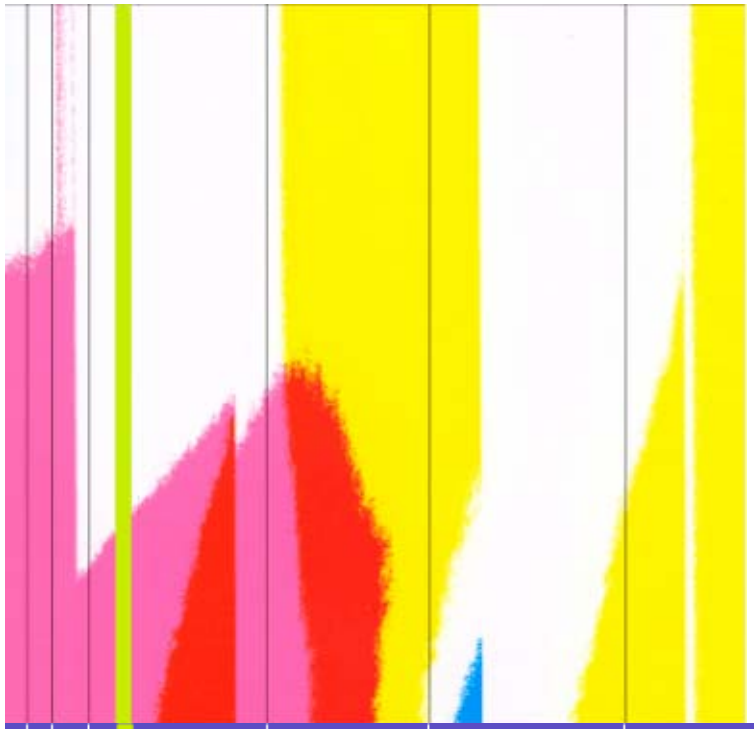


Computer assisted privacy protection

Daniel Le Métayer

PRIAM Workshop
May 22-23 2008

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA
RHÔNE-ALPES

Why computer assistance?

- The “Invisibility” principle of pervasive computing prevents case by case interactions with the subject. Even when not strictly impossible, too frequent interactions undermines the awareness of the subject and leads to privacy relinquishment.

- Philosophy: increased automation from the “invasion side”, why not also use automation to improve the position of the “defense”?

- But: a number of legal requirements have to be met.

Summary of the legal analysis

- Consent is a unilateral act rather than a contract between the subject and the controller.

- Consent should be free, specific, informed and unambiguous.

- Written form is not always required, but desirable for a better protection of the subjects (for proof if not for validity). Electronic documents are admitted as well as hand-written documents.

- Software agents should be considered as any software components (objects rather than legal entities).

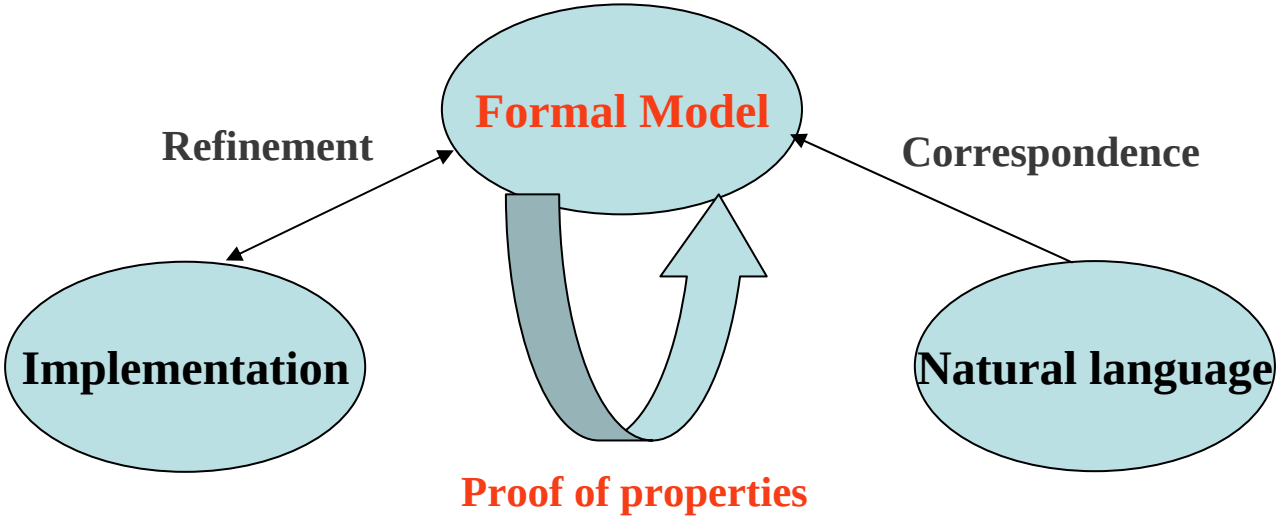
Our position

- Be pragmatic: this is a risk analysis problem. The introduction of software agents should reduce risks for the subjects.

- To reduce risks it is necessary to identify all actors involved and define precisely their commitments and responsibilities.

