Interacting State Machines for Mobility

Thomas A. Kuhn and David von Oheimb
Corporate Technology
Siemens AG
{Thomas.Kuhn|David.von.Oheimb}@siemens.com

11 September 2003

Outline

- Introduction, Motivation and Goals
- ISMs Definition and Semantics
  (Generic ISMs, AmbISMs, dAmbISMs)
- Distributed Accumulation Example
  (Isabelle Representation)
- Applications and Conclusion

Introduction

Present an approach which combines:

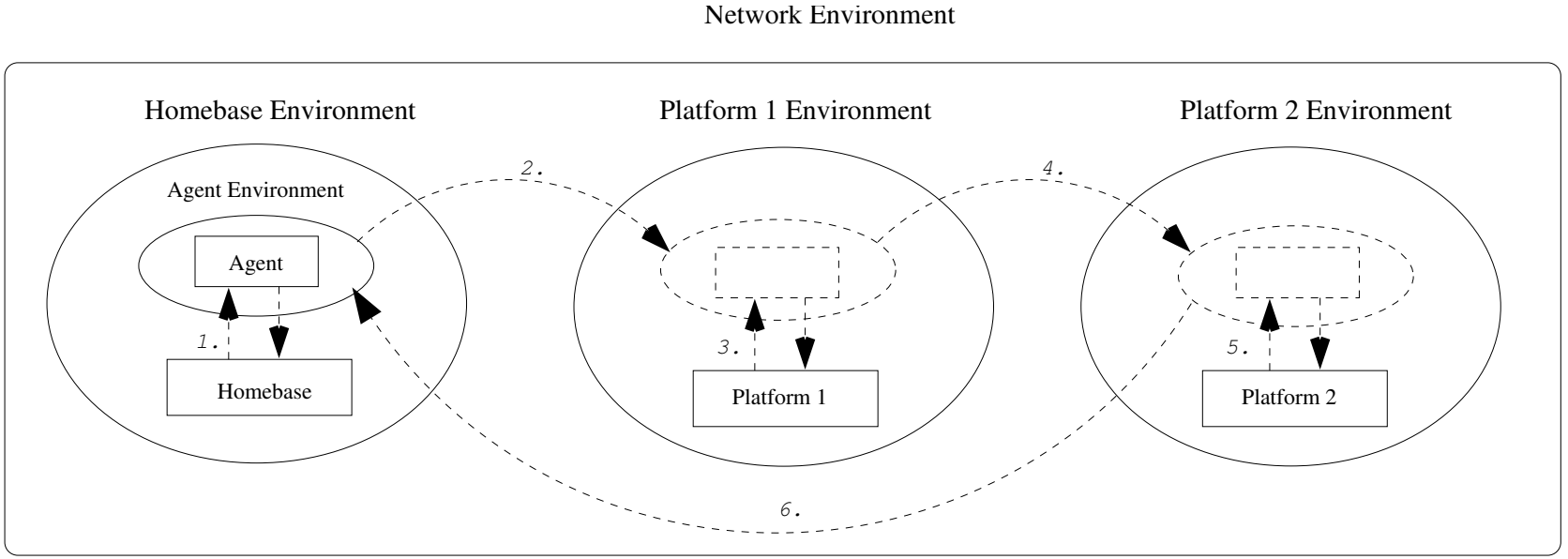
- Boxed ambients
- State-oriented modelling technique

Mobile Security Modelling

- Focus on mobile systems (ISM application domain extension)
- Additional features are hierachical environments, migration and locality constraints on communication
- Proof support through Isabelle/HOL

Motivation (1) - Distributed Accumulation Example

- mobile agent system
- home platform, mobile agent, agent platforms (1 and 2)
- visit agent platforms, get value, count sum, give sum back



Motivation (2) - Approaches

- Pi-Calculus
- I/O Automata
- (Boxed) Ambient Calculus

Problems with the (boxed) ambient calculus:

