Towards a policy enforcement infrastructure using eFLINT

Lu-Chi Liu, Milen G. Kebede, Thomas van Binsbergen, Tom van Engers

Informatics Institute, University of Amsterdam
l.liu@uva.nl

April 11, 2022

Motivation

Problem

- Assuring compliance is labour intensive, costly and complex
- Compliance processes are predominately manual
- Example: Human error, broad interpretations of the GDPR, inability to show compliance proof

Goal

- Formalize wide variety of normative sources such as regulations and contracts
- Automate required monitoring, control and enforcement of such norms

eFLINT

High-level, declarative, and modular domain-specific language

- Makes it easy to adapt to changing interpretations of laws/norms/policies
- Enables analyses such as model checking or property testing

Tracking changes of unfolding scenario's

- Automated assessment of concrete scenarios
- Scencario's can be specified in the language directly (e.g. for testing, debugging)
- Or can be produced dynamically as system runs (monitoring events/actions)

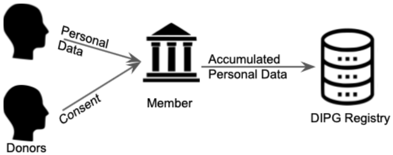
Specifying policies at various levels of abstraction

- Example: GDPR regulation, consortium data sharing agreement, access control
- The policies can be explicitly linked, enabling reasoning across layers

Compliance questions

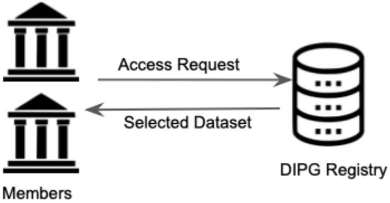
According to the GDPR and the DIPG regulatory document:

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



```
?Enabled(write(<X>,<Y>))
```

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



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

Examples

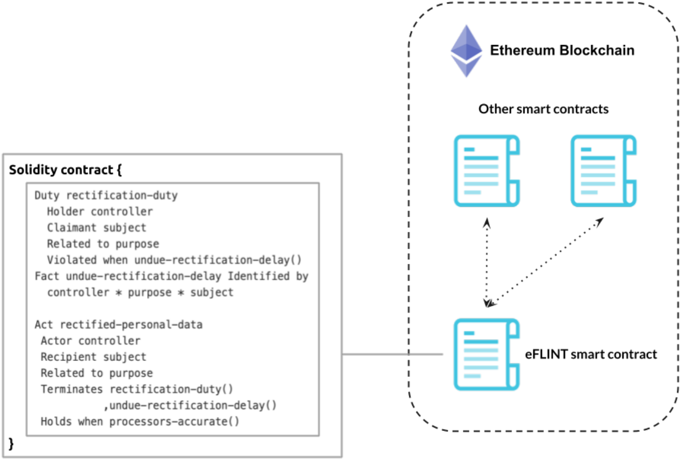
- Writing to the registry is considered making data available (consortium notion)

```
Extend Act write Holds when (Exists member:
  make-data-available(member, DCOG, asset)
    && affiliated-with(actor, member))
```

- Making data available is considered collecting personal data (GDPR notion)

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

Application: eFLINT Smart Contracts



Solidity contract {

```
Duty rectification-duty
  Holder controller
  Claimant subject
  Related to purpose
  Violated when undue-rectification-delay()
Fact undue-rectification-delay Identified by
  controller * purpose * subject

Act rectified-personal-data
 Actor controller
 Recipient subject
 Related to purpose
 Terminates rectification-duty()
           ,undue-rectification-delay()
 Holds when processors-accurate()
```

}

Why are we interested in blockchain?

- Transactions/data are stored (automatically) on blockchain
- Transactions/data cannot be changed or deleted once recorded
- Offer transaction history that can be used for dispute settlement
- Monitor and enforce based on the programmed contracts
- Smart contracts are not contracts - **Expressiveness**

Integrate eFLINT with blockchain

- Raise the level of abstraction
- Reduce the gap between legal documents and smart contracts
- Translate eFLINT into smart contract languages (e.g., Solidity)

From eFLINT to Solidity

eFLINT specifications

```
Fact subject Identified by Alice
Fact controller Identified by institution
Act collect-personal-data
  Actor controller
  Recipient subject
  Related to data, processor, purpose
  Conditioned by accurate-for-purpose(data, purpose), subject-of(subject,data)
  Creates processes(processor, data, controller, purpose)
  Holds when consent(subject, controller, purpose)
```

Solidity scripts

```
type Subject is bytes32;
Subject Alice = Subject.wrap("Alice");
type Controller is bytes32;
Controller Institution = Controller.wrap("Institution");
function collect-personal-data (Controller _controller, Subject _subject,
    Data _data, Processor _processor, Purpose _purpose) public {
    require (
         accurate-for-purpose(data, purpose), "inaccurate purpose"
    ); ...
}
```

From eFLINT to Solidity

eFLINT specifications share similar structures to Solidity contracts.

