PRIAM
Privacy Specification Model

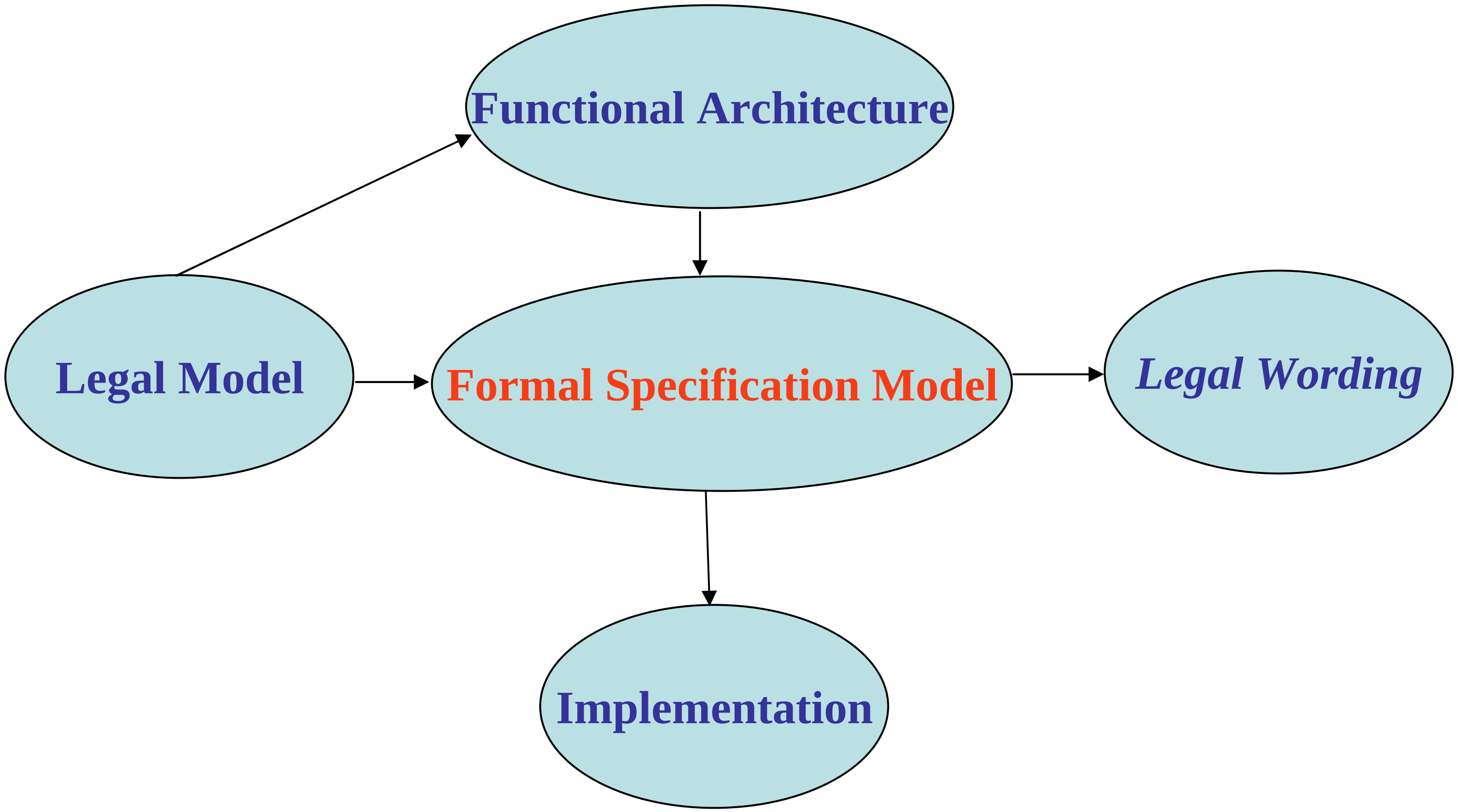
December 6-7 2007

Daniel Le Métayer

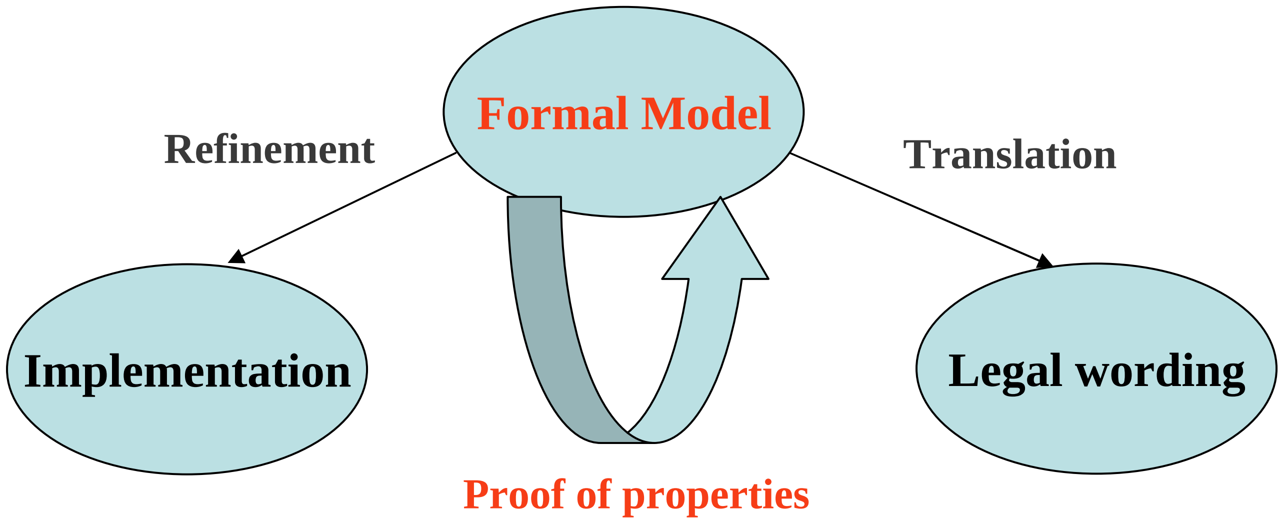
INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA
RHÔNE-ALPES

PRIAM approach



Formal model as a link between technology and law



Need for a formally defined privacy policy language

Case by case consent is impossible $\Rightarrow$ need for a generic way to express privacy requirements and policies

A formal model is useful to

- Avoid ambiguities in the expression of the policies and requirements (internal consistency)
- Ensure that the combination of techniques used to implement the policy is indeed sufficient (completeness)
- Check consistency between policies requested by data subjects and policies implemented by data controllers (compliance)

Overall goal: strengthen the liability (and trust) of all the actors involved

Possible approaches

Use an existing formal language (e.g. process calculus):

- Pros: semantics and proof system available, possibly a refinement theory

- Cons: not necessarily well suited, possibly too general or complex

Define a dedicated language:

- Pros: hopefully well suited, minimal

- Cons: need to define semantics, proof system and refinement theory

Also: possibly too specific (difficult to cope with new privacy policies or assumptions)

Requirements

Meet the challenges posed by the formalization of privacy for ubiquitous computing :

- Broadcast asynchronous communications
- Dynamic set of agents (agents can become active or inactive, permanently or temporarily)
- Obligations as well as rights
- Deal with time
- Sticky data policies
- A priori as well as a posteriori checks

Our approach

Three tier approach for a maximal level of reusability:

- Definition of a kernel language: computation and communication issues

- Models in this language: privacy policy frameworks (agent specifications)

- Parameters of these models: specific privacy policies (agent policy and data policy)

Benefits

Properties can be proven (and reused) at each level:

- Universal properties at the language level

  Example: conditions for property preserving refinements

- General privacy properties at the model level

  Example: if the policy associated with data D of subject S requires that D cannot be forwarded by a collector, then, for any possible trace T and any index i, such that in Ti the state of agent A contains D, then there exists an index j < i such that in Tj, A receives D from S

- Specific privacy properties with parameters

  Example: for any possible trace T and any index i, the state of agent A in Ti does not contain D

Kerlan: Kernel language

Basic notions:

