Static Checking of GDPR-Related Privacy Compliance

SHUKUN TOKAS

DOCTORAL RESEARCH FELLOW,

RELIABLE SYSTEMS (PSY) GROUP,

UNIVERSITY OF OSLO

Outline

- GDPR
- GDPR Formalization
  - Role, Purpose, & Access Restrictions
- Policy Definition Language
- Formalization of Policy Compliance
- Type-effect System
  - *Static Compliance Checking : Case study*
- Runtime System
- Conclusion

Research Problem

To investigate how to formalize fundamental privacy principles and to provide a built-in ability (in language) to fulfill data protection obligations under the GDPR.

General Data Protection Regulation (GDPR)

- Data Protection and Privacy Regulation in **EU**

- **Central Principle** : “Organizations collecting and processing personal data must be explicit about
  - how the data will be used and
  - that the data is actually used for the purposes for which it was collected”.

  - *Privacy by Design* (article 25)
    - Identify privacy requirements at design stage => embed them into subsequent implementation

GDPR Formalization

- Principals
- Purpose
- Access Rights

- Consent (Static Notion)

Principals

- describes who can access sensitive data
- given by interface
- Organized in open ended hierarchy
  - **Any** : predefined, least specialized interface
  - **I < J**, I is a subinterface of J
  - e.g. Doctor < Nurse

Interface and Cointerface

```
interface Patient{
begin
  with Doctor // Doctor is a cointerface
    void updatePrescription(String newPresc);
end
}
```

- Interface Patient has Doctor as cointerface, restricting callers to Doctor.
- Callee can restrict Callers
- Cointerfaces describe roles.

Purpose

- describes what the system intends to do with sensitive data
- purpose hierarchy, to ensure data will be used only for stated purposes
- Natural language statement => purpose title (unique in application context)
- Explicit language construct

Access Rights

- Describes how sensitive data can be accessed, distinguishing between different operations
- Self, read, write, rincr etc
- Lattice
  - meet and join

Principal + Purpose + Access Right = Policy

- Privacy Policy, in this setting is a statement that expresses permitted use of the sensitive information by declared program entities.
  - restricting *who*, *why*, and *what*
  - Implicit policy for default access
  - implicit policy to support article 15
- Programmer specify these policy on
  - Type Definition : upper bound on permitted use of sensitive information
    - Policy set
  - Method : upper bound on actions performed by program entities
    - Single policy
- Default Policy : no-sensitive information (denoted by ●)

Privacy in Software Ecosystem

- How privacy is situated within a large IT Project?
  - Multiple professions that all interact during various stages of s/w development
- Roles that hold a stake in how the software is developed:
  - Project managers – ensure resource availability, team communication
  - Lawyers – track regulatory issues, aligning software with legal norms
  - Requirement engineers – collect, analyze and manage s/w requirements
  - Designers – translate requirements into design, privacy-related requirements
  - **Programmers –** translate software design into source code
  - Tester – validate, discover ways for privacy violations

Language Setting & Program Constructs (I)

- Setting: active object languages for distributed concurrent OO systems
  - Compositional semantics
- Approach : Type system based enforcement

  - ***How to link policy constructs in programs?***

Ommitted Slides:

- Policy Definition Language, BNF syntax of language
- Policy Compliance Formalization, on policy and policy sets
- Meet and join over policy sets, closure
- Language definition, integration of policy definition with core language
- Static-type system,
  - Class, Interface, Method and statements.
- Case study
  - Policy specification
  - Integrating policy with core language
  - Static compliance checking

Runtime Compliance

- Formalize Policy Soundness
  - Policy level obtained at runtime is greater or equal to one calculated by static policy typing
  - i.e syntactic compliance implies semantic compliance
- Theoretical results
  - Soundness
  - Runtime compliance
  - Progress

Conclusion

- Investigated opportunities with the GDPR from language–based perspective.
- Focused on *Privacy by Design* principle.
- Defined language for formulating policy.
- Formalized
  - Static privacy policies.
  - Notion of policy compliance.
  - Rules for policy compliance (given by extended type and effect system).
  - Runtime Compliance.

Healthcare Example

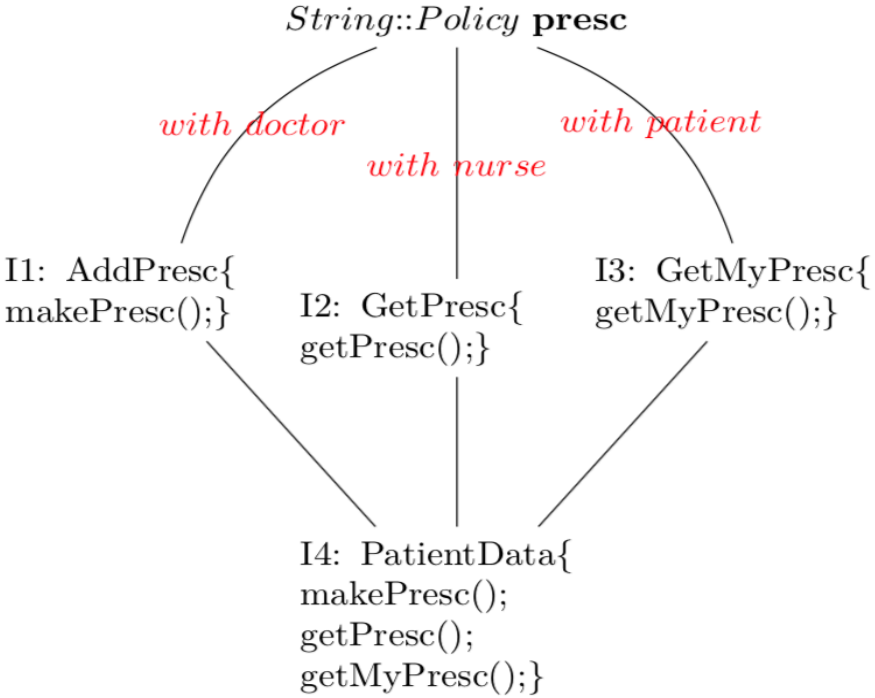

Figure 1: Healthcare Example