Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 52 (2015) 59 – 66

Procedia Computer Science

The 6th International Conference on Ambient Systems, Networks and Technologies
(ANT 2015)

Policy-Carrying Data: A Step Towards Transparent Data Sharing

Julian Padget[a], Wamberto W. Vasconcelos[b,*]

[a]*Dept. of Computer Science, University of Bath, Bath, BA2 7AY, U.K. j.a.padget@bath.ac.uk*
[b]*Dept. of Computing Science, University of Aberdeen, Aberdeen, AB24 3LT, U.K., w.w.vasconcelos@abdn.ac.uk*

Abstract

The emerging research and application domains of the Internet-of-Things and Big Data, together with socio-technical phenomena such as social networking, as well as mobile phones (equipped with sensors, GPS, etc.) make us – companies, research centres, people in general – all (sometimes unwittingly, possibly unwillingly) producers and consumers of data. A sensitive issue for data providers concerns control over access, sharing, dissemination and use of data. By control, we mean placing limitations on *who* can access the data, *when* data can be accessed, *how* data can be accessed, and so on. We propose means to capture the expression of controls over data and to associate that indivisibly with the data via what we call "policy-carrying data" (PCD). The PCD establishes permissions for what the consumer may do to the data, but also – in a novel addition for such policies – establish what the consumer can/must do subsequently with the data. We formalise PCD and illustrate how it can meet potential stakeholder requirements.

Keywords: Policies; data access; Internet-of-things

1. Introduction

The ever-reducing cost of sensors, their increasing variety, fuelled by advances in smart materials, and their incorporation in every day devices are indeed realising the Internet of Things and are capable of creating the foreseen data deluge. The availability and the volume of data motivate first-order questions such as (i) what data can be sensed physically and (ii) what information can be recovered, although second-order issues such as (i) to whom the data belong and to whom information derived from the data belongs (ii) to whom the knowledge derived from this information belongs and to what risks stakeholders are exposed as a result of this information and this knowledge remain less explored. These less explored issues may hinder progress of the first-order issues through the creation of regulation, where, although there is plenty of privacy legislation, none was designed for these circumstances; legal thinking and interpretation operate at a much slower pace than technological innovation.

[*] Corresponding author.
E-mail address: w.w.vasconcelos@abdn.ac.uk

doi:10.1016/j.procs.2015.05.020

We propose means to capture the expression of controls over (derived) data and to associate that indivisibly with the data via what we call "policy-carrying data" (PCD[1]). Following policy conventions, in defining the who, the when and the how, the PCD establishes permissions for what the consumer may do to the data. A novel aspect of our proposal is the establishment of obligations concerning what the consumer should do with the (derived) data and it is these that are the foundation of transparency. Such obligations are the transactional unit for a non-pecuniary data economy, where access to and use of data may be traded for obligations that act as a form of user-definable, liquidity at-point-of-use community currency[14]. These obligations may pertain directly to actions of data consumers or – and this is another significant novelty of our approach – indirectly to the policy associated either with the extracted data or the data derived from them.

A feature of the current data landscape is the relative freedom of movement of data from individuals to the data silos used in cloud computing and thence between silos, which could be viewed as contributary to the disempowerment that individuals might feel over their own data – privacy controls aside[5]. The situation is potentially further complicated since the platform may enable the collection and interpretation of those data, thus adding value to them, as in the case of activity-monitoring devices or home energy monitors. The PCD concept associates data with bespoke policies: for example, framework policies might be defined by legislation, while specific policies for individual needs would have to satisfy the norms established at the primary level[13]. In this way, crisp but unworkable definitions of issues such as "When do data stop being private?" and "How to decide if data revelation is in the public interest?" can be blurred as distinctions are established to meet the needs of a given situation.

Legislation will necessarily lag behind practice, but also, we contend, can only ever offer a high-level framework because of the potentially huge variety of regulations that seem likely to be required. Furthermore, by its nature legislation (and its revision) tends to take a retrospective view rather than being defined with foresight in mind. The concept of data wrapped up in a policy – policy-carrying data (PCD) – forms the basis for the realization of a data-sharing economy based on the transfer of obligations to provide an achievable combination of privacy and transparency rather than unachievable – and arguably undesirable – absolute privacy. Thus, we offer a formal model of PCD and we show how this model can be put to use via reasoning mechanisms and illustrative PCDs.

The rest of the paper is organized as follows. We put forward a formal and a computational model for the combination of policies and data, presenting and justifying a reference architecture in Section 2. In Section 3 we present our formalism for PCDs – its syntax and operational semantics. In Section 4 we outline reasoning mechanisms which stakeholders can make use of when using PCDs. We discuss our approach and contrast it with related work in Section 5 and we conclude in Section 6, setting out some avenues for future work.

2. A Framework for Policy-carrying Data

In this section we set out a reference framework within which we situate and connect stakeholders, PCDs and an information model. We illustrate our framework in Fig. 1, where we show stakeholders (circles), processes (arrows) and information model (boxes within central box). The stakeholders envisaged are (i) data owners/producers who make data/information available (represented as the left-hand circle) and, in a richer version of the model, these may be separated into those that assert rights over the data and those that publish it; (ii) data consumers who want to access data (represented as the right-hand circle) and again in a richer version of the model, there may be entities that are both consumers and producers of data, either by offering aggregation services or by adding value in some way; (iii) monitor/police who are responsible for monitoring/policing the publication and access activities (represented by the upper circle in the middle).

