Definition of Data Sharing Agreements
The case of Spanish Data Protection Law

Marina Egea[1], Ilaria Matteucci[2], Paolo Mori[2], and Marinella Petrocchi[2]

[1] Atos Research & Innovation, Madrid, Spain
[2] Istituto di Informatica e Telematica, CNR, Pisa, Italy

Abstract. Electronic sharing of data among different parties, including groups of organizations and/or individuals, while protecting their legitimate rights on these data, is a key both for business and societal transactions. However, data sharing clauses are usually specified in legal documents that are far from being amenable of automated processing by the electronic platform that should enforce them. Furthermore, different parties usually pursue different interests. This may lead to conflicts that need to be solved for the agreements to succeed. Addressing this problem, in this paper we i) discuss a proposal for the definition of a machine processable electronic data sharing multilateral contract (e-DSA); ii) recall a controlled natural language (CNL4DSA) developed for expressing e-DSA clauses, in particular, authorizations and obligations policies on data; iii) instantiate a resolution process that can solve potential conflicts posed by different stakeholders' clauses, e.g., legal, organizational, and end-users' clauses, according to specific criteria. We illustrate our approach on a realistic e-Health scenario derived from one described by a Spanish medical institution. The main novelty of this paper are the reference to the Spanish Data Protection Law (S)DPL as the basic source of policies regulating data exchange and the idea of a multi-step e-DSA definition phase that incrementally increases the contract granularity. To the best of our knowledge, this is one of the first attempts to investigate how a real DPL can be translated into privacy rules electronically manageable by a devoted e-DSA-based infrastructure.

Keywords. Electronic data sharing, multilateral agreements, data protection law, privacy, conflict resolution, e-health.

1 Introduction

Sharing data among groups of organizations and/or individuals is a key necessity in modern web-based society and at the very core of business and societal transactions. However, data sharing poses several problems including trust, privacy, data misuse and/or abuse, and uncontrolled propagation of data. Hence, who produces the data would like to protect them by imposing some constraints on the operations that are allowed. Often organizations use legal documents (contracts) to regulate the terms and conditions under which they agree to share data among themselves. A similar approach can be used when data is shared

between a user and an organisation. A key problem, in the digital world, is that the constraints expressed in such (not digital) contracts remain inaccessible from the software infrastructure supporting the data sharing and management processes and, consequently, they cannot be automatically enforced. Instead, they still need to be interpreted and translated (primarily by humans) into meaningful technical policies, to ensure degrees of enforcement and auditing. What usually happens, when end-users data are going to be processed by organisations, is that end-users are asked to accept online a series of regulatory clauses on the terms of data processing, by simply clicking on a "Review and Accept the Terms and Conditions" button, and no further controls are performed on the operations that are actually executed on such data. Neither the users receive any information about how these data are processed or stored. Namely, the processing remains completely opaque for them.

In the following, we will focus on a multi-step definition process of electronic Data Sharing Agreements (e-DSA), which can be exploited to electronically represent (and manage) traditional legal contracts, by properly defining, among other fields, the parties defining and signing the agreements, the data covered and the validity time of the agreement. In particular, the different parties will be entitled to define the privacy policies regulating the sharing of the covered data. In such a way, the resulting object will contain policies defined by distinct subjects. The e-DSA will pass a validation phase to check if the various policies could be in conflict one with each other. Upon validating the e-DSA, it will be eventually enforced (meaning that the data access requests will be subject to an access control phase, according to the policies regulating the data access and defined in the validated e-DSA).

The main novelty of this paper is the reference to the Spanish Data Protection Law as the basic source of policies regulating data exchange. Starting from the conceptualisation of the currently applicable law regulating data protection in Spain, we try to model some of the original clauses in a controlled natural language format, amenable for automatic verification. Also, we depict a scenario in which both medical organisations and patients contribute in defining the e-DSA, by expressing their own constraints and preferences on the data possibly being shared. The scenario is realistic too and it derives from a real Spanish medical institution that provided it within the European project CoCo-Cloud [10]. To the best of our knowledge, this is one of the first attempts to investigate how a real Data Protection Law can be translated into a set of privacy policies electronically manageable by a devoted e-DSA-based infrastructure. Hence, the local laws and regulations of the countries in which medical data are produced and stored impose some specific constraints on them; furthermore, the organization which produced the data may want to impose its own policies on them, and, finally, the end-users (e.g., the patients, for the scope of this paper) have also the right to define their constraints on the data referring to them. Indeed, the European Directive on Data Protection 95/46/EC, and its recent reform IP/12/46 of January 25, 2012, embraced by the legislation of different European

countries, recognize the right of the individuals to consciously control the use of their personal data.

Last but not least, throughout the paper, we will show how to finalise the definition of a "conflict-free" e-DSA, by applying an appropriate conflict resolution strategy to conflicting policies possibly defined by Law, organisations, and end-users.

The remainder of the paper is as follows: Section 2 reports an informal, yet quite complete conceptualization of the Spanish Data Protection Law. Section 3 introduces the notion of e-DSA, focusing on e-DSA management over its whole lifecycle and introducing the controlled natural language we have defined in the past for expressing the e-DSA privacy policies. In Section 4, we give examples of conflicting scenarios that could realistically arise when more than one actor define their own preferences over the same dataset. Section 5 presents a technique for solving conflicts among applicable policies evaluating to different results (i.e., one allowing data access, the other one denying it). Section 6 exemplifies the presented techniques in the healthcare scenario we refer to. Finally, Section 7 discusses related work and Section 8 draws some conclusions.

2 Conceptualizing the Spanish Data Protection Law

In this section, we provide a summary of the Spanish Data Protection Law (SDPL), i.e., Organic Law 15/1999 of 13 December on the Protection of Personal Data, as an illustration of the regulations that affect data in European member countries, with an emphasis on the regulations explicitly stated on health data processing. We cover all titles of the law but the title VI that regulates the creation and organization of the data protection agency, since this information is not relevant to the purpose of this paper. For the same reason, we skip chapter 1 of title IV, since it regulates how Spanish public administration should handle personal data. Figure 1 depicts a conceptual metamodel that captures the main concepts handled by the law in order to concisely bring them to software and security engineers.

2.1 Summary of the SDPL

In the first title, the law specifies its subject and scope of application and also provides a list of general definitions to build the common ground. More concretely, the subject reads

> [. . .] this law is intended to guarantee and protect the public liberties and fundamental rights of natural persons, and in particular their personal and family privacy, with regard to the processing of personal data.

Regarding scope, the law applies to personal data recorded on a physical support which makes them capable of processing, and to any type of subsequent use of such data all over the Spanish territory. Also, it applies when data are processed

by organizations that are subject to Spanish regulations and that use for the processing means established in Spain. The law does not apply to data in transit, i.e., data that do not reside in Spain (neither its processing organization does) but goes across Spain in their trip to other countries.

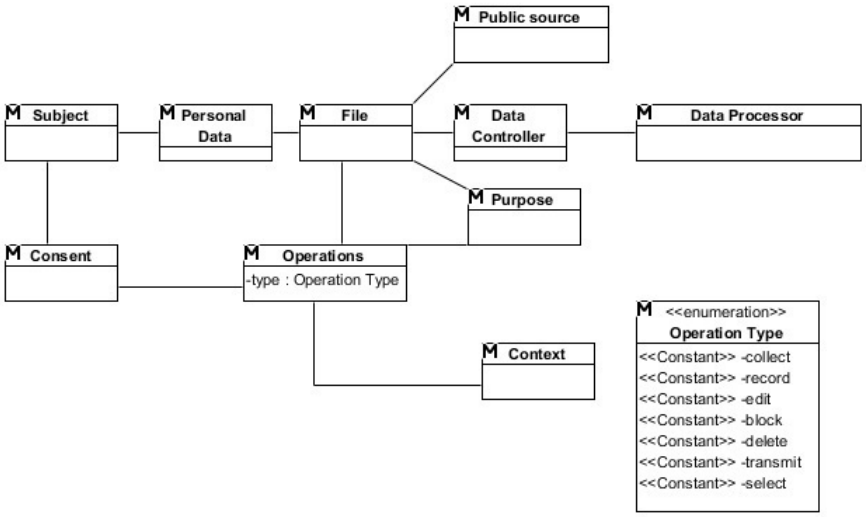

Fig. 1. SDPL conceptual metamodel

The following definitions apply for the purposes of this law, and we have tried to capture them in Figure 1.

