DSAP: Data Sharing Agreement Privacy Ontology

Mingyuan Li and Reza Samavi

Computing & Software, McMaster University, Hamilton, Ontario, Canada
{lim107,samavir}@mcmaster.ca

Abstract. In this paper we propose a flexible and extensible ontology for expressing and enforcing privacy policies described in medical research data sharing agreements (DSA). We demonstrate that our ontology is capable of supporting the DSA in a collaborative health research data sharing scenario through providing the appropriate vocabulary and structure to log privacy events in a linked data based audit log. Furthermore, through querying the audit log, we can answer privacy queries relevant to the data sharing agreements.

Keywords: Privacy · Data Sharing · Semantic Web · Ontology

1 Introduction

A data sharing agreement (DSA) is a written document composed of static text expressing the constraints that researchers must follow when sharing data containing personal information [20]. The composition of the expressed constraints, or privacy policies, is guided by the goal to foster data sharing but also to protect research subjects' personal information. The most common form to express DSA policies is using natural language. However, inherent ambiguity in natural languages is a source of problem. While a number of policy languages are proposed to express DSA policies in unambiguous manner (e.g., [2,3,15]), their lack of flexibility, extensibility, and ease of use made their applications limited in collaborative medical research environments [15]. In addition, there are aspects of privacy constraints (e.g., purpose definition [2], data transfer [15] and retention [3] policies) that the proposed languages are not able to express. The objective of this work is to design a semantic model with necessary concepts and properties that allows DSA privacy policies to be expressed unambiguously. The model also supports transparency and auditibility of actions of researchers on private data and consequently making the responsible participants accountable.

We motivate the need for such a semantic model using a typical collaborative medical research scenario as depicted in Fig. 1. Bob is a researcher at a public university (PU-A) and conducts only publicly-funded research. His colleague, Charlie, is also a researcher at PU-A but he also holds an appointment at a private pharmaceutical company (PP-A). The collaborative research involves a genetics dataset bound by a DSA privacy constraint that limits the genetics data to be only shared with publicly funded research studies. Charlie appears

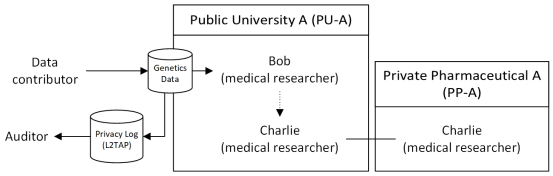

Fig. 1: Motivating Scenario

to be in a legitimate position to receive data in his capacity as a researcher at PU-A, but his private appointment at PP-A raises an issue. The confusion of responsibilities is necessary to be resolved as mishandling of patient data can harm the patients and can be a liability for the researchers [17]. Bob and Charlie expect to have an unambiguous and efficient way of answering their questions regarding their data protection responsibilities.

In this paper, we propose the DSAP (Data Sharing Agreement Privacy) ontology, as a semantic model consisting of explicitly defined concepts and relations [12], to provide medical researchers with a language to express privacy policies of DSAs. To design the ontology we adapted the methodology described by [13] and utilized concepts and relations derived from surveyed privacy policies of a diverse number of DSAs and guidelines. We designed the ontology with a hierarchical structure, lightweight expressiveness, and with capability of reusing other ontologies. These characteristics make our ontology flexible and extensible to be used by different medical disciplines. We demonstrate that using our ontology we can create a Linked Data [14] based privacy log [19,18] to answer privacy queries. The full DSAP ontology is available on *http://l2tap.org/dsap*.

The paper structure is as follows. In Section 2, we discuss the ontology requirements and provide an overview of the DSAP ontology. In Section 3, we demonstrate how DSAP can be used for logging events related to data sharing. In Section 4, we report the evaluation of our ontology. The related work is discussed in Section 5, and we conclude in Section 6.

