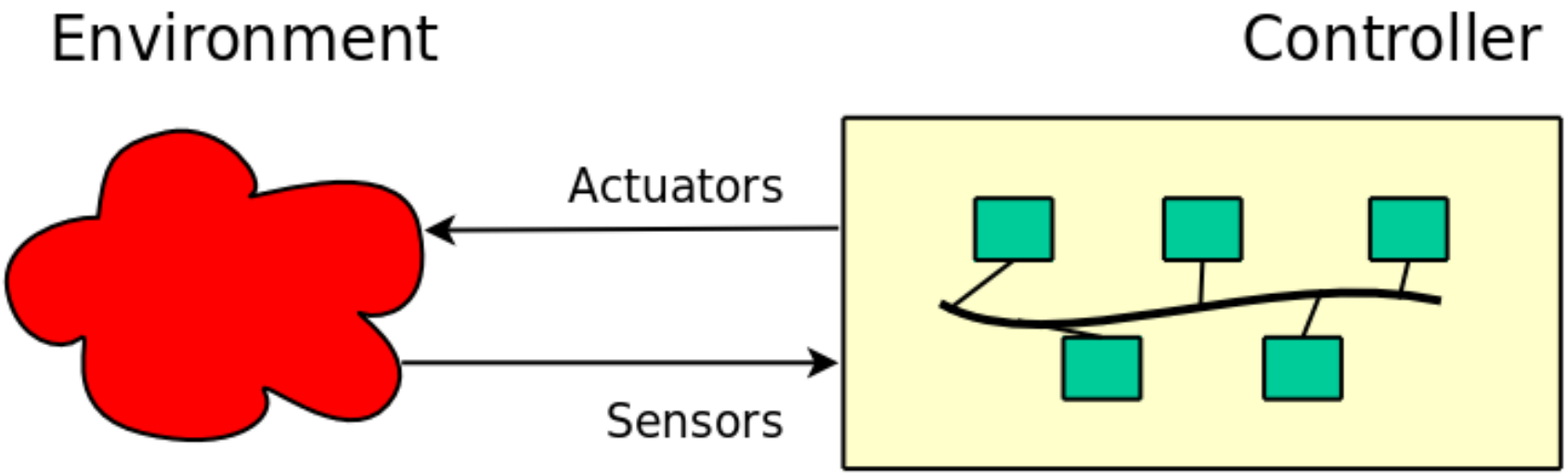
Embedded Systems Specification and Design

David Kendall

Introduction

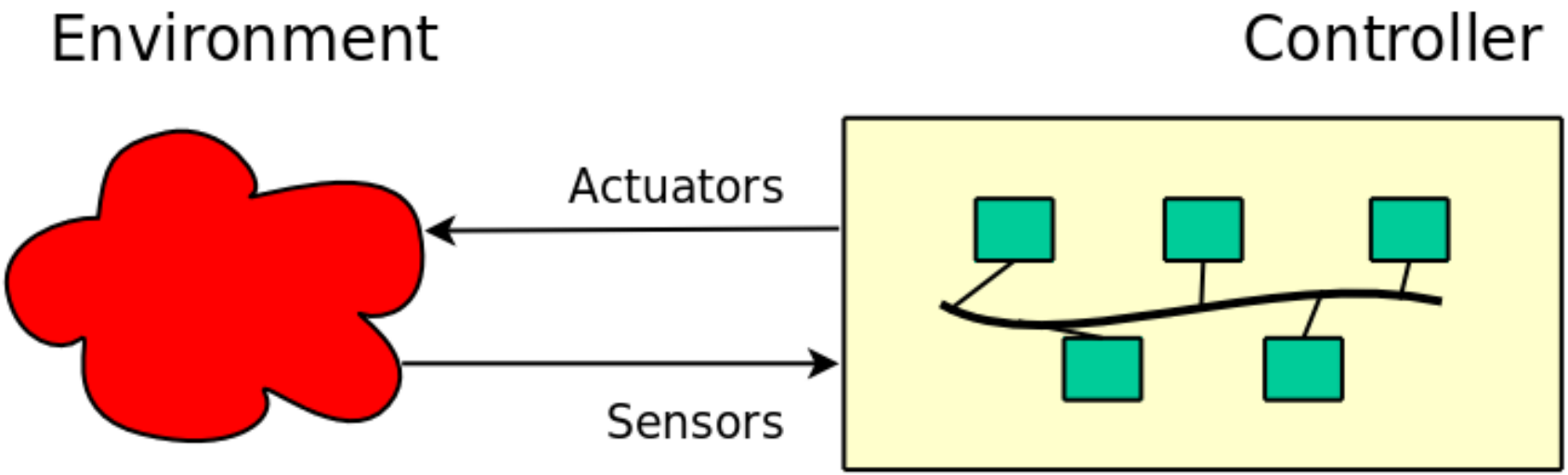
- What is an embedded system?
- What do we mean by specification and design?
- What are the topics covered in this module?
- What learning resources are available?

