UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Dynamic generation of access control policies from social policies

van Binsbergen, L.T.; Kebede, M.G.; Baugh, J.; van Engers, T.; van Vuurden, D.G.

DOI
10.1016/j.procs.2021.12.221

Publication date
2022

Document Version
Final published version

Published in
Procedia Computer Science

License
CC BY-NC-ND

Link to publication

Citation for published version (APA):
van Binsbergen, L. T., Kebede, M. G., Baugh, J., van Engers, T., & van Vuurden, D. G. (2022). Dynamic generation of access control policies from social policies. *Procedia Computer Science*, *198*, 140-147. https://doi.org/10.1016/j.procs.2021.12.221

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:11 Oct 2026

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.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the Conference Program Chairs

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

1. Introduction

Health-care organisations collect data for various purposes, including the execution of data-driven research projects to improve the efficacy of (personalised) care. The collection and processing of data is governed by (national) laws, (international) regulations, such as the General Data Protection Regulation (GDPR) of the European Commission [5], contracts, and organisational policies – referred to as 'social policies' in this paper. Assuring compliance is labour intensive, costly and complex, especially in situations where data is processed by external parties. To reduce labour and costs, and to raise the compliance level of systems that provide access to or process healthcare-data, it is desirable to automate compliance checking. To achieve this, interpretations of applicable norms have to be formalised and cor-

* Corresponding author. Tel.: +0031 20 525 8626. Email: ltvanbinsbergen@acm.org

1877-0509 © 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the Conference Program Chairs
10.1016/j.procs.2021.12.221

responding enforcement mechanisms should be available within the software ecosystem. Access control is a popular mechanism to enforce policies, especially within the healthcare domain. Conventional access control techniques, however, are shown not to be sufficiently capable to capture the complex conditions encountered in laws, regulations and sharing agreements [3, 11]. Moreover, laws and regulations may differ across jurisdictions and are subject to change, requiring localised variants of software and frequent updates to software.

As a remedy, we employ domain-specific languages to formalise norms as specifications from which software components – referred to as *regulatory services* – can be derived that administer and enforce complex sets of norms. In doing so, we achieve a level of separation of concern that makes it possible to adjust to changing (legislative and jurisdictional) contexts and improve transparency with respect to the applied interpretations of norms and situations.

To experiment with the formal representation of norms, and the derivation of regulatory services, we have developed eFLINT, a domain-specific language for formalising interpretations of norms [23]. The language is based on the fundamental legal concepts of Hohfeld's framework for legal reasoning [7] and on transition system semantics. This combination makes it possible to formalise various types of policies and to assess concrete cases statically (offline) or dynamically (online) for compliance. Moreover, the underlying transition system can be queried to determine the current 'normative positions' of actors, e.g. to establish whether they hold certain permissions or obligations. As such, eFLINT can be used for access control.

In this paper, we extend eFLINT with two new constructs for formalising the dependencies between social policies – such as the role of the GDPR in data sharing agreements – as well as the connection between social policies and system-level policies. In this way, social policies become enforceable in software systems and, conversely, system-level policies acquire legal status. We demonstrate the features of eFLINT through a use-case provided by the European Society for Paediatric Oncology (SIOPE) that has established the DIPG/DMG Network, a data-sharing consortium supporting research on a rare form of brain cancer in children. In particular, we will show how eFLINT can be used to formalise aspects of the GDPR, the DIPG/DMG Network's Regulatory Document, and their interconnection, as well as how the resulting specifications can be used to automate compliance decisions.

2. Preliminaries

EFLINT is designed for formalising norms that can typically be found in social policies, laws, regulations, organisational policies and contracts [23]. As an example, consider the following fragment, adapted from the formalisation of GDPR articles in [23].

```
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)
```

The fragment shows that personal data can be collected for a specific purpose if consent has been given for that purpose (Article 6(1)(a) of the GDPR [5]) and that the data must be accurate for this purpose (Article 5(1)(d)). The former is specified as a derivation rule in the `Holds when` clause as it is *one* of the several conditions under which personal data can be collected lawfully. The latter is specified as a pre-condition as it is *always* required for lawful processing.