- State (*record*)
- Environment (*multiset of tuples*)
- Condition : BooleanExpression | [Pattern*]
- Action: StateField := Expression | [Expression*]
- Agent : <Condition*, Action*, Priority>*
- System: Agent*

Example of specification in Kerlan

Agent state: [Identity, AgentPolicy, Time, DataSpace, Trace]

Agent environment: {Message}

AgentPolicy: DataType → DataPolicy

DataPolicy: [Deletion, Use, Transfer, SRights]

Deletion: Nat | ∞

Use: [Purposes, Information, Consent]

Transfer: [Right, Information, Consent]

SRights: [DataAccess, ValueModification, PolicyModification, TraceAccess, Deletion]

DataSpace: {[Data, Time]}

Data : [Identity, DataType, Value, DataPolicy]

Message: [MessageType, Identity, Identity, Content]

Specification of agent behaviours (1/5)

[SendData, x, y, d]

y = Identity

AgentPolicy(d.DataType) ≤ d.DataPolicy

Time = t

→

DataSpace := DataSpace U {[d,t]}

Specification of agent behaviours (2/5)

$[d,t] \in DataSpace$

$t + d.DataPolicy.Deletion = t'$

$Time = t'$

$\rightarrow$

$DataSpace := DataSpace - \{[d,t]\}$

Specification of agent behaviours (3/5)

[RequestData, x, y, [z,type]]

y = Identity   z ≠ y

[d,t] ∈ DataSpace

d.DataType = type

d.Identity = z

d.DataPolicy.Transfer.Right = True

d.DataPolicy.Transfer.Information = False

d.DataPolicy.Transfer.Consent = False

→

[SendData, y, x, d]

Specification of agent behaviours (4/5)

[RequestData, x, y, [z,type]]

y = Identity $z \neq y$

$[d,t] \in$ DataSpace

d.DataType = type

d.Identity = z

d.DataPolicy.Transfer.Right = True

d.DataPolicy.Transfer.Information = True

d.DataPolicy.Transfer.Consent = False

$\rightarrow$

[SendData, y, x, d], [TransferInfo, y, z, [x,d]]

Specification of agent behaviours (5/5)

[RequestData, x, y, [z,type]]

y = Identity z ≠ y

[d,t] ∈ DataSpace

d.DataType = type

d.Identity = z

d.DataPolicy.Transfer.Right = True

d.DataPolicy.Transfer.Information = Flase

d.DataPolicy.Transfer.Consent = True

→

[TransferRequest, y, z, [x,d]]

Semantics of Kerlan

Trace semantics:

- Semantics of a system: set of all possible execution traces
- Each execution trace is a sequence of tuples of triples: T**
- Ti: (Definition, Environment, State) for agent i

Essential features:

- Communications through the environments
- Non determinism
- Priority to local actions to ensure the execution of obligations
- Intermittent agents
- Simple treatment of time: True → Time := Time + 1
- No sequentiality !

Back to requirements

Meet the challenges posed by the formalization of privacy for ubiquitous computing :

- Broadcast asynchronous communications
- Dynamic set of agents (agents can become active or inactive, permanently or temporarily)
- Obligations as well as rights
- Deal with time
- Sticky data policies
- A priori as well as a posteriori checks

Additional features

- Specification of incompatibilities between data types (e.g. “no collection of both profession and town”)

- Level of flexibility in privacy policies (limited form of negotiation)

- Types of roles and types of agents (to qualify use and transfer rights)

- Order relationship between types (data, roles, agents)

- Additional sensors (e.g. location)

Future work

- Full definition of realistic privacy policies (limitations?)
- Formal definition of refinement and associated liability assumptions (no other action on collected data, secure communications, etc.)
- Translation into “natural” legal language and integration within a legal framework (need for third parties?)
- Extensions (identity management, trust management) ?

PRIAM position

- Ambient Intelligence context:

  Pragmatic approach: no other solution than Flexibility + Responsibility

- Tighten the link between privacy rights and technology:

  Top-down approach: Law → Formal Model → Implementation

- Reestablish the balance between data owners and controllers

  Technology can also be used to strengthen citizen rights : require the use of dedicated tools and their protection by law