algorithms

MDPI

Article

The Need for Machine-Processable Agreements in Health Data Management †

George Konstantinidis [1,*], Adriane Chapman [1], Mark J. Weal [1], Ahmed Alzubaidi [1], Lisa M. Ballard [2], and Anneke M. Lucassen [2]

[1] School of Electronics and Computer Science, University of Southampton, Southampton, UK; adriane.chapman@soton.ac.uk (A.C.); mjw@ecs.soton.ac.uk (M.J.W.); aa4u16@soton.ac.uk (A.A.)

[2] Clinical Ethics and Law at Southampton, Centre for Cancer Immunology, University of Southampton, Southampton, UK; l.ballard@soton.ac.uk (L.M.B.); a.m.lucassen@soton.ac.uk (A.M.L.)

* Correspondence:g.konstantinidis@soton.ac.uk

† This paper is an extended version of G.K.'s invited talk abstract [2] published in the Proceedings of the Second International Workshop on Semantic Web Meets Health Data Management (SWH 2019), Auckland, New Zealand, October 26, 2019.

Received: 29 February 2020; Accepted: 4 April 2020; Published: 7 April 2020

Abstract: Data processing agreements in health data management are laid out by organisations in monolithic "Terms and Conditions" documents written in natural legal language. These top-down policies usually protect the interest of the service providers, rather than the data owners. They are coarse-grained and do not allow for more than a few opt-in or opt-out options for individuals to express their consent on personal data processing, and these options often do not transfer to software as they were intended to. In this paper, we study the problem of health data sharing and we advocate the need for individuals to describe their personal contract of data usage in a formal, machine-processable language. We develop an application for sharing patient genomic information and test results, and use interactions with patients and clinicians in order to identify the particular peculiarities a privacy/policy/consent language should offer in this complicated domain. We present how Semantic Web technologies can have a central role in this approach by providing the formal tools and features required in such a language. We present our ongoing approach to construct an ontology-based framework and a policy language that allows patients and clinicians to express fine-grained consent, preferences or suggestions on sharing medical information. Our language offers unique features such as multi-party ownership of data or data sharing dependencies. We evaluate the landscape of policy languages from different areas, and show how they are lacking major requirements needed in health data management. In addition to enabling patients, our approach helps organisations increase technological capabilities, abide by legal requirements, and save resources.

Keywords: data sharing; consent; privacy policies; privacy languages; genomic data; genomic medicine; health data management

1. Introduction

Data sharing in health data management is often a tedious and manual process. In different organisations there are usually processes in place so that physicians can share data regarding a particular patient (e.g., a diagnosis, or medical tests' and results) with other medical professionals that care for the same patient. Medical data sharing, either for patient care or for clinical research, requires bilateral, custom-crafted, lengthy data access agreements, written in natural language, and set between clinicians, researchers, and the health institutions. These agreements leave very little space for customisation. More importantly, individual patients have very little voice in how their data is

used and shared beyond a high-level consent form that states data may be shared to other consultants. At best, patients are offered a few opt-in/out choices on coarse-grained features of the sharing.

This landscape becomes much more complicated due to the nature of medical data, especially in medical genetics where the boundaries of ownership and control of data are blurry and where additional people not initially involved in the clinical process might have an interest or even a right to obtain information and data. For example, the English case of ABC vs. St George's NHS Trust and others was discussed in the High Court in November 2019. The patient (XX) received a diagnosis of Huntington's disease (HD) – a hereditary condition with a 50% chance of passing it on to offspring – with no cure and likely early death. The patient's pregnant daughter (ABC) did not know of XX's condition and that it was heritable, and his clinicians discussed whether she had a right to know. They discussed this with XX, who refused consent for ABC to be told as he was concerned she would terminate her pregnancy. ABC, who was later tested and found to have inherited the gene mutation for HD, claims that XX's clinicians were neglectful for not communicating his diagnosis to her. Situations like this case will occur more often as genetic testing and genomic medicine is adopted more widely in general medical practice, highlighting the consideration of if and when it may be appropriate to make relatives aware of risks discovered through the testing of a single person, which can be done without breaching confidentiality and without the patient's consent [1].

