Logics for the Ambient Calculus - and model checking mobile ambients

Helena M. Meyer

Mobile Computing Seminar

December 2, 2004

Plan

- Motivation
- The formal model for the ambient logic
- The ambient logic
- Logical formulas and notation of satisfaction
- Model checking of mobile programs

Motivation

Goal:

- √ *formally* describe properties of mobile computations
- √ reasoning about these properties
- √ ensure *correctness* of programs written in the ambient calculus

Motivation

Solution:

- Cardelli and Gordon: the ambient logic
- a modal logic with temporal and *spatial* modalities
- designed to specify properties of distributed and mobile computations programmed in the ambient calculus
- formal model:
  the ambient calculus without private names

Motivation

Specification logic for mobility:

- √ describe the structure of spatial configurations
- √ describe their evolution through time
- √ $\Rightarrow$ a modal logic of space and time

The ambient logic can be:

- √ very precise - describing the structure of locations
- √ imprecise - describing things that happen *eventually* or *sometime*

Motivation

Spatial configurations = ambient configurations consisting only of spatial operators

Natural **interpretation** as

- ✓ unordered edge-labelled trees
- ✓ edge labels = names of sublocations
- ✓ subtrees = sublocations

Motivation

Spatial specifications:

- √ the configuration looks initially like tree T1
- √ and eventually like tree T2

Temporal specifications:

- √ “there is always at most one agent called n here”
- √ “eventually the agent crosses the firewall”

Plan

- √ Motivation
- √ **The formal model for the ambient logic**
- √ The ambient logic
- √ Logical formulas and notation of satisfaction
- √ Model checking of mobile programs

A model for the ambient logic

- a modified version of the basic Ambient Calculus
- no name restriction $(\nu n)P$

$$
\begin{array}{lll}
P, Q ::= & \text{processes} & \\
\quad \mathbf{0} & \text{inactivity} & \multirow{4}{*}{} \\
\quad P|Q & \text{composition} & \\
\quad !P & \text{replication} & \\
\quad M[P] & \text{ambient} & \\
\quad M.P & \text{action} & \\
\quad (n).P & \text{input action} & \\
\quad \langle M \rangle & \text{output action} &
\end{array}
\left.\begin{array}{l} \mathbf{0} \;\; \text{inactivity} \\ P|Q \;\; \text{composition} \\ !P \;\; \text{replication} \\ M[P] \;\; \text{ambient} \end{array}\right\} \text{spatial}
\qquad
\left.\begin{array}{l} M.P \;\; \text{action} \\ (n).P \;\; \text{input action} \\ \langle M \rangle \;\; \text{output action} \end{array}\right\} \text{temporal}
$$

A model for the ambient logic

Syntax (continued):

$$
\begin{array}{lll}
M ::= & & \text{expressions} \\
\quad n & \text{name} \left.\right\} \text{names} & \\
\quad in\ M & \text{can enter M} & \\
\quad out\ M & \text{can exit M} & \text{capabilities} \\
\quad open\ M & \text{can open M} & \\
\quad \epsilon & \text{null} & \\
\quad M.M' & \text{composite} & \text{paths}
\end{array}
$$

Semantics of the ambient calculus

The semantics is given by the relations:

- ✓ $P \equiv Q$ and $P \rightarrow Q$

Structural congruence, $P \equiv Q$,

- ✓ is a relation between processes
- ✓ used in the ambient logic
- ✓ used in the reduction semantics
- ✓ the rules for **structural congruence** are sound and complete for equivalence of edge-labelled trees
- ✓ completeness result motivates the choice of axioms for structural congruence

Semantics of the ambient calculus

The **reduction relation**, $P \rightarrow Q$,

- ✓ describes the dynamic behaviour of ambients
- ✓ defines the evolution of processes over time

Auxiliary relation:

- ✓ **sublocation relation**, $P \downarrow Q$,
- ✓ defines the spatial distribution of processes
- ✓ holds when Q is the whole interior of a top-level ambient in P