2 DSAP Ontology Design

The key objective in designing the DSAP ontology is to support transparency and accountability with respect to the treatment of personal data when multiple stakeholders (participants) are collaborating in a health research context. The ontology should provide necessary structure to support encoding: 1) the static semantic: i.e. the privacy terms and conditions and what participants must perform in order to comply with these terms, and 2) the dynamic semantic: i.e. what participants have performed and the actual occurrences of data usage access activities with respect to the DSA privacy constraints. Although the focus of our DSAP ontology is only on the static semantics of DSAs, we are interested to design the ontology such that it supports the dynamic semantics through extension. Therefore extensibility of the DSAP ontology to capture dynamic semantics by using other ontologies is a core design requirement.

2.1 Ontology Users and Requirements

The main participants in a medical research are data contributors, medical researchers (and their associated hospitals or research centers) and privacy audi-

tors [9]. Data contributors are usually patients and while they contribute data to the studies, their main concern is the preservation of their privacy throughout the medical research lifecycle (i.e., collection, use and process, and disclosure). Privacy auditors are concerned with the compliance of medical researchers with privacy policies governing the use of personal data. Patients can also be an auditor when they are enabled to audit the process of data sharing. Meanwhile, in addition to sharing data among themselves, medical researchers need to communicate privacy policies regarding the treatment of data with their collaborators. The researchers are also concerned with their own compliance with privacy policies specified by their collaborators. To achieve compliance, the researchers first need to understand their own privileges and obligations w.r.t. the health data.

Flexibility in expressing privacy policies. Different medical research disciplines may need different types of privacy policies to be expressed. For instance, with respect to data retention policies, public health researchers may only work with aggregated and anonymized data. Therefore, public health research DSAs may simply require the researcher to dispose the data after a year. On the other hand, genetics researchers work with more sensitive personal data. As a result, genetics research DSAs may specify more strict policies to include an encryption requirement in addition to disposal requirement. Therefore, the ontology will need to be flexible to be able to express different types of privacy policies.

Extensibility to work with other ontologies. With an extensible ontology, medical researchers will have the freedom to utilize other ontologies with more or less expressive power. By extending, more vocabulary can be utilized to express diverse privacy policies as well as allowing the enforcement of privacy policies to be achieved. Ontology reuse has many benefits and is encouraged in the ontology design literature [16]. Saving the labour involved in creating an ontology from scratch, improving the quality of ontology, and reducing ontology maintenance overhead [16] are among these advantages. We design our ontology with reuse of other ontologies in mind where appropriate.

Lightweight expressivity. An important requirement of the DSAP ontology is decidability. Our ontology is organized into a hierarchical structure with the key advantage of supporting inheritance for complex concepts. The transitive closure of inheritance supports decidable reasoning. In addition, when reading a hierarchy, the medical researcher can intuitively organize the concepts and relations based on parent-child relationships. The simple and self-explanatory organization of concepts in a hierarchy simplifies the complexities of DSAs. The hierarchy structure also promotes extensibility as the concepts within the hierarchy can be extended without the need to disrupt the rest of the ontology. Another distinct advantage of a lightweight ontology is its extensibility and flexibility. For example in our ontology, if we use a hierarchy to express different roles, using the `rdfs:subClassOf` allows a more complex role hierarchy from another ontology to be directly plugged into our ontology.

Semantic Technology support Our ontology makes use of a limited number of concept and properties in RDFS [4], and RDF [1], e.g., `rdfs:domain`, `rdfs:range`, `rdfs:subClassOf` and `rdfs:subPropertyOf`. Using these ontology

languages, with limited expressive power, has multiple advantages over highly expressive ontology languages such as first order logic (FOL). First, there are scalable off the shelf semantic technology support (e.g., IBM DB2) for RDF data model. Second, compared to FOL, our ontology is decidable as we rely on the limited reasoning power of the transitive closure of `rdfs:subClassOf` and `rdfs:subPropertyOf`. In addition, RDF type ontology is relatively easier to understand by a non-technical person such as a medical researcher and is typically used for description or classification purposes [11].

2.2 DSAP Core Structure

We adapted the ontology design methodology described by [13] with the following four steps: 1) *Describe Motivating Scenario* as described in the Introduction section. 2) *Determine Competency Questions.* Competency questions are questions that our ontology is required to answer. 3) *Derive Concepts and Relations.* Concepts and relations are derived from vocabulary gathered from a survey of DSA templates and guidelines. 4) *Evaluate the Ontology.* We evaluated our ontology to support the static semantics of DSA (by directly using DSAP concepts) and dynamic semantics of DSAs (through the extension described in Section 4).