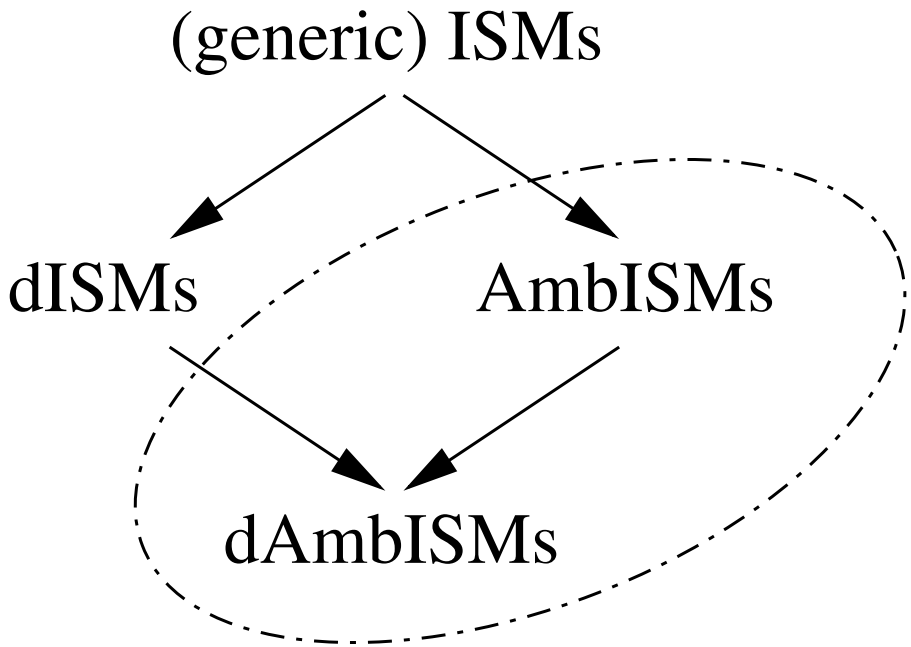
- No direct handling state information
- Cumbersome expression of non-sequential control flow
- No concept of named ports or channels
- Cumbersome expression and verification of local computation

Goals

Properties for formal security analysis:
(particularly suited for industrial use)

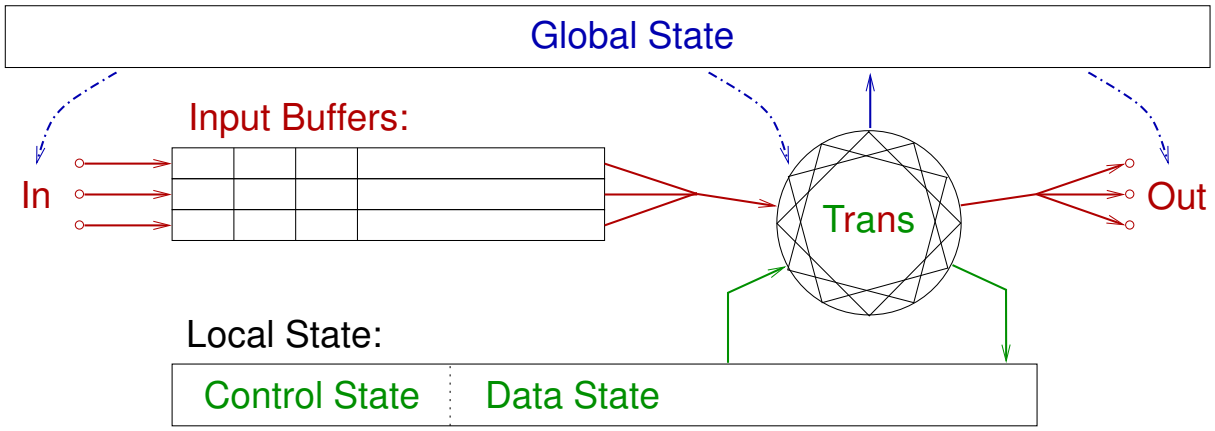
- Expressiveness
- Flexibility
- Simplicity
- Graphical capabilities
- Maturity of the semantics
- Availability of tools