To address such complex scenarios and requirements, in this paper we advocate the need, and present an initial approach, for machine-processable data sharing agreements in the health data management domain. This paper includes and extends [2]. We evaluate several related approaches and languages for expressing privacy preferences, sharing policies and consent on data-sharing. We find that most of these approaches specify coarse-grained policies, using predefined options, present mostly "accept/reject" agreements that do not offer flexibility for partial access, and, most importantly, do not capture the pragmatic necessities that we identify in health, and particular genetic data sharing.

A simple scenario which is not expressible in other approaches, and that we would like to capture, is the fact that clinicians might *suggest* or *advise* patients to share their data with relatives but the final choice usually lies with patient. People who have genetic tests for hereditary diseases are advised to inform close relatives who may also have inherited the same condition. However, up to 42% of relatives remain uninformed [3–8]. The result of an individual's genetic test can suggest that close relatives may have the same result, and with this information they can make informed decisions about their own testing, screening programs, or preventative treatment.

To understand the processes that take place, we have developed an application, myKinMatters, that allows us to explore how users can specify permissions and consent. myKinMatters, shown in Figure 1 offers patients a secure way of sharing their genetic test results electronically with their relatives and has a variety of functions i.e. supports the patient with creating a family tree; allows the patient to take their family tree to their clinician who can mark on the app which relatives could benefit from being given the information; allows the patient to upload their genetic test result document along with any other documents they wish to share; send documents electronically to relative's email or mobile phone; and track which relatives have opened the documents. We co-designed myKinMatters with clinicians and health professionals at a University Hospital Trust, and have deployed the application with test users in simulated scenarios as part of intervention trials.

Through our analysis we have identified five major requirements that are particular to health data sharing: (i) the need to define complex, previously undefined, classes of individuals to share data with, e.g., "share data with all females on the patient's mother's side", (ii) data sharing dependencies, for example, a patient wants to share with their siblings only if their mother learns first, or in contrast, only if their siblings are not going to share it further, (iii) data sharing instigation, e.g., the clinician might suggest or advise that the patient should share the data with someone in their family and we would like to record this action, as well as whether the patient actually shares the data or not, (iv) multi-party privileges on data, e.g., control of genetic data might not belong to one individual in particular but to

an entire family, or a clinician might have an obligation to override a patient's preference, and (v) areas where permission to share may be different where the data is anonymized.

Our first requirement points us to use an ontology-based approach, ontologies being shared vocabularies, machine-processable conceptualisations of a domain of interest (https://www.w3.org/standards/semanticweb/ontology). For example, our approach allows us to employ a pedigree genealogical ontology such as [9], which in combination with familial relationship inference rules, e.g., "all female siblings of a mother are maternal aunts", enables us to offer complex sharing customisation options to patients and clinicians. We perform a comparative analysis that shows that our identified requirements are not captured by existing approaches or languages, and so in this paper we investigate the use of an ontology-based data sharing framework, designed in modular way so as to support our identified challenges as well as future unforeseen requirements. Some aspects of medical data sharing prove deeper and more challenging than others, such as the support of multi-party privileges on data which can lead to conflicting statements, for which we also present a conflict detection algorithm and present a conflict resolution intervention that will work with humans (clinicians and patients) to reach a solution.

Our contributions are the following:

1. We discuss the case of health data ownership and point out the intricacies and complexities of this matter that are particular to health data.
2. We construct and deploy an application for sharing familial data to study user and clinician policies, preferences and consent. Using this system, and literature on patient-family data sharing, we gather a set of real health-based use cases that reveal challenges and requirements of data sharing agreements (Section 2). This informs the design of the building blocks of our framework.
3. We present an ontology to describe the actors, data types and processes that require capturing as part of a permissions-based genetic data sharing system in Section 4 and an ontology-based policy language in Section 5 that can work with our or any other ontology and allows multiple owners, fine-grained identification of resources and sharing requirements, definition of complex classes of individuals and data sharing policies. Multiple privileges might end up in a conflict.
4. We present a review of data sharing policies and machine-enforceable contracts. We map related approaches and identify their disadvantages in Section 6. We compare these languages to our framework and do an analysis with respect to meeting use case needs and detecting and dealing with conflicts in a real world setting.