- *Personal data*: any information concerning identified or identifiable natural persons.
- *File*: any structured set of personal data, whatever the form or method of its creation, storage, and digital access.
- *Processing of data*: operations and technical processes, which allow the collection, recording, storage, adaptation, modification, blocking and cancellation of data. Also, the assignments of data resulting from communications, consultations, interconnections, and transfers.
- *Data controller*: natural or legal person, whether public or private, or administrative body which determines the purpose, content, and use of data processing.
- *Data subject*: the natural person whom the personal data undergoing the processing refer to.
- *Dissociation procedure*: any processing of personal data carried out in such a way that the information obtained cannot be associated with an identified or identifiable person.

- *Data processor*: the natural or legal person, public authority, service or any other body which alone or jointly with others processes personal data on behalf of the controller.
- *Consent of the data subject*: any free, unequivocal, specific, and informed indication of wishes by which the data subjects consent to the processing of personal data relating to them.
- *Communication of data*: any disclosure of data to a person other than the data subject.
- *Sources accessible to the public*: those files which can be consulted by anyone, which are not subject to restrictive legislation, or which are subject only to payment of a consultation fee.
  More concretely, the law considers public sources the publicity register, telephone directories, and the lists of persons belonging to professional associations containing only data on the name, title, profession, activity, academic degree, address, and an indication of their membership to the association. Also, newspapers, official gazettes, and the media.

Moreover, in Figure 2 we show how to extend the core conceptual metamodel to refine it with concepts that clearly affect the access to files, but are not present in the core definitions of the law, thus are not represented in Figure 1. For instance, the law does not apply to domestic files, classified files, and terrorism related files. Also, a file and a data controller are linked explicitly to a right, which may turn at run time into a permission to execute operations on a file when certain authorization rules are met (e.g., explicit consent of processing to receive an economic assessment) or conditions are met (e.g., for the next 24 hours). The last extension example is the subtype system below the operation meta class that tries to capture a concept that is more relevant to IT engineers than to lawyers, i.e., there may be hundreds of operations coming from different systems that could be classified using this taxonomy.

Principles of data protection

In this section, we outline, the principles that should govern personal data management in any processing environment subject to SDPL.

Quality of data. Personal data may be collected for processing only if they are adequate, updated, relevant and not excessive for the legitimate purposes for which they were obtained. These data shall not be retained longer than necessary. Processing of the data for historical, statistical or scientific purposes is considered compatible with other legitimate purposes.

Consent of the data subject. Processing of personal data shall require the unambiguous consent of the data subject, unless laid down otherwise by law. Exceptions are public administrations (PAs) for the exercise of their functions, for business contracts or its maintenance, in the action of protecting data subject vital interest or when contained in public sources, unless rights of the data subject are jeopardized.

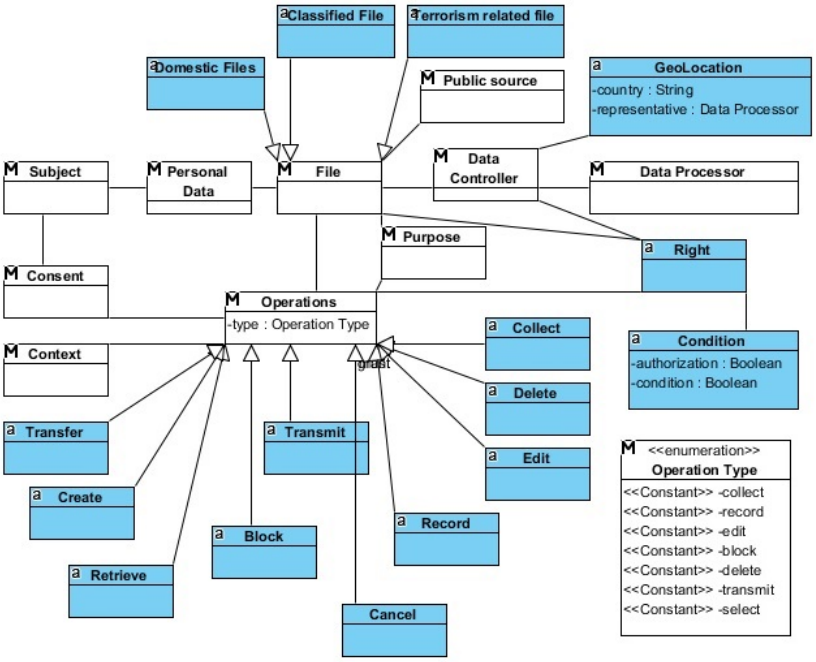

Fig. 2. SDPL extended conceptual metamodel

Right of information in the collection of data. As we mentioned above, processing of personal data requires unambiguous consent of the data subject. Furthermore, the data subject from whom personal data are requested must previously be informed explicitly about existence of a file or processing operation, its purpose, the recipients of the information, and the possible consequences of sharing information, its rights of access, rectification, erasure, and objection.

Data with special protection. Nobody may be obliged to state his ideology, religion, or beliefs. Personal data which reveal the ideology, trade union membership, religion and beliefs may be processed only with the written consent of the data subject. Exceptions shall be files maintained by political parties, trade unions, churches, religious confessions, associations, foundations, etc., as regards the data relating to their associates or members, but the communication of such data always require prior consent of the data subject. Also in the needed of medical care, the management of health care services, provided that such data is used by a health professional subject to professional secrecy.

Data on health. Public and private health-care institutions and centers, and their professionals, may process personal data relating to the health of persons consulting them, in accordance with legislation on health care.

Data security. The data controller and processor(s) shall adopt the measures necessary to ensure the security of the personal data and prevent their alteration, loss, or unauthorized access taking into account the context of handling.

Duty of secrecy. The controller and processors managing personal data shall be subject to professional secrecy as regards such data and to the duty to keep them.

Communication of data. Personal data may be communicated to third parties only for purposes directly related to the legitimate functions of the transferor and transferee with the prior consent of the data subject, except if the transfer is authorized by law, done under legal relationships, data come from public sources, or health data transfer is necessary for resolving an emergency.

Access to data on behalf of third parties. Access to data by a third party shall not be considered communication of data when such access is necessary for the provision of a service to the data controller. This processing shall be regulated in a contract and expressly laid down that the processor processes the data only following instructions of the controller, and shall not communicate them. The contract shall also set out the security measures which the processor is obliged to implement.

Rights of persons

Right of access. The data subject shall have the right to request and obtain free of charge information on his/her personal data subjected to processing, its origin and its (intended) communication. The right of access may be exercised only at intervals of twelve months, unless the data subject can prove a legitimate interest in the access.

Right of rectification or cancellation. The controller is obliged to implement the right of rectification or cancellation of the data subject within a period of ten days. Cancellation shall maintain data at the disposal of the public administrations, judges and courts, for the purpose of determining any liability. Afterwards, they shall be deleted.

If the rectified or canceled data have previously been communicated, the controller shall also notify the change so as the processor also rectifies or cancels the data.