Competency Questions Competency questions define the queries to be answered by the DSAP ontology. These questions can be prohibitive or prescriptive. Prohibitive questions are questions asked by a researcher or an auditor to ensure a certain task is permitted to perform on private data. Examples are: Is Bob allowed to share the dataset D_1 with Charlie? Should I perform task T_1 (e.g., sending a notification) at the point in time t_1?. This type of questions usually demands a simple yes or no answer. Prescriptive questions in contrast are questions that demand a certain piece of instruction from the DSA. Examples of prescriptive questions are: What are my obligations if I access the dataset D_1? For what purposes can I disclose the dataset D_1?

DSAP Ontology Concept Derivation For deriving the main concepts of the ontology in addition to using the usage scenario recommended in the methodology, we used DSA templates and guidelines. We surveyed 9 data sharing agreements and 10 data sharing guidelines. The obtained documents included from both Europe and North America representing a range of medical research topics such as cancer and genetics research. First, a word pool was created by parsing the data sharing guidelines and data sharing agreements through a word counter [10]. A total of 48,241 words were parsed. Next, after synonyms and duplicates were removed, 72 unique words were identified and were used to create a preliminary vocabulary list from which our ontology was constructed. Nouns were used as candidate concepts while verbs were used as candidate relations. Following word analysis, we conducted a whole-document analysis of five data sharing agreements from Canada and Europe. The policy clauses covering intellectual property rights and authorships were excluded and privacy-related clauses were analyzed for required concepts and relations to be expressed.

Overview of DSAP Ontology We use UML class diagram to represent the core concepts of DSAP Ontology: `Agent`, `Role`, `Action`, `Dataset`, and `Policy` as depicted in Figure 2 and described below:

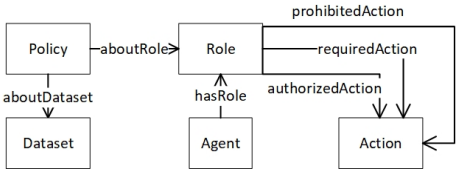

Fig. 2: DSAP Ontology Core Concepts

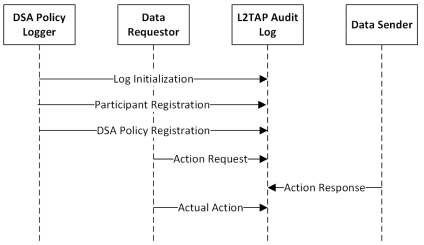

Fig. 3: Events in L2TAP Log [18]

`Agent`: An agent is a person or a thing that is responsible for the occurrence of an action. In our scenario, Bob and Charlie are examples of agents.

`Role`: The role identifies the expected actions of an agent. Bob and Charlie both assume the role of a researcher. As researchers, they are expected to do something with data, such as access data, share data, and analyze data. Bob also takes on the role of data sender while Charlie takes the role of data requestor.

`Action`: An action is something that is performed by an agent on an item (e.g., a dataset). Destroying a dataset is an example of an action. In our scenario, Charlie is required to destroy the dataset at the end of agreement.

`Dataset`: The dataset captures any collection of private data, such as data containing personal health information or any personal identifiable information. The genetics data in our scenario is an example of dataset.

`Policy`: Policy refers to each individual clause in the DSA expressing a privacy constraint. "*Bob is only allowed to share the genetics dataset with publicly funded research*" is an example of a policy.