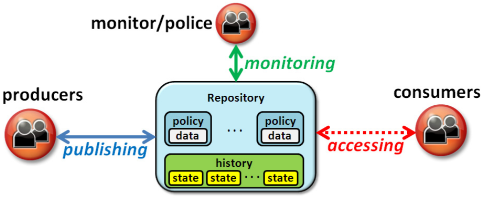

Fig. 1: A Framework for Policy-carrying Data

We note that the first two types of stakeholders can be institutions or people as well as computational entities/devices such as sensors, programs, databases, and so on. The monitor/police works as a third-party authority ensuring that activities (publishing and accessing) follow policies and dealing with violations. Each of these stakeholders has

[1] PCD also stands for "policy-carrying data collection" and we use PCDs (in the plural) to indicate a set of policy-carrying data collections.

their specific ways to interact via the repository: (i) Publishing (represented by the blue solid line) is the process whereby data owners/producers make their data available but "wrapped" within a policy, that is, they publish, in a repository, some policy-carrying data. (ii) Accessing (represented by the red dotted lines) is the process whereby data consumers *attempt* to obtain access to data mediated via policies. (iii) Monitoring (represented by the green line) concerns observing activities and checking for policy compliance or violation, and dispensing rewards or sanctions.

Our framework relies on an information model comprising the PCD – a policy and a data collection made available through the policy – and a history – a collection of events (*i.e.*, a record of activities carried out) gathered at particular time points, denoted as the *states* of the repository. We present our framework in a conceptual, technology-agnostic, and more abstract viewpoint for the sake of generality and to enable a more succinct description of essential features which technicalities would be a distraction. The framework would support stakeholders carrying out the cycle of publish-access-monitor activities supported by a Web server. Although we introduce our framework using a single server, this can be replicated across many servers, provided these can synchronize their information. Servers should be equipped with functionalities to enable the policing of those accessing and uploading PCD, keeping records of usage and (non-)compliance, and enforcing the policies' access control. We envisage programmatic access to PCD, whereby programs and functionalities developed with specific technologies can access any PCD, interacting with the server via pre-establised protocols.

A typical PCD would express something like "Lab managers can access 500 records of my data". If an interested party requested 1,000 records, the server would (i) check the credentials of the requester (who needs to be registered); (ii) grant access to 500 records (a message would provide reasons for not providing the 1,000 records); (iii) update the record of that requester with respect to that PCD. Further requests from the same party would be rejected with a suitable justification. For such control to be in place, the server requires a record of events – an explicit account of the history of the PCD, how they have been used, by whom and when. We observe that PCD can also be used as a means to "wrap" a sensor or other data source, including whole sensor networks. We can have, for instance, a policy establishing that "anyone is permitted to request the temperature reading of the sensor once every hour".

3. A Language for Policies-carrying-Data

Much research has been carried out on data access policies since the early UNIX file systems[8,20,21]. Some notable features our approach are as follows. We include means to refer to a history of events; examples are "the first 10 people can use my data" and "anyone is permitted to use n records of my data". We provide fine-grained control over who is to access the data, and under what circumstances; for instance, "invididual i_{285} is forbidden to access my data" and "anyone from company x may use my data after 6PM". Additionally we capture dynamic aspects of data usage, examples being "whoever accesses D_1 should not access D_2" and "anyone who uses my data should provide data". We adapt and extend existing proposals on normative reasoning for multi-agent systems[7,9]: we represent roles (of participants), data-related events (such as accessing records or publishing data collections), authorship of events and attempts thereof, activation and deactivation conditions of policies, and the object of the policy, namely, the data collection itself.

Syntax for Policy-carrying Data. We introduce a language for policies and a representation for policies-carrying-data, and use these to build a simple operational semantics using states and histories. In our presentation we make use of a countably infinite set $\mathcal{P}$ of propositional variables p, q, r, t with or without subscripts. For simplicity, we only use two logical operators, namely, negation $\neg$ and conjunction $\wedge$. We regard a set of propositions $\{p_1, \ldots, p_n\}$ as the *conjunction* of its elements, that is, $\{p_1, \ldots, p_n\} \overset{def}{=} \bigwedge_{i=1}^{n} p_i$. Given two sets of (possibly negated) propositions Σ, Σ', we define an entailment relation $\Sigma \models \Sigma'$ which holds if, and only, if the following two conditions hold *i)* for all $p \in \Sigma$ then $p \in \Sigma'$; and *ii)* for all $\neg p \in \Sigma$ then $p \notin \Sigma'$. Our definitions are shown in Figure 2, and explained below.

We introduce our language of policies via Def. 1; these are the "policy" part of our PCDs. We make use of a subset $\mathcal{P}^{\alpha} \subseteq \mathcal{P}$ of propositional variables to create a vocabulary of action labels α which are the target of the policies. These are formally stated in Def. 2 and we note that *accessSome*(n) and *provideSome*(n) are shorthands for the more standard representation for propositional variables $accessSome_n$ and $provideSome_n$ respectively. A sample policy is $\langle\{\neg accessSome(50)\}, \{accessSome(50)\}, \mathsf{P}_{all} accessSome(50)\rangle$ which stipulates that anyone (described by the role *all*) is permitted to access 50 records of a data collection; the norm is activated if the records haven't yet been accessed, and the norm is deactivated when 50 records are accessed. We explain below that policies are instantiated to individuals: although the policy is stated in general terms, for policing/monitoring purposes (and for sanctioning/rewarding when this is the case), we must keep a record of individuals' activities and the policies which are applicable to them (via their roles). We explain below how authorship of actions and roles are captured. PCDs are formally defined as Def. 3: we are not specific about what the data collections are – these can be individual records of a database, files, readings from a sensor, and so on. Very importantly, rather than having data collections replicated in every PCD referring to them, there could be only one copy of the data collection and all PCDs regulating its access would make use of a unique locator such as a URL.