What is an embedded system?



- Computing system *embedded* in physical environment for purpose of monitoring and/or control

Typical characteristics of embedded system



- *concurrent* – composed of multi-tasking and/or distributed processes
- *communicating* processes are specialized to perform some specific task
- *real-time* – timing requirements established by the environment
- *resource-constrained* – limited resources: processing, memory, peripherals, power, . . .

Embedded Systems Development

- Requirements specification
  - Physical system modelling
  - Control law design
- Embedded control system design
  - Model-Driven Design (MDD)
- Implementation and testing

Model-Driven Design

- Design by creating models of (important aspects of) the system to be developed
  - system architecture
  - synchronization
  - communication protocols
  - real-time behaviour
- Analyse the models to check if they have desired properties
- Iterate
  - Refine models
  - Further analysis

Formal Methods

- Applied discrete mathematics: set theory, logic
- Languages, analysis techniques, tools
- Applicable to development of computer hardware and software
- Useful for
  - Description: e.g., specifying requirements
    precise, concise, unambiguous
  - Analysis: e.g., demonstrating program function implements design
- Rigour of mathematics helps to develop convincing argument using calculation
- Reduces reliance on human intuition and judgement

The essence of a formal method

- Informally

The system satisfies its specification

- Formally

Sys $\models$ Spec

- A formal language has well-defined:
  - syntax,
  - semantics, and
  - rules for reasoning about relationship between expressions

Model Checking

- Pragmatic engineering approach to use of formal methods in embedded systems design.
- Sys – finite state transition model
- Spec – propositions of temporal logic
- $\models$ – model check

Temporal Logic

Amir Pnueli

A. Pnueli, *The Temporal Logic of Programs*, 18th Annual Symposium on Foundations of Computer Science, pp. 46–57, 1977

Turing Award Winner 1996

Model-checking

Ed Clarke

E. Allen Emerson

Joseph Sifakis

Edmund M. Clarke, E. Allen Emerson, *Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic*, Proceedings of Workshop on Logic of Programs, pp. 52-71, 1981.

J.P. Queille and J. Sifakis, *Specification and verification of concurrent systems in CESAR*, Proceedings of International Symposium on Programming, LNCS 137, pp. 337–351, 1982

