CORPORATE TECHNOLOGY

Formal Security Analysis

at Siemens Corporate Technology

2nd APPSEM II Workshop

April 13th - 15th 2004, Tallinn, Estonia

David von Oheimb and Volkmar Lotz

Siemens CT IC Sec

{david.von.oheimb|volkmar.lotz}@siemens.com

Information &
Communications
Security

Overview

- Our Department
  - Siemens CT IC Sec
- Description and proof techniques
  - Motivation
  - Interacting State Machines
  - Tool Support
  - ISM Extensions
  - ISM Applications
  - Information Flow
  - Cryptographic Protocols
- Selected References
- Summary

SIEMENS

CORPORATE TECHNOLOGY

IT Security at
Siemens CT IC Sec

Contact: Dr. Stephan Lechner
Phone: +49 89 636 46 888
E-mail: stephan.lechner@siemens.com

Competencies

E-Business / E-Commerce security
Internet- / Multimedia security
Mobile Communications security
Cryptography and **Formal Methods**

- **Design, implementation and integration** of security architectures and solutions
- **Security consulting:** Internet, Multimedia, E-Commerce, WLAN, UMTS, mobile devices,...
- **Development and assessment** of cryptographic algorithms
- **Formal analysis** of security specifications and properties

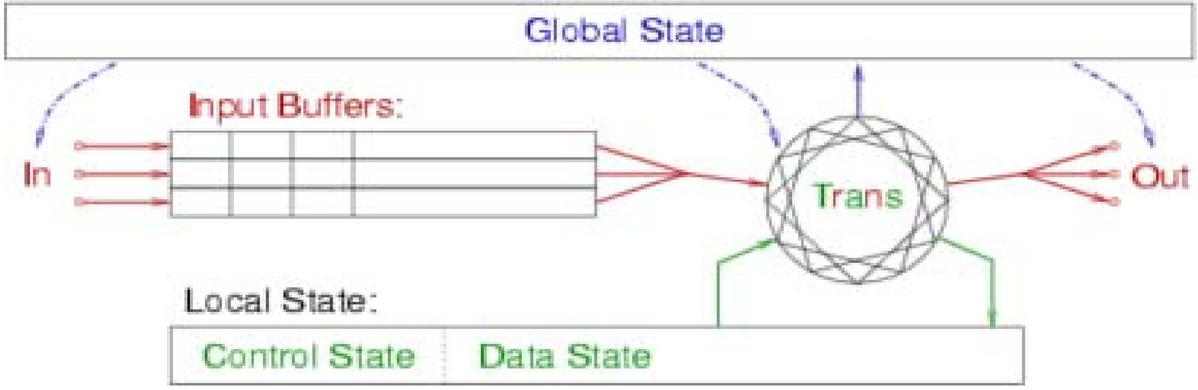
Information & Communications Security

Motivation of Formal Analysis

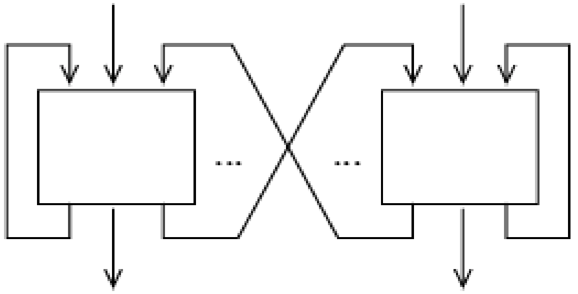
- **Need for quality assurance/improvement of security solutions**
  - highly complex systems and policies
  - ambiguos and incomplete specifications
  - uncertainty about adequacy of solutions
- **Design and assessment of new mechanisms and solutions**
  - New restrictions by the environment (low cost solutions, performance, memory), e.g., car keyless go, smartcard systems
  - New applications (mobility, ad-hoc networking, agent systems, ...)
- **Laws and Regulations**
  - Certification according to Common Criteria

Interacting State Machines (ISMs)

- state transitions (maybe non-deterministic)
- buffered I/O simultaneously on multiple connections