We use a finite and non-empty set of role labels $R = \{r_1, \ldots, r_t\}$ with a *subsumption* relation $\sqsubseteq$: $R \times R$. The subsumption allows us to capture relations among roles, such as power or, more related to the work presented here, "is-a" relations. For instance, given $R = \{employee, scientist, manager\}$, we can define a subsumption relation $scientist \sqsubseteq employee$ and $manager \sqsubseteq employee$ to represent that scientists and managers are employees. In our work we assume a finite and non-empty set of individuals $A = \{a_1, \ldots, a_s\}$ uniquely identified. Moreover, individuals are associated with a non-empty set of roles and a function $roles : A \times R \mapsto 2^R, roles(a, R) = R'$ provides the set of roles $R' \subseteq R$ with which a is associated.

Operational Semantics. Definitions 4–9 explain the operational semantics connecting the syntax of our policies with an underlying computational model. We use the operational semantics to define mechanisms to check properties. Our underlying computational model is a sequence of *states*, captured in Def. 4. The states represent "snapshots" of actual events; each event is recorded as a proposition p. For compactness, we do not allow negated propositions in our states, thus adopting the *closed world assumption*[16] which establishes that what is not stated/proven as true is deemed false. For each proposition $p \in \Sigma$ we associate an agent $a \in A$, $perf(p, \Sigma) = a$, thus representing who brought about p. A sequence of states represents a *history*, formalised in Def. 5: histories record sequences of states, providing a linear account of events and how they are temporally related.

We formally connect policies with histories. We define means to check if a policy was active in a history, in Def. 6. We establish the conditions for policy *compliance* with the three cases below. We address the compliance of obligations, prohibitions and permission with Def. 7, Def. 8, and Def. 9 respectively. We denote policy compliance as the relation $comply(\pi, \mathcal{H})$. We say that a policy has been violated in a history, denoted as $violate(\pi, \mathcal{H})$ if, and only if, $active(\pi, \mathcal{H})$ holds and $comply(\pi, \mathcal{H})$ does not hold; that is, the policy was active and the restrictions on what should be done by whom have been violated.

Def. 1. *A policy π is of the form $\langle \Sigma^a, \Sigma^d, \mathsf{D}_r \alpha \rangle$ where Σ^a, Σ^d, are sets of (possibly negated) propositions representing the activation and deactivation conditions, respectively;* $\mathsf{D} \in \{\mathsf{O}, \mathsf{F}, \mathsf{P}\}$ *is a deontic modality; $r \in$ Roles is a role label; α is an action label (see Def. 2 below)*

Def. 2. *An action label α is accessAll (all records of a data collection are accessible); accessSome(n) (n records of a data collection are accessible); provideAll (all records of a data collection are provided); or provideSome(n) (n records of a data collection are provided).*

Def. 3. *A policy-carrying data collection PCD is a pair $\langle \pi, D \rangle$ where π is a policy (cf. Def. 1) and $D = \{d_1, \ldots, d_n\}$ is a set of data items.*

Def. 4. *A state $\Sigma = \{p_0, \ldots, p_n\}$ is a possibly empty and finite set of propositions $p_i, 0 \leq i \leq n$.*

Def. 5. *A history $\mathcal{H} = \langle \Sigma_0, \ldots, \Sigma_m \rangle$ is a possibly empty and finite sequence of states $\Sigma_j, 0 \leq j \leq m$.*

Def. 6. *A policy $\pi = \langle \Sigma^a, \Sigma^d, \mathsf{D}_r \alpha \rangle$ was active in history $\mathcal{H} = \langle \Sigma_1, \ldots, \Sigma_n \rangle$ if, and only if, the following conditions hold: i) $\Sigma_1 \models \Sigma^a$ (the policy became active at state* 1*), ii) $\Sigma_n \models \Sigma^d$ (the policy became inactive at state n), and iii) $\Sigma_i \not\models \Sigma^d, 1 < i < n$ (the policy was not deactivated in the intervening states). We represent policy activation as the relation active$(\pi, \mathcal{H})$.*

Def. 7. *A policy $\pi = \langle \Sigma^a, \Sigma^d, \mathsf{O}_r \alpha \rangle$ (an obligation) was complied with in history $\mathcal{H} = \langle \Sigma_1, \ldots, \Sigma_n \rangle$, if, and only if, the following conditions hold:* i) *active$(\pi, \mathcal{H})$ (the policy was active in the history)* ii) *$\Sigma_j \models \{\alpha\}$ for some $j, 1 < j < n$ (the action label occurs in some state j), and* iii) *$perf(\alpha, \Sigma_j) = a, r \in roles(a, R)$ (the agent who performed the action had role r associated with it).*