2. Real Health Use Cases and Requirements of Data Sharing

In this work2 we focus on genetic health information, as experts have advised that the treatment of some of this data should be considered to be being owned by the family, not the individual [10]. Consider genetic diseases that have a known hereditary pattern, such as Huntington's, in which children have a 50% likelihood of inheriting the disease from a parent who has it. Current court decisions in the UK indicate that genetic risk of Huntington's can be considered familial, rather than confidential to an individual [10]. The actual diagnosis of the disease is still an individual's private data, but the risk and identified genetics can effectively be considered owned by the a family.

In order to fully understand how patients and consultants understand and interact with genetic health information, we built myKinMatters (Fig. 1). myKinMatters is designed to explore the decision making processes around the sharing of genetic test results. Users can construct family trees using a simple form-based interface. These trees can then form the basis of discussion with clinicians as to who in the family tree information should be shared with. myKinMatters allows users to share test result information via email with members on the family tree. The system allows the anonymous sharing of test results, providing an email handshake system enabling identity confirmation before sharing.

Through co-design with clinicians and observations of patient usage, which are ongoing as part of an NHS-based user study, the following requirements were identified with respect to individuals' setting sharing-policies over their data:

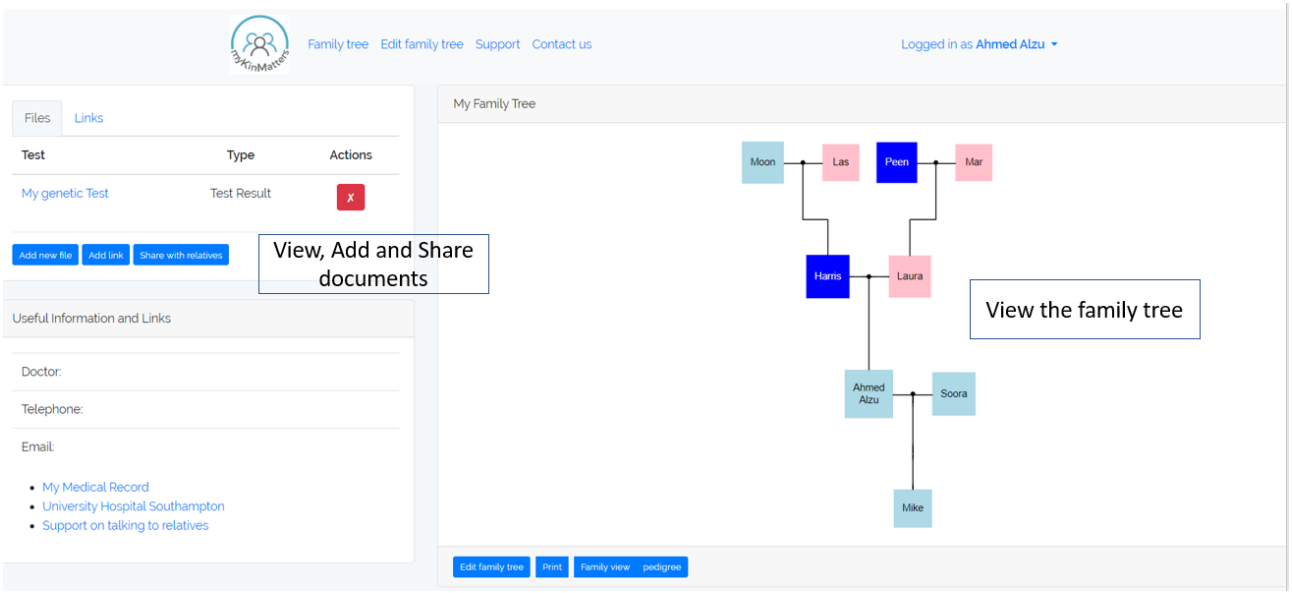

Figure 1. myKinMatters provides patients and clinicians with the ability to view genetic information and choose the data sharing policies that are appropriate for the individual, their family and medical care.