Overview of ISM Hierachy



(Generic) ISM Concepts (1) - Basics

- Automata (single, composed)
- Interactions (buffered communication)
- Ports (receiver, sender)
- Composition in parallel (ISM System)
- States (local and global, initial local state)
- Transitions ($TRANS(C, \Sigma) = P((MSGs \times \Sigma) \times C \times (MSGs \times \Sigma))$



Ambient ISMs (AmbISM) Concepts

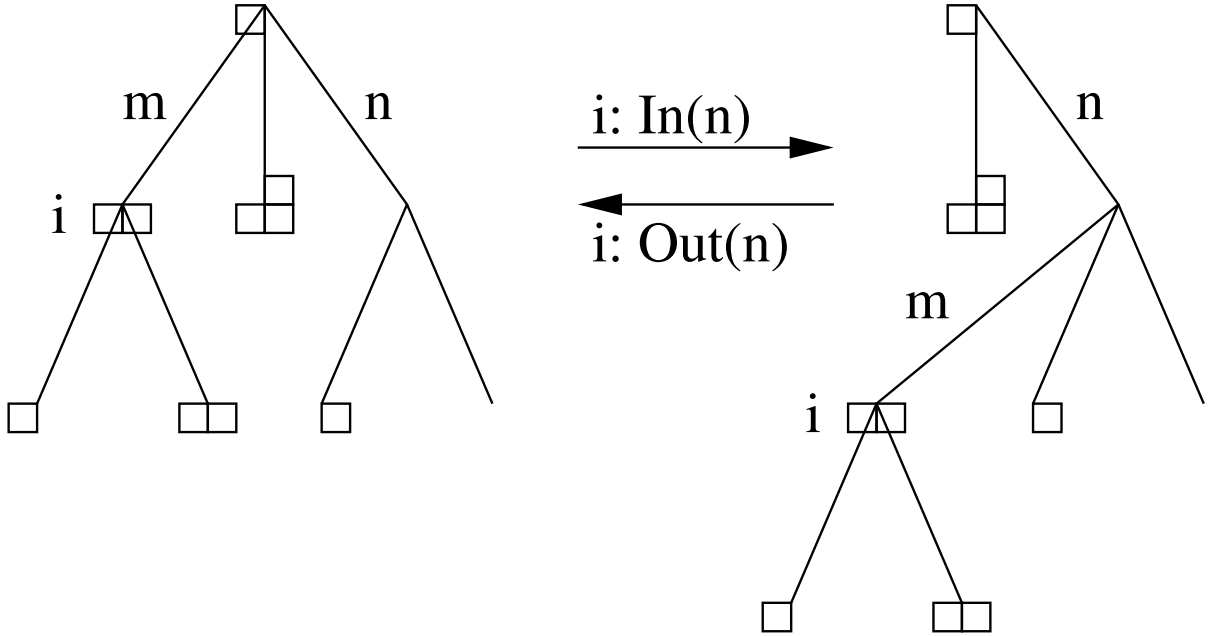
- Ambients (nested administration domains)
- Ambient tree/forest
- Enhanced local communication (parent, child)
- Global states have the form