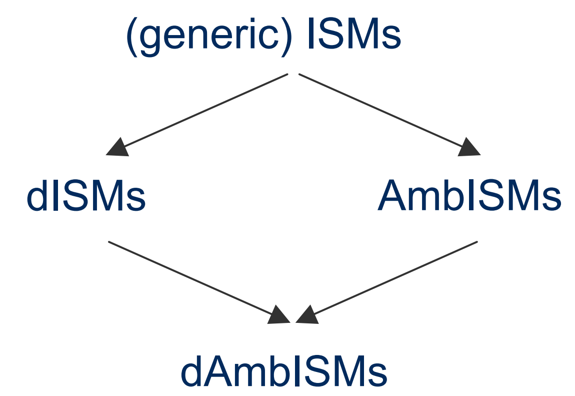


- finite trace semantics
- modular (hierarchical) parallel composition



Extensions to ISM Concepts

- **Generic ISMs:** global/shared state
- **Dynamic ISMs:** changing availability and communication
- **Ambient ISMs:** mobility with constrained communication
- **Dynamic Ambient ISMs:** combination



- **Application:** German BMWA lead project MAP
  "Mobile workplace of the future" (Thomas Kuhn)

- **Future work on ISMs:** refinement, test case generation

Tool Support for ISMs

- **AutoFocus: CASE tool for graphical specification and simulation**
  - syntactic perspective
  - graphical documentation
  - type and consistency checks
- **Isabelle/HOL: powerful interactive theorem prover**
  - semantic perspective
  - textual documentation
  - validation and correctness proofs
- AutoFocus drawing → Quest file → Isabelle theory file

  Within Isabelle: ism sections → standard HOL definitions

Applications of ISMs

- LKW model for Infineon SLE 66 smart card processor
- Infineon SLE 88 memory management
- mobile agent case study for MAP project
- access control for medical information system
- document management system for aircraft industry

Information Flow Control

- **Motivation:** analysis of SLE66
- **Starting point:** noninterference à la [Rusby92]
- **Extensions:** nondeterminism etc.
- **Notions:** nonleakage, noninfluence
- **Focus:** state-oriented systems
- **Related:** language-based security
- **Speciality:** intransitive policies

Verification of Cryptographic Protocols

- Scalable Approach supported by a portfolio of languages / tools
  - Finite-state model checking: CASPER / FDR
  - Infinite-state model checking: HLPSL,IF / OFMC, CLMC, SATML
  - (Interactive) theorem proving: CSP, ISMs / Paulson's inductive method (Isabelle)
- Current work: EU project AVISPA (Jorge Cuellar)

  modeling languages and automatic proof techniques suitable for analyzing industrial-scale protocols from IETF, IEEE, ITU, and 3GPP standards
- Examples:
  - UMTS authentication and key agreement protocol
  - IKE (internet key exchange protocol)
  - H.530 authentication for mobile multi-media applications (S. Mödersheim, Haykal Tej)
  - EAP (extensible authentication protocol): AKA, Archie, IKEv2, SIM, TLS, TTLS, PEAP

Selected References

- V. Lotz, V. Kessler, G. Walter, "A Formal Security Model for Microprocessor Hardware", IEEE Transactions on Software Engineering, 2000
- D. v.Oheimb, V. Lotz, "Formal Security Analysis with Interacting State Machines", ESORICS 2002, Springer LNCS 2502, 2002
- T. Kuhn, D. v.Oheimb, "Interacting State Machines for Mobility", FM 2003
- D. v.Oheimb, V. Lotz, "Extending Interacting State Machines with Dynamic Features", ICFEM 2003
- D. v.Oheimb, G. Walter, V. Lotz, "A Formal Security Model for the Infineon SLE88 Smartcard Memory Management", ESORICS 2003
- D. Basin, S. Mödersheim, L. Viganò: An On-the-Fly Model-Checker for Security Protocol Analysis, ESORICS 2003
- D. v.Oheimb, "Information flow control revisited: Noninfluence = Noninterference + Nonleakage", submitted for publication, 2004

Conclusion

Formal Methods for Security Analysis

Utilizing mathematically precise techniques for the specification of security requirements and the verification of security properties