Creation. Files in private ownership containing personal data may be created when it is necessary for the success of the legitimate activity and purpose.

Notification and entry in the register. Any person or body creating files of personal data shall first notify the Data Protection Agency (DPA). The notification includes the name of the controller, the purpose of the file, its location, the type of personal data, the security measures (either basic, medium or high level), any transfers intended (also to third countries). The DPA must be informed of any changes in the purpose of the computer file, the controller and the address of its location.

Communication of transfers of data. When making the first transfer of data, the controller must communicate this to the data subjects, indicating the purpose of the file, the nature of the data transferred and the name and address of the transferee.

Processing for the purpose of publicity and market research. Those involved in publicity, distance selling, market research, etc. shall use personal data when

they feature in public sources or when they have been provided by the data subjects themselves or with their consent. In exercising the right of access, data subjects shall have the right to know the origin of their personal data. Data subjects shall have the right to ask for stop the processing of their data.

Standard codes of conduct. By means of sectorial agreements, administrative agreements or company decisions, publicly and privately owned controllers and the organizations to which they belong may draw up standard codes of conduct. Codes of conduct must be deposited in the General Data Protection Register.

International movement of data

General rule. There may be no temporary or permanent transfers of personal data to countries which do not provide a level of protection comparable to that provided by the DPL, which is assessed by DPA.
Derogation. The provisions of the preceding paragraph shall not apply where:

- The international transfer is the result of applying agreements to which Spain is a party.
- The transfer provides international judicial aid.
- The transfer is necessary for medical care or health services management.
- Whereas the transfer of data is related to money.
- The data subject has given consent to the transfer.
- The transfer is necessary for starting or ending a contract or pre-contractual measures between the data subject and the controller.
- The transfer is necessary for ending a contract, in the interest of the data subject, between the controller and a third party.
- The transfer is necessary or legally required to safeguard a public interest, e.g. taxes.
- The transfer is necessary in legal proceedings.
- The transfer takes place at the request of a person with a legitimate interest, from a public register, and the request complies with the purpose of the register.
- The transfer takes place to a Member State of the European Union or to a country which the EU has declared to ensure an adequate level of protection.

3 Data Sharing Agreements

An electronic Data Sharing Agreement (e-DSA) is a human-readable, yet machine-processable contract, regulating how organizations and/or individuals share data. It is essentially a multilateral agreement consisting of:

- Predefined legal background information (which is usually available from a template, following, e.g., the textual template of traditional legal contracts);
- Dynamically defined information, including the definition of the validity period, the parties participating in the agreement, the data covered and, most importantly, the statements that constrain how data can be shared among the parties (such statements usually include authorization and obligation policies).

In the following, we illustrate the main phases of what we envisage to be an e-DSA definition phase, with reference to the sharing of medical data. We imagine a scenario in which the agreement draft is shown to the patient instantiating terms of law for the sharing of that kind of data, and the data sharing rules specific for that health care organization already defined by the policy experts.

In this paper, we propose a three-step phase for e-DSA definition, so that:

- (a) predefined legal background information are already filled and available in the initial e-DSA template, which encodes terms of Law for the sharing of personal data. For example, in the case of Spanish legislation, the initial e-DSA template is already filled with rules regulating the disclosure of personal data, according to SDPL (see Section 2);
- (b) policy experts at the hospital edit the initial e-DSA draft to set rules specific for that hospital, over covered data, according to internal regulations of the organization. It is worth noticing that editing the rules at this second step is not a frequent task, i.e., it is not requested that policy experts write a part of the e-DSA each time a patient wants to download a document. The policies internally defined at the organization are static ones and embrace categories of data, rather than the particular examination a patient may require in the future, and should be changed rarely over time. At this point of e-DSA definition, covered data are instantiated as belonging to categories of data (e.g., medical reports, payment receipts, administrative data, etc.). Also, we may envisage that e-DSA also instantiate the purpose for which it is being issued (like "clinical investigation", "publicity", "marketing", and so on).
- (c) As a third and final step in the editing phase, end-users complete the e-DSA either accepting or neglecting the previously-defined rules. In particular, the end-user has the opportunity to 1) give consent to the sharing of data, as expressed by Law and organizations-specific rules, and 2) edit some other adjustment to control her data disclosure. At this final point, the e-DSA is instantiated with the specific identifier of the end-user, and the specific identifier of the data over which the patient is expressing her sharing preferences. Given that it is unreasonable to think of end-users (e.g., patients) as policy experts, opportune authoring wizards should be designed to help the user in this last phase of e-DSA definition.

Given the three-step authoring phase depicted above, an e-DSA analysis phase is necessary before actually deploying the e-DSA. During the analysis phase, appropriate verification tools will detect possible conflicts between 1) in force law rules encoded in the initial e-DSA template at step (a), 2) organizations-specific rules, defined by policy experts at organization side at step (b), and 3) the customization of the end-user expressed in the last phase of e-DSA definition (c). However, distinct priority levels could be assigned to distinct rules. Hence, if the two conflicting rules have different priority levels, the conflict resolution is straightforward.

If the set of rules in the resulting e-DSA are conflict-free, then a deployment phase follows, where a set of enforceable policies are derived from the e-DSA and

deployed within the organization IT infrastructure. E-DSA enforcement mechanisms are used to ensure that requests to access and process confidential data happen consistently with the agreed e-DSA, both during the interactions with specific services and in other contexts, including attempts of employees and/or other applications to use the data and/or disclose it to third parties.

In the following, the focus is on CNL4DSA, a controlled natural language specifically designed for expressing and analyzing e-DSA rules. We remind this language, highlighting its capability to encode some data protection principles from SPDL.

3.1 CNL4DSA

In order to be able to express e-DSA rules in a processable but, at the same time, human readable way, work in [23] has introduced a controlled natural language for electronic DSA, named CNL4DSA.

The CNL4DSA language has been thought to express *Authorizations*, *Prohibitions*, and *Obligations* policies referring to data and involving parties specified in the e-DSA. It expresses the rules in a way that is pretty understandable by humans, and, at the same time, it allows to derive a formal specification of the rules, that is the input for automatic analyzers.

Rules (and set of rules, i.e., policies) are expressed in terms of *subject*, *object* (or resource), *action*, and *environment*. Notices that these concepts are inline with those shown in Figure 2: `Subject`, `File`, `Operation`, and `Context` (resp.). Similarly, the eXtensible Access Control Markup Language (XACML), the well known, de facto, standard for access control [27], relies on similar assumptions. We take advantage of this alignment to be able to enforce CNL clauses (in particular, SDPL originated clauses) using XACML. Hence, we consider a e-DSA policy as a set of rules that are evaluated, for each access request, to decide whether a given subject is allowed to perform a given action on a given resource, in a given environment. The features of the four elements, i.e., subjects, objects, actions, and environment, are expressed through *attributes* in XACML. Although, the enforcement of metamodel based policies would be probably different in other settings.

For each element, a (not exhaustive) list of attributes follows, especially referring to a health care scenario.

Subject. Attributes for subjects can be: ID, Role, and Organization.

- IDs express unique identifiers of the subject, e.g., "*abcde*123".
- Role specifies the functions and the capabilities of a subject in an organization. According to her role, a subject has different access privileges in a system. For example:
  - *general practitioner* is the role covered by that doctor who has a general view of the medical history of a patient;
  - *psychiatrists, orthopedists, radiologists, ...* identify doctors that are working in different medical specialty;

  - *rescue team member* provides first aids at the incident location and retrieves the first health information about the patient;
  - *patient* is used when the subject acts as a patient.
- Organization represents the organization the subject belongs to, e.g., "Red Cross" or "Hospital ABC".