1. The ability for both patients and clinicians to express with whom data should be shared. When describing individuals to share data with, we can either define groups of users or allow patients to use complex classes and relationships, e.g., "all of my mother's sisters". The clinicians also use complex classes and relationships, but might use different terminology, e.g., "maternal aunts".
2. Record the process both patients and clinicians work through to determine what data should be shared with whom. In many of the observed interactions, the clinician must prompt the patient and suggest that the patient should share the data with a family member. This advice is not always taken, but the clinician's instigation is recorded.
3. Patients construct family trees in terms of genealogy, while clinicians express advice based on pedigree family trees; the system needs to be able to understand the relationship between these.
4. The system needs to understand what sharing has taken place so far as this may impact on future sharing decisions.
5. There will be generic information related to conditions that users might wish to include alongside shared test results, such as disease information or support information.
6. Patients have complex data sharing dependencies and wish to carefully manage the rate and spread of data, based on what other family members know, and when they know it.
7. With genetic data, some genetic information is owned by the family, while the diagnosis and disease progression is specific to a patient.

Based on observations of patient interactions, we can see several core examples of genetic information sharing among patients, families and clinicians. These were simplified and translated into Use Cases in Table 1.

UC1 highlights the very basic concerns that can be expressed in a family, and showcases how conflicts can arise. In this case, Alice was diagnosed with an autosomal dominant genetic disease as an adult. In such a case, the clinician would recommend that all first degree relatives be informed and tested. Alice wishes to share with all of her siblings, but her identical sister, also an owner of the data, does not wish their brother to know. Note, this is not necessarily a malicious statement, but can arise in complex family situations. Moreover, because Alice's identical twin sister has identical genetic information, the clinician's statement of "first degree relatives" applies to both Alice and her sister, even though this would include Alice's second degree relatives (i.e. her sister's children).

Table 1. Genetic sharing use cases with multi-party consent needs.

id	Name	Disease Pattern	User	Sharing Policy
UC1	Basic	Autosomal Dominant	Patient Alice	Share with my siblings and children.
			Alice's twin	Never share with Brother Ali.
			Clinician	Share with first degree relatives
UC2	Family vs. Pedigree	X-linked Recessive	Patient Bob	Share all information with all siblings.
			Clinician	Share with mother and her children.
UC3	Age Concerns	Autosomal Dominant Females at Risk	Patient Cathy	Share with my children, siblings.
			Cathy's partner	Share with the children after they are 15.
			Clinician	Share with first degree relatives
UC4	Anonymous	X-linked Recessive	Patient Doug	Share with all my children, but anonymize information.
			Clinician	Share with mother and her children.
UC5	Conditions	Gonadal Mosaicism	Patient Erica	Share all information with all my children. Don't share the results with my nieces and nephews directly, share with their moms.
			Clinician	Share with children.
UC6	Order Matters	Autosomal Recessive	Patient Fran	Share with my sister only after my mother has been informed.
			Clinician	Share with spouse. If carrier, share with all children.
UC7	Groups of Documents	Somatic mutation	Patient Greg	My children to see one of, not both the results.
			Clinician	– (not heritable)

UC2 shows the difference between patient and consultant terminology. Bob uses language from the family tree "siblings", which could include adopted sisters, half-brothers, etc. Because the disease is X-linked recessive, if Bob has it, his mother, Barbara, could be a carrier. Barbara's biological daughters could be carriers, and her biological sons will have the disease. To a clinician, the pedigree relationships of interest are based on inheritance patterns (e.g., mother and her biological children).