The normative foundations of the language are found in Hohfeld's framework for analysing courthouse activities [7]. Hohfeld's framework centres around power-liability and duty-claim relations between actors, stating that for every power holder (`Actor`, above) there is a recipient (`Recipient`, above) affected by the power and for every duty holder there is a claimant. Additional objects parameterising the relation are specified in the `Related to` clause.

The semantics of the language are formalised as a transition system in which states are sets of facts (i.e. Boolean variables) and actions and events trigger transitions between states by terminating and creating facts (i.e. setting the variables to false or true respectively). Notions of 'action-compliance' and 'duty-compliance' are defined over traces in the transition system based on whether actions are enabled and duties are violated. An action is enabled if it holds true (i.e. one of its derivation rules holds true) and if all its pre-conditions hold. A duty is violated if it holds true and if one of its violation conditions holds. The normative relations between actors are derived from the actions and duties that hold true at any given time.

In the next section, we show how eFLINT is used to formalise parts of the regulatory document governing the SIOPE DIPG/DMG Network. The reader is referred to [23] for a more detailed description of the syntax and semantics of the language, including also the formalisation and application of duties (which are not occurring in this paper).

3. The SIOPE DIPG/DMG Network Regulatory Document

Diffuse Intrinsic Pontine Glioma (DIPG), also known as diffuse midline glioma (DMG), is a rare pediatric brain cancer for which there is no curative treatment, despite decades of clinical trials [10]. Children suffering from DIPG face a dismal prognosis, with a median overall survival (OS) of eleven months and a two-year OS of less than 10% [6]. In order to advance DIPG research, the SIOPE DIPG/DMG Network and Registry were established in 2011. The registry holds information on DIPG patients across Europe and a partner registry in North America – the International DIPG/DMG Registry – includes patient data primarily from the USA, Canada and Australia, with additional international members [2]. The registry serves to improve DIPG research by granting members (conditional) access to selected datasets in order to perform analyses with more data points and thus higher efficacy. Members submit clinical data from their institution to the DIPG Registry via a secure online CRF-structured web application and database[1]. Imaging data is uploaded into a separate repository and review system[2]. All data is coded and de-identified at upload.

In this section, we present eFLINT fragments that capture certain terms and articles defined in the SIOPE DIPG/DMG Regulatory Document, a consortium agreement between network members available online [22]. The document is especially interesting for this work, as it makes explicit connections to the GDPR (detailed further in Section 4) and regulates the process of acquiring access to data.

The first section defines several terms that we formalise as fact-type definitions in eFLINT. For example, the following fragment gives a formalisation of the definition of the term “Researcher”.

```
Fact person
Fact institution
Fact member     Identified by institution
Fact affiliated Identified by person * institution
Fact researcher Derived from institution  When member(institution)
                           ,person        When affiliated(person, institution) && member(institution)
```

Fact-type definitions without an `Identified by` clause have atoms as values, such as `"Alice"`, whereas facts with an `Identified by` clause establish either a predicate over another type (e.g. `member`) or establish a relation between several types (e.g. `affiliated`). The derivation rules of the fact-type `researcher` establish that a Researcher is a person or an institute for which holds that, it is either a member (as an institute) or affiliated with a member (as a person).

The following fragment expresses that members can make data available under the condition that the data is ”Coded” – defined as “pseudonymised pursuant to the GDPR”.

```
Fact coded Identified by dataset
Act make-data-available
  Actor           institution
  Recipient       dcog
  Related to      dataset
  Conditioned by  coded(dataset)
  Holds when      member(institution)
```

The Dutch Childhood Oncology Group (DCOG) facilitates the SIOPE DIPG/DMG Registry and is therefore listed as recipient. Note that we assume a mechanism to determine whether a dataset is properly coded, i.e. whether `coded(dataset)` holds for a given dataset is to be provided as input.