Assessment

- of security solutions with formal models

Evaluation

- according to ITSEC and Common Criteria

Verification

- of (safety and) security properties

Scalable, tool-based approach

- theorem prover, CASE tool, model checkers

Wide range of application domains

- requirements, architectures, mechanisms

CORPORATE TECHNOLOGY

Notes on Industrial Formal Requirements Engineering

2nd APPSEM II Workshop

April 13th - 15th 2004, Tallinn, Estonia

David von Oheimb and Volkmar Lotz

Siemens CT IC Sec

{david.von.oheimb|volkmar.lotz}@siemens.com

Information & Communications Security

Requirements for Modeling Formalism

- adequately expressive, abstract
- simple, graphical
- modular, scalable, flexible
- precise
- state-oriented
- tool-supported

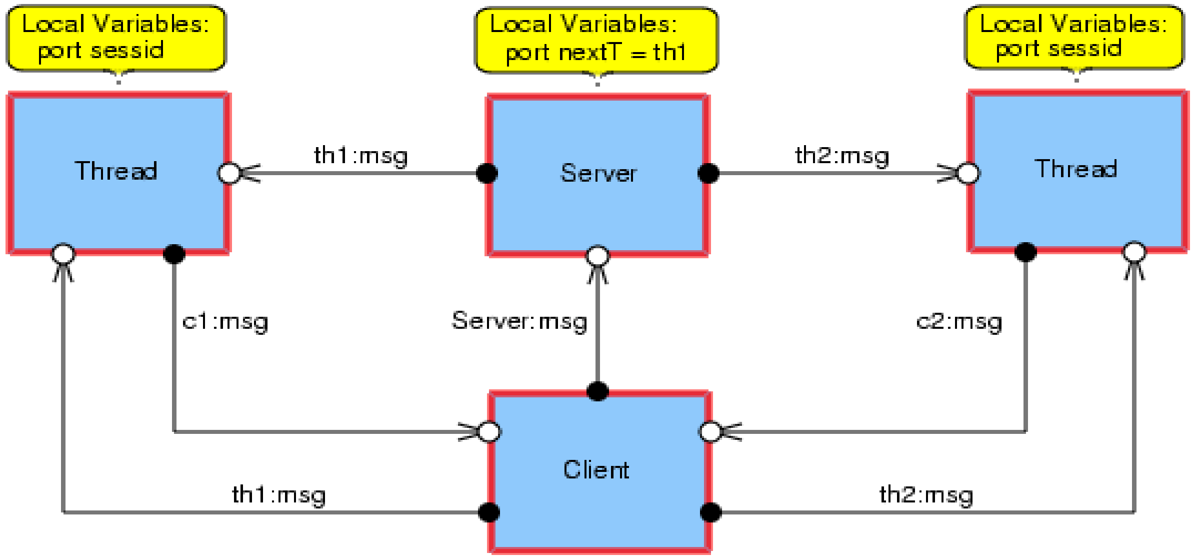
Basic ISMs in Isabelle/HOL

```
ism name ((param_name :: param_type))* =
    ports pn_type
      inputs    I_pns
      outputs   O_pns
    messages msg_type
     states     [state_type]
     [control cs_type [init cs_expr0]]
     [data      ds_type [init ds_expr0] [name ds_name]]
   [transitions
     (tr_name [attrs]]: [cs_expr -> cs_expr']
     [pre  (bool_expr)+ ]
     [in    ([multi] I_pn  I_msgs)+ ]
     [out  ([multi] O_pn O_msgs)+ ]
     [post ((lvar_name := expr)+ | ds_expr')] )+ ]
```

