Guidelines for a European Graduate Curriculum on Embedded Software and Systems

Paul Caspi

VERIMAG (CNRS)

http://www-verimag.imag.fr

October 2003

Motivation and History

Embedded systems have an ever growing economic importance

The ARTIST Noe is focused on embedded software and systems

The question of curricula for embedded systems has not yet been clearly addressed

Motivation and History

Embedded systems have an ever growing economic importance

The ARTIST Noe is focused on embedded software and systems

The question of curricula for embedded systems has not yet been clearly addressed

⇒ A work-package has been devoted to the subject

The first deliverable is visible at:

http://www.artist-embedded.org/Education/

(*hypertext document with links to course and curricula information*)

Overview

⇒ Motivation and history of the education document

⇒ Limitations and principles

⇒ Main recommendations

⇒ How to proceed ?

Limitations and Principles

The composition of the Artist consortium

⇒ Diversity of education

⇒ Diversity of European education systems and styles

⇒ Diversity of the embedded system domain

Diversity of Education

Present day students will be active for the next forty years. During this period, techniques will drastically evolve and this evolution will be dealt with by continuous training :

- seminars, in-house training, tool vendor training, . . .

Diversity of Education

Present day students will be active for the next forty years. During this period, techniques will drastically evolve and this evolution will be dealt with by continuous training :

- seminars, in-house training, tool vendor training, . . .

This is not likely to provide bases for a true understanding of the domain

University is the time where foundations should be laid down

Diversity of European Education

- Despite the Bologna Declaration, there is still a large diversity of education systems in Europe

  e.g. French engineering schools

- and a diversity of styles:

  inductive : from practice to theory

  deductive : from theory to practice

$\Rightarrow$ It is difficult to propose precise courses and curricula

We rather intend to define large bodies of knowledge that should be part of curricula

Diversity of the Embedded System Domain

- diversity of actors :

  avionics, space, ground transportation, nuclear, automotive, control, telecoms, consumer electronics, . . .

  most of these actors have their own education systems

- diversity of practices and implementations

  hardware, software, control-based design tools, software-based design tools, synchronous, asynchronous, time-triggered, event-triggered . . .

Diversity of the Embedded System Domain

- the variety of design choices is poorly exploited
- poor mobility between application domains
- fragmentation of research

Computer Science

What can be the role of Computer Science in this landscape ?

Computer Science

What can be the role of Computer Science in this landscape ?

- Address fundamental aspects:

  design methods, algorithmics, semantics, verification,...

- Provide unification :
  - compare different approaches
  - find commonalities between application domains

Computer Science

What can be the role of Computer Science in this landscape ?

- Address fundamental aspects:

  design methods, algorithmics, semantics, verification,...

- Provide unification :
  - compare different approaches
  - find commonalities between application domains

but this requires being more aware of application domains

Recommendations

- ⇒ Control and signal processing
- ⇒ Computing theory
- ⇒ Real-time
- ⇒ Distributed systems
- ⇒ Evaluation and optimisation of extra-functional properties
- ⇒ System engineering and architecture
- ⇒ Practice

Control and Signal Processing

- Many embedded systems deal with the control of physical environments and knowing how to model and reason about them is important
- Overall properties of closed-loop systems such as stability are fundamental
- Control-based design tools like Simulink are de facto standards in many application domains

  (*avionics, automotive, signal processing*)

Control and Signal Processing

- Many embedded systems deal with the control of physical environments and knowing how to model and reason about them is important
- Overall properties of closed-loop systems such as stability are fundamental
- Control-based design tools like Simulink are de facto standards in many application domains

  (*avionics, automotive, signal processing*)

hint: try to jointly address continuous and discrete event control

Computing Theory

- It is possible to formally reason about computer behaviours as with physical environments (semantics)
- Computers are not omnipotent (computability, complexity)
- Formal verification tools and methods are useful for designing safer systems

Computing Theory

- It is possible to formally reason about computer behaviours as with physical environments (semantics)
- Computers are not omnipotent (computability, complexity)
- Formal verification tools and methods are useful for designing safer systems

Having control and computer theory in the same curriculum can lead to interesting convergences

Real-Time Computing

Is core knowledge for embedded systems

Should try to address the questions:

- when and why use compiled concurrency (synchronous language)?
- when and why use interpreted concurrency (tasking, threading)?
- when, why and which scheduling technique?

Real-Time Computing

Is core knowledge for embedded systems

Should try to address the questions:

- when and why use compiled concurrency (synchronous language)?
- when and why use interpreted concurrency (tasking, threading)?
- when, why and which scheduling technique?

Broadening the space of design choices

Distributed Computing

Also a core subject

Distributed algorithmics is a fundamental issue for answering questions such as:

- Can a consensus be reached among fault-free computing units?
- How many faults can be tolerated with a given strategy?
- Is clock synchronisation needed?
- CAN or TTA ?

Also broadening the space of design choices

Evaluation and optimisation

For measuring and evaluating designs:

- Performance
- Dependability
- Power consumption
- ...

Fundamental knowledge in any engineering practice

System Engineering and Architecture

A transverse topic needed in order to

- apply all these theories in a coherent way
- cover the whole life-cycle of products

Quite difficult to teach

Hints:

- Component-based design
- Platform-based design
- taxonomy of applications

Practice

In order to

- apply the theories
- introduce practical issues

Choose subjects that cover many aspects:

- hard and soft real-time,
- distribution and/or fault-tolerance,
- evaluation,
- formal verification, …

Conclusion

What do we provide:

- a higher level point of view that unifies what is currently done,
- a framework for implementation or improvement,
- emphasis on unification:
  - Control and Computer theories
  - Synchronous and asynchronous languages and systems
  - Events and time in control and systems
  - Architecture

Difficult to implement

How to proceed ?

How to improve and disseminate these ideas?

How to proceed ?

How to improve and disseminate these ideas?

- Via a European label for master curricula?

  But what about the autonomy of universities ?

How to proceed ?

How to improve and disseminate these ideas?

- Via a European label for master curricula?

  But what about the autonomy of universities ?

- Promoting discussions and convincing colleagues?

  But this is a slow process

How to proceed ?

How to improve and disseminate these ideas?

- Via a European label for master curricula?

  But what about the autonomy of universities ?

- Promoting discussions and convincing colleagues?

  But this is a slow process

- What else ?

The first deliverable is visible at:

http://www.artist-embedded.org/Education/

Comments can be sent to :

caspi@imag.fr