Object. Attributes for objects could be: ID, Issuer, and Category.
- ID is a code that expresses the identifier of the object, *e.g.*, "*xyz*".
- Issuer is the ID of the subject who produces that object;
- Category could be *medical*, including documents that collect medical information about the patient, and *administrative*, including documents collecting personal information, such as the patient's name, surname, address, date and place of birth, etc.

Action. We consider their IDs only, e.g., "Process", "Cancel", "Rectify", "Access".

To specify authorizations and obligations, we introduce the notion of *fragment* denoted as $f, f_1, \ldots$, and ranged over the set $\mathcal{F}$. The *fragment* is a tuple consisting of three elements, $f = \langle s, a, o \rangle$, where s is the subject, a is the action, o is the object, expressing that "the subject s performs the action a on the object o". The terms representing the action element a could be instantiated from a predefined list, e.g., as in the `Operation Type` enumerated meta-class of the meta-model represented in Figure 1 and Figure 2.

Referring to SPDL, examples of fragment could be "health care institutions process personal data" and "data subjects access their personal data" where "health care institutions" and "data subjects" are the subject of the fragments, "process" and "access" are actions and "personal data" represents the object.

Usually, fragments are evaluated within a specific *context*. In CNL4DSA, a *basic context* is a predicate c that characterizes the elements of the policies, like, e.g., environmental condition. Simple contexts are, e.g., temporal clauses: "within a period of ten days", or location clauses: "inside the health care centre". In order to describe complex agreements, contexts need to be composable. Hence, starting from the basic context c, we use the boolean connectors *and*, *or*, and *not* for describing a *composite context* C (ranged over the set $\mathcal{C}$) which is defined inductively as follows:

$$C \ := c \mid C \ and \ C \mid C \ or \ C \mid not \ c$$

As attributes of the environment we can consider, e.g.:

- Time, with the obvious meaning;
- Location, which represents a physical position (contextualizing, it could be either of the object or of the subject);
- Status, which specifies the exceptional nature of a situation, such as an emergency situation.

More complex expressions are generated by combining fragments. We refer to such expressions as *composite fragments*, and we denote them as F (ranged over the set $\mathcal{CF}$). We distinguish two disjoint sets of composite fragments: *authorization/prohibition* fragments, denoted by F_A and ranged over the set $\mathcal{AUTH}$, and *obligation* fragments, denoted by F_O and ranged over the set $\mathcal{OBL}$.

Authorization/Prohibition Fragment. The syntax of a composite authorization fragment is inductively defined as follows:

$$F_A := nil \mid can\ (cannot)\ f \mid F_A; F_A \mid if\ C\ then\ F_A \mid after\ f\ then\ F_A \mid (F_A)$$

The intuition for the composite authorization/prohibition fragment is the following:

- *nil* can do nothing.
- *can (cannot) f* is the atomic authorization (prohibition) fragment. Its informal meaning is *the subject s can (cannot) perform the action a on the object o. can f* expresses that f is allowed, but not required. Dually, *cannot f* expresses that f is not allowed, hence it is required that f does not happen.
- $F_A; F_A$ is a list of composite authorization fragments. The list constitutes the authorization section of the considered e-DSA. Whenever one term of the list performs a f-transition, then that term evolves to the correspondent derivative.
- $if\ C\ then\ F_A$ expresses the logical implication between a composite context C and a composite authorization/prohibition fragment: if C holds, then F_A is applied.
- $after\ f\ then\ F_A$ represents the temporal sequence of fragments. Informally, after f has happened, then the composite fragment F_A is applied.

Work in [23] also associates a formal semantics to CNL4DSA syntax, based on a *modal transition system* MTS [17, 16], making the language amenable for automated processing and analysis, see, e.g., [22].

Example 1. The various clauses presented in Section 2 can be encoded in the aforementioned CNL4DSA language. As an example, we consider, from "Principles of data protection", the *Consent of the data subject* paragraph. In particular, the clauses 1) "Processing of personal data shall require the unambiguous consent of the data subject", and 2) "Exceptions are PAs for the exercise of their functions [...] in the action of protecting data subject vital interest or when contained in public sources"could be expressed in CNL4DSA as follows:

1) **if** *hasDataCategory*(Data, Personal) and **if** *belongsTo*(Data, DataSubject) and **if** not *consentGiven*(Data, DataSubject) and **if** not *hasOrganization*(DataProcessor, PA) **then** DataProcessor **cannot** *process* Data.˙
2) **if** *hasdatacategory*(Data, Personal) and **if** *belongsTo*(Data, DataSubject) and **if** *hasOrganization*(DataProcessor, PA) and **if** `hasPurpose`(DataProcessor,Protection) **then** DataProcessor **can** *process* Data.

Obligation Fragment. Similarly to authorisation fragments, the syntax of a composite obligation fragment is inductively defined as follows:

$$F_O := nil \mid must\ f \mid F_O; F_O \mid if\ C\ then\ F_O \mid after\ f\ then\ F_O \mid (F_O)$$

The intuition for the composite obligation fragment is the following:

- *nil* expresses no obligation.
- *must* f is the atomic obligation fragment. Its meaning is *the subject s must perform action a on the object o*. Thus, the f-transition is required.
- $F_O; F_O$ represents a list of composite obligation fragments. The list constitutes the obligation section of the considered DSA. Whenever one term of the list performs a f-transition, then that term evolves to the correspondent derivative.
- $if\ C\ then\ F_O$ expresses the logical implication between a context C and a composite obligation fragment. It means that if C holds, then F_O is required.
- $after\ f\ then\ F_O$ represents the temporal sequence of fragments. It means that after that f is performed, then F_O is required.

Example 2. In Section 2, we focus on the clauses about "Right of persons". If we consider the *Right of rectification or cancellation*: The controller is obliged to implement the right of rectification/cancellation of the data subject within a period of ten days. [...], this can be expressed in CNL4DSA as follows:

if *hasRole*(User, DataController) and **if** *hasDate* (RectifyRequest, Date) and **if** *timeLessThen*(CurrentDate, Date+tenDays) **then after** DataSubject *send* RectifyRequest **then** User **must** *rectify* Data

Quite obviously, if the data subject would like to cancel, the rule is similar.

4 Example Scenario

Here, we concentrate on the health care scenario, giving examples of rules that could be set through different steps of the e-DSA definition phase. As introduced in Section 3, rules definition occurs both statically and dynamically. First, generic rules are embedded in the initial e-DSA template according to legislation prescriptions on the protection of personal data. Then, policy experts at the medical organization set internal rules that are in force at their specific institution. Such rules will have a finer degree of granularity, with respect to the generic rules encoding terms of legislation. We may envisage that they will tell about actions allowed by subject covering roles over categories of data, e.g., "Those doctors operating at this medical institution can access medical examinations of patients in care at the same medical institution". This kind of rules is supposed to be statically defined, or, at least, we envisage they do not change very frequently over time. A third step, instead, takes place when the end-user (in the e-health scenario she will often collapse with the "patient") is going to interact with the medical institution to, e.g., book a clinical investigation, or negotiate the terms for a diagnosis collection. We envisage a scenario in which the patient is asked to accept the terms and regulations of law, as well as the internal rules previously set by the institution. The patient will also have the possibility to customize her own preferences regarding, e.g., identifiers of people allowed to collect the examinations on her stead.

It is worth noticing that, in the most general case, we envisage an e-DSA authoring phase in which the degree of granularity increases step after step. Indeed, when setting the initial e-DSA draft, generic rules regarding legislation of personal data processing are in place. Then, according to the kind of covered data, the e-DSA will increase its level of granularity since the policy experts will speak about, e.g., clinical diagnoses. Also, the agreement will have at this point one of the parties declared (i.e., the name of the medical organization which the policy experts belong to), and we may moreover suppose that various kind of e-DSA exist such that different data sharing purposes are defined. For example, specific e-DSA templates available at the hospital could be designed for "clinical investigation" purposes, while others for "marketing and publicity" or "analytics" purposes.