Def. 8. *A policy $\pi = \langle \Sigma^a, \Sigma^d, \mathsf{F}_r \alpha \rangle$ (a prohibition) was complied with in history $\mathcal{H} = \langle \Sigma_0, \ldots, \Sigma_n \rangle$ if, and only if, the following conditions hold:* i) *active$(\pi, \mathcal{H})$ (the policy was active in the history)* ii) *For all $j, 1 < j < n$, such that $\Sigma_j \models \{\alpha\}$ we have $perf(\alpha, \Sigma_j) = a, r \notin roles(a, R)$ (if α was performed, then the agent who performed it did not have role r associated with it).*

Def. 9. *A policy $\pi = \langle \Sigma^a, \Sigma^d, \mathsf{P}_r \alpha \rangle$ (a permission) was complied with in history $\mathcal{H} = \langle \Sigma_0, \ldots, \Sigma_n \rangle$ if, and only if, the following conditions hold:* i) *active$(\pi, \mathcal{H})$ (the policy was active in the history)* ii) *for all $j, 1 < j < n$, such that $\Sigma_j \models \{\alpha\}$ we have $perf(\alpha, \Sigma_j) = a, r \in roles(a, R)$ (if α was performed then the agent who performed it had role r associated with it).*

Fig. 2: Definitions for Policy-carrying Data: Syntax (Defs. 1–3) and Operational Semantics (Defs.4–9)

In open systems autonomous software agents are free to actually perform forbidden actions, but in a data-sharing context we want to rule out any policy-violating behaviour. We thus consider an *attempt* to access data as evidence of policy violation: consumers may try to access data they are not entitled to, and this attempt counts as if the data had been accessed, even though our PCD will prevent this from happening. Our policy violation above is interpreted under this light: the prohibited event is recorded but it did not actually happen.

The definitions of policy compliance/violation above allows us to relate policies following the usual deontic abbreviations[15], namely $\mathsf{O}p \equiv \mathsf{F}\neg p$ (p being obliged is equivalent to not achieving p being prohibited) and $\mathsf{F}p \equiv \neg\mathsf{P}p$ (p being prohibited is equivalent to not being permitted p). We sketch a proof below that such equivalences also hold in our operational model. Without loss of generality, we assume that i) our policies have the same activation and deactivation conditions and thus are active or not in exactly the same histories; and ii) the roles of our policies are the same. This means condition 1 ($active(\pi, \mathcal{H})$) of Defs. 7–9 holds, and so does $active(\pi, \mathcal{H})$ in the definition of violation; thus we only need to check if $comply(\pi, \mathcal{H})$ does not hold:

Claim: For any $\mathcal{H}$, $\pi_1 = \langle\Sigma^a, \Sigma^d, \mathsf{O}_r\alpha\rangle$ is complied with, iff, $\pi_2 = \langle\Sigma^a, \Sigma^d, \mathsf{F}_r\alpha\rangle$ is violated.

Proof: ($\Rightarrow$) If $\pi_1 = \langle\Sigma^a, \Sigma^d, \mathsf{O}_r\alpha\rangle$ was complied with then $\Sigma_j \models \{\alpha\}$ for some $j, 1 < j < n$ and $perf(\alpha, \Sigma_j) = a, r \in roles(a, R)$. This means that $\pi_2 = \langle\Sigma^a, \Sigma^d, \mathsf{F}_r\alpha\rangle$ was violated since $comply(\pi, \mathcal{H})$ does not hold as there is one $j, 1 < j < n, \Sigma_j \models \{\alpha\}, perf(\alpha, \Sigma_j) = a$ but $r \in roles(a, R)$. ($\Leftarrow$) If $\pi_2 = \langle\Sigma^a, \Sigma^d, \mathsf{F}_r\alpha\rangle$ has been violated then there is one $j, 1 < j < n, \Sigma_j \models \{\alpha\}, perf(\alpha, \Sigma_j) = a$ and $r \in roles(a, R)$. These are precisely conditions 2–3 of Def. 7 describing when $\pi_1 = \langle\Sigma^a, \Sigma^d, \mathsf{O}_r\alpha\rangle$ is complied with. ■

Claim: For any $\mathcal{H}$, $\pi_1 = \langle\Sigma^a, \Sigma^d, \mathsf{F}_r\alpha\rangle$ is complied with, iff, $\pi_2 = \langle\Sigma^a, \Sigma^d, \mathsf{P}_r\alpha\rangle$ is violated.

Proof ($\Rightarrow$) For π_1 (a prohibition) to be complied with, two possibilities may occur: *i)* there is a $j, 1 < j < n, \Sigma_j \models \{\alpha\}$ but $perf(\alpha, \Sigma_j) = a, r \notin roles(a, R)$ (α was performed by someone who did not have role r associated with it) or *ii)* for all $j, 1 < j < n, \Sigma_j \not\models \{\alpha\}$ (no-one performed the prohibited action). Case *i)* goes against condition 2 of Def 9 (α was performed by someone not permitted to do so) so implies in $\pi_2 = \langle\Sigma^a, \Sigma^d, \mathsf{P}_r\alpha\rangle$ being violated. Case *ii)* will only imply in a violation of the permission if, and only if, all actions have to be explicitly permitted. ($\Leftarrow$) If $\pi_2 = \langle\Sigma^a, \Sigma^d, \mathsf{P}_r\alpha\rangle$ has been violated then there is one $j, 1 < j < n, \Sigma_j \models \{\alpha\}, perf(\alpha, \Sigma_j) = a$ and $r \notin roles(a, R)$. This is precisely condition 2 of Def. 8 describing when $\pi_1 = \langle\Sigma^a, \Sigma^d, \mathsf{F}_r\alpha\rangle$ is complied with. ■