Turing Award Winners 2007

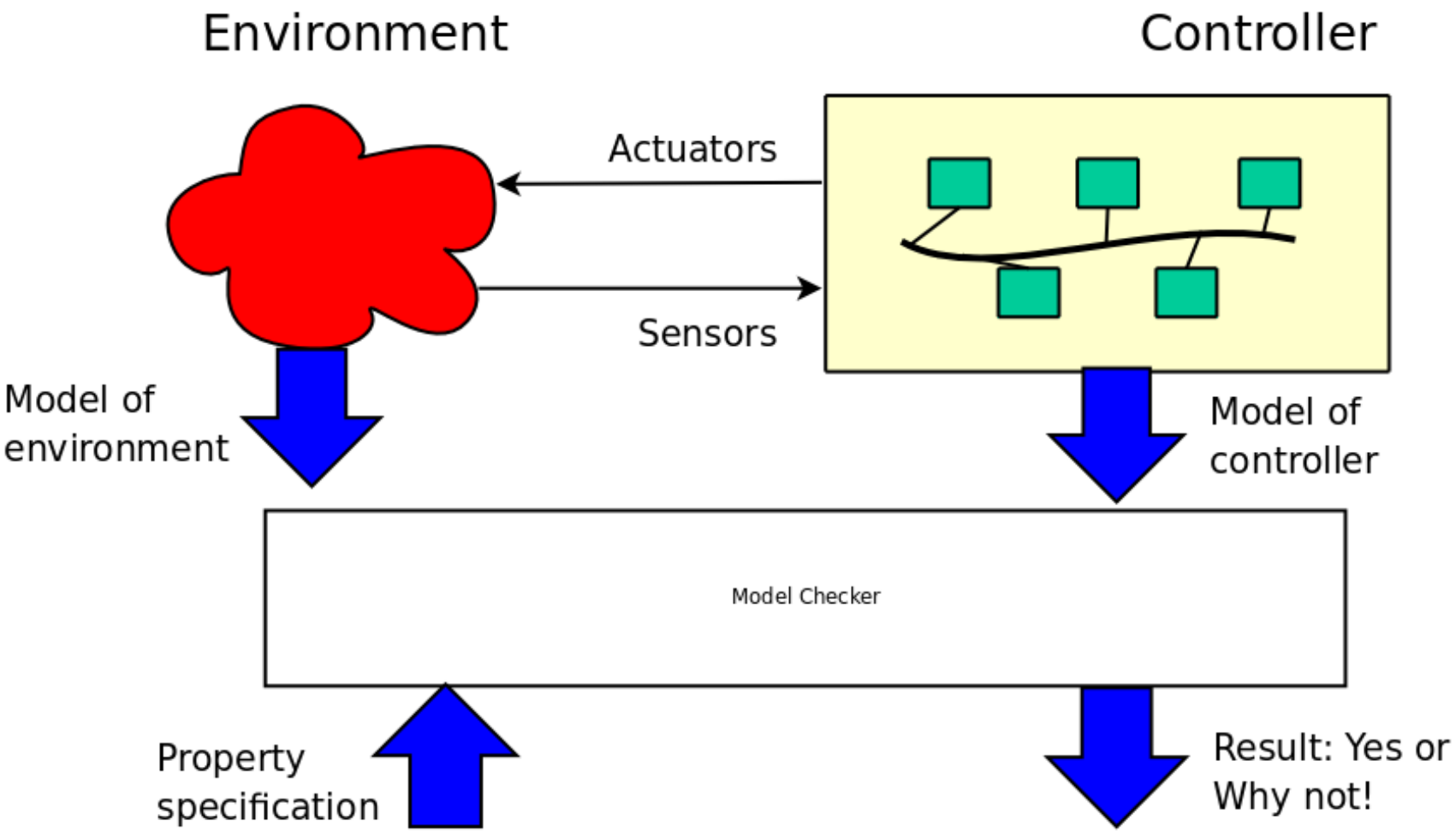
A language-theoretic formulation of model-checking

Moshe Vardi

Pierre Wolper

M. Vardi, P. Wolper, *An automata-theoretic approach to automatic program verification*, Proceedings of the First Symposium on Logic in Computer Science, pp. 322–331, 1986

Model Checking for Embedded Systems



Topics and Resources

- Theory and practice of
  - SPIN – concurrency and communication
  - UPPAAL – real-time
  - PRISM – probabilistic modelling and analysis

http://hesabu.net/cm0604/

Summary

- Model-driven design approach
- Employ formal methods to model and analyse systems
- Detect bugs early in design lifecycle
- Model-checking can
  - Demonstrate that a model has required properties
  - Be used in practice by engineers with only limited mathematical knowledge
- SPIN is a model checker that is freely available, widely used and highly respected