As a final step, after the patient has given consent to prior rules in the e-DSA and she has customized particular preferences over one (or more) of her specific documents, the patient name will be added as the other party of the agreement, and a validity time will be set up.

Hereafter, we give examples of rules that can be defined within the three-step e-DSA definition phase.

Scenario: Let us consider a radiology examination report. For the SPDL conceptualized in Section 2, this kind of document is classified as personal data, since it naturally contains elements that are useful to identify the data subject. Hence, the e-DSA associated to this document includes the CNL4DSA clauses imposed by the law, as for instance, the following one (that is similar to the first clause presented in previous Section 3):

if *hasDataCategory*(Data, Personal) and **if** *belongsTo*(Data, DataSubject) and **if** not *consentGiven*(Data, DataSubject) **then** DataProcessor **cannot** *retrieve* Data.

This clause states that the subject DataProcessor cannot perform operations on data whose category is Personal, if the DataSubject does not express her consent for the operation on such data. It is worth noticing that action *process* refers to generic actions that can be performed on a document, e.g., cancel, edit, transmit, record, etc. (as it was shown in the meta-model of Figures 1 and 2). Legal rules are general and are introduced in the e-DSA to safeguard the Law while managing personal data.

Internal rules at a specific health care organization can be such that doctors who produces the data within the organization can ask a second opinion from doctors of the same organization, in order to provide a better diagnosis to the patient (this rule is inspired from a real case study described in [13]). Hence, in this scenario, the policy experts at the health care organization which a doctor belongs to may add, at the second level of the e-DSA draft, a rule stating that "each doctor can read the medical documents produced within the organization

that doctor belongs to" . This rule can be written in CNL4DSA as follows:

if *hasDataCategory*(Data, Medical) and **if** *hasRole*(DataProcessor, Doctor) and **if** *hasOrganization*(DataProcessor, Hospital ABC) and **if** *hasIssuer*(Data, Hospital ABC) **then** DataProcessor **can** *retrieve* Data.

As a third step, the radiological report is released to the patient. According to the law, the patient is the DataSubject of the report (that is the Personal data), thus the patient should give the consent to the rules regulating the management of that report by the organization. We imagine that the patient agrees (or disagrees) with the application of such rules by subscribing the e-DSA or not, e.g., by ticking the consent box.

On the one hand, if the patient does not tick the consent box, when a doctor, belonging to the same health-care organization where the report has been produced, tries to access such report, both the prohibition rule set by the law (i.e., "nobody can process the data") and the rule by the organization ("doctor can process data") apply. Obviously, such conflict must be solved to obtain an enforceable access decision. Appropriate techniques for automatically solving conflicts are described in Section 5. Assuming that prohibition rules set by the Law must be always enforced, in this particular case the conflict resolution strategy is straightforward, because the prohibition is prioritized during the definition phase (one could indeed set the maximum priority level for that rule).

On the other hand, if the patient ticks the consent box, the policy set by the law is not applicable (since the consent has been given). Now, the patient can express some preferences, likely though a simple authoring interface. These extra rules refine the access rights she gives on her data. Let us suppose that the patient gives her consent to access her report (whose ID is, e.g., *RadiologicalReportXY*), but she wants to allow one person only (e.g., Doctor Paul Smith) to access it. Thus, the following rules are automatically added to the e-DSA:

if *hasDataID*(Data, RadiologyReportXY) and **if** *hasRole*(DataProcessor,Doctor) and **if** *hasID*(DataProcessor, Paul Smith) **then** DataProcessor **can** *retrieve* Data.

if *hasDataID*(Data, RadiologyReportXY) and **if** *hasRole*(DataProcessor,Doctor) and **if** not *hasID*(DataProcessor, Paul Smith) **then** DataProcessor **cannot** *retrieve* Data.

For the sake of completeness, we recall that, after the third authoring step, the e-DSA will be finalized with patient name and validity time if no conflicts arise among all the set of rules defined in the three steps.

Instead, in this particular example, when a doctor who is not Doctor Paul Smith tries to access the report with ID *RadiologicalReportXY*, a conflict occurs between the rule defined by the hospital and the ones of the patient, because the former allows doctors from that hospital to retrieve the report while the rules of

the patient do not allow its disclosure to other subjects but Doctor Paul Smith. Next section will introduce a technique for conflict detection and resolution.

5 Conflict Detection and Resolution

This section describes two approaches to perform, resp., analysis and resolution of conflicts among a set of policy rules.

5.1 e-DSA Analysis

The analysis process allows to detect conflicts between rules forming a policy. In particular, it checks if the rules set is conflict-free, by performing pairwise analysis over all pairs of rules in the e-DSA. The analyzer exhaustively simulates all the possible access requests, under a set of contextual conditions, defined by a policy expert (e.g., she can set date and time of the access request, the role of the subject, the data category, etc.). Thus, the analyzer checks if there exist at least two rules that, simultaneously, allow and deny the same subject to perform the same action on the same object, under the given set of contextual conditions.

The analysis tool we have proposed in [24] takes as input CNL4DSA rules. The formal engine performing the analysis of policies is Maude [5] that is an executable programming language that models distributed systems and the actions within those systems. The choice of using Maude for policy analysis was driven by the fact that rewrite rules (which Maude build upon) are a natural way to model the behaviour of distributed systems, and we see a policy exactly as a process where different subjects may interact with each other, possibly on the same set of objects. Maude comes with built-in commands allowing to search for allowed traces, i.e., sequence of actions, of a rule specified in CNL4DSA. These traces represent the sequences of actions that are authorized, or denied, by the rule. Also, exploiting the implementation of modal logic over the CNL4DSA semantics, as done in [33, 6] for process algebras such as CCS [25], it is possible to prove that a modal formula, representing a certain query, is satisfied by the Maude specification of the rule (or set of rules). The analyzer shows the analysis results through a user interface deployed as a Web Application. It allows the user both to query Maude specification and visualize human-readable results.

5.2 e-DSA Conflict Resolution

Here, we describe a methodology introduced in [21] applicable to solve rules' conflicts. It is based on the Analytical Hierarchy Process (AHP), a well known multi-criteria decision system [30, 31]. AHP is a multi-criteria decision making technique, which has been largely used in several fields of study. Given a decision problem, within which different *alternatives* can be chosen to reach a *goal*, AHP returns the *most relevant* alternative with respect to a set of *criteria*. The adoption of AHP to solve conflicts among rules has been described in [21, 18].

Within the e-DSA scenario, the technique is applied when the conflicting rules have been defined with the same priority level.

The AHP approach requires to subdivide a complex problem (i.e., ranking conflicting rules) into a set of sub-problems, equal in number to the chosen *criteria*, and then computes the solution (*i.e.*, choose the applicable rule) by properly merging all the local solutions for each sub-problem. In Figure 3, we show a possible instantiation of the AHP hierarchy for conflict resolution. The goal (the box on top of the hierarchy in Figure 3) is "select the rule" among conflicting ones, *e.g.*, rule1 and rule2 in the boxes at the bottom of the hierarchy. As usual, rules are expressed in terms of *subject*, *object*, *action*, and *environment* and they are evaluated according to the value of these attributes in order to determine which rule can be applied to each access request.