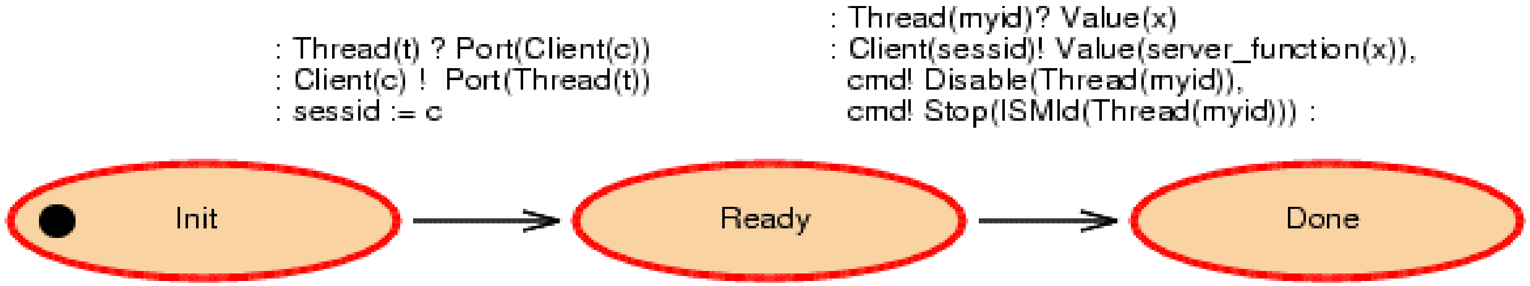
CORPORATE TECHNOLOGY

ISM representation in AutoFocus

- **System Structure Diagram:** client/server



- **State Transition Diagram:** worker thread



Information &
Communications
Security

Benefits for Requirements Engineering

- "nasty questions" leading to
  - better understanding
  - clarification
  - explicitness
- thus detection of
  - gaps
  - hidden assumptions
  - inconsistencies
- optimizations, bug fixes
- results limited by
  - **scope of modeling**
  - **validity of model**

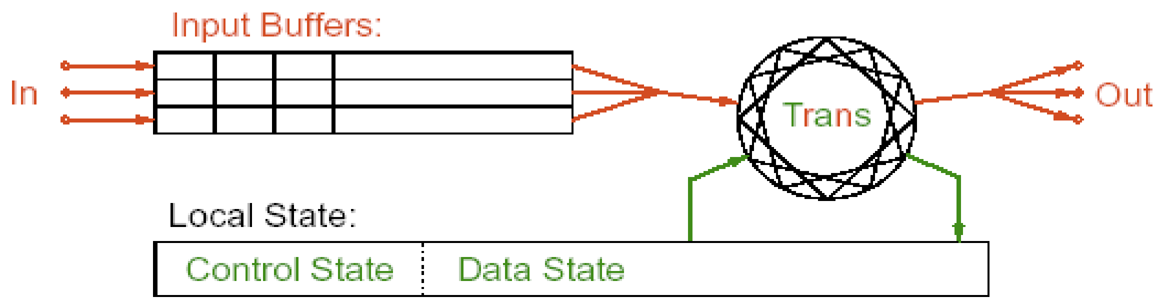
Backup Slides

- Formal definition of Interacting State Machines
- Isabelle/HOL
- Graphical representation of ISMs (Example: LKW model)
- dynamic Ambient ISMs
- Ambient ISM Example
- Project MAP
- CASPER/FDR and HLPSL/OFMC
- Modeling the H.530 protocol

CORPORATE TECHNOLOGY

Formal Definition of Basic ISMs

$MSGs = \mathcal{P} \to \mathcal{M}^*$	family of messages $\mathcal{M}$, indexed by port names $\mathcal{P}$
$CONF(\Sigma) = MSGs \times \Sigma$	configuration with local state Σ
$TRANS(\Sigma) = \wp(CONF(\Sigma) \times CONF(\Sigma))$	transitions
$ISM(\Sigma) = \wp(\mathcal{P}) \times \wp(\mathcal{P}) \times \Sigma \times TRANS(\Sigma)$	ISM type
$a = (In(a), Out(a), \sigma_0(a), Trans(a))$	ISM value a



Information & Communications Security

Open runs

$$Runs(a) \in \wp(\Sigma^*)$$

$$\langle \sigma_0(a) \rangle \in Runs(a)$$

$$\frac{ss^\frown\sigma \in Runs(a) \qquad ((i,\sigma),(o,\sigma')) \in Trans(a)}{ss^\frown\sigma^\frown\sigma' \in Runs(a)}$$

CORPORATE TECHNOLOGY

Parallel Runs (Interaction)