  $\alpha = (parent(\alpha), home(\alpha))$
- Global transition relation

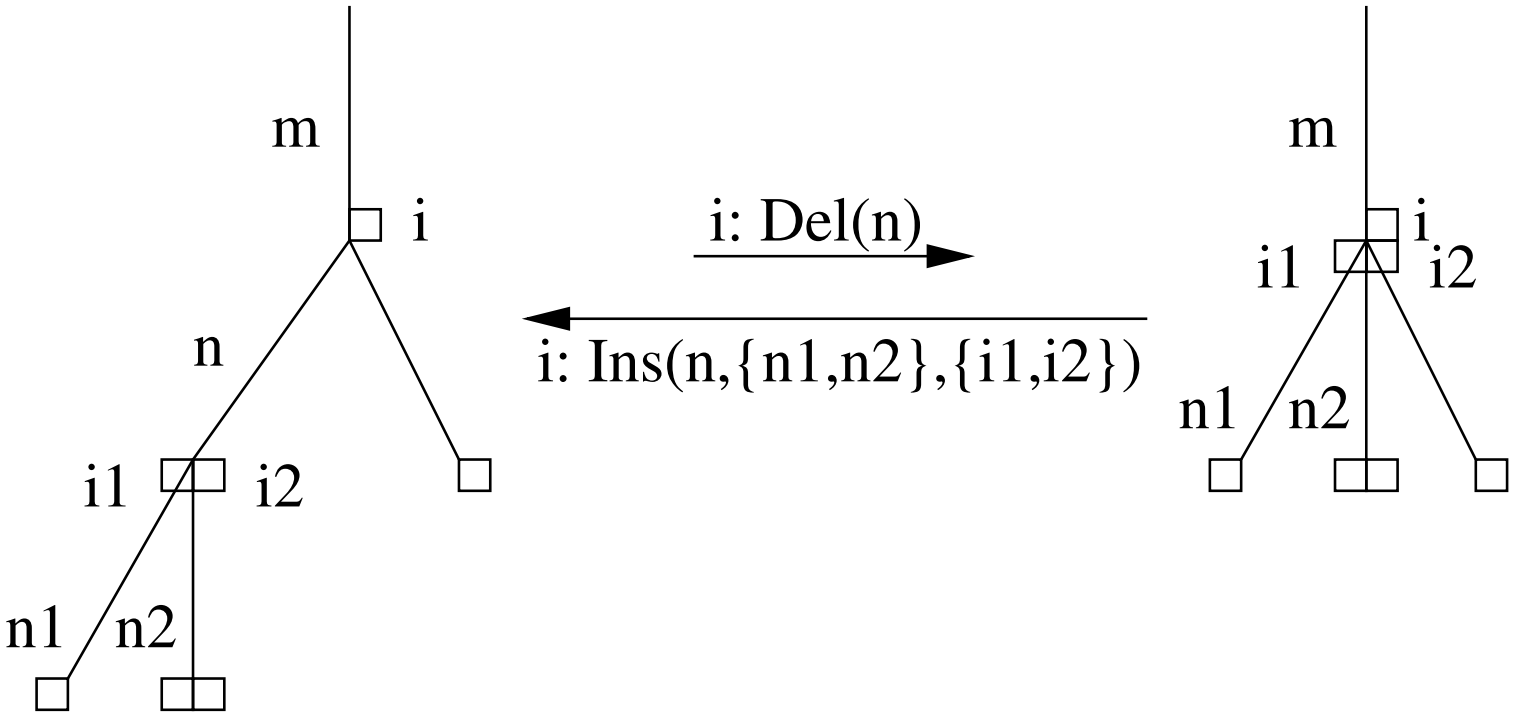
  $\{(\alpha, acmds, \alpha') \mid i \in dom(home(\alpha)) \wedge \ \alpha \xrightarrow{i:acmds}{}^{*} \alpha'\}$
- Commands for tree/forest motification

  (In, Out, Del, Ins, $Assign$)

AmbISMs Semantics (1)



AmbISMs Semantics (2)



Distributed Accumulation (1) - Agent Plattform

ism *AP (N::nat)* =
  ports *"port"*
    inputs *"{Request}"*
    outputs *"UNIV"* — the universal set (of port names)
  messages *"message"*
  commands *"A_cmds"* **default** *"[]"*
  states *state*
    control *"AP_state"* **init** *"Loop"*
  transitions
   request: — the platform gets the reply channel and sends the value

    Loop $\longrightarrow$ *Loop*
    in *"Request" "[Port p]"*
    out *"p" "[Value N]"*

Distributed Accumulation (2)- Further Transitions

- Agent is placed in its environment

```
start:
  Start -> Instruct
  cmd "[Ins AG_amb {} {}, Assign AG AG_amb]]"
```

- Agent gets the route imprinted

```
out "AGData" "[Route [HB_amb, AP_amb 1,
                AP_amb 2, HB_amb]]"
```

- Agent migrates to the next agent platform on the route

```
migrate:
  Migrate -> Decide
  pre "route s = r#rs"
  cmd "[Out (here s), In r]"
  post here := "r", route := "rs"
```