Having all actions explicitly permitted (if not they are forbidden) might be necessary in some data-sharing scenarios. Moreover, this will not lead to an excessive number of unnecessary policies, in that we can relate roles (hence creating policies with a role r subsuming other roles $r_i, 1 < i < n$) as well as relate actions (hence creating policies with actions α such that α *implies* $\alpha_j, 1 < j < m$), so one single policy $\langle\Sigma^a, \Sigma^d, \mathsf{P}_r\alpha\rangle$ stands for a set of policies $\bigcup_{i=1}^{n} \bigcup_{j=1}^{m} \{\langle\Sigma^a, \Sigma^d, \mathsf{P}_{r_i}\alpha_j\rangle\}$.

We note that the operational semantics provides a counterpart to the usual Kripke semantics used in (modal) deontic logics[15]. This enables us to draw parallels between deontic equivalences and relationships among our policies, as shown above.

PCD and Individual Agents. PCD are ultimately aimed at individuals, although they are specified in general terms. The role of a policy is ultimately adopted by individual agents; actions are performed by agents, this being captured by the $perf(p, \Sigma) = a$ function which provides the agent a responsible for performing $p \in \Sigma$. Compliance and violation are also associated to individuals.

The compliance definitions (Defs. 7–9) can be extended to obtain the identity of the individual agents responsible for complying with the policy. Given $\pi = \langle\Sigma^a, \Sigma^d, \mathsf{O}_r\alpha\rangle$ (an obligation) and $\mathcal{H} = \langle\Sigma_1, \ldots, \Sigma_n\rangle, comply(\pi, \mathcal{H})$ such that $\Sigma_j \models \{\alpha\}$ and $perf(\alpha, \Sigma_j) = a, r \in roles(a, R)$, then a was responsible for complying with the policy. Similar definitions can be provided for prohibitions and permissions. We denote the compliance of an individual a to policy π in history $\mathcal{H}$ as $comply(\pi, \mathcal{H}, a)$. Since more than one agent may comply with the policy, we can compute them all as $complyAll(\pi, \mathcal{H}, A'), A' \subseteq A$, such that, for all $a \in A', comply(\pi, \mathcal{H}, a)$.

In realistic settings we need to consider longer histories in which a policy is complied with or violated many times. We introduce an operator "$\bullet$" to merge/split histories; we say that $\mathcal{H} = \mathcal{H}_1 \bullet \mathcal{H}_2 \bullet \cdots \bullet \mathcal{H}_n$ holds if, and only if, $\mathcal{H}_i = \langle\Sigma^i_1, \ldots, \Sigma^i_{m_i}\rangle, 1 \leq i \leq n$, and $\mathcal{H} = \langle\Sigma^1_1, \ldots, \Sigma^1_{m_1}, \Sigma^2_1, \ldots, \Sigma^2_{m_2}, \ldots, \Sigma^n_1, \ldots, \Sigma^1_{m_n}\rangle$. With this operator, we can compute, given a history, all the sub-histories in which a policy was complied with (or violated): $comply^*(\pi, \mathcal{H}, \{\mathcal{H}_1, \ldots, \mathcal{H}_p\})$ holds if, and only if, $\mathcal{H} = \mathcal{H}' \bullet \mathcal{H}_i \bullet \mathcal{H}'', comply(\pi, \mathcal{H}_i), 1 \leq i \leq p$, that is, we decompose $\mathcal{H}$ into all those sub-histories $\mathcal{H}_i$ in which the policy was complied with. A similar computation can be defined for violations: $violate^*(\pi, \mathcal{H}, \{\mathcal{H}_1, \ldots, \mathcal{H}_p\})$ holds if, and only if, $\mathcal{H} = \mathcal{H}' \bullet \mathcal{H}_i \bullet \mathcal{H}'', violate(\pi, \mathcal{H}_i), 1 \leq i \leq p$. We

Algorithm 1: Individual Access

```
input: PCDs, $A, R$
output: $DAs = \{\langle D_1, A_1\rangle, \ldots, \langle D_n, A_n\rangle\}$
function INDIVACCESS(PCDs, $A, R$)
  $DAs \leftarrow \emptyset$
  for $PCD \in$ PCDs do
    if $PCD = \langle\langle \Sigma_a, \Sigma_e, \mathsf{P}_r\alpha\rangle, D\rangle$ then
      $DAs \leftarrow DAs \cup \{\langle D, \emptyset\rangle\}$
      for $a \in A$ do
        if $\exists r_a \in roles(a, R), r_a \sqsubseteq r$ then
          $DAs \leftarrow DAs \setminus \{\langle D, A_D\rangle\}$
          $A_D \leftarrow A_D \cup \{a\}$
          $DAs \leftarrow DAs \cup \{\langle D, A_D\rangle\}$
  return $DAs$
end function
```

