Control and Customization of Mobile Systems Under Dynamic Constraints: A Proposal for Collaboration in ARTIST with VERIMAG

Magnus Egerstedt
Sung-Kyu Lim
Krishna Palem
Sudhakar Yalamanchili

Center for Research on Embedded Systems and Technology
School of Electrical and Computer Engineering
Georgia Institute of Technology
http://www.crest.gatech.edu
From Specifications to HW/SW Architectures

- Embedded Customization Design Flow:

  Requirements (time, power, memory)
  
  Executable specifications
  
  Code (e.g. TimeC)
  
  Compiler
  
  Machine code (customized instr. set)
  
  Architecture description
  
  HW synthesis


NYU (Goldberg, Pnueli)

GaTech
From Specifications to HW/SW Architectures

- Given high-level specifications (including real-time, power, and memory constraints) can the customization process be automated?
  - Design automation via innovative, architecture-level design space exploration
  - Using compiler optimizations as a tool for designing workload-specific architectures
  - Integration with front-end system specification & verification flows and back-end HW/SW design activities

- For mobile, reactive systems what are the special constraints associated with multi-modal control strategies?

- Can dynamical constraints such as memory and power resources be modeled in a similar manner as temporal constraints?
From Specifications to HW/SW Architectures: An Approach to Collaboration with VERIMAG

- Embedded Customization Design Flow:
  - Requirements (time, power, memory)
  - Executable specifications
  - Code (e.g. time C)

- Tools are needed in order to achieve correctness in each step
- Languages, verification, testing, and modeling?
At each step, the resulting objects must be provably correct in the sense that the required properties are preserved.

Customization tools are necessary for achieving this:
- Formal methods for going from high-level specifications to executable code
- Compiler optimization tools for designing/exploring the architectures
- VLSI tools for HW synthesis

It is of key importance that the initial specifications are expressive enough to capture not just timing considerations, but also other physical design constraints.
High-Level Specifications

- We use LSCs (Live Sequence Charts) for modeling temporal orderings.
- From these, Timed Automata can be obtained as executable specifications.
- However, other physical constraints (e.g. memory and power) must be specifiable as well.
- Calls for an extended syntax in the LSCs as well as the introduction of Hybrid Automata rather than Timed Automata at the next specification level.
- Team: Egerstedt, Goldberg, Pnueli.
High-Level Specifications: Challenges

- Upper bounds on the memory and power consumption available to each component during a given part of the task can be modeled by differential equations.
- As the system transitions to different modes of operation, new constraints are in force (e.g. cell phones during dialing, calling, and idling modes).
- This model allows for dynamic, physical implementation constraints to be incorporated already at the high-level specifications.
- When the system to be customized is a mobile, reactive system, this provides a unified treatment of the constraints and physical states of the system (described via controlled differential equations).
Customizing Platforms at CREST

- Requirements Specification
- Traceback
- Executable Specification
- System Analysis & Verification
- Constraints on Architecture
- Constraints, Application Code & Architecture Specification
- Architecture Design
- HDL and System Specifications
- System Implementation

Faculty:
- K. Palem
- S. Yalamanchili
- V. Mooney
- S. K. Lim
Customization via Design Space Exploration

- **Fix program**
- **User specifies design constraints**
- **Optimizations and exploration tools search design space**
- **Best design is chosen**

Design space exploration for concurrent optimization an architecture and associated compiler optimizations!
Architecture Design Methodology

Trimaran Infrastructure

Application Code

Architecture Specification

Compiler Specification

Compiled code

Simulation & Evaluation

Constraints (cost, power, throughput, etc)

Software

• Architecture Specification
• Compiler Specification
• Run-time Specification

Customization Decisions

Intermediate Representation

ADEPT

Performance Results

Architecture Cost Models

Genesys

Empirical Models

October 2003
Once code has been produced from the specifications:

- Compiler optimization tools can be used for determining the workload specific hardware, software architectures
- Design space exploration allows for optimization based hardware synthesis
- Compilers provide a unified framework for exploring and finding architectures as well as compile to the appropriate architecture (TRIMARAN)
- Customized instruction sets should be selected

Questions

- Highly accurate cost modeling in the design space exploration?
- Decomposition of the computations in such a way that the total energy required to perform an algorithm with specified bounds will be met by the best QoS possible?
- Empirical power models of individual instructions and instruction sequences in terms of supply voltage, technology parameters such as threshold voltage, instruction delay / timing / clock speed and input data values (best-, worst-, avg-cases)?
Reactive, Mobile Systems (SPIDER)

- For mobile systems (e.g. autonomous mobile robots) the executions can be characterized by strings of control modes (so-called hybrid words)

\[ S = \sigma_1 \sigma_2 \sigma_1 \]

or

\[ S \in (\sigma_1 \sigma_2)^* \sigma_1 \]

- Each mode consists of a reference to a particular control law, the interrupt condition, and a specification of the temporal sensitivity of that particular mode (unstable modes, safety critical switches, …)

- The interrupts together with the temporal sensitivities can be modeled in LSCs.

**SPIDER:** Specifiable and Provable Integrated Design of Embedded systems for Robotics
In Summary: Collaboration Opportunities with VERIMAG

- Tools are needed in order to achieve correctness in each step
- Characterizations of specifications from multi-modal control design?
  - Critical modes and interrupts?
  - Power, memory, and real-time constraints?
- “Hybrid” LSCs to Hybrid Automata?
  - Equivalence?
  - Automatic transition or post model validation?
- From Hybrid Automata to code?
  - Timed languages?
  - Languages for control tasks?
- Languages, verification, testing, and modeling?
- Design space exploration algorithms?