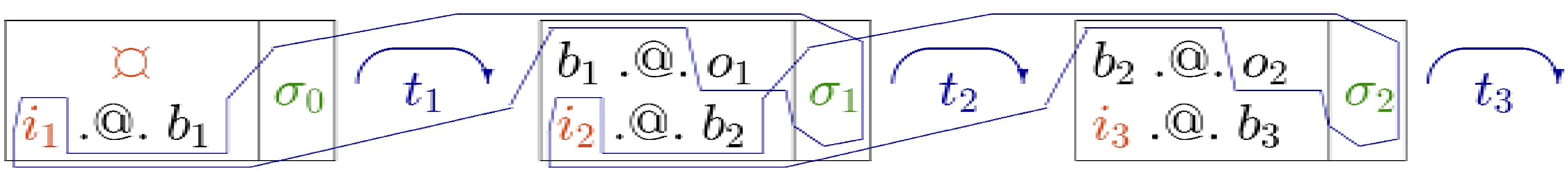
Let $A = (A_i)_{i \in I}$ be a family of ISMs.

$CRuns(A)$ of type $\wp((CONF(\Pi_{i \in I}\Sigma_i))^*)$

$$\frac{}{\langle(\text{¤}, \Pi_{i \in I}(\sigma_0(A_i)))\rangle \in CRuns(A)}$$

$$\frac{\begin{array}{c} j \in I \\ cs \frown (i\ .@.\ b, (S[j := \sigma])) \in CRuns(A) \\ ((i, \sigma), (o, \sigma')) \in Trans(A_j) \end{array}}{cs \frown (i\ .@.\ b, S[j := \sigma]) \frown (b\ .@.\ o, S[j := \sigma']) \in CRuns(A)}$$

$S[j := \sigma]$ replaces the j-th component of the tuple S by σ.

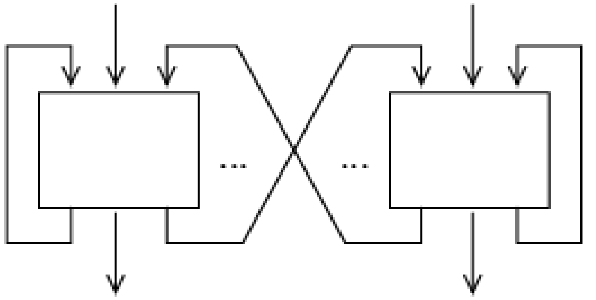


Information & Communications Security

(Parallel) Composition of ISMs

Let $A = (A_i)_{i \in I}$ be a family of ISMs. Their *parallel composition* $\|_{i \in I} A_i$ is an ISM of type $ISM(CONF(\Pi_{i \in I} \Sigma_i))$ being defined as

$$(AllIn(A) \setminus AllOut(A),\ AllOut(A) \setminus AllIn(A),\ (\emptyset, S_0(A)),\ PTrans(A))$$



where

- $AllIn(A) = \bigcup_{i \in I} In(A_i)$
- $AllOut(A) = \bigcup_{i \in I} Out(A_i)$
- $S_0(A) = \Pi_{i \in I}(\sigma_0(A_i))$ is the Cartesian product of all initial local states
- $PTrans(A) \in TRANS(CONF(\Pi_{i \in I} \Sigma_i))$ is the parallel composition of their transition relations, defined as ...

Parallel State Transition Relation

$$\frac{\begin{array}{c} j \in I \\ ((i,\sigma),(o,\sigma')) \in \mathit{Trans}(A_j) \end{array}}{\begin{array}{c} ((i_{|\overline{AllOut(A)}}, (i_{|AllOut(A)} \;.@.\; b, S[j:=\sigma])), \\ (o_{|\overline{AllIn(A)}}, \;(b \;.@.\; o_{|AllIn(A)}, \;S[j:=\sigma']))) \in \mathit{PTrans}(A) \end{array}}$$