We consider as *criteria* (second group of boxes starting from the top of the hierarchy) the *specificity* of the elements that constitute a rule: i) *Specificity of the subject*, in which we evaluate the attributes exploited in the two conflicting rules to identify the subject, to determine which of them define a more specific set of subjects; ii) *Specificity of the object* in which we evaluate the attributes exploited in the two rules to identify the object; iii) *Specificity of the environment* in which we evaluate the attributes to identify the environment. Furthermore, AHP features the capability to further refine each criterion in sub-criteria, by considering the attributes that identify each element, e.g., for the subject: ID, Role, and Organization. The attributes' set depends on a given scenario.

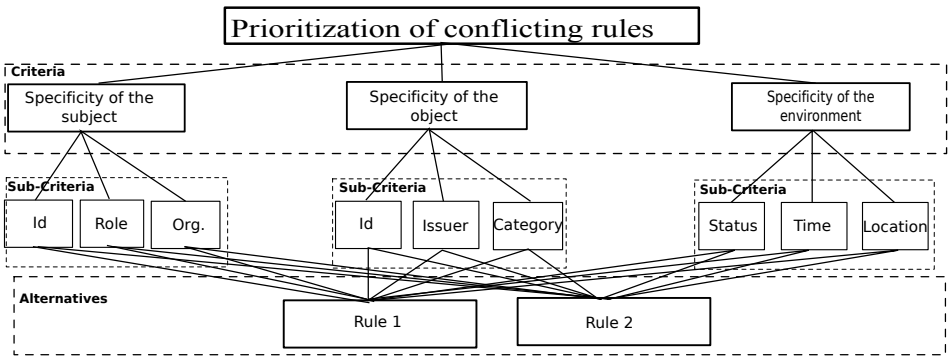

Fig. 3. AHP Hierarchy for Conflict Resolution.

Figure 3 represents the hierarchy here considered. However, the methodology is general enough to allow the insertion of further criteria and sub criteria that may be helpful to evaluate the alternatives.

Once the hierarchy is built, the method performs pairwise comparison, from the bottom to the top, in order to compute the relevance, hereafter called *local priority*: i) of each alternatives with respect to each sub-criteria, ii) of each sub-criterion with respect to the relative criterion, and finally, iii) of each criterion

Table 1. Fundamental Scale for AHP

Intensity	Definition	Explanation
1	Equal	Two elements contribute equally to the objective
3	Moderate	One element is slightly more relevant than another
5	Strong	One element is strongly more relevant over another
7	Very strong	One element is very strongly more relevant over another
9	Extreme	One element is extremely more relevant over another

with respect to the goal. Note that, in case of a criterion without sub-criteria, the local priority of each alternative is computed with respect to the criterion.

Comparisons are made through a scale of numbers typical to AHP (see Table 1) that indicates how many times an alternative is *more relevant* than another.

Computation of local priorities. Let the reader suppose that rule1 and rule2 are two conflicting rules. They become the two alternatives in the hierarchy and they are evaluated with respect to sub criteria. To this aim, k 2x2 pairwise comparison matrices, where k is the number of sub criteria (in our case, k=9), are built according to a very simple approach, based on the presence of the attributes in the rules. Given that a_{ij} is the generic element of one of these matrices:

- rule1 and rule2 contain (or do not contain) attribute A: then $a_{12} = a_{21} = 1$.
- If only rule1 contains A, than $a_{12} = 9$, and $a_{21} = \frac{1}{9}$.
- If only rule2 contains A, than $a_{12} = \frac{1}{9}$, and $a_{12} = 9$.

Once a comparison matrice has been defined, the local priorities can be computed as the normalized eigenvector associated with the largest eigenvalue of such matrice [29].

Then, moving up in the hierarchy, we quantify how subcriteria are relevant with respect to the correspondent criterion. Hence, we evaluate how the attributes are relevant to identify the subject, the object and the environment. In particular, in our example we use the matrices in Table 2, with the local priorities shown in the last column of each matrice. As an example, in the matrice that compares the subject's attributes (the left-most one in Table 2), we write $a_{12} = 9$ since we think that the subject ID allows to identify the subject *extremely* better than the subject role. Indeed, the subject ID exactly identifies one subject. For the same reason, we put $a_{13} = 9$ (ID *vs* the organization the subject belongs to). More details are in [21].

We remark that the values in these matrices simply represent the perception of the authors on the relative relevance of the attributes. Other values could have been chosen as well.

Finally, we quantify how the three criteria are relevant for achieving the goal of solving conflicts. Without loss of generality, we hypothesize that all the criteria equally contribute to meet the goal. In this straightforward case, the pairwise comparison matrice is a 3x3 matrice with all the elements equal to 1,

Table 2. Comparison matrices and local priorities for subcriteria *w.r.t.* criteria

SUBJ	ID	role	organiz.	$\bar{p}_{Subj}$
ID	1	9	9	0.818182
role	$\frac{1}{9}$	1	1	0.0909091
org	$\frac{1}{9}$	1	1	0.0909091

OBJ	ID	issuer	category	$\bar{p}_{Obj}$
ID	1	5	7	0.7454
issuer	$\frac{1}{5}$	1	$\frac{4}{3}$	0.1454
category	$\frac{1}{7}$	$\frac{3}{4}$	1	0.1091

ENV	status	time	location	$\bar{p}_{Env}$
status	1	7	7	0.777778
time	$\frac{1}{7}$	1	1	0.111111
location	$\frac{1}{7}$	1	1	0.111111

and the local priorities of the criteria with respect to the goal are simply 0.33 each. Hence, for the computation of the global priorities, $p_g^{c_j} = 0.33$, $j = 1,\ldots,3$ (see below).

Computation of global priorities. Once all local priorities are computed, the following formula computes the global priorities. For the sake of simplicity, we have in mind a hierarchy tree where the leftmost $n1$ criteria have a set of sub-criteria each, while the rightmost $n2$ criteria have no sub-criteria below them, and $n1 + n2 = n$ is the number of total criteria.

$$P_g^{a_i} = \sum_{w=1}^{n1} \sum_{k=1}^{q(w)} p_g^{c_w} \cdot p_{c_w}^{sc_k^w} \cdot p_{sc_k^w}^{a_i} + \sum_{j=1}^{n2} p_g^{c_j} \cdot p_{c_j}^{a_i} \tag{1}$$

$q(w)$ is the number of sub-criteria for criterion c_w, $p_g^{c_w}$ is the local priority of criterion c_w with respect to the goal g, $p_{c_w}^{sc_k^w}$ is the local priority of sub-criterion k with respect to criterion c_w, and $p_{sc_k^w}^{a_i}$ is the local priority of alternative a_i with respect to sub-criterion k of criterion c_w. $p_{c_w}^{sc_k^w}$ and $p_{sc_k^w}^{a_i}$ are computed by following the same procedure of the pairwise comparisons matrices illustrated above.

It is worth noticing that, in our approach, we do not consider as a decisional criterion the specificity of the action. This is because we evaluate the action only according to its ID, always present in a policy. So the evaluation of the alternative rules with respect to the criterion *action* is constant, and it does not add any meaningful information for taking the final decision.

In [18], we have developed a prototype implementation of the conflict solver based on the rules' specificity, highlighting a twofold advantage. First, the prototype is specifically based on the XACML engine and it extends the native XACML combining algorithms for conflict resolution, aiming at a finer granularity in the evaluation of conflicting rules. Secondly, we experienced good results in terms of execution time, negligible to human beings up to a quite large amount of conflicting rules (for example, execution time is 275 milliseconds with 64 conflicting rules composed by three attributes each).

6 Resolution Strategy Example

We refer to the example scenario in Section 4, where a doctor at Hospital ABC would like to share with another doctor a particular radiological report.