In order to get access to data, Researchers must send project proposals for consideration by the executive committee of the network, reviewing the proposal for merit, feasibility and ethical approval. If approved, a formal approval letter is issued by the executive committee outlining conditions of use. Prior to data transfer, a data sharing agreement, included in the approval letter, must be signed and returned by the Researcher. The registry coordinator will then select and send data appropriate for the project. This process is visualised in Figure 3. Section 6 demonstrates the

[1] https://dipgregistry.eu
[2] https://mdpe-hit.de/

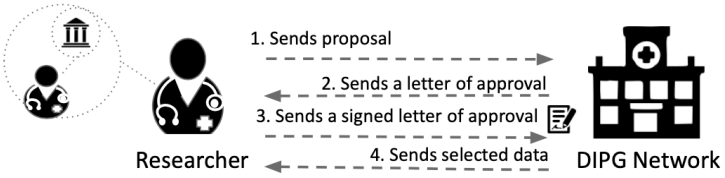

Fig. 1. The process of having projects approved for access to data

automation of access to selected data sets by applying the eFLINT specifications developed throughout this paper. The formalisation of the act of selecting data, given below, is key to this automation.

```
Act select-data
  Actor        ecommittee
  Recipient    researcher
  Related to   project, dataset
  Creates      selected(dataset, project)
  Terminates   duty-to-select-data(ecommittee, researcher, project)
  Holds when   approved(project, researcher)
```

Data can only be selected for a Researcher when related to a project approved for that Researcher[3]. The effects of executing an instance of `select-data` are the creation of an instance of `selected`, recording which dataset has been selected for which project, and the termination of an instance of the duty `duty-to-select-data` (not given).

4. Modular type extensions and synchronisation clauses

This section formalises the interconnections, described in the Regulatory Document, between certain GDPR concepts and concepts in the Regulatory Document and do so by applying new features of the eFLINT language. The following fragment shows how derivation rules (here given as `Derived from` clauses) are used to 'generate' instances of GDPR concepts (such as `subject` and `controller`) based on existing instances of DIPG concepts (such as `donor` and `member`). For example, the third line states that member institutions are considered processors of personal data in addition to the DCOG (represented by the string `"DCOG"`). The fourth line states that projects are considered as purposes with respect to the GPDR, i.e. purposes for which patients (donors) may choose to give consent.

```
Extend Fact subject    Derived from donor
Extend Fact controller Derived from member.institution
Extend Fact processor  Derived from "DCOG", member.institution
Extend Fact purpose    Derived from "DIPGResearch", project
Extend Fact data       Derived from dataset
```

The `Extend` keyword is a novel addition to eFLINT for adding derivation clauses to type definitions (as shown above for fact-types), adding pre- and post-conditions to action-types (and synchronisation clauses, introduced later), and to adding violation conditions to duty-types. Adding a derivation rule or condition to a type-definition via `Extend` or as part of the original definition is equivalent. An instance of a type is considered to hold true if there exists a derivation rule that generates that instance. Similarly, an instance is created or terminated by an instance of an action-type (or event-type) if there exists a post-condition demanding this. In this sense, derivation rules and post-conditions are 'disjunctive'. The same applies to violation conditions of a duty; a duty is violated if it holds true and (at least) one of its violation conditions holds. The pre-conditions of action-types are 'conjunctive', i.e. for an action to be permitted, all its pre-conditions must hold. The `Extend` keyword is associative and commutative and enables modular and declarative extension of types. The extension mechanism is declarative because the order in which extensions are given does not matter. The extension mechanism is modular because, in term of this example, the GDPR concepts, the DIPG concepts and the rules connecting these concepts have been defined in isolation. This permits reuse between specifications as well as defining alternative specifications that, when implementing the same 'interface', can be used

[3] That Researcher is listed as a recipient does not indicate that they *receive* the selected dataset; instead they receive access, as shown in Section 5

as drop-in replacements. The type-checking performed on declarations is performed also on extending clauses. The main limitation is that eFLINT does not currently have a module system for introducing separate namespaces.