Several relations are important in our core ontology. The `hasRole` relation captures the `Role` that an `Agent` assumes. In our scenario, Charlie `hasRole` researcher. With `aboutRole`, we can also associate a `Policy` with `Role`. The `prohibitedAction`, `requiredAction`, and `authorizedAction` relations capture the prohibition, requirement, and authorization of a `Role` to perform a certain `Action` respectively. For instance, the `requiredAction` can capture that Charlie as a researcher is required to destroy data (an `Action`) at the end of agreement. Finally, the `aboutDataset` relation mints the `Dataset` to a `Policy`. For example, our genetics data can be associated with DSA privacy policies through the `aboutDataset` relation. There are additional concepts and relations such as `dsap:Purpose`, `dsap:eligiblePurpose`, `dsap:Consent`, etc. The full ontology and potential mapping of some DSAP concepts (e.g., Role, Agent, Consent) to the concepts in other ontologies is available at *http://l2tap.org/dsap*.

In Table 1, we demonstrate our ontology's ability to express privacy policies in DSAs by translating a DSA excerpt into RDF format serialized in the Turtle syntax [7]. The policy has three components: 1) An obligation clause stating that the receiver of the data must destroy data captured using the `dsap:requiredAction` property (lines 1-2); 2) The clause indicating that the specific dataset to be destroyed captured using the `dsap:aboutDataset` property (line 3-4); and 3) A temporal clause stating the date at which dataset must be destroyed encoded using the `tl:atDate` property from the Timeline Ontology

(line 5). In the example shown in Table 1, ambiguity resulting from the vague expression: "tangible materials" is eliminated when the DSA author is forced to declare specific datasets using the `dsap:aboutDataset` relation in our ontology.

Table 1: Expressing Data Retention Privacy Policy in RDF Triples

DSA Excerpt	**RDF Triples**

Upon termination of this Agreement, receiving party shall promptly destroy all documents, files and other tangible materials representing disclosing party's information.	`:receiver a dsap:DataReceiver;`
	`  dsap:requiredAction :deleteData.`
	`:deleteData a dsap:DestroyAction;`
	`  dsap:aboutDataset :healthData;`
	`  tl:atDate "2018-08-01"^^xsd:date.`

3 DSAP Ontology and Dynamic Semantics of DSAs

In this section, we describe how the DSAP ontology can be extended to capture dynamic semantics of DSAs using other Linked Data based ontologies such as L2TAP model [18,19]. For our motivating scenario (Fig. 1) we show that privacy events can be recorded in a linked data based audit log [19]. The audit log can be then published and queried using SPARQL queries to answer privacy questions.

We use a UML sequence diagram (Fig. 3) adapted from [19] to describe the participants and the order of privacy events that occur in a collaborative data sharing environment. The main participants of the privacy audit logging process are DSA Logger, Data Requestor (Charlie), the L2TAP Audit Log, the Data Sender (Bob), and the Auditor. Similar to the scenario described in [19], for each event generated by the participants in our data sharing scenario (Fig. 1), we can generate an L2TAP log event. The order of events is also similar to the L2TAP log in general as the logger needs to initialize log and register all participants (including data senders, requestors, auditors and even patients as the auditors). Then privacy policies and terms and conditions of a DSA (static semantics of the DSA) will be registered by the logger using the DSAP ontology. After these three events are logged the log can be used to record all different actions that occur during the lifecycle of a DSA. To show how DSAP can be used with the extended semantics of L2TAP, below we describe the event of a privacy policy registration for our motivating scenario. A complete list of all events are available at our ontology website (*http://l2tap.org/dsap*).

Each privacy event in the L2TAP log contains a log header and a log body. The log header encodes assertions about the provenance of an event (who, when and what). For the log header we plugin the L2TAP-core module (with the **l2tap** namespace) as described in [19]. For example, in the privacy policy registration event (Listing 1.1), the header is shown in lines 1-12. In our header, `exlog:logevent3` is an instance of `l2tap:PrivacyEvent` (line 1) and a member of `exlog:log1` (line 2). `exlog:log1` is the URI for the group of log events that together captures all events in our data sharing scenario. Line 3 captures that the logger for this event is `exlog:dsapLogger`. lines 4-5 capture the timestamps for the log using two subproperties of `l2tap:timeStamp` and utilizing the Timeline Ontology and `xsd:dateTime` in lines 7-12. Until now we captured the *who* and *when* assertions of the log event. The *what* of a log event (in this example,

the privacy policy) is the payload of the event and captured in line 6 using the `l2tap:eventData` property. Note that the URI that this property is pointing to, is a named graph [6]. The L2TAP core ontology only captures that the body of the event is a graph without speaking about the detail semantics of the triples inside the graph. This L2TAP feature allows the DSAP ontology to be extended to use the L2TAP ontology to capture dynamic semantics of a DSA.

