Dynamic generation of access control policies from social policies

L. Thomas van Binsbergen, Milen G. Kebede, Joshua Baugh, Tom van Engers, Dannis G. van Vuurden

Informatics Institute, University of Amsterdam
m.g.kebede@uva.nl

November 3, 2021

Motivation

Problem

- Assuring compliance is labour intensive, costly and complex
- Conventional Access Control Techniques have limitations in capturing and enforcing policies from social norms such as the General Data Protection Regulation(GDPR)
- There are not many policy specification languages for specifying both social policies and system-level policies

Goal

- Enable the implementation of legally-aware data sharing infrastructure

Overview

1. What is eFLINT?
2. Extensions to eFLINT
3. The DIPG registry use-case
4. Evaluation
5. Conclusion

Domain Specification Language

- Formalizes norms in social policies (for e.g. GDPR, DSA) and System-level policies(access control policies).
- Normative foundation in Hohfeld's framework - power-liability and duty-claim relations

Interpretations and scenarios

- eFLINT semantics are formalized as transition systems
- Facts, actions, events and duties change over time due to the effects of actions and events

Assessing Compliance

- Action-Compliance : every action labelling a transition is enabled in the source configuration
- Duty-Compliance : all duties in all configuration are not violated

eFLINT Example

GDPR – Article 6(1)(a):

> *Personal data can be collected for a specific purpose if consent has been given for that purpose*

GDPR – Article 5(1)(d):

> *Data must be accurate for purpose specified*

```
1 Act collect-personal-data
2   Actor controller
3   Recipient subject
4   Related to data, processor, purpose
5   Conditioned by accurate-for-purpose(data, purpose), subject-of(subject,data)
6   Creates processes(processor, data, controller, purpose)
7   Holds when consent(subject, controller, purpose)
```

Extensions to the eFLINT language

Extend Keyword

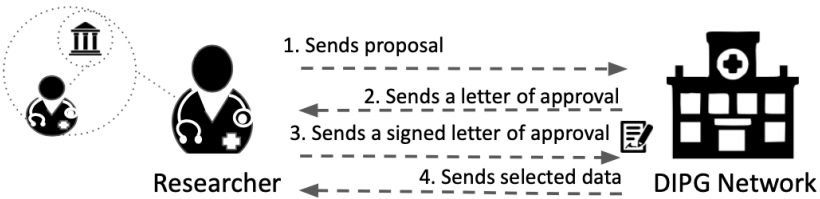
- To add derivation clauses to type definitions, to add pre- and post-conditions to action types and to adding violation conditions to duty types
- Modular and declarative extension of types

Sync Keyword

- Automate high level compliance decisions through lower level enforcement mechanisms

Use-case DIPG Registry

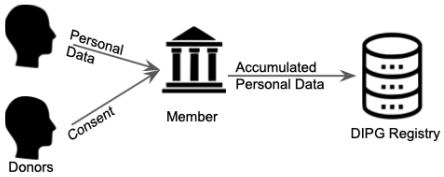
- Diffuse Intrinsic Pontine Gliomas(DIPG) registry: rare disease repository that allows researchers to access patient data that can lead to discovering new treatment and prognosis factors.



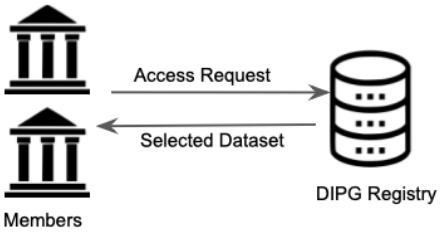
Compliance questions

According to the GDPR and the DIPG regulatory document:

1. What conditions need to be fulfilled by a member before making data available?



2. What conditions need to be fulfilled when accessing data from the registry?



Compliance Question 1

DIPG Regulatory document – Article 4(2):

Members should transfer data to the DIPG registry in a coded form only

```
1 Fact coded Identified by dataset
2 Act make-data-available
3   Actor institution
4   Recipient  dcog
5   Related to  dataset
6   Conditioned by  coded(dataset) Holds when member(institution)
```

Compliance Question 1

```
1 Extend Act make-data-available Syncs with (Foreach donor:
2   collect-personal-data(controller = institution
3                         ,subject    = donor
4                         ,data       = dataset
5                         ,processor  = "DCOG"
6                         ,purpose    = "DIPGResearch")
7     When subject-of(donor, dataset))
```

An institution can make a dataset available when (for each donor (subject) in the dataset):

- The institution should be a member of the consortium
- Data should be coded
- Consent is given by the donor for the processing of their personal data by the DCOG for the purpose of DIPGResearch
- Data should be accurate for the purpose DIPGResearch

Compliance Question 2

```
1 Fact actor
2 Fact recipient
3 Fact asset
4 Act  access Actor actor Recipient recipient Related to asset
5     Holds when read(actor ,recipient ,asset), write(actor ,recipient ,asset)
6 Act read Actor actor Recipient recipient Related to asset Syncs with
      access(actor,recipient,asset)
7 Act write Actor actor Recipient recipient Related to asset Syncs with
      access(actor,recipient,asset)
```

Read and write action are instances of access action

```
1 Extend Act read Holds when (Exists project, institution:
2   selected(asset,project) && approved(project,institution) &&
      affiliated(actor, institution))
```

An actor can *read* an asset when (there exists a project and an institution for which):

- The asset is selected for the project
- The project is approved for the institution
- The actor is affiliated with the institution

More Examples

Granting read and write permission to dataset owners

```
1 Fact owner-of Identified by institution * dataset
2 Extend Act make-data-available Creates owner-of(institution, dataset)
3 Extend Act write Holds when affiliated(actor,institution)
4                             && owner-of(institution, asset)
5 Extend Act read  Holds when affiliated(actor,institution)
6                             && owner-of(institution, asset)
```

An actor can write or read an asset when:

- The actor is affiliated with an institution
- The institution is the owner of the asset
- the previous extension to read holds

Evaluation

Experimentation based on Haskell implementation of eFLINT

First scenario

Members make data available to the registry with eFLINT deciding on compliance according to DIPG network regulatory document and GDPR

- Member institutions can run an instance of eFLINT server

```
?Enabled(make-data-available(<X>),DCOG,<Y>)
```

Second scenario

Researcher of a member attempting to read a dataset in the DIPG registry with eFLINT determining whether an access request is permitted

- An eFLINT server running alongside the registry

```
?Enabled(read(<X>,DCOG,<Y>))
```

Conclusion

Answer compliance questions

- eFLINT as a Policy Administration Point(PAP) by generating access control policies such as XACML or ODRL policies
- eFLINT is used to specify higher-level concepts (`collect-personal-data`) and lower-level concepts (`read` and `write`)
- We introduced eFLINT extensions that enable us to connect higher-level and lower-level concepts : as demonstrated by the generation of access control rules
- The approach in the paper is for a centralised solution

Future work

- Develop a decentralised solution

Dynamic generation of access control policies from social policies

L. Thomas van Binsbergen, Milen G. Kebede, Joshua Baugh, Tom van Engers, Dannis G. van Vuurden

Informatics Institute, University of Amsterdam
m.g.kebede@uva.nl

November 3, 2021