Semantics of the ambient calculus

The **reduction relation**:

$$\left.\begin{array}{lcl} n[in\, m.P|Q]|m[R] & \rightarrow & m[n[P|Q]|R] \\ m[n[out\, m.P|Q]|R] & \rightarrow & n[P|Q]|m[R] \\ open\, n.P|n[Q] & \rightarrow & P|Q \end{array}\right\} \text{mobility reductic}$$

$$(n).P|\langle M\rangle \;\rightarrow\; P\{n \leftarrow M\}\;\}\;\text{local communication}$$

$$P \rightarrow Q \;\Rightarrow\; n[P] \rightarrow n[Q]\;\}\;\text{ambient}$$

$$P \rightarrow Q \;\Rightarrow\; P|R \rightarrow Q|R\;\}\;\text{parallel}$$

$$P' \equiv P,\; P \rightarrow Q,\; Q \equiv Q' \;\Rightarrow\; P' \rightarrow Q'\;\}\;\equiv$$

reduction

Plan

- √ Motivation
- √ The formal model for the ambient logic
- √ **The ambient logic**
- √ Logical formulas and notation of satisfaction
- √ Model checking of mobile programs

The Ambient Logic

- **Goal:** formalise assertions
- **Solution:** modal logics
- **Benefits:** support model checking
- **Properties:**

consider *spatial modalities* for properties:

- hold at a certain location
- hold at some location
- hold at every location

The Ambient Logic

The ambient logic can talk about

- √ time
- √ space
- √ i.e. spatial properties and spatial specifications

Modal logic: truth of a formula is relative to a state

Ambient logic: truth is relative to the current time i.e.

- √ the current state of execution
- √ the current location

The Ambient Logic

Correspondence between

- ✓ spatial constructs in the ambient calculus
- ✓ certain formulas of the ambient logic

Processes in ambient calculus	**Formulas of the logic**	
$\mathbf{0}$	$\mathbf{0}$	**"there is nothing here"**
$n[P]$	$n[\mathcal{A}]$	**"there is one thing here"**
$P\|Q$	$\mathcal{A}\|\mathcal{B}$	**"there are two things here"**

Logical formulas

Syntax for the ambient logic:

η	a name n or a variable x
$\mathcal{A}, \mathcal{B} ::=$	formula
$\quad \mathbf{T}$	true
$\quad \neg\mathcal{A}$	negation
$\quad \mathcal{A} \vee \mathcal{B}$	disjunction
$\quad \mathbf{0}$	void
$\quad \eta[\mathcal{A}]$	location
$\quad \mathcal{A} \vert \mathcal{B}$	composition

Logical formulas

Syntax continued:

$\forall x.\mathcal{A}$	universal quantification over names
$\Diamond\mathcal{A}$	sometime modality
$\triangle\mathcal{A}$	somewhere modality
$\mathcal{A}@\eta$	location adjunct
$\mathcal{A} \triangleright \mathcal{B}$	composition adjunct

The Ambient Logic

Examples:

- the formula $n[\mathbf{0}]$: *there is currently an empty location called n*
- $n[\mathcal{A}]$: represents a single step in *space* (the place one step down into n)
- $\triangle\mathcal{A}$: an arbitrary number of steps in space
- $\diamond\mathcal{A}$: temporal eventuality operator
- no name binders
- only quantifiers bind variables
- a formula $\mathcal{A}$ is *closed* if $fv(\mathcal{A}) = \emptyset$

The satisfaction relation

$P \models \mathcal{A}$

- √ process P satisfies the closed formula $\mathcal{A}$
- √ is defined inductively
- √ temporal modality: semantics is given by *reductions*
- √ spatial modality: semantics is given by the *sublocation relation* $P \downarrow P'$
  - ⋆ $P \downarrow P'$ iff $\exists n,\ P'' \ :\ P \equiv n[P']|P''$
  - ⋆ $P \downarrow^* P'$ is the reflexive and transitive closure ("P contains P' at some nesting level")

The satisfaction relation

Notation:

- Π is the sort of processes ($P : \Pi$)
- Φ is the sort of formulas ($\mathcal{A} : \Phi$)
- Λ is the sort of names ($n : \Lambda$)
- Γ is the sort of variables ($x : \Gamma$)