The concern over the protection of minors, and when children can and should be informed about their bodies is shown in UC3, in which Cathy wishes to share the information with her siblings and children, while her partner believes the children should not be informed before they are fifteen years old. Because the disease itself is autosomal dominant, Cathy's children have a 50% chance of inheriting it. There are other flavours of sharing in restricted ways beyond setting an age limit upon access of information. UC4 acknowledges that because of the privacy concerns of each individual, there need to be gradations in what is actually shared. In this case, Doug specifically asks for an anonymized version of the resource to be distributed instead of the actual resource itself.

Continuing with the concern for minors theme, in UC5, Erica wishes to share information with her children, but instead of sharing with her nieces and nephews, she is only comfortable sharing with their mothers, and letting them decide how much to divulge. Because her disease is caused by a gonadal mosaicism, the only individuals who could be affected by the disease are Erica's children; therefore, the clinician recommends sharing only with them.

UC6 illustrates cases where sharing may have specific time or disclosure-order conditions such as not sharing until certain conditions are true. In this case, an autosomal recessive disease is found in Fran. She has a 25% chance of passing this disease on to her children, but only if her partner is also a carrier. As such, the clinician has a series of dependencies in the testing and sharing of results. The clinician recommends that Fran's partner be shared with and tested. If Fran's partner is a carrier, then all children should be shared with and tested. At the same time, Fran has other time-sensitive sharing concerns. She does not want her sister to know anything until her mother is informed.

Finally, UC7 acknowledges that clinical test results can take several forms, from full tests results, through to summaries and anonymized results. In this case, Greg has a somatic mutation that has resulted in a disease. A series of tests have identified that Greg does not have an inheritable genetic disease, and he only wishes to share some pieces of that information. In fact, Greg is willing for *either*

the genetic test showing no chromosomal abnormalities *or* the screen for specific genetic markers to be shared, but not both. The clinician does not recommend sharing with any additional individuals because the underlying cause was found to be a somatic mutation, not an inherited genetic variant.

This analysis exposes several requirements with major ones being:

1. The need to define complex descriptions of individuals which you cannot always predetermine. A privacy policy management and data sharing system should have abilities to define complex classes such as the complex classes in ontologies or description logics [11].
2. The need to impose sharing dependencies, such as decisions to share based on history of sharing, or imposing a particular time order to the sharing of data or information.
3. The need to capture, actions, obligations but of equal importance, suggestions or instigations for sharing. The fact that a clinician suggests that the patient should share information with several individuals needs to be recorded as a first-class citizen in the events that are taking place in the data management system. This might be important for historic, legal or accountability reasons.
4. The need to capture different privileges or even ownership rights on the data. There is an ongoing policy discussion about ownership of genetic data and any data management and sharing system should take into account this peculiarity which might even lead to conflicts in the policy.
5. In our use cases some data might be anonymized and some not and this at different times for different parties. Decisions about sharing might depend on the anonymization of the data but also the converse: decisions about anonymizing depend on the sharing preferences. This dynamic interplay of anonymization and sharing is particular in health data management.

Notice that in many of the Use Cases, conflict exists between individuals. For example, in UC1, Alice, her twin and the clinician all conflict on the exact set of individuals to share with. Alice wishes to share with a greater set than the clinician recommends; Alice's twin with a narrower set than the clinician recommends. In this case, how do we resolve this conflict? One of the immediate outcome of interviews with patients and clinicians is the statement that conflict resolution and sharing should not happen automatically. Instead, the conflicts should be identified so that individuals can maintain agency. In addition to maintaining individual control, the exact ownership and distribution is dependent upon laws and regulations. Instead, we should allow individuals to express priority in terms of what sharing is most important.

To address these challenges in this paper we introduce an ontology-based, multi-privileges data access policy language.