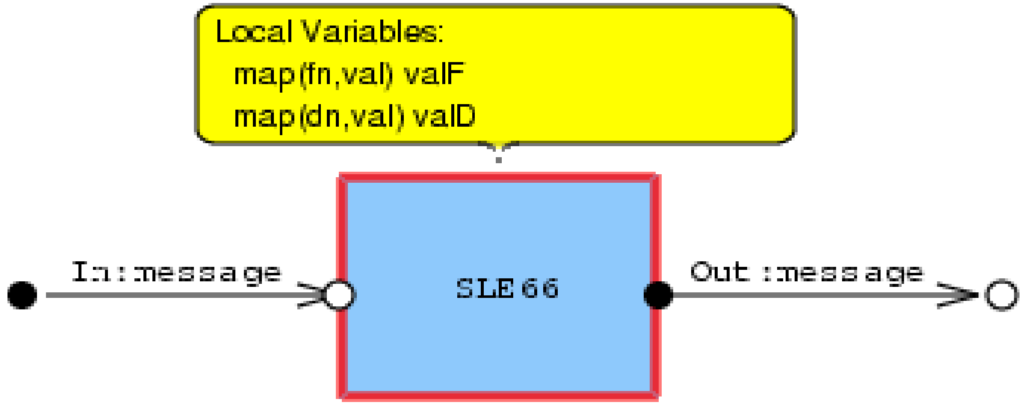
where

- $S[j:=\sigma]$ replaces the j-th component of the tuple S by σ
- $m_{|P}$ denotes the restriction $\lambda p.$ **if** $p \in P$ **then** $m(p)$ **else** $\langle\rangle$ of the message family m to the set of ports P
- $o_{|\overline{AllIn(A)}}$ denotes those parts of the output o provided to any outer ISM
- $o_{|AllIn(A)}$ denotes the internal output to peer ISMs or direct feedback, which is added to the current buffer contents b

Isabelle/HOL

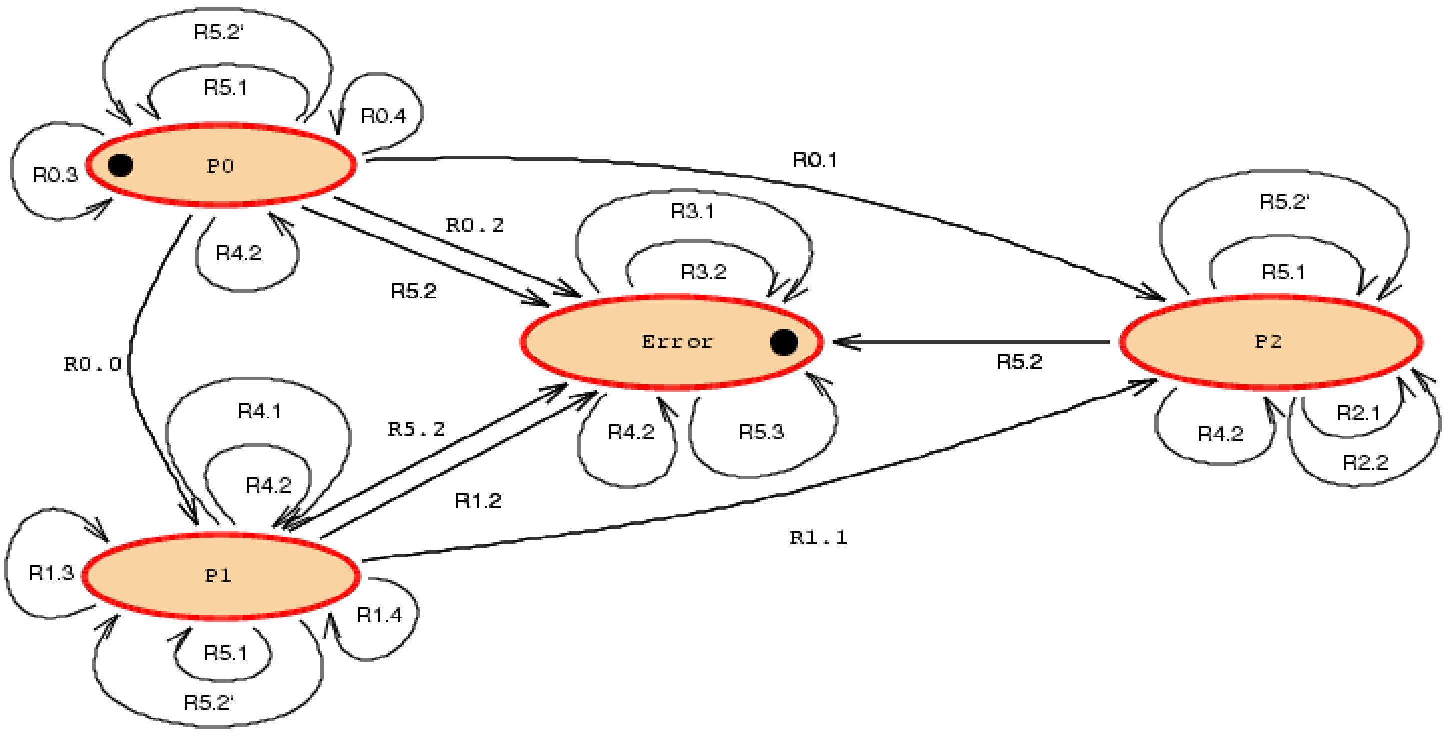
- generic interactive theorem prover
- most popular object logic: Higher-Order Logic (HOL)
(for its expressiveness + automatic type inference)
- HOL: predicate logic based on simply-typed lambda-calculus
- proofs with semi-automatic tactics including rewriting
- user interface: Proof General, integrated with XEmacs
- well-documented and supported, freely available (open-source)