We use concepts and relations in the DSAP ontology to describe the log body which is enclosed within a named graph (lines 15-28). Each DSA privacy policy is encoded by the `dsap:Policy` concept. For simplicity, our data sharing scenario will have two DSA privacy policies. The first registered policy (lines 15-22) specifies that Bob as a researcher from PU-A is permitted to share genetics data using the `dsap:aboutRole` and `dsap:aboutDataset` property. The permission to share data is captured using the `dsap:authorizedAction` property to a `dsap:ShareAction` concept (line 18). Lines 21-22 assert that the sharing of genetics data can only happen to publicly funded research captured using the `dsap:permittedFunding` property. The second DSA policy (lines 24-26) identifies that the policy is about Charlie and genetics data which is captured using the `dsap:aboutRole` and `dsap:aboutDataset` properties, respectively. The policy further specifies that Charlie is required to destroy the data in line 27 using the `dsap:requiredAction` property. Line 28 uses `dsap:hasDate` to indicate that the data must be destroyed at the agreement end date.

```
1  exlog:logevent3 a l2tap:ParticipantRegistrationEvent;
2    l2tap:memberOf exlog:log1;
3    l2tap:eventParticipant exlog:dsapLogger;
4    l2tap:receivingTimestamp exlog:logevent3-time1;
5    l2tap:publicationTimestamp exlog:logevent3-time2;
6    l2tap:eventData exlog:log-namedgraph3.
7  exlog:logevent3-time1 a tl:Instant;
8    tl:atDateTime "2018-07-01T12:00:02Z"^^xsd:dateTime;
9    tl:OnTimeLine exlog:tlphysical.
10 exlog:logevent3-time2 a tl:Instant;
11   tl:atDateTime "2018-07-01T12:00:03Z"^^xsd:dateTime;
12   tl:OnTimeline exlog:tlphysical.
13 exlog:log-namedgraph3 a rdfg:Graph.
14 exlog:log-namedgraph3 {
15   exlog:policy1 a dsap:Policy;
16     dsap:aboutRole exAgreement:univAResearcher1;
17     dsap:aboutDataset exAgreement:geneticsData.
18   exAgreement:univAResearcher1 dsap:authorizedAction exAgreement:shareData.
19   exAgreement:shareData a dsap:ShareAction;
20     dsap:aboutDataset exAgreement:geneticsData;
21     dsap:permittedFunding dsap:publicFunding.
22   exStudy:publicFunding a dsap:publicFunding.
23   exlog:policy2 a dsap:Policy;
24     dsap:aboutRole exAgreement:univAResearcher2;
25     dsap:aboutDataset exAgreement:geneticsData.
26   exAgreement:univAResearcher2 dsap:requiredAction exAgreement:destroyData.
27   exAgreement:destroyData a dsap:DestroyAction;
28     dsap:hasDate exAgreement:endDate.}
```

Listing 1.1: DSA Privacy Policy Registration Event

After the initialization of the log and the registration of participants and privacy policies into an L2TAP log, the researchers can start to share the data. Privacy events that occur during the sharing of data are captured in three steps: 1) Action Request (request for sharing by a data requestor, 2) Action Response

(response to the sharing request by a data sender), and 3) Actual Action (receipt of data by the data requestor). For a complete listings of all events, please see our ontology website (*http://l2tap.org/dsap*).

4 DSAP Ontology Evaluation

We evaluate three aspects of the DSAP ontology: 1) static aspect: its ability to express privacy policies in DSAs, 2) dynamic aspect: its ability to log privacy events during the data sharing process, and 3) application aspect: its ability to answer privacy competency questions.