3. The need for new policy languages and policy management systems

Our use cases expose a need different to traditional data privacy or security. Traditional data privacy approaches such as privacy[12], k-anonymity[13], or l-diversity[14], offer top-down, coarse-grained and authoritative privacy policies. These approaches coming from the Web, Cubersecurity, Artificial Intelligence or Data Management communities focus on either suppressing or altering (e.g., by masking, generalizing or inserting noise) information before it is released in order to achieve de-identification or ensure confidentiality of individuals' identities. Differently, approaches that focus on denying access rather than distorting information, are access control approaches, such as the common role-based [15] and attribute-based [16]. These approaches aim to completely disallow access to certain data, depending on roles/attributes of users and purposes, in order to maintain confidentiality of private information.

The basic assumption made by privacy technologies is that they protect against a non-trusted party, or an adversary. Thus they restrict data access or aim to protect a secret; such as the identity of an individual. In contrast there is not much focus on supporting automated approaches for privacy enforcing against relatively trusted parties. Limited focus in this direction was given by the so-called Privacy Enhancing Technologies, that offer languages for large scale systems and the World Wide Web. The now deprecated W3C standard of the P3P language (https://www.w3.org/P3P/) would be used

by web servers to specify their data handling practices in policies and web clients to state their own privacy preferences; based on an automatic match/mismatch between the two parts a webpage would be visited or not. These languages, similarly to "Terms and Conditions", are top-down, coarse-grained and they do not offer much customisation. Moreover they present "accept all or nothing" options and do not offer negotiation or partial access. These reasons led to an abandonment of most approaches and the proliferation of legal, natural language, "Terms of Use" agreements.

In this paper, we advocate the need for a paradigm shift, and suggest the design of machine-processable privacy language able to express sharing agreements, tailored to health data sharing and based on semantic web technologies which can provide common shared vocabularies for data sharing intentions and agreements. We also believe such a language should come with the algorithmic machinery that is needed to process these agreements. One can build on top of more general schemas such as FOAF, and schema.org that can be used to describe persons and personal data, or DICOM [17] for modeling healthcare and medical imaging metadata. Mulitple ongoing approaches use knowledge graphs to capture aspects of data sharing agreements. The Open Digital Rights Language (ODRL) [18] is a policy expression language that models content, services, actions, prohibitions, and obligations. PROV-O [19] models provenance information generated in different systems and under different contexts. More interestingly, the SPECIAL [20] project provides a vocabulary for expressing consent together with data processing workflows which take such consent into account, while in [21] the authors develop an ontology that models privacy policies described in actual medical research data sharing agreements. Starting from such Knowledge Graphs we envision different entities able encode their preferences and intentions of data usage in a machine understandable way and have data processing algorithms automatically enforce these preferences. In order to achieve this, the developed vocabularies have to be backed by the development of generic and re-applicable algorithms; possibly borrowing from data integration [22] or ontology-based query answering [23].

There is a need to develop theoretical results, algorithms and systems for expressing, supporting and managing detailed privacy preferences, personal consent and bilateral agreements of data access, usage and sharing. Systems need to achieve the following:

1) Create a bottom-up setting where individuals and organisations can create data sharing policies backed-up by a formal machine-processable technical language. This is in contrast to the classic service-client data sharing model where clients commit their data to a service provider after being presented with coarse-grained "Terms and Conditions" written in natural language, and on which the clients get only a few and simple opt-in/opt-out options. 2) Enable rich and more dynamic expressions of access and usage policies. Current access control systems are based on a predefined set of static rules in order to prohibit access to certain fields of a data repository; these access control rules cannot express sharing contracts such as a policy that prohibits a data item to be shared depending on its history of sharing, or on the amount of data already given to the data requester. 3) Develop algorithms that will allow data requesters to obtain the maximal result set of the query that still abides by the contracts of the data owners. 4) Support the goal of data auditing [24] which is to determine if private information was disclosed in answering queries, as well as the goal of accountability [25] which aims to understand the responsibilities of different parties in data processing. Data sharing agreements would be central in data auditing and accountability scenarios, since they inform exactly how to (or not to) use a particular dataset.