Another feature added to eFLINT is illustrated by the following code fragment, providing an extension of the `make-data-available` type defined in the previous section.

```
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))
```

The type is extended with a 'synchronisation clause' that demands that when an instance of the action is executed, this instance must synchronise with certain instances of the `collect-personal-data` type (given in Section 2). In terms of eFLINT's transition system semantics, the transitions encoded by these act-type instances are to be performed simultaneously. As a result, the effects of the synchronised actions accumulate (i.e. the actions happen simultaneously) and the pre-conditions and post-conditions of all actions are relevant. In the example above, `make-data-available` effectively inherits the pre- and post-conditions of certain instances of `collect-personal-data`. Which instances these are, is determined by the body of the `Foreach` expression of the synchronisation clause. In this case, there is an instance for each donor for which holds that the donor is a subject of the dataset being made available (assuming there is a mechanism to detect for which donors consent is required and whether they have given their consent). Informally, the extension states that the action of making data available is to be understood (also) as collecting personal data under the GDPR, with the controller being the member institution making the data available, the DCOG processing the data on behalf of the member, and doing so for the purpose of DIPG research. An important consequence of this synchronisation is that instances of `make-data-available` are not permitted if one of the affected donors has not given explicit consent for the processing of their personal data for the purpose of DIPG research. In this way, the corresponding part of the DIPG Regulatory Document is formalised and can be used to enforce consent as a condition on making data available.

5. Deriving access control rules

In this section, we use the new features of eFLINT to connect the specifications concerning Regulatory Document and GDPR norms with access control rules. The connection is made by defining read and write action-types in eFLINT and subsequently extending their definitions. The separation between initial definition and subsequent extension is made to emphasise that the read and write actions are not application-specific, whereas the extensions are specific to the combined DIPG-GDPR specifications. The read and write actions are defined as follows:

```
Fact actor
Fact recipient
Fact asset
Act  access Actor actor Recipient recipient Related to asset
     Holds when read(actor,recipient,asset), write(actor,recipient,asset)
Act  read   Actor actor Recipient recipient Related to asset Syncs with access(actor,recipient,asset)
Act  write  Actor actor Recipient recipient Related to asset Syncs with access(actor,recipient,asset)
```

The derivation rules and synchronisation conditions establish a relation between the `access`, `read` and `write` types in which reading or writing an asset is considered accessing an asset.

In the following fragment, `make-data-available` is extended to reflect that a member making data available is considered the owner of that data. Read and write access is granted to persons affiliated with owners of datasets.

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

The following fragment specifies that researchers of other institutions receive read access to a dataset when the institution they are affiliated with has had a project approved for which this dataset has been selected:

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

In the next section we reflect on how the established specifications can be used to automate compliance.

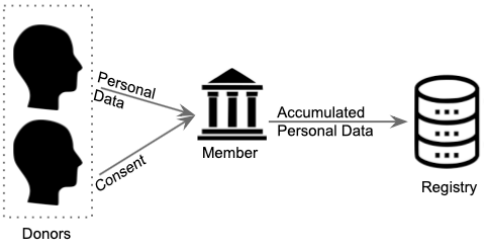

Fig. 2. The process of making data available

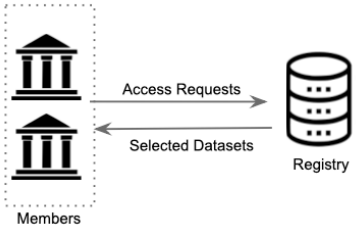

Fig. 3. The process of accessing data

6. Evaluation

The code fragments presented previously are taken (and modified slightly for clarity) from larger projects in which DIPG and GDPR concepts are formalised, and are supplemented with the code for interconnecting GDPR, DIPG and access control specifications. To demonstrate the feasibility and desirable properties of our approach, we ran experiments based on the Haskell implementation of eFLINT [23]. This implementation provides a (TCP and HTTP) server to which commands can be sent triggering the execution of actions and the creation/termination of facts, possibly resulting in violations (of powers or duties) returned as part of the response.

Experiments have been performed based on simulations of two scenarios. The first scenario involves members of the SIOPE DIPG/DMG Network making data available to the Registry, with eFLINT deciding on compliance according to the presented formalisation of the DIPG Regulatory Document and the GDPR. To automate this decision, an instance of the eFLINT server can be running at every member institution. The server is asked to evaluate the query `?Enabled(make-data-available(<X>,DCOG,<Y>))`, with `<X>` and `<Y>` placeholders for the values identifying the member itself and the relevant dataset respectively. To answer this query, the eFLINT server needs to be able to determine whether the dataset is coded, which patients of the member contributed to dataset, whether these patients have given consent and whether the data is accurate for the purpose of DIPG research, as shown by the (extended) definitions of `make-data-available` and `collect-personal-data`. This information is sent to the eFLINT server as input alongside or prior to the query. For example, if a member institution runs a portal on which patients can modify their consent, then changes in consent can be communicated with the eFLINT server automatically.