The static aspect of our ontology was evaluated by translating DSA privacy policies into RDF triples. The example shown in Section 3 demonstrates the ability of our ontology to unambiguously capture the static aspect of DSAs. The dynamic aspect of our ontology was evaluated by generating a linked data based privacy audit log for our motivating scenario as described in Section 3. Finally, to evaluate the application aspect of our ontology, a testing environment was set up on Virtuoso 06.01.3127 server[1] with quad store installed on 64-bit Ubuntu 14.04 LTS virtual machine with 1GB of memory and one Intel-i5 CPU core. A total of 10,000 RDF triples were generated in the log to mock the amount of activity in a real data sharing scenario. SPARQL queries were run against the populated log to answer privacy competency questions.

```
1 ASK
2 WHERE{ exlog:accessRequest1 scip:dataSender ?bobRole.
3   ?bobRole dsap:authorizedAction ?bobAuthorizedAction.
4   ?bobAuthorizedAction a dsap:ShareAction.
5   ?bobAuthorizedAction dsap:aboutDataset exAgreement:geneticsData.
6   exlog:accessRequest1 scip:dataRequestor ?requestorRole.
7   ?requestorRole dsap:partOfStudy ?study.
8   ?study dsap:hasFunding ?funding.
9   ?funding a dsap:PublicFunding.}
```

Listing 1.2: SPARQL Query to Answer a Prohibitive Competency Question

Listing 1.2 shows the SPARQL `ASK` statement (line 1) used to answer the prohibitive competency question: *Is Bob allowed to share the genetics dataset with Charlie?*. The statement tests for the existence of RDF triple patterns registered within the L2TAP log returning `TRUE` or `FALSE`. Lines 2-3 check for actions that Bob is authorized to perform per DSA. Line 4 checks if any of the authorized action is a `ShareAction`. Should Bob be authorized to share data, line 5 checks if Bob is authorized to share the genetics dataset that Charlie requested. Since the DSA specifies that the genetics dataset can only be shared with publicly funded research, a check is added to ensure that Charlie is working on a publicly funded research project (lines 6-9). The execution of Listing 1.2 returns `TRUE`, meaning *Bob is permitted to share the genetics dataset with Charlie*. Due to limited space in this paper, we present other SPARQL queries on *http://l2tap.org/dsap*.

5 Related Work

Work to translate natural language privacy policies into unambiguous machine language exist [5]. Brodie *et al.* [5] proposed SPARCLE Policy Workbench as a

[1] `https://virtuoso.openlinksw.com`

means to parse natural languages into a machine-readable XML expression of policy elements. While the accuracy of this system in parsing healthcare privacy constraints in DSAa is high ($\geq$ 91%) [5], the accuracy is not perfect, so manual intervention and verification is required. Event-B specification language has been adapted by Arenas *et al.* [2] to express DSA constraints. They represent temporal, protection, and sharing constraints in obligation clauses with linear temporal logic (LTL), a linear sequence of events defined by time. Implementation support for this model is available through the Rodin platform (www.event-b.org/platform.html), which utilizes the ProB animator and model checker to analyze the DSA for conflicts and to validate a principal's actions with obligations outlined in the DSA. Although Event-B can be used for unambiguously interpretation of DSA conditions, some privacy specific clauses such as purpose definition, could not be expressed with this model. In contrast to the LTL modelling of DSAs, Swarup *et al.* [20] modelled obligation clauses in DSAs using distributed temporal logic predicates over data resources, data stores and data flows, which allows reasoning about properties of several components of the system for past and future events. Swarup *et al.*'s DSA framework is also able to express penalties if an obligation is not complied with. Electronic implementation of this framework is underway [20].

One drawback of policy languages is their domain dependency [2,3,15]. The lack of flexibility makes policy languages unable to adapt for data sharing scenarios across various medical research domains. Moreover, the machine-oriented syntax of policy languages force policy languages to be highly structured and difficult to extend.

6 Conclusion

In this paper, we proposed an ontology to address the gap to express privacy constraints in DSAs unambiguously while also allowing flexibility. Our ontology was designed using RDF and RDFS with hierarchical structure to take advantage of transitive closure under inheritance. With the reuse of other ontologies, we showed both the static and dynamic semantics of DSA privacy policies can be captured using the DSAP ontology.