Towards these general objectives as well as to implement our particular health data sharing requirements we make steps to present an ontology-based framework and a policy language to share health information.

4. Our Ontological Framework

Through exploration of the possible use cases illustrated in Section 2 previously, and through detailed co-design sessions with clinicians exploring the processes of results' production and sharing, we have constructed an ontology to model the foundational information that will be required to

support the use cases. Previous work has shown how co-design of ontologies can be achieved with domain experts through the use of controlled natural languages [26]. The purpose of this conceptual high-level domain ontology is to articulate the different actors, entities and relationships that occur throughout the genetic testing and sharing process.

4.1. An Ontology for Genetic Relationships

As part of the myKinMatters intervention, those with genetic tests are asked to create a family tree that expresses their family structure. Participants will typically create a family tree in a genealogical sense, the interface allowing them to express family connections that aren't just pedigree familial connections (blood relatives.) The myKinMatters software understands how to present family relationships in both genealogical and pedigree formats, with clinicians typically working with the later. Pedigree family trees are a well established risk assessment tool in clinical medicine [27]. The Pedigree Standardization Working Group (PSWG) has been developing a standardised vocabulary and graphical represention for pedigree family trees [28]. The standard allows for expressions of the relationship between individuals and additional pedigree information down to the granularity of recording whether multiple gestation is monozygotic, dizygotic etc.

Our domain ontology does not seek to redefine detailed family tree ontologies as existing ontologies will be used such as that created by Santos et. al., [9].

4.2. An Ontology for Familial Consent

We present the conceptual domain ontology and not reifying it into a physical one, in order to ease presentation. Our ontology can represent data in the system and allow inferences to be used to understand conflicts within the system. Figure 2 describes the main classes and relationships that are required to represent the concepts inherent in expressing consent for sharing in a genetic information context. The ontology is designed to model both the data being shared but also the workflows of the processes followed by clinicians and patients in sharing test results. It has previously been demonstrated that capturing semantic descriptions of intent can enhance workflow practices [29].

A genealogical ontology, such as one reused from [29] is not pictured, but essentially defines a set of relationships and proprieties associated with the Person class depicted in Figure 2. The domain also recognises two sub_classes of Person, Patient and Clinician. This introduces the notion of roles within the system; more may become apparent over time. In addition our ontology is modelling groups of users, such as a group of patients taking part in the same study and have a collective right on results, or a team of clinicians. When highlighting conflicts in expressed permissions, or building systems that might make decisions based on data in the system, the roles involved could provide information on priorities, weighting of policies and conflict resolution strategies. For example, when highlighting conflicts, the policies of the patient might be given higher priority than the policy of the clinician to reflect the current legal polemics. Our system is not seeking to automate decision making at this point but rather to highlight conflict, so prioritising is not essential, but could be layered on top of the current ontological structure.

The class Test Results is a key concept in the ontology. A Test Result may comprise of several Resources. These might be Letters, Full Test Results, Summary Results or Information Resources. These may have specific properties such as being anonymous, that can be referred to in sharing policies. Policies are expressed about sharing Test Results. A Person expresses a Policy that relates_to a Test Result and concerns a Patient. Resources can be anonymous versions of results or summaries. In order to express that only anonymous results should be shared, as illustrated in the previous use cases (Use Case 4), it is necessary to express a sharing policy about an anonymous resource. Properties on the resource entity are used to identify it as anonymous to allow systems built on top of the ontology to highlight which resources are anonymous when users construct their policies.

Although out of scope of the use cases presented within this paper, our ontology allows the capturing of specific Diagnoses of a Condition that may be related to a Test Result.