- To define these commitments the language issue is crucial: need to use the appropriate language for each task, to ensure that each actor understands his declarations (consent, commitment, etc.) and that such declarations are unambiguous.

Formal model as a link between technology and law



Our proposal

1. Global architecture (actors and responsibilities)
2. Restricted natural language for declarations (SIMPL)
3. Formal model based on execution traces
4. Translation of declarations (consent of the subject and declaration of the controller) into the formal model
5. Link between the formal model and software agent implementations

NB: links with the formal models are necessarily partial (some aspects are not amenable to logic and need to be checked manually)

Global architecture

- Actors: Personal Data Authority (PDA), Software Agent Providers (SAP), Controllers and Subjects:

- Roles of the Personal Data Authority :
  - Certify the meaning of the natural language for declarations (link SIMPL – Formal Model and legality of disclosure policies)
  - Possibly: certify specific Software Agents (link Formal Model – Implementation) or approve independent evaluation centers for the certification of Software Agents

- Roles and commitments of the Software Agent Providers :
  - Deliver and warrant Software Agents (consistency with Formal Model) through an agreement with their customer (Subject or Controller)
  - Possibly: submit Software Agents for certification

- Commitments of the Controllers:
  - Use the Software Agents faithfully (no access to personal data other than through the Software Agent, no execution traces tampering, etc.)
  - Ensure compliance with his declarations
  - Ensure the security of personal data

SIMPL : a SIMple Privacy Language

- Pattern-based language used to define disclosure policies (Subject) and collection policies (Controller)

- The interface of the Software Agent provides a way for the user to define his privacy policy (disclosure or collection) and displays the policy to the user before signature (e.g. validation using a PIN)

- NB: proof of concept language rather than definite solution

SIMple Grammar (excerpt for Subjects)

Consent → I consent to disclose data of category *Category* to a third party only if *Condition-D*

Condition-D → *Party-OK* [and *State-OK*]

State-OK → *Var* is [less than ‖ more than] *Val* [and *State-OK*]

Category → *String*

Var → *String*

Val → *String*

Purposes → *List*

Categories → *List*

List → *String* ‖ *String* [*End-List*]

End-List → , *String* [*List*]

Unit → *Day(s)* ‖ *Week(s)* ‖ *Year(s)*

SIMple Grammar

Party-OK → the aforementioned third party has provided the following pieces of information pursuant to this disclosure of data:

1. His identity [with certificate from Privacy Certification Authority in *List*] [and such identity belongs to *List*.]
2. His verification level [with certificate from Privacy Certification Authority in *List*] [and such verification level is at least *Number* (see definitions below).]
3. …

SIMple Grammar

3. His privacy policy with respect to the aforementioned category of data and such policy includes the following commitments :
   - Use only this data for the following purpose(s): *Purposes*.
   - [Delete this data within a delay of *Number Unit*.]
   - [Not transfer or disclose this data to any other third party. ‖ Transfer this data always accompanied with the present privacy and only to third parties allowed to receive this data according the present privacy policy after commitment of such third party to respect this privacy policy [provided I am previously informed of such disclosure and the identity of the recipient of the data [and I have given my consent before such disclosure.]]]
   - [Ensure that any Valid Request from my side to access such data will be satisfied [within a delay of *Number Unit*.]]
   - [Ensure that any Valid Request from my side to erase such data will be satisfied [within a delay of *Number Unit*.]]
   - [Ensure that any Valid Request from my side to modify such data will be satisfied [within a delay of *Number Unit*.]]

Agreement between SAP and Subject

The Software Agent shall provide an appropriate interface for the Subject

(i) to define his Disclosure Policy in the Certified Language set forth below and

(ii) to accept it using his PIN code after the complete Disclosure Policy has been displayed or printed and he has been asked to confirm his approval.

The Subject shall be able to revise his Disclosure Policy at any time. In the case where the Subject would revise it, the new Disclosure Policy shall be applied by the Software Agent as soon as it has been confirmed by the Subject.

The Software Agent Provider warranties that no personal data from the device (data which is subject to the current Disclosure Policy) shall be disclosed to any third party by the Software Agent unless such disclosure meets all the requirements set forth in the then current Disclosure Policy.

Formal Model

Each Software Agent is associated with sets of tuples of execution traces :

- Event trace E1, …, En
- Time trace T1, … Tn
- State trace S1, …, Sn
- Data Space trace D1, …, Dn

State: Variables → Values

Variables include context variables (e.g. localization) as well as policy parameters such as My-identity, My-authority, My-level, δ (reply delay)

Data Space: Categories → Values $\oplus$

(Categories, Identities) → (Times, Values, Sticky-policies, Entities)

Compliance property for Subject Agents

$\forall$i, Ei = Data-disclosure(En1, En2, Id, Ca, Va, Po) $\Rightarrow$

$\exists$ j, j = MRi (S-Define-disclosure (*), E) and

Ej = S-Define-disclosure (Dp) and

$\exists$ k, k = JBi (Disclosure-request (En1, En2, Ca, *, *, *),
Data-disclosure (En1, En2, Id, Ca, *, *), E) and