Algorithm 2: Individual Obligations

```
input: PCDs, $a, R$
output: $\alpha Ds = \{\langle \alpha_1, D_1\rangle, \ldots, \langle \alpha_n, D_n\rangle\}$
function OWNOBLS(PCDs, $a, R$)
  $\alpha Ds \leftarrow \emptyset$
  for $PCD \in$ PCDs do
    if $PCD = \langle\langle \Sigma_a, \Sigma_e, \mathsf{O}_r\alpha\rangle, D\rangle$ then
      if $\exists r_a \in roles(a, R), r_a \sqsubseteq r'$ then
        $\alpha Ds \leftarrow \alpha Ds \cup \{\langle \alpha, D\rangle\}$
  return $\alpha Ds$
end function
```

Algorithm 3: Access Control

```
input: PCDs, $a, R, \alpha, D, \mathcal{H}$
output: $GrantAccess \in \{\top, \bot\}$
function CHKACCSS(PCDs, $a, R, \alpha, D, \mathcal{H}$)
  let $\mathcal{H} = \langle \Sigma_1, \ldots, \Sigma_n\rangle$
  $GrantAccess \leftarrow \bot$
  if $\langle\langle \Sigma_a, \Sigma_e, \mathsf{P}_r\alpha\rangle, D\rangle \in$ PCDs &
      $active(\langle \Sigma_a, \Sigma_e, \mathsf{P}_r\alpha\rangle, \langle \Sigma_k, \ldots, \Sigma_n\rangle)$ then
    if $\exists r_a \in roles(a, R), r_a \sqsubseteq r$ then
      $GrantAccess \leftarrow \top$
      if $\langle\langle \Sigma'_a, \Sigma'_e, \mathsf{F}_{r'}\alpha\rangle, D\rangle \in$ PCDs &
          $active(\langle \Sigma'_a, \Sigma'_e, \mathsf{F}_{r'}\alpha\rangle, \langle \Sigma_l, \ldots, \Sigma_n\rangle)$ then
        if $\exists r_a \in roles(a, R), r_a \sqsubseteq r'$ then
          $GrantAccess \leftarrow \bot$
  return $GrantAccess$
end function
```

Fig. 4: Reasoning mechanisms

also define means to compute those individuals responsible for policy compliance/violation: $comply^*(\pi, \mathcal{H}, \{\mathcal{H}_1, \ldots, \mathcal{H}_p\}, \{a_{\mathcal{H}_1}, \ldots, a_{\mathcal{H}_p}\})$ if, and only if, for all $i, 1 \leq i \leq p, comply(\pi, \mathcal{H}_i, a_{\mathcal{H}_i})$. Again, there might be more than one agent responsible for policy compliance/violation in each sub-history, and we can obtain these as $complyAll^*(\pi, \mathcal{H}, \{\mathcal{H}_1, \ldots, \mathcal{H}_p\}, \{A_{\mathcal{H}_1}, \ldots, A_{\mathcal{H}_p}\})$ where for all $i, 1 \leq i \leq p, A_{\mathcal{H}_i} \subseteq A, complyAll^*(\pi, \mathcal{H}_i, A_{\mathcal{H}_i})$. With these basic operations, we can define policing mechanisms to dispense rewards and sanctions to individuals, based on histories of states and policies; we present one such mechanism in Alg. 3 in Section 4.

We make use of our formalism to represent typical examples of PCD; these are shown in Fig. 3. PCD (1) captures a simple permission to access all records of a data collection. The activation condition specifies that the permission is in place if the records have not yet been accessed, and the policy is deactivated when the records are accessed; the policy stipulates a "one-off" access to the data. The *all* role stipulates that anyone can take advantage of this policy. PCD (2) illustrates a useful way to inter-relate policies. It states that anyone who accesses D_1 is forbidden to access D_2. The "⊥" is the "false" proposition which never occurs in any state; as a deactivation condition, it stipulates that the policy will never be deactivated once it is activated. This creates a "chain" of events relating PCDs: if someone makes use of the permission to access D_1 (established by PCD 1) then it is forbidden to access D_2. We also specify PCD (3), stating that those who access D_1 are obliged to provide 300 records to (be added to) D_2. The obligation is deactivated after the agent who accessed D_1 provides some data.

$$\langle \overbrace{\langle \underbrace{\{\neg accessAll(D_1)\}}_{activation}, \underbrace{\{accessAll(D_1)\}}_{deactivation}, \underbrace{\mathsf{P}_{all} accessAll(D_1)}_{target} \rangle}^{policy}, \overbrace{D_1}^{data} \rangle \quad (1)$$

$$\langle\langle \{accessAll(D_1)\}, \{\bot\}, \mathsf{F}_{all} accessAll(D_2)\rangle, D_2\rangle \quad (2)$$

$$\langle\langle \{accessAll(D_1)\}, \{provideSome(300, D_2)\}, \mathsf{O}_{all} provideSome(300, D_2)\rangle, D_2\rangle \quad (3)$$

Fig. 3: Sample PCDs

4. Reasoning with/about PCDs

We present in Fig. 4 three mechanisms to enable stakeholders to reason with and about their PCDs. Initially, we show in Alg. 1 a process whereby publishers of PCDs can obtain the identity of individual agents who have access (via their associated roles) to data collections. The algorithm describes function INDIVACCESS with input parameters comprising a set of PCDs, a set of agents and their roles (roles associated to agent can be obtained via the *roles* function introduced early in Section 3). The function returns a possibly empty set of pairs $\langle D, A_D\rangle$, D being a (reference to a) data collection of a PCD, and $A_D \subseteq A$ a (possibly empty) set of individual agent identities; these are the agents which have access to the various data collections.