The second scenario involves a researcher of a member attempting to read a dataset in the SIOPE DIPG/DMG Registry. An eFLINT server running alongside the registry can be used to determine whether such an access request is to be permitted (thus behaving as a policy decision point) by answering the query `?Enabled(read(<X>,DCOG,<Y>))` where `<X>` and `<Y>` are placeholders for the values identifying the researcher and the dataset respectively. To answer this query, the eFLINT server needs to know with which (member) institution the researcher is affiliated (if any) and whether that institution has a project approved for which this dataset is selected by the executive committee, as shown by the extended definition of `read`. This information can be provided to the eFLINT server automatically if the registry runs software that supports the process of proposing/approving projects and selecting data for projects. Events in this software system can then trigger input being sent to the local eFLINT server. These scenarios show that with eFLINT, access control is dynamic in the sense that permissions are subject to change depending on (user) actions within the SIOPE DIPG/DMG (member or registry) software. The (legal) powers that give rise to changes in permissions are specified explicitly and are assigned to human actors or legal entities in eFLINT, whereas in attribute-based access control systems changes to attributes are not explicitly associated with (legal) powers.

In our experiments, we have executed the scenarios described above by simulating the SIOPE DIPG/DMG Registry software, member software, and human actions in order to generate various concrete instances of these scenarios as test-cases. Real-world prototyping is to be executed in future work. However, this is expected to demonstrate that the current eFLINT implementation is not practical for this purpose – the implementation is derived directly from the formal operational semantics to guarantee correctness without concerns for runtime efficiency.

Rather than using eFLINT for policy decision making directly, it is also possible to use eFLINT in a policy administration point by generating access control policies such as XACML [12] or ODRL policies [9]. The generation

of such policies can either be event-driven, i.e. whenever there is a change to the knowledge base maintained by the eFLINT server, or request-driven, i.e. whenever an access requests is made.

The scenarios described here demonstrate how compliance decisions regarding data management can be automated as part of current SIOPE DIPG/DMG Registry practices. However, a decentralised solution without central registry to collect all data is preferred, increasing control members have over the data they contribute to the SIOPE DIPG/DMG Network. In future work we wish to demonstrate the application of eFLINT in a decentralised solution based on the 'digital datamarket' concept [20] in which analysis algorithms travel to data (rather than vice versa) and members (as well as their patients) can at any moment withdraw their data or change their conditions for access (such as consent).

7. Related work

Access control is the process of determining the permissibility of access requests such as read or write actions on a data object based on policies. Several access control models have been proposed in literature. Early models include Discretionary Access Control (DAC) [19], Mandatory Access Control (MAC) [13], and Role-Based Access Control (RBAC) [18]. These models fail to capture dynamic and temporal aspects of access permissions. The attributes of Attribute-Based Access Control (ABAC) [8] from which permissions are computed can change in between requests, the Usage CONtrol (UCON) model considers conditions to hold during access events [15], and in the access control model proposed by Bertino et al. [3], periodic intervals are associated with permissions. The eXtensible Access Control Markup Language (XACML) is a popular attribute-based access control language and enforcement engine [12]. In the context of the Health Insurance portability and Accountability Act (HIPAA), [4] provides arguments as to why XACML cannot meet certain requirements from privacy law. The Open Digital Rights Language (ODRL) is a policy expression language to describe conditions under which certain actions are permitted on (digital) assets [9]. ODRL has been used in various domains such as licensing [17] and privacy [24]. The fact-type definitions of eFLINT can be used to represent attributes, roles and ownership and the concept of power supports the dynamic creation and termination of permissions. However, eFLINT events are discrete and can therefore not be used for usage control in the way described by the UCON model. To which extent this is a limitation of eFLINT is to be explored in future work.