Graphical Representation: System Structure Diagram



CORPORATE TECHNOLOGY

Graphical Representation: State Transition Diagram



Information & Communications Security

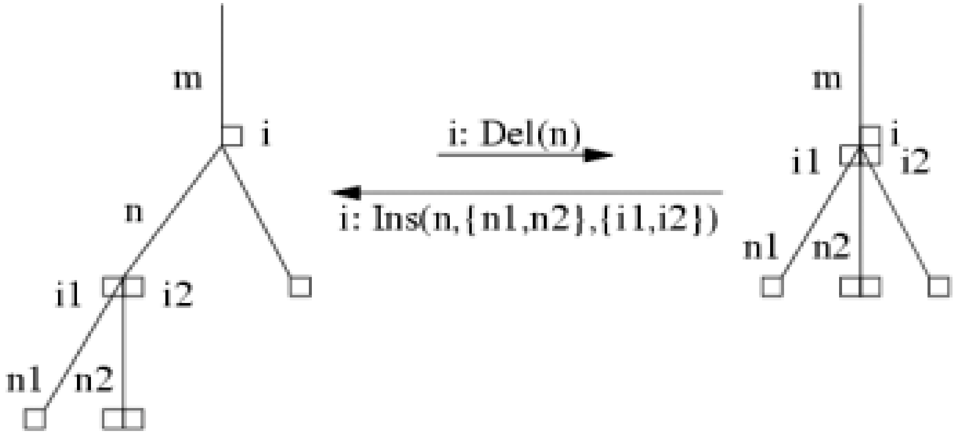
dynamic Ambient ISMs

- Dynamic commands:
  `Run(i), Stop(i), Enable(p), Disable(p), New(p), Convey(p,i)`
- Additional structure:

  ambient tree

  with locality

  constraints



- Mobile commands:
  `Assign(i,n), In(n), Out(n), Del(n), Ins(n,ns,is)`
- Operational semantics of Ambient Calculus