In Alg. 2 we introduce a mechanism which analyses a set of PCDs and gathers, for a given individual a, all the obligations which might apply to it. The algorithm describes a function OWNOBLS which gathers a possibly empty set of pairs $\langle \alpha, D\rangle$ representing actions α which a is obliged to carry out with respect to data collection D. The algorithm assumes that activation conditions will hold and hence the obligation will be in place. This function is invoked by consumers of PCDs, so that they can assess the responsibilities they might bring upon themselves when registering with specific roles.

Finally, we illustrate how our PCDs and their operational semantics can be used in policing. We relate permissions and prohibitions for data sharing in a pragmatic fashion. Permissions explicitly indicate who can access the data; if the agent is not permitted, then it will not have access to the data and any attempt to access the data will be recorded as a potential violation. According to this view, one would think that prohibitions would no longer be needed since anything that is not explicitly permitted is forbidden. However, prohibitions can be interpreted as permissions being *revoked* under special circumstances. In this interpretation, prohibitions take precendence over permissions, thus making permissions void under certain circumstances. An example would be a permission to access D and a prohibition to rule out its access at certain times.

A mechanism to police data access factoring in this relation is presented in Alg. 3. It takes as input a set of PCDs, an agent id a, the set R of roles, an action α, a target data collection D and a history $\mathcal{H}$. The history is used as a "sliding window" from a state Σ_k (or Σ_l) in the past to the current state Σ_n. The mechanism initially assumes access is prevented (line 3), then it carries out an analysis of existing PCDs. In step 4 the mechanism checks if, in the set of PCDs, there is a permission on action α concerned with data D (given as a parameters) and with associated role r; it also checks if the permission is currently valid within the window $\langle \Sigma_k, \ldots, \Sigma_n \rangle$. Step 5 checks if the permission is applicable to agent a (via one of its roles r_a); if it is, then access is granted (provisionally). Step 7 checks if a prohibition on action α over D and with associated role r' exists in the set of PCDs; it also checks that the policy is active within a window $\langle \Sigma_l, \ldots, \Sigma_n \rangle$. If such a PCD exists, then in step 8 we check if the prohibition applies to a (via one of its roles r_a); if it is applicable, then access is denied, and we record a's attempt to perform α in D.

All reasoning mechanisms terminate since all loops are over finite sets, and all tests (namely $\sqsubseteq$ and *active*) also terminate. Alg. 1 is correct: it exhaustively checks all policies of PCD which grant permission to D (with any action α), and it considers if each agent in any of their roles (and their subsuming roles) is addressed by the permission; if it is, then the agent is added to those agents with access to D. Similarly, Alg. 2 is correct in that it exhaustively checks the set PCD looking for obligations which apply to any of the roles r_a (or their subsuming roles) associated with an agent. Finally, Alg. 3 is correct in that, within the period covered by history $\mathcal{H}$, it finds at least one active permission applicable to a (with provisional access granted) for which there is not one active prohibition applicable to a.

5. Related Work

Berners-Lee has called for a bill of rights or magna carta[12] to address issues of privacy, censorship and control of the internet. While that is an on-going and evolving debate, the proposal here seeks to provide a potential mechanism at the operational level that can capture specific features reflecting the normative principles that Berners-Lee promotes both in terms of the web architecture[4] and actors' behavioural constraints.

Research on security and privacy explored alternatives for authentication and authorization, including the popular role-based access control (RBAC). These assume, however, that the principal can only act on the subject in a context where the principal's actions can be observed and controlled. This clearly does not hold in an environment in which data is shared and propagated largely without oversight, although[6,11] begin to address this scenario. Nevertheless, once the data is outside the domain in which the policy can be enforced, the guarantees that a security framework such as RBAC provides almost certainly cannot be upheld and encryption probably only delays access. Thus, expectations about the treatment of data must be revised to accept transparency in place of privacy, although this too cannot necessarily be assured, some of the practicalities of which are discussed in[17]. Hansen[10] sets out higher level requirements: "unlinkability when possible and desired, transparency on possible and actual linkages, and the feasibility for data subjects to exercise control or at least intervene in the processing of data." We notice "where possible": there cannot be absolute guarantees, only best efforts, hence our notion of satisficing security.

Others have independently used the term "policy-carrying data". The research presented in[24] and[18] introduces similar concepts, however there is little or no detail about the policy languages they support or indeed their semantics, instead, their presentations concentrate on encryption aspects, architecture and information models, and how their approaches can be implemented/integrated with specific technologies.

At the mathematical and detail level, our work draws upon research on normative multi-agent systems[1], especially on proposals for norm specification[7,19,23] and normative (practical) reasoning[3,9]. Our notation is heavily inspired by existing work[7,9,23] but we simplify the components of our policies, leaving out aspects such as deadlines and sanctions/rewards. A rule-based language such as[9], being Turing-complete, would allows us to represent arbitrary concepts, but its expressiveness would render reasoning mechanisms more complex. We note that our semantics – the explicit recording of states of the computation – has been used in the literature, either as Kripke structures (providing the usual underpinning of modal deontic logics[15]) or as operational semantics[9,22]. However, our proposal is the first attempt at adapting and extending work on normative multi-agent systems to regulate data access.