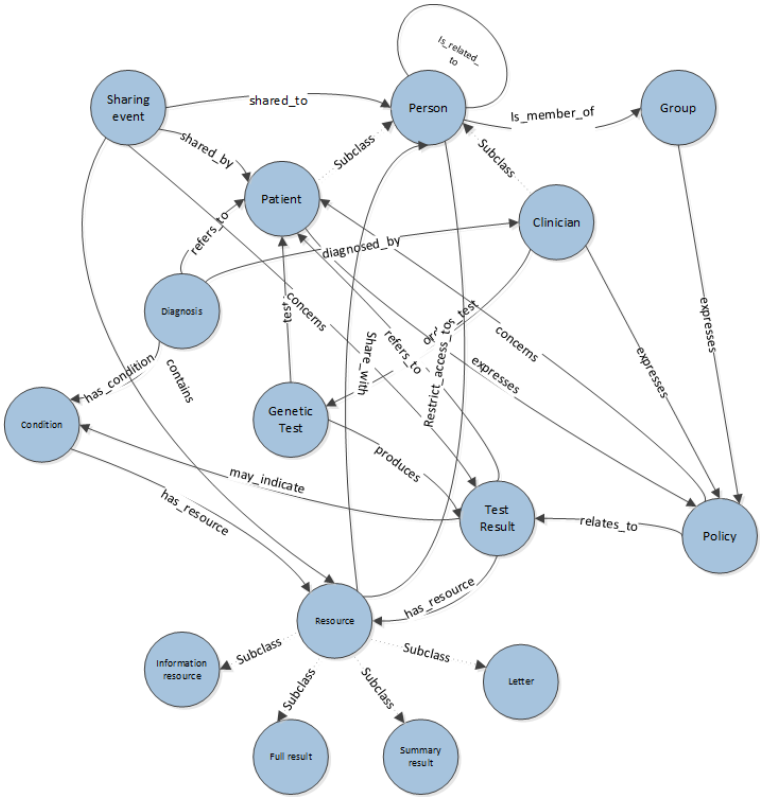

Figure 2. The myKinMatters conceptual ontology

In addition to information specifically about the Test Result, the ontology also allows the representation of data about the sharing of results. A Sharing Event captures the notion of Resources of a Test Result being shared with a Person. This information is required in order to express some of the conditions that might be included as part of a Policy statement, for example that a result should only be shared with a niece once it has been shared with their parent.

4.3. Rules and Conditions

Existing pedigree ontologies may contain relationships such as has_grandparent and has_child. These basic building blocks allow for the expression of complete family trees however a gap exists between relationships modelled in these ontologies and higher level constructs such as the notion of siblings, cousins, first order relatives.

With the classes and properties defined in our conceptual ontology we can revisit the use cases presented earlier with a view of expressing some of the additional vocabulary that is associated with the conditions attached to the policies. We will use the protege representation of SWRL [30] to represent these additional relational constructs for compactness.

Clinicians and Patients will likely wish to express policies around broad categories of individuals as can be seen from the illustrative examples in Table 1. The pedigree ontology may have explicit representations of some of these, but we are able to use rule based inferences to define additional relationships if required. Examples of these might be notions of 1st degree relatives, cousins, and siblings. A first degree relative is clinically defined as a person's parent, sibling, or child. This can be defined using the following simple set of rules:

$$Person(?x), hasParent(?x, ?y) -> isFirstDegreeRelative(?x, ?y)$$

$$Person(?x), hasChild(?x, ?y) -> isFirstDegreeRelative(?x, ?y)$$

```
author authorid
resource [URIs] {
share
   relation:conditions,
   relation:conditions
restrict
   relation:conditions:instead,
   relation:conditions:relation:conditions
}
```

Figure 3. High-level grammar for multi-owner data sharing

$$Person(?x), hasParent(?x, ?y), hasChild(?y, ?z) -> isFirstDegreeRelative(?x, ?z)$$

In a similar fashion we can define rules for the notion of siblings.

$$Person(?x), hasParent(?x, ?y), hasChild(?y, ?z) -> isSibling(?x, ?z)$$

which allows for a simplification of our third first degree relative rule to:

$$Person(?x), isSibling(?x, ?y) -> isFirstDegreeRelative(?x, ?y)$$

The cousin relationship can then be defined as:

$$Person(?x), hasