TMC
Theorie des systemes mobiles et communicants

Uwe.Nestmann@epfl.ch – INR 317

mobile communications ?
mobile communication systems ?
mobile computing ?
mobile computing platforms ?
mobile computation ?
mobile X ?

Oct 25, 2001 12:46

What this course is about

mobile communications ?
mobile communication systems ?
mobile computing ?
mobile computing platforms ?
mobile computation !
mobile X ?

Example: Overtaking Cars

Example: Hand–Over Protocol

Motivation

"Theories usually arise to explain practise"
(Robin Milner, Turing–Award 1991)

Milner 89: "***Communication & Concurrency***"

- *closed* communication protocols
- parallel computation

Milner 99: "***Communicating and Mobile Systems***"

- *open/dynamic* communication protocols
- distributed and *mobile* computation, Internet

Motivation II

The ***practise*** *of concurrent/communicating systems was/is:*

- not a stable craft, nor a well–established science
- not eqipped with standard repertoire of constructions (which do exist to some extent for sequential systems)

Explain it! Theory!

The **"naïve"** way: extend known sequential models

The **foundational** way:

- build a dedicated foundational model without "legacy"
- develop and study its theory
- analyze what arises "naturally" within this model.

Communication Media (I)

Communication Media (II)

What this course really is about

process calculi/algebras (CCS, pi–calculus):
- small formal modeling languages
- few concise powerful language primitives

structural operational semantics:
- how to formally express how computation proceeds
- *formal reasoning* about process systems: *equivalence*
- axiomatic counterparts

type systems as programming disciplines:
- classify patterns of interactive behaviour
- avoiding stupid errors in complex interactions

Syntax and Formal Semantics

Objectives

Why *communicating/mobile/distributed/secure* systems ?
- increasing number of existing systems
- tend to be complex
- tend to be error–prone (even small ones)

Why *calculi* ?
- *compositional*: break **big** things into **several small** things
- *algebraic*: ease mechanical verification
- *syntactic*: provide basis for programming languages

Objectives II

Course preparing for **projects** on:

- conception and implementation of verification tools
- design of "high–level" languages for
  – mobile/distributed systems
  – cryptographic protocols
- modeling and analyzing systems of this kind
- meta–theoretical problems
- ....

Objectives III

after *the course, participants* ***should be able to****:*

- easily read syntax & **operational semantics**
- **compare/evaluate** languages based on their semantics
- **reason formally** about (toy) examples
- "play" with languages
  – **write your own formal semantics** for a given language
  – **argue for it !** (i.e., analyze its properties)

Course material

a web site:
http://lampwww.epfl.ch/cours/tmc01/

an introductory book (by R. Milner):
communicating and mobile systems: the pi–calculus

- 3 copies in the DI–library
- 5 copies "soon" in the LAMP–library

an advanced book (by D. Sangiorgi, D. Walker):
The pi–calculus: A Theory of Mobile Processes

slides (by myself)
papers ...

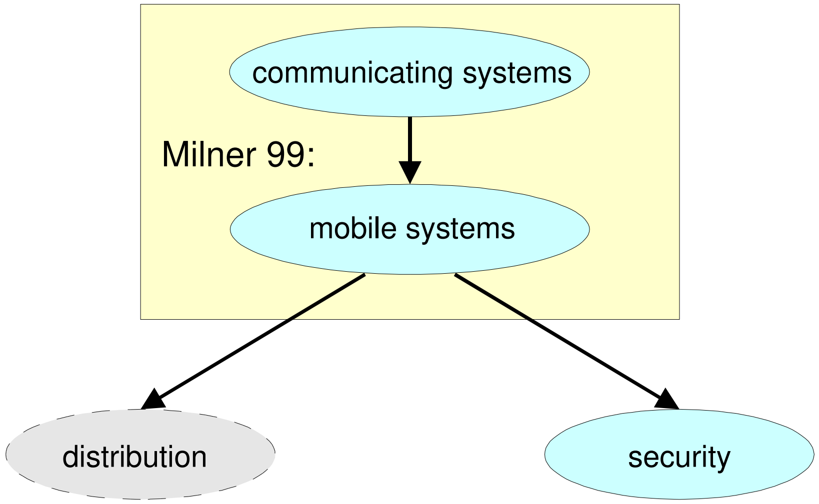
Newsgroup?

forum for online interaction:

- among participants
- between participants and the teacher

epfl.di.cours.tmc

Course Overview



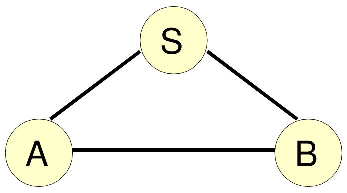
Milner, again ...

"This book (course) is not about design,
for example, it will *not* teach you
how best to design a concurrent operating system.

Instead, we will try to isolate concepts
which allow designers (engineers) to **think clearly**,
not only when analyzing interactive systems
but even when expressing their designs in the first place."

➔ examples, not large but small ones,
illustrating key notions, problems, and solutions

A Simple Cryptographic Protocol



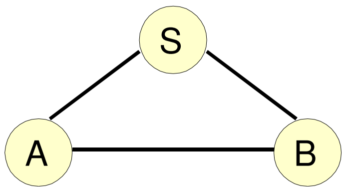
"the wide–mouthed frog"

1) $A \rightarrow S : c_{AB}$ (on c_{AS})
2) $S \rightarrow B : c_{AB}$ (on c_{SB})
3) $A \rightarrow B : M$ (on c_{AB})

How can you argue that such a system is correct ?
What does "correct" mean ?

A Simple Cryptographic Protocol (II)

Formulate specification as a term in the **spi calculus**
Formulate implementation as a term ...
Compare the two, **algebraically**.



$$P = (\nu c_{AS} c_{BS})(\ A \mid S \mid B\ )$$
$$S = \quad c_{AS}?c_{AB}\ ;\ c_{SB}!\ c_{AB}$$
$$A = (\nu c_{AB})(\ c_{AS}!\ c_{AB}\ ;\ c_{AB}!\ M\ )$$
$$B = \quad c_{BS}?c \quad ;\ c\ \ ?\ M$$

Tool Support

verification tools:

- concurrency workbench (CCS)
- **mobility workbench** (pi calculus)
- (security workbench)

calculus–based programming languages:

- (Nomadic) Pict
- join–calculus (Jocaml)
- funnel

Credits

- **oral exam** + **one option** during the semester

- **option 1**: presentation
  of a paper, or
  of some selected material from one of the books

- **option 2**: miniproject
  – modeling and analysis for a chosen problem
  – meta–theoretical proofwork

- "size" of the options
  will depend on the number of participants