6. Conclusions and Future Work

The increasing volume of personal data being created by the internet of things is making it unavoidable to find adequate answers to questions related to data access rights and responsibilities. The volume and variety of data coupled with the inappropriateness of existing legislation risk an unregulated "free-for-all" environment. What happens might be somewhat different, but until the consumers and producers of data are governed by a framework that offers equitable trade-offs between rights and responsibilities and delivers sufficient transparency, the risk remains. This has been the motivation for the principle of policy-carrying data put forward here. We have intentionally kept features to a minimum and focussed on establishing the mathematical foundations of PCD, and illustrating what can be achieved with it, in terms of policy descriptions and how to reason with and about them. We make a case for transparency, in lieu of security and privacy, since either of the latter is likely to be delivered in terms of their traditional black-and-white interpretations. This work makes a concrete contribution towards laying the foundations for a transparent data economy. For future work we shall explore extensions to the expressiveness of PCD (*e.g.*, adding deadlines, sanctions and rewards) and study reasoning mechanisms which factor these extensions in. We also want to apply model-checking[2] techniques to study properties in sets of PCDs such as "which obligations (if any) cannot be achieved?"

References

1. Andrighetto, G., Governatori, G., Noriega, P., and van der Torre, L. W. N., editors (2013). *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.
2. Baier, C. and Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press.
3. Balke, T., De Vos, M., and Padget, J. A. (2013). Evaluating the cost of enforcement by agent-based simulation: A wireless mobile grid example. In *PRIMA*, pages 21–36.
4. Berners-Lee, T. (1999). *Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor*. Orion Business. ISBN-13: 978-0752820903.
5. Brandimarte, L., Acquisti, A., and Loewenstein, G. (2013). Misplaced confidences: Privacy and the control paradox. *Social Psychological and Personality Science*, 4(3):340–347.
6. Cheng, Y., Park, J., and Sandhu, R. S. (2012). A user-to-user relationship-based access control model for online social networks. pages 8–24.
7. Şensoy, M., Norman, T. J., Vasconcelos, W. W., and Sycara, K. (2012). OWL-POLAR: A framework for semantic policy representation and reasoning. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12-13.
8. Ferraiolo, D., Atluri, V., and Gavrila, S. (2011). The policy machine: A novel architecture and framework for access control policy specification and enforcement. *Journal of Systems Architecture*, 57(4).
9. García-Camino, A., Rodríguez-Aguilar, J. A., Sierra, C., and Vasconcelos, W. W. (2009). Constraint rule-based programming of norms for electronic institutions. *Autonomous Agents and Multi-Agent Systems*, 18(1):186–217.
10. Hansen, M. (2012). Top 10 mistakes in system design from a privacy perspective and privacy protection goals. In *Privacy and Identity Management for Life*, volume 375 of *IFIP Adv. in Inf. & Comm. Techn.*, pages 14–31. Springer.
11. Karjoth, G., Schunter, M., and Waidner, M. (2003). Platform for enterprise privacy practices: privacy-enabled management of customer data. In *Procs. 2nd Int'l Conf. on Privacy-enhancing technologies (PET'02*. Springer.
12. Kiss, J. (2014). An online Magna Carta: Berners-Lee calls for bill of rights for web. Web content. `http://www.theguardian.com/technology/2014/mar/12/online-magna-carta-berners-lee-web`, retrieved 20141218.
13. Li, T., Balke, T., De Vos, M., Padget, J. A., and Satoh, K. (2013). A model-based approach to the automatic revision of secondary legislation. In *International Conference on Artificial Intelligence and Law*. ACM.
14. Litaer, B. (2002). *The Future of Money: Creating New Wealth, Work and a Wiser World*. Century.
15. McNamara, P. (2006). Deontic logic. In *Logic and the Modalities in the Twentieth Century*, volume 7. North-Holland.
16. Reiter, R. (1978). On closed world databases. In *Logic and Databases*. Plenum Press, NY, USA.
17. Sackmann, S. and Kähmer, M. (2008). ExPDT: A policy-based approach for automating compliance. *Wirtschaftsinformatik/Angewandte Informatik*, 50:366–374.
18. Saroiu, S., Wolman, A., and Agarwal, S. (2015). Policy-carrying data: A privacy abstraction for attaching terms of service to mobile data. In *HotMobile'15*. ACM Press.
19. Savarimuthu, B. T. R., Padget, J., and Purvis, M. (2013). Social norm recommendation for virtual agent societies. In *PRIMA*.
20. Suhendra, V. (2011). A survey on access control deployment. In *Security Technol.*, volume 259 of *Comm. in Comp. & Inf. Science*. Springer.
21. Tonti, G., Bradshaw, J. M., Jeffers, R., Montanari, R., Suri, N., and Uszok, A. (2003). Semantic web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. In *Procs. ISWC 2003*, volume 2870 of *LNCS*. Springer.
22. Vasconcelos, W. W., García-Camino, A., Gaertner, D., Rodríguez-Aguilar, J. A., and Noriega, P. (2012). Distributed norm management for multi-agent systems. *Expert Syst. & Appl.*, 39(5):5990–5999.
23. Vasconcelos, W. W., Kollingbaum, M. J., and Norman, T. J. (2009). Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152.
24. Wang, X., Yong, Q., Dai, Y., Ren, J., and Hang, Z. (2013). Protecting outsourced data privacy with lifelong policy carrying. In *IEEE Int'l Confs. on High Perf. Comp. & Comm. and Embedded & Ubiquitous Comp. (HPCC-EUC)*.