The satisfaction relation

$$
\begin{array}{lll}
P \models \mathbf{T} & & \\
P \models \neg\mathcal{A} & \text{iff} & \neg P \models \mathcal{A} \\
P \models \mathcal{A} \vee \mathcal{B} & \text{iff} & P \models \mathcal{A} \ \vee \ P \models \mathcal{B} \\
P \models \mathbf{0} & \text{iff} & P \equiv \mathbf{0} \\
P \models n[\mathcal{A}] & \text{iff} & \exists P' : \Pi.\ P \equiv n[P'] \wedge P' \models \mathcal{A} \\
P \models \mathcal{A}|\mathcal{B} & \text{iff} & \exists P', P'' : \Pi.\ P \equiv P'|P'' \wedge \\
 & & P' \models \mathcal{A} \wedge P'' \models \mathcal{B} \\
P \models \forall x.\mathcal{A} & \text{iff} & \forall m : \Lambda.\ P \models \mathcal{A}\{x \leftarrow m\} \\
P \models \Diamond\mathcal{A} & \text{iff} & \exists P' : \Pi.\ P \rightarrow^{*} P' \wedge P' \models \mathcal{A} \\
P \models \triangle\mathcal{A} & \text{iff} & \exists P' : \Pi.\ P \downarrow^{*} P' \wedge P' \models \mathcal{A}
\end{array}
$$

The satisfaction relation

Continued:

$$P \models \mathcal{A}@n \quad \text{iff} \quad n[P] \models \mathcal{A}$$
$$P \models \mathcal{A} \triangleright \mathcal{B} \quad \text{iff} \quad \forall P' : \Pi.\ P' \models \mathcal{A} \Rightarrow P|P' \models \mathcal{B}$$

- ✓ @ and $\triangleright$ can be used to express assumption/guarantee specifications (security properties)
- ✓ $P \models \mathcal{A}@n$ means that P satisfies $\mathcal{A}$ even when placed into a location n
- ✓ $\mathcal{A} \triangleright \mathcal{B}$ means that P satisfies $\mathcal{B}$ under any possible attack by an opponent that it bound to satisfy $\mathcal{A}$

The satisfaction relation

- Note:
  the definition of satisfaction is based heavily on the structural congruence relation
- Structural congruence is easily decidable (useful in model checking applications)

Derived connectives

Other properties expressible in the logic:

$\mathbf{F}$	iff $\neg\mathbf{T}$	false
$\mathcal{A} \wedge \mathcal{B}$	iff $\neg(\neg\mathcal{A} \vee \neg\mathcal{B})$	conjunction
$\mathcal{A} \Rightarrow \mathcal{B}$	iff $\neg\mathcal{A} \vee \mathcal{B}$	implication
$\exists x.\ \mathcal{A}$	iff $\neg\forall x.\ \neg\mathcal{A}$	existential quantification
$\Box\mathcal{A}$	iff $\neg\Diamond\neg\mathcal{A}$	everytime modality
$\triangledown\mathcal{A}$	iff $\neg\vartriangle\neg\mathcal{A}$	everywhere modality

Example:

- ✓ $\Box n[\mathbf{0}]$: *there is always a location called n*

Plan

- √ Motivation
- √ The formal model for the ambient logic
- √ The ambient logic
- √ Logical formulas and notation of satisfaction
- √ **Model checking of mobile programs**

Model checking

Given:

- √ program P in ambient calculus
- √ property F specified as an ambient logic formula
- √ determine automatically whether $P \models F$

Requirement:

- √ the model of the program must be *finite*

Model checking

Model checking algorithm

- for the *fragment* of the ambient calculus:
  - no replications
  - no dynamic name generations
  - called the *replication-free* or *finite-state* fragment of the ambient calculus
- against a *fragment* of the ambient logic:
  - no guarantee operators ($\mathcal{A} \triangleright \mathcal{B}$)

Model checking

Why these fragments?

Replication and *guarantee* are both sources of infinity:

- $\surd$ replicated process $\equiv$ infinite array of replicas
- $\surd$ guarantee formula $\equiv$ a certain infinite quantification over processes

Conclusion:
not possible to extend the model checking algorithm, because these extensions leads to **undecidability**

Model checking

- √ Cardelli and Gordon give a model checking algorithm for this decidable sublogic
- √ Express and automatically verify properties of mobile code
- √ No official implementation of a model checker for mobile ambients (?)

References

- L. Cardelli and A. Gordon: *Anytime, Anywhere: Modal Logics for Mobile Ambients*, 2000
- W. Charatonik, A. Gordon et al: *Model Checking Mobile Ambients*, 2002

A study of the logic extended with constructs for describing private names:

- L. Cardelli and A. Gordon: *Logical properties of name restriction*, 2001

Questions?