Ek = Disclosure-request (En1, En2, Ca, Id', Ve', Co') and

Id = Si (My-identity) and Di (Ca) = Va

Po = (Id, Ve, Co, Cx) and and Co' = Co

$\forall$ Ca", Ca $\subseteq$ Ca"

Po $\Rightarrow$ Dp (Ca") and

Dp(Ca") = (Id", Ve", Co", Cx") and

Id' $\surd$ Id" and Ve' $\surd$ Ve" and Si $\surd$ Cx" and

Event matching

- MRi (EP,E) = greatest j such that j < i and Match(Ei,EP)
- JBi (EP1, EP2, E) = greatest j such that j < i and Match (Ei,EP1)

  and $\forall$ k, j < k < i , $\neg$ Match (Ei,EP2)

  EP, EP1, EP2 $\in$ Event-Patterns

The definitions of $\Rightarrow$ and $\surd$ are based on set inclusion (if A $\subseteq$ B then A $\Rightarrow$ B), natural number ordering (if n ≤ m then n $\Rightarrow$ m) and $\perp \Rightarrow X \Rightarrow \nabla$.

Global compliance property

Authorization to keep data:

If the value of a subject is in the data space of a controller, then the subject must have defined at some stage a privacy policy allowing this controller to receive this data.

$\forall$ Id, Id', Ca, i

Trace(Id) = (E, T, S, D) and

Trace(Id') = (E', T', S', D') and

Di(Ca, Id') $\neq \perp$

$\Rightarrow$

$\exists$ j, E'j = S-Define-disclosure(Dp) and

Dp(Ca) = (Id", *, *, *) and

Id $\surd$ Id"

Link with Software Agent implementations

Only assumptions on the implementation language: trace semantics

- NB: requires the implementation of all context variables

Abstraction function:

Concrete Traces → Consistent Abstract Traces

NB: the abstraction function leaves out from the concrete traces all information which is not relevant at the abstract level (e.g. negotiation steps or operation unrelated to personal data management).

Verification of compliance of Software Agent implementations

Formal semantics of the implementation language:

Software Agent Implementation → Set of Concrete Traces

Abstraction function:

Set of Concrete Traces → Set of Abstract Traces

What has to be proved: the abstracts traces associated with the Software Agent Implementation satisfy the local compliance properties

Benefits: modular and abstract reasoning

Generality of the approach

Key issue:

No constraint on the abstraction function between concrete and abstract traces

Consequence:

The abstract model does not make any assumption with respect to the concrete communication scheme (point to point or broadcast, synchronous or asynchronous, etc.) or execution model (synchronous, asynchronous)

Benefits: Generality and separation of issues

Example: Broadcast communication scheme through a tuple space $\Rightarrow$ the abstraction function maps output/input actions onto communication event pairs in the abstract traces (not necessarily one-to-one mapping)

Complementary verifications

Aspects which cannot be verified formally have to be documented and checked manually:

- User interface (faithful presentation of information and unambiguous interactions with the user)
- Mapping from concrete to abstract traces (faithful correspondence, no hidden access to personal data or masked interactions)
- Implementation of access control (faithful implementation of purpose restrictions).

Both formal and informal verifications can be part of a certification process.

Back to the legal analysis (1/3)

Consent is a unilateral act rather than a contract between the Subject and the Controller:

No contractual relationship between the Subject and the Controller in our solution, but unilateral declarations.

Software Agents should be considered as any software components (objects rather than legal entities):

Contract between the Software Agent Provider and the Subject: the Provider should be liable for the behavior of the Software Agent.

Additional measure: the Software Agent can be certified by an independent entity.

Back to the legal analysis (2/3)

Consent should be:

Free: the specification of consent is independent of any other consideration (e.g. negotiation or business transaction)

Specific: the logic used to specify consent is as precise as possible (hierarchies of categories, of purposes, context, time, etc.)

Informed: the framework provides two levels of information: at the time of policy definition (to the human – Subject or Controller) and before disclosure (to the Software Agent)

Unambiguous: simple natural language with a well defined mathematical semantics (possibly certified by an independent authority)

Back to the legal analysis (3/3)

Written form not always required, but desirable for a better protection of the subjects (for proof if not for validity). Electronic documents are admitted as well as hand-written documents:

Execution traces required for Controllers and can be checked by auditors (either automatic or human).

Conclusion on legal issues

Our claim: Subjects are better protected with a reliable Software Agent because they are not overwhelmed with repeated requests $\Rightarrow$ they have time to take informed, unambiguous and well considered decisions concerning the general management of their personal data.

But: “reliability” of the agent is crucial; the objective of the technical framework is to reduce risks (or enhance trust).

Further work

- Integration of security issues (confidentiality, authenticity, integrity): at the formal model level (security protocol) and at the implementation level (including code integrity?)

- Identity and/or trust management

- Examples of refinement towards specific implementation languages (point to point communication and broadcast)

- Implementation of the framework (Software Agents, interfaces, policy verification)

- Semi-formal specification of the certification process (inspiration from business models, Common Criteria)

- Quantitative risk analysis ?