Various languages have been introduced to formally specify social policies and contracts. Symboleo is designed to specify contracts and has similar foundations as eFLINT, being based on the event calculus, state machine models and Hohfeld's legal framework [21]. Symboleo has embedded notions of contract states such as suspension and termination triggered through the execution of powers. The Institutional Action Language (InstAL) [14] is another domain-specific language for modelling both regulative and constitutive norms in terms of duties and powers, as well as a set of tools for executing models via a translation to Answer Set Programming (ASP) [14]. LegalRuleML is an extension of RuleML based on deontic positions (permissions and obligations) for describing norms through rules, considering important legal and logical aspects such as defeasibility, the semantics of negation, temporal properties and jurisdictions [1]. DAPRECO is a repository of LegalRuleML rules representing provisions of the GDPR [16].

8. Conclusion

In this work, we presented the formalisation of rules from legislation and other regulatory documents using the eFLINT language, taking the GDPR and the regulatory document of the SIOPE DIPG/DMG Network as a use case. We have also introduced extensions to eFLINT that enable the formalisation of dependencies between, in our use case, the GDPR and SIOPE DIPG/DMG Regulatory Document and the interconnections of these regulations with system-level policies such as access control policies. As a result, we can automate high-level compliance decisions through lower-level enforcement mechanisms. The centralised solution for the SIOPE DIPG/DMG Network discussed in this paper marks a first step towards an envisioned decentralised solution to be presented in future work.

Acknowledgements. This work is part of the EPI project supported by NWO in the Commit2Data – Data2Person program (628.011.028) and the AMdEX Fieldlab project supported by Kansen Voor West EFRO (KVW00309).

References

[1] Athan, T., Governatori, G., Palmirani, M., Paschke, A., Wyner, A., 2015. LegalRuleML: Design Principles and Foundations. Springer International Publishing, Cham. pp. 151–188. doi:10.1007/978-3-319-21768-0_6.

[2] Baugh, J., Bartels, U., Leach, J., Jones, B., Chaney, B., Warren, K.E., Kirkendall, J., Doughman, R., Hawkins, C., Miles, L., et al., 2017. The international diffuse intrinsic pontine glioma registry: an infrastructure to accelerate collaborative research for an orphan disease. Journal of neuro-oncology 132, 323–331. doi:10.1007/s11060-017-2372-5.

[3] Bertino, E., Bettini, C., Ferrari, E., Samarati, P., 1998. An access control model supporting periodicity constraints and temporal reasoning. ACM Trans. Database Syst. 23, 231–285. doi:10.1145/293910.293151.

[4] Chowdhury, O., Chen, H., Niu, J., Li, N., Bertino, E., 2012. On XACML's Adequacy to Specify and to Enforce HIPAA, in: Gunter, C.A., Peterson, Z.N.J. (Eds.), 3rd USENIX Workshop on Health Security and Privacy, USENIX Association. doi:10.5555/2372366.2372381.

[5] Council of the EU, 2016. General Data Protection Regulation. Official Journal of the European Union 59.

[6] Hoffman, L.M., Van Zanten, S.E.V., Colditz, N., Baugh, J., Chaney, B., Hoffmann, M., Lane, A., Fuller, C., Miles, L., Hawkins, C., et al., 2018. Clinical, radiologic, pathologic, and molecular characteristics of long-term survivors of diffuse intrinsic pontine glioma (dipg): a collaborative report from the international and european society for pediatric oncology dipg registries. Journal of Clinical Oncology 36, 1963. doi:10.1200/JCO.2017.75.9308.

[7] Hohfeld, W.N., 1917. Fundamental legal conceptions as applied in judicial reasoning. The Yale Law Journal 26, 710–770. doi:10.2307/786270.

[8] Hu, V.C., Kuhn, D.R., Ferraiolo, D.F., 2015. Attribute-based access control. Computer 48, 85–88. doi:10.1109/MC.2015.33.

[9] Iannella, R., Villata, S., 2018. ODRL information model 2.2. W3C Recommendation .

[10] Jansen, M., Van Vuurden, D., Vandertop, W., Kaspers, G., 2012. Diffuse intrinsic pontine gliomas: a systematic update on clinical trials and biology. Cancer treatment reviews 38, 27–35. doi:10.1016/j.ctrv.2011.06.007.