Distributed Accumulation (3) - Verification

Exists a run sequence where a port gives back $Reply = [Value\ 3]?$

constdefs

```
 System :: "(id, (A_cmds, port, message, state) ism) family"
"System  ≡ (λi. case i of AG   ⇒ Agent.ism
                        | HB   ⇒ Homebase.ism
                        | AP n ⇒ if n = 1 then AP1.ism else AP2.ism,
            {AG, HB, AP 1, AP 2})"

 Runs :: "((port, message, (id, ambient) astate × (id ⇒ state))
             conf list) set"
"Runs ≡ Amb_comp_runs System
(|parent = empty(HB_amb↦NW_amb)(AP_amb 1↦NW_amb  )(AP_amb 2↦NW_amb  ),
 home   = empty(HB    ↦HB_amb)(AP     1↦AP_amb 1)(AP     2↦AP_amb 2)|)"
```

theorem `"∃r∈Runs. ∃(b,as,st) ∈ set r. b Reply = [Value 3]"`

Applications of the ISM Approach

- LKW model for Infineon SLE66 smart card processor
- Infineon SLE88 memory management model
- Security models for healthcare applications
- Distributed accumulation (with delegation)
- Authentication protocol modelling for MAP agent plattform

Conclusion

- Expressiveness
  (from very abstract to very fine-grained, refinement, information flow, concepts for expressing mobility)
- Flexibility
  (ease of further enhancements for special purpose system requirements)
- Simplicity
  (compositional state oriented view, designer can disburden from overloaded additional structure)
- Availability of Tools
  (open-source modelling and proof system Isabelle)

Thank you for your attention!

Backup Slides

Backup Slide: Related Work

- Process Calculi
- Dynamic FOCUS
- Mobile TLA
- Mobile Unity (Disadvantages)
- MobiS (Coordination Model based Specification Language)

Backup Slide - Boxed Ambient Example

$$
\begin{array}{l}
network[\\
\quad h[\ (\nu a)\ (route[\ in\ a.\langle h\rangle^{\uparrow}.\langle a1\rangle^{\uparrow}.\langle a2\rangle^{\uparrow}.\langle h\rangle^{\uparrow}.(result)^{storage}.\ out\ a.\langle result\rangle^{\uparrow} \\
\qquad\quad a[\ \langle 0\rangle^{storage} \mid \langle h\rangle^{place} \mid \langle continue\rangle^{semaphore} \mid \\
\qquad\qquad !((cont)^{semaphore}.(here)^{place}.(next)^{route}.\langle next\rangle^{place}.\ out\ here. \\
\qquad\qquad\ in\ next.\ (r[\ out\ a.\ \langle a\rangle\ ] \mid (value)^{\star}.\ (accu)^{storage}. \\
\qquad\qquad\ \langle accu + value\rangle^{storage}.\langle continue\rangle^{semaphore}\ )) \mid \\
\qquad\qquad semaphore[\ !(k).\langle k\rangle\ ] \mid \\
\qquad\qquad place[\ !(i).\langle i\rangle\ ] \mid \\
\qquad\qquad storage[\ !(j).\langle j\rangle\ ] \\
\qquad\quad]) \\
\qquad] \mid \\
\qquad a1[\ !(p)^{r}.\langle 1\rangle^{p}\ ] \mid \\
\qquad a2[\ !(p)^{r}.\langle 2\rangle^{p}\ ] \\
\quad]
\end{array}
$$

Backup Slide: Semantics of Command

Example of description of the semantics of the command Out:

$$\frac{home(\alpha, i) = m \wedge n \neq m \wedge parent(\alpha, m) = n}{\alpha \xrightarrow{i:Out(n)} \alpha(\!|parent := (parent(\alpha))(m := parent(\alpha, n))|\!)}$$

Backup Slide: dAmbISM - Basic Properties

- locality of Ambient ISMs further restricts outputs of dynamic ISMs
- enabledness and the running state of dynamic ISMs restrict the transitions of Ambient ISMs, in particular their outputs
- locality restricts dynamic ISM manipulation
- composite runs preserve the well-formedness of parallel composition