Let R_1 be the hospital rule:

if *hasDataCategory*(Data, Medical) and **if** *hasRole*(DataProcessor, Doctor) and **if** *hasOrganization*(DataProcessor, Hospital ABC) and **if** *hasIssuer*(Data, Hospital ABC) **then** DataProcessor **can** *retrieve* Data.

Instead, the patient would like to share that data with Doctor Paul Smith only (R_2):

if *hasDataID*(Data, RadiologyReportXY) and **if** *hasRole*(DataProcessor,Doctor) and **if** not *hasID*(DataProcessor, Paul Smith) **then** DataProcessor **cannot** *retrieve* Data.

If the doctor at ABC tries to show the radiology examination to a doctor different from Paul Smith a conflict occurs because both rule1 and rule2 are applicable, but with opposite effects.

According to what discussed in the previous section, each criterion is statically evaluated with respect to the goal. We recall that the local priorities of the uppermost two levels of the AHP hierarchy in Figure 3 are stitched to the rules themselves. They are defined according to the scenario when the policies are created, and they do not change until the rules change. In our case, we hypothesize that, for the uppermost level, the local priorities are all equal to 0.33, while the local priorities for the middle level have been specified in Table 2.

Instead, the local priorities of the lowest level are evaluated at runtime, when someone tries to access the data. The evaluation is simply based on the presence, or the absence, of an attribute in the conflicting rules. In our example, we have that:

- R_1 identifies the subject through role and organization, while R_2 through role and ID.
- R_1 identifies the object through category and issuer, while R_2 through the object ID.
- Neither R_1 nor R_2 specifies constraints on the environment.

Table 3 shows an example of the simple 2x2 matrices that compare R_1 and R_2 with respect to the presence of the attribute ID of the element object. Since R_2 specifies the object through the ID, while R_1 does not, we give 9 to R_2 and $\frac{1}{9}$ to R_1.

Similar 2x2 matrices are built for evaluating R_1 and R_2 with respect to all the sub-criteria (we have 9 matrices). The global priorities are calculated according

Table 3. R_1 and R_2 evaluated w.r.t. the presence of the attribute ID of the object

ID_{Obj}	R_1	R_2	$\bar{p}_{ID_{Obj}}$
R_1	1	$\frac{1}{9}$	0.1
R_2	9	1	0.9

to expression 1 and instantiated as in 2:

$$\begin{aligned} P_g^{R_1} &= 0.33 \cdot ((p_{Subj}^{ID} \cdot p_{ID}^{R_1}) + (p_{Subj}^{Role} \cdot p_{Role}^{R_1}) + (p_{Subj}^{Org} \cdot p_{Org}^{R_1}))+ \\ &\quad 0.33 \cdot ((p_{Obj}^{ID_{Obj}} \cdot p_{ID_{Obj}}^{R_1}) + (p_{Obj}^{Iss} \cdot p_{Iss}^{R_1}) + (p_{Obj}^{Cat} \cdot p_{Cat}^{R_1}))+ \\ &\quad 0.33 \cdot ((p_{Env}^{Stat} \cdot p_{Stat}^{R_1}) + (p_{Env}^{Time} \cdot p_{Time}^{R_1}) + (p_{Env}^{Loc} \cdot p_{Loc}^{R_1})) \\ &= 0.34 \end{aligned} \tag{2}$$

where $p_{Subj}^{(\_)}$, $p_{Obj}^{(\_)}$, and $p_{Env}^{(\_)}$ are the value of vectors of local priorities shown in Figure 2 (rightmost column of each matrix), while $p_{ID}^{R_1}$, $p_{Role}^{R_1}$, ... are the local priorities of rule R_1 against all the subcriteria, as the result of the nine 2x2 matrices. For example, $p_{ID_{Obj}}^{R_1} = 0.1$, see Table 3. Complementary, for rule R_2 we obtain $P_g^{R_2} = 0.66$. Hence, the result of the decision strategy shows a preference for the execution of rule R_2.

7 Related Work

To the best of our knowledge, the main novelty of this paper is the translation of a real Data Protection Law, i.e., the Spanish Data Protection Law (SDPL), into privacy electronically manageable rules by a devoted e-DSA-based infrastructure. Indeed, this is the first attempt to refer to the SDPL as the basic source of the design of multi-lateral e-DSA that regulates data exchange among different entities.

On the other hand, over the last decades, researchers have investigated several solutions for (platform-independent) policy-based infrastructures, to specify, analyze, and deploy privacy, security, and networking policies. Hereafter we revise some work focused on validation and policy conflict detection and resolution.

Data protection policy analysis is essential to detect inconsistencies and conflicts before the actual enforcement. In [3], it is shown that the Event-B language (www.event-b.org) can be used to model obliged events. The Rodin platform provides animation and model checking tool set for analyzing specifications written in Event-B, thus leading to capability of obligations analysis [2]. The authors of [26] propose a comprehensive framework for expressing highly complex privacy-related policies, featuring purposes and obligations. Also, a formal definition of conflicting permission assignments is given, together with efficient conflict-checking algorithms and with a set of experimental results which show the performances of such algorithms. The Policy Design Tool [28] offers a sophisticated way for modeling and analyzing high-level security requirements in a business context and create security policy templates in a standard format.

Hence, there exists generic formal approaches that could a priori be exploited for the analysis of some aspects of data protection policies. As an example, the Klaim family of process calculi [8] provides a high-level model for distributed systems, and, in particular, exploits a capability-based type system for programming and controlling access and usage of resources. Also, work in [12] exploits a static analyzer for a variant of Klaim.

Policy conflict detection is generally followed by resolutions of conflicts. Not necessarily tied to data protection, existing work concerns general conflict resolution methods for access control in various areas. The approach adopted by the eXtensible Access Control Markup Language (XACML) [27] is a very general one, defines standard rule-combining algorithms: Deny-Overrides, Permit-Overrides, First-Applicable, and Only-One-Applicable. As an example, the Deny-Overrides algorithm states that the result of the policy is Deny if the evaluation of at least one of the rules returns Deny. A classification of anomalies that may occur among firewall policies is presented in [1]. In the same work, an editing tool allows a user to insert, modify, and remove, policy rules in order to avoid anomalies.

In [14], the authors propose a conflict resolution strategy for medical policies, by presenting a classification of conflicts and suggesting a strategy based on high level features of the policy as a whole (such as the regency of a policy). If such characteristics are not sufficient for deciding which policy should be applied, the *default deny* approach is applied.

Work in [11] identifies a number of conflict types, using examples from the military and aerospace domain, and discusses how to prevent and resolve such conflicts for different classes of them.

In [19, 32] the authors deal with both the detection and resolution of conflicts. Work in [19] defines a policy precedence relationship that takes into account the following principles: *a)* Rules that deny the access have the priority on the others; *b)* Priorities could be explicitly assigned to policy rules by the system administrators; *c)* Higher priority is given to the rule whose distance with the object it refers to is the lowest, where a specific function should be defined to measure such distance; and *d)* Higher priority is given to the rule that is more specific according to the domain nesting criteria. In [32], the authors investigate policy conflict resolution in pervasive environments. They discussed different strategies for conflict detection but the part dedicated to the conflict resolution strategy just refers to quite standard strategies, *i.e.*, role hierarchies override and obligation precedence. Also in [9], four different strategies for solving conflicts are considered. They distinguish among solving conflicts at compile-time, at run-time, in a balanced way leaving to run-time only potential conflicts, or in ad-hoc way accordingly to the particular conflicts. In general they take into account the role of the requester for deciding which policy wins the conflict. Also in this case, the strategy is based only on one criterion. The approach in [19, 32] is extended in [20]. Indeed, the authors introduce the definition and employment of the precedence establishment principals in a context-aware-manner, *i.e.*, according

to the relation among the specificity of the context. The decision criterion is a unique one that groups a set of contextual conditions.