[11] Kirrane, S., Mileo, A., Decker, S., 2017. Access control and the Resource Description Framework: A survey. Semantic Web 8, 311–352. doi:10.3233/SW-160236.

[12] OASIS eXtensible Access Control Markup Language (XACML) Technical Committee, 2017. eXtensible Access Control Markup Language (XACML) Version 3.0 Plus Errata 01.

[13] Osborn, S.L., 1997. Mandatory access control and role-based access control revisited, in: Youman, C.E., Coyne, E.J., Jaeger, T. (Eds.), Proceedings of the Second Workshop on Role-Based Access Control, RBAC 1997,November 6-7, 1997, ACM. pp. 31–40. doi:10.1145/266741.266751.

[14] Padget, J., Elakehal, E.E., Li, T., De Vos, M., 2016. InstAL: An institutional action language, in: Social coordination frameworks for social technical systems. Springer, pp. 101–124. doi:10.1007/978-3-319-33570-4_6.

[15] Park, J., Sandhu, R., 2002. Towards usage control models: beyond traditional access control, in: Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery. p. 5764. doi:10.1145/507711.507722.

[16] Robaldo, L., Bartolini, C., Palmirani, M., Rossi, A., Martoni, M., Lenzini, G., 2020. Formalizing GDPR Provisions in Reified I/O logic: The DAPRECO Knowledge Base. Journal of Logic, Language and Information 29, 401–449. doi:10.1007/s10849-019-09309-z.

[17] Rodríguez-Doncel, V., Villata, S., Gómez-Pérez, A., 2014. A dataset of RDF licenses, in: Hoekstra, R. (Ed.), Legal Knowledge and Information Systems - JURIX 2014: The Twenty-Seventh Annual Conference, Jagiellonian University, Krakow, Poland, 10-12 December 2014, IOS Press. pp. 187–188. doi:10.3233/978-1-61499-468-8-187.

[18] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E., 1996. Role-based access control models. Computer 29, 38–47. doi:10.1109/2.485845.

[19] Sandhu, R.S., Munawer, Q., 1998. How to do discretionary access control using roles, in: Youman, C.E., Jaeger, T. (Eds.), Proceedings of the Third ACM Workshop on Role-Based Access Control, RBAC 1998,October 22-23, 1998, ACM. pp. 47–54. doi:10.1145/286884.286893.

[20] Shakeri, S., Maccatrozzo, V., Veen, L., Bakhshi, R., Gommans, L., de Laat, C., Grosso, P., 2019. Modeling and Matching Digital Data Marketplace Policies, in: 15th International Conference on eScience, eScience 2019, San Diego, CA, USA, September 24-27, 2019, IEEE. pp. 570–577. doi:10.1109/eScience.2019.00078.

[21] Sharifi, S., Parvizimosaed, A., Amyot, D., Logrippo, L., Mylopoulos, J., 2020. Symboleo: Towards a Specification Language for Legal Contracts, in: Breaux, T.D., Zisman, A., Fricker, S., Glinz, M. (Eds.), 28th IEEE International Requirements Engineering Conference, RE 2020, August 31 - September 4, 2020, IEEE. pp. 364–369. doi:10.1109/RE48521.2020.00049.

[22] SIOPE DIPG Network, 2018. DIPG Registry and Imaging Repository – Regulatory Document. URL: https://dipgregistry.eu/Content/files/2018-10-10SIOPEDIPGRegistry-RegulatoryDocument_v%202.0_final.pdf. [Online, accessed 1st July 2021].

[23] Van Binsbergen, L.T., Liu, L., van Doesburg, R., van Engers, T., 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, ACM. pp. 124–136. doi:10.1145/3425898.3426958.

[24] Vos, M.D., Kirrane, S., Padget, J.A., Satoh, K., 2019. ODRL Policy Modelling and Compliance Checking, in: Fodor, P., Montali, M., Calvanese, D., Roman, D. (Eds.), Rules and Reasoning - Third International Joint Conference, RuleML+RR 2019, Bolzano, Italy, September 16-19, 2019, Proceedings, Springer. pp. 36–51. doi:10.1007/978-3-030-31095-0_3.