
- Smaller, more numerous and more intimately connected
- Ushers in a new kind of application
- Ultimately used in many ways not previously imagined

Number Crunching
Data Storage

productivity
interactive

streaming
information
to/from physical
world

"ITR Project Overview", S. Sastry
Instrumenting the world
The Sensor Network Challenge

Monitoring & Managing Spaces and Things

Miniature, low-power connections to the physical world

"ITR Project Overview", S. Sastry
Traditional Systems

- Well established layers of abstractions
- Strict boundaries
- Ample resources
- Independent Applications at endpoints communicate pt-pt through routers
- Well attended
by comparison ...

- Highly Constrained resources
  - processing, storage, bandwidth, power
- Applications spread over many small nodes
  - self-organizing Collectives
  - highly integrated with changing environment and network
  - communication is fundamental
- Concurrency intensive in bursts
  - streams of sensor data and network traffic
- Robust
  - inaccessible, critical operation
- Unclear where the boundaries belong
  - even HW/SW will move
# Mote Evolution

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>AT90LS8535</td>
<td>ATmega163</td>
<td>ATmega128</td>
<td>TI MSP430</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program memory (KB)</td>
<td>8</td>
<td>16</td>
<td>128</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAM (KB)</td>
<td>0.5</td>
<td>1</td>
<td>15</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active Power (mW)</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sleep Power (μW)</td>
<td>45</td>
<td>45</td>
<td>75</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wakeup Time (μs)</td>
<td>1000</td>
<td>36</td>
<td>180</td>
<td>180</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonvolatile storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip</td>
<td>24LC256</td>
<td>AT45DB041B</td>
<td>ST M24M01S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection type</td>
<td>I²C</td>
<td>SPI</td>
<td>I²C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size (KB)</td>
<td>32</td>
<td>512</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radio</td>
<td>TR1000</td>
<td>TR1000</td>
<td>CC1000</td>
<td>CC2420</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data rate (kbps)</td>
<td>10</td>
<td>40</td>
<td>38.4</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modulation type</td>
<td>OOK</td>
<td>ASK</td>
<td>FSK</td>
<td>O-QPSK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive Power (mW)</td>
<td>9</td>
<td>12</td>
<td>29</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmit Power at 0dBm (mW)</td>
<td>36</td>
<td>36</td>
<td>42</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Consumption</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum Operation (V)</td>
<td>2.7</td>
<td>2.7</td>
<td>2.7</td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Active Power (mW)</td>
<td>24</td>
<td>27</td>
<td>44</td>
<td>89</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Programming and Sensor Interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expansion</td>
<td>none</td>
<td>51-pin</td>
<td>51-pin</td>
<td>none</td>
<td>51-pin</td>
<td>19-pin</td>
<td>51-pin</td>
<td>10-pin</td>
</tr>
<tr>
<td>Communication</td>
<td>IEEE 1284 (programming) and RS232 (requires additional hardware)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated Sensors</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Evolution of Motes Continued

Dot motes, MICA motes, smart dust, Telos motes
NEST Final Experiment: Sensor Node

**Telos B mote**
- 8MHz TI MSP430 microcontroller
- RAM: 10kB; Flash: 48kB
- Chipcon CC2420 Radio: 250kbps, 2.4GHz, IEEE 802.15.4 standard compliant
- Radio range of up to 125 meters

**Trio Sensor Board**
- Features a microphone, a piezoelectric buzzer, x-y axis magnetometers, and four passive infrared (PIR) motion sensors
- Solar-power charging circuitry

**Trio Node**
Multiple Target Tracking

• Goal
  – Track an unknown number of multiple targets using a sensor network of binary sensors without classification information
  – Coordinate multiple pursuers to chase and capture multiple evaders in minimum time using a sensor network
    • Done in simulation due to physical and time constraints
Simulation: Multiple-Target Tracking & Pursuit Evasion Games in Sensor Networks

Detection

Fusion

Estimated Tracks and Pursuer-to-evader Assignment

"ITR Project Overview", S. Sastry

Chess Review, Nov. 21, 2005
Overall Architecture

Multi-agent coordination algorithm

- Minimize time to capture all evaders
- Robust Minimum Time Control (MTC)
NEST Final Experiment: System

• Software
  – TinyOS
  – Deluge
    • Network reprogramming
  – Drip and Drain (Routing Layer)
    • Drip: disseminate commands
    • Drain: collect data
  – DetectionEvent
    • Multi-moded event generator
  – Multi-sensor fusion and multiple-target tracking algorithms
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
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