Work in [4] presents a formal model, based on deontic logic, to detect and, possibly, solve conflicts among security policies. An implementation of the model is left as future work. In [7], the authors present *Or BAC*, a methodology to manage conflicts occurring among permissions and prohibitions. Within this approach, rules are grouped according to the organizations that emit them. The advantage of this proposal is to reduce the problem of redundant policies.

The procedure known as *Break the Glass* [15] may be applied in extraordinary situations, bypassing all the existing applicable rules. As an example, by applying this methodology, rescue team members can access patient medical documents in an emergency situation, whatever the policies related to those documents are. A proper audit support should be used to monitor the accesses.

8 Conclusions

Protecting personal data from unauthorized disclosure to third parties is an issue regulated by the legislation of different European countries, with the support of common European directives. Technically, data access, processing, and sharing can be regulated defining (and enacting) appropriate machine processable data sharing multilateral contracts, named e-DSA. Based on the protection of data, an e-DSA is written by all the entities that have a jurisdiction on that data. It collects rules that cover several aspects, from legal constraints to user-preferences.

Hence, in this paper we provided an overview of the incremental construction phases of an e-DSA. These phases follow the agreement procedure that takes into account clauses coming from a default template stating the legislation of application; the clauses introduced by an organization, e.g., a health-care company, and finally, the decisions of the end-users with respect to the use of their data (e.g., including consent, purpose, restrictions, ...). As an e-DSA construction process evolves, the clauses specified in the e-DSA reach a finer granularity. The main novelty of this paper is the reference and conceptual modelling of the Spanish Data Protection Law (S)DPL as the basic source of policies regulating data exchange. To the best of our knowledge this is also the first attempt of specifying rules of SPDL as clauses for electronic processing in a controlled natural language (CNL4DSA).

The e-DSA construction procedure is not restricted to the authoring procedure but it also includes a conflict detection and, eventually, resolution phase. We also introduce an e-DSA validator and illustrate its usage and the conflict detection and resolution process through a realistic e-Health scenario, based on a real one described by a Spanish medical institution.

In the immediate future, we will expand our studies about the conceptualization and formal analysis of the SPDL. In particular, we will validate them using scenarios of data exchange in different settings and domains where they may be classified at different sensitive levels. In this way, we will also get further

feedback about the practical application of the methodology presented in this paper.

9 Acknowledgements

The research leading to these results has been partially funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant no. 610853 (Coco Cloud).

References

1. E. S. Al-Shaer and H. H. Hamed. Firewall policy advisor for anomaly discovery and rule editing. In *IFIP/IEEE Integrated Network Management*, pages 17–30, 2003.
2. A. Arenas et al. An Event-B Approach to Data Sharing Agreements. In *Integrated Formal Methods*, pages 28–42. Springer, 2010.
3. J. Bicarregui, A. Arenas, B. Aziz, P. Massonet, and C. Ponsard. Towards Modelling Obligations in Event-B. In *Abstract State Machines, B and Z*, volume 5238 of *Lecture Notes in Computer Science*, pages 181–194, 2008.
4. L. Cholvy and F. Cuppens. Analyzing consistency of security policies. In *IEEE Symposium on Security and Privacy*, pages 103–112, 1997.
5. M. Clavel et al., editors. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *LNCS*. Springer, 2007.
6. M. Colombo, F. Martinelli, I. Matteucci, and M. Petrocchi. Context-aware analysis of data sharing agreements. In *Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services*, pages 99 – 104, 2010.
7. F. Cuppens, N. Cuppens-Boulahia, and M. B. Ghorbel. High level conflict management strategies in advanced access control models. *ENTCS*, 186:3–26, 2007.
8. R. De Nicola, G. L. Ferrari, and R. Pugliese. Programming Access Control: The KLAIM Experience. In *CONCUR*, pages 48–65, 2000.
9. N. Dunlop, J. Indulska, and K. Raymond. Methods for conflict resolution in policy-based management systems. In *Enterprise Distributed Object Computing*, pages 98–109. IEEE, 2003.
10. EU FP7 grant no. 610853. Confidential and Compliant Clouds (Coco Cloud) project, 2013. `http://www.coco-cloud.eu`.
11. M. Hall-May and T. Kelly. Towards conflict detection and resolution of safety policies. In *Intl. System Safety Conf.*, pages 687–695, 2006.
12. R. R. Hansen, F. Nielson, H. R. Nielson, and C. W. Probst. Static Validation of Licence Conformance Policies. In *ARES*, pages 1104–1111, 2008.
13. Hewlett-Packard Italiana (Editor). Coco-Cloud Deliverable 7.1: Definition of pilot requirements, 2014. `http://www.coco-cloud.eu/deliverables`.

14. J. Jin, G.-J. Ahn, H. Hu, M. J. Covington, and X. Zhang. Patient-centric authorization framework for electronic healthcare services. *Computers & Security*, 30(2-3):116–127, 2011.
15. Joint NEMA/COCIR/JIRA Security and Privacy Committee (SPC). Break-glass: An approach to granting emergency access to healthcare systems, 2004.
16. K. G. Larsen. Modal specifications. In *Automatic Verification Methods for Finite State Systems*, pages 232–246, 1989.
17. K. G. Larsen and B. Thomsen. A modal process logic. In *LICS*, pages 203–210, 1988.
18. A. Lunardelli, I. Matteucci, P. Mori, and M. Petrocchi. A Prototype for Solving Conflicts in XACML-based e-Health Policies. In *Computer-Based Medical Systems*, pages 449–452. IEEE, 2013.
19. E. C. Lupu and M. Sloman. Conflicts in policy-based distributed systems management. *IEEE Trans. Softw. Eng.*, 25(6):852–869, 1999.
20. A. Masoumzadeh, M. Amini, and R. Jalili. Conflict detection and resolution in context-aware authorization. In *Security in Networks and Distributed Systems*, pages 505–511. IEEE, 2007.
21. I. Matteucci, P. Mori, and M. Petrocchi. Prioritized Execution of Privacy Policies. In *DPM/SETOP*, pages 133–145, 2012.
22. I. Matteucci, P. Mori, M. Petrocchi, and L. Wiegand. Controlled data sharing in E-health. In *STAST*, pages 17–23, 2011.
23. I. Matteucci, M. Petrocchi, and M. L. Sbodio. CNL4DSA: a Controlled Natural Language for Data Sharing Agreements. In *SAC: Privacy on the Web Track*, pages 616–620. ACM, 2010.
24. I. Matteucci, M. Petrocchi, M. L. Sbodio, and L. Wiegand. A design phase for data sharing agreements. In *DPM/SETOP*, pages 25–41, 2011.
25. R. Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.
26. Q. Ni, , E. Bertino, J. Lobo, C. Brodie, C.-M. Karat, J. Karat, and A. Trombetta. Privacy-aware Role-based Access Control. *ACM Transactions on Information and System Security*, 13(3):24:1–24:31, 2010.
27. OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0, January 2013.
28. Policy Design Tool. http://www.alphaworks.ibm.com/tech/policydesigntool, 2009.
29. T. L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234–281, 1977.
30. T. L. Saaty. Decision-making with the AHP: Why is the principal eigenvector necessary. *European Journal of Operational Research*, 145(1):85–91, 2003.
31. T. L. Saaty. Decision making with the Analytic Hierarchy Process. *International Journal of Services Sciences*, 1(1):83–98, 2008.
32. E. Syukur. Methods for policy conflict detection and resolution in pervasive computing environments. In *Policy Management for Web (WWW05)*, pages 10–14. ACM, 2005.
33. A. Verdejo and N. Martí-Oliet. Implementing CCS in Maude 2. *ENTCS 71*, 2002.