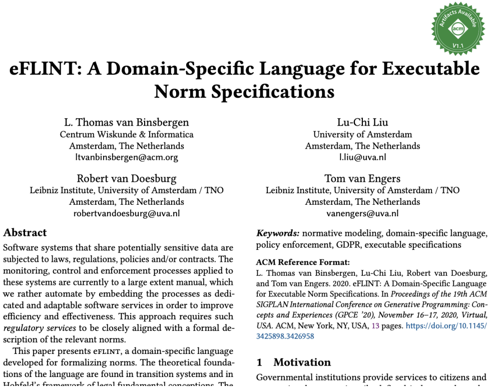
Type variables and assigned identifiers

```
Fact subject Identified by Alice, Bob
Fact controller Identified by institution
```

Function definition: function name, list of parameters, and a statement block

```
Act collect-personal-data
  Actor controller
  Recipient subject
  Related to data, processor, purpose
  Conditioned by accurate-for-purpose(data, purpose), subject-of(subject,data)
  Creates processes(processor, data, controller, purpose)
  Holds when consent(subject, controller, purpose)
```

Towards a policy enforcement infrastructure using eFLINT



eFLINT: A Domain-Specific Language for Executable Norm Specifications

L. Thomas van Binsbergen
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
ltvanbinsbergen@acm.org

Lu-Chi Liu
University of Amsterdam
Amsterdam, The Netherlands
l.liu@uva.nl

Robert van Doesburg
Leibniz Institute, University of Amsterdam / TNO
Amsterdam, The Netherlands
robertvandoesburg@uva.nl

Tom van Engers
Leibniz Institute, University of Amsterdam / TNO
Amsterdam, The Netherlands
vanengers@uva.nl

Abstract

Software systems that share potentially sensitive data are subjected to laws, regulations, policies and/or contracts. The monitoring, control and enforcement processes applied to these systems are currently to a large extent manual, which we rather automate by embedding the processes as dedicated and adaptable software services in order to improve efficiency and effectiveness. This approach requires such *regulatory services* to be closely aligned with a formal description of the relevant norms.

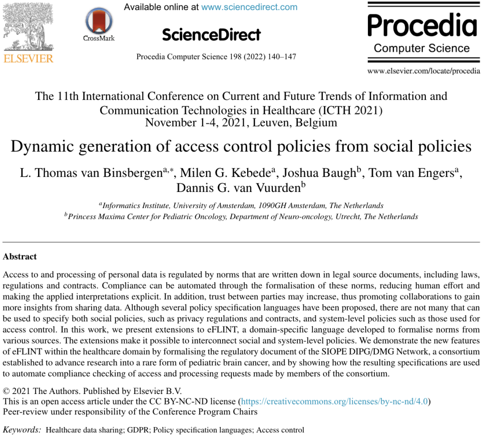
This paper presents eFLINT, a domain-specific language developed for formalizing norms. The theoretical foundations of the language are found in transition systems and in Hohfeld's framework of legal fundamental conceptions. The

Keywords: normative modeling, domain-specific language, policy enforcement, GDPR, executable specifications

ACM Reference Format:
L. Thomas van Binsbergen, Lu-Chi Liu, Robert van Doesburg, and Tom van Engers. 2020. eFLINT: A Domain-Specific Language for Executable Norm Specifications. In *Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences (GPCE '20), November 16–17, 2020, Virtual, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3425898.3426958

1 Motivation

Governmental institutions provide services to citizens and



Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 198 (2022) 140–147

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2021)
November 1-4, 2021, Leuven, Belgium

Dynamic generation of access control policies from social policies

L. Thomas van Binsbergen[a,*], Milen G. Kebede[a], Joshua Baugh[b], Tom van Engers[a], Dannis G. van Vuurden[b]

[a]*Informatics Institute, University of Amsterdam, 1090GH Amsterdam, The Netherlands*
[b]*Princess Maxima Center for Pediatric Oncology, Department of Neuro-oncology, Utrecht, The Netherlands*

Abstract

Access to and processing of personal data is regulated by norms that are written down in legal source documents, including laws, regulations and contracts. Compliance can be automated through the formalisation of these norms, reducing human effort and making the applied interpretations explicit. In addition, trust between parties may increase, thus promoting collaborations to gain more insights from sharing data. Although several policy specification languages have been proposed, there are not many that can be used to specify both social policies, such as privacy regulations and contracts, and system-level policies such as those used for access control. In this work, we present extensions to eFLINT, a domain-specific language developed to formalise norms from various sources. The extensions make it possible to interconnect social and system-level policies. We demonstrate the new features of eFLINT within the healthcare domain by formalising the regulatory document of the SIOPE DIPG/DMG Network, a consortium established to advance research into a rare form of pediatric brain cancer, and by showing how the resulting specifications are used to automate compliance checking of access and processing requests made by members of the consortium.

Keywords: Healthcare data sharing; GDPR; Policy specification languages; Access control