Ambient ISM Example

- **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"
```

Project MAP

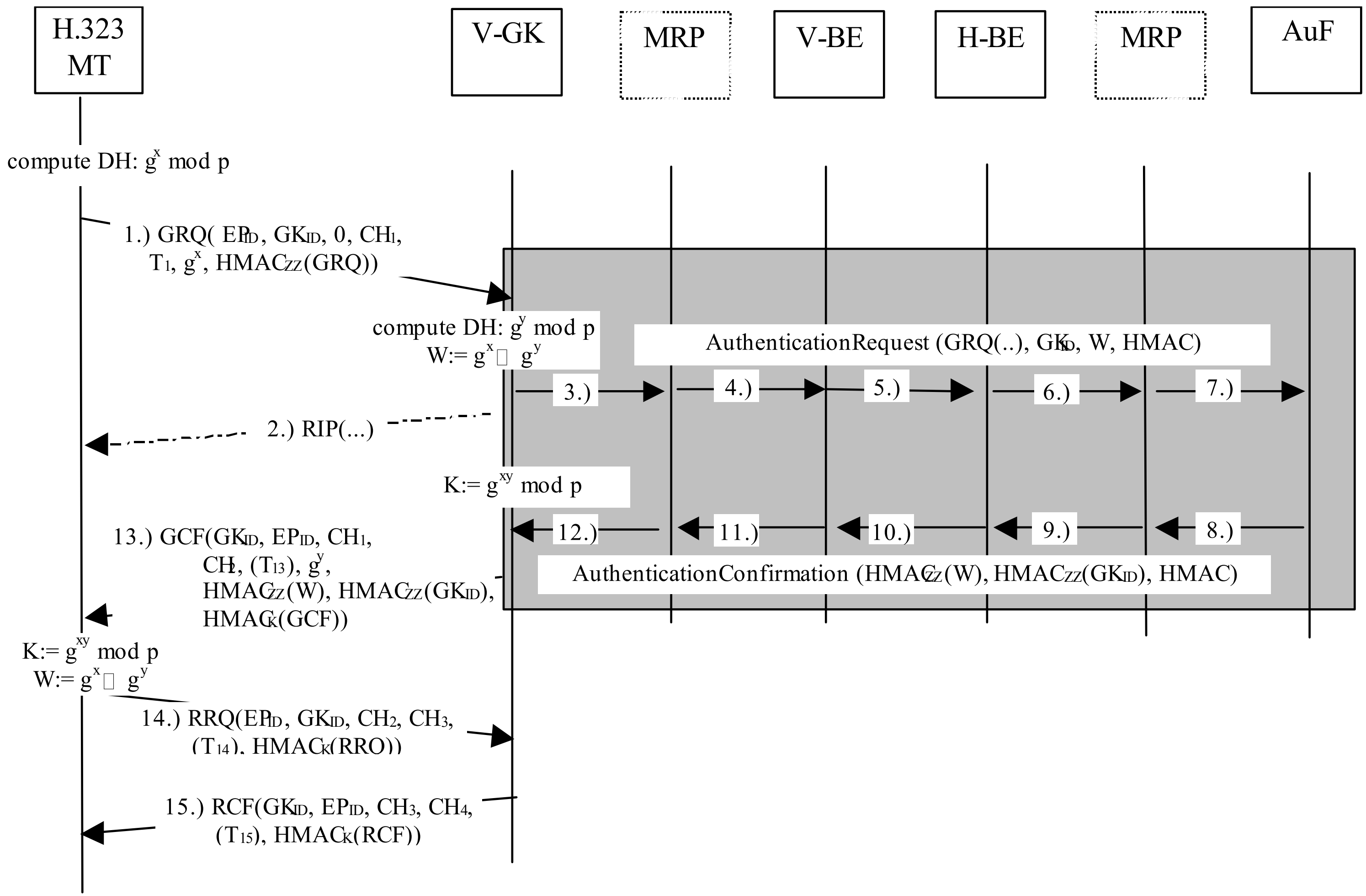
- **MAP: „Multimedia Arbeitsplatz der Zukunft“**
- **One of the six main projects in the area of**
  Integrating Man and Machine in the Knowledge Society
  sponsored by the German Federal Ministry of Economics and Labor
- **Partners: Industrial (9), SME (5), Academic (6)**
- **Aim: develop novel concepts and a basis for future mobile, multi-media based work places**
- **Methods from**
  - security technology
  - man-machine interaction
  - agent technology
  - Mobility support

CASPER/FDR and HLPSL/OFMC

- CASPER: high-level specification language for crypto protocols
- Finite-state model checker FDR *(Failure-/Divergence-Refinement)*
  - automatic refinement checks
  - complexity limitations
- HLPSL: High-Level Protocol Specification Language
- Infinite-state model checker OFMC *(On-the-Fly Model Checker)*
  - built-in “lazy” intruder model
  - partial-order reduction, heuristics
  - easy-to-use
  - copes well with complex protocols

CORPORATE TECHNOLOGY

Modeling the H.530 protocol



Information & Communications Security