Our ontology design suffers from a number of limitations motiviating future research. The concepts derived in our ontology was limited to only the surveyed DSAs. The confined focus may limit the vocabulary used for our DSAP Ontology's concepts and relations. The `OWL:sameAs` property [8] can be used to capture equivalent classes in different research domains that may use different terminologies for DSAs. Moreover, to cover legal terms governing health data sharing within different jurisdictions, the `dsap:Jurisdiction` concept may be extended and the `dsap:requiredJurisdictionCompliance` property can be used to capture specific jurisdictional requirements. To make our ontology usable to the medical researcher, an interface shell needs to be developed. The shell can be designed similar to other medical ontology viewers such as the Systematized Nomenclature of Medicine Clinical Terms Ontology Browser (*http://browser.ihtsdotools.org/?*).

Acknowledgments

Financial support from NSERC and help from Andrew Sutton are acknowledged.

References

1. Resource Description Framework (RDF). Online (02 2014), https://w3.org/RDF/
2. Arenas, A.E., Aziz, B., Bicarregui, J., Wilson, M.D.: An event-B approach to data sharing agreements. In: International Conference on Integrated Formal Methods. pp. 28–42. Springer (2010)
3. Aziz, B., Arenas, A., Wilson, M.: SecPAL4DSA: a policy language for specifying data sharing agreements. In: FTRA International Conference on Secure and Trust Computing, Data Management, and Application. pp. 29–36. Springer (2011)
4. Brickley, D., Guha, R.: RDF Schema 1.1. Online (02 2014), https://www.w3.org/TR/rdf-schema/
5. Brodie, C.A., Karat, C.M., Karat, J.: An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In: Proceedings of the second symposium on Usable privacy and security. pp. 8–19. ACM (2006)
6. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: Proceedings of the 14th international conference on World Wide Web. pp. 613–622. ACM (2005)
7. David Beckett, T.B.L.: Turtle - Terse RDF Triple Language (03 2011), https://www.w3.org/TeamSubmission/turtle/
8. Dean, M., Schreiber, G.: OWL Lite. Online (Feb 2004), https://www.w3.org/TR/owl-ref/#OWLLite
9. Denny, S.G., Silaigwana, B., Wassenaar, D., Bull, S., Parker, M.: Developing ethical practices for public health research data sharing in South Africa: The views and experiences from a diverse sample of research stakeholders. Journal of Empirical Research on Human Research Ethics **10**(3), 290–301 (2015)
10. FreeOnlineWordCounter: Online (Apr 2018), http://countwordsfree.com/
11. Giunchiglia, F., Zaihrayeu, I.: Lightweight ontologies. In: Encyclopedia of Database Systems, pp. 1613–1619. Springer (2009)
12. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2), 199–220 (1993)
13. Grüninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. Workshop on Basic Ontological Issues in Knowledge Sharing (1995)
14. Heath, T., Bizer, C.: Linked data: Evolving the web into a global data space. Synthesis Lectures on the Semantic Web: Theory and Tech. **1**(1), 1–136 (2011)
15. Kagal, L., Finin, T., Joshi, A.: A policy language for a pervasive computing environment. In: Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on. pp. 63–74. IEEE (2003)
16. Lonsdale, D., Embley, D.W., Ding, Y., Xu, L., Hepp, M.: Reusing ontologies and language components for ontology generation. Data & Knowledge Engineering **69**(4), 318–330 (2010)
17. O'herrin, J.K., Fost, N., Kudsk, K.A.: Health Insurance Portability Accountability Act (HIPAA) regulations: effect on medical record research. Annals of surgery **239**(6), 772 (2004)
18. Samavi, R., Consens, M.P.: Publishing L2TAP Logs to Facilitate Transparency and Accountability. In: LDOW (2014)
19. Samavi, R., Consens, M.P.: Publishing privacy logs to facilitate transparency and accountability. Journal of Web Semantics **50**, 1–20 (2018)
20. Swarup, V., Seligman, L., Rosenthal, A.: A data sharing agreement framework. In: International Conference on Information Systems Security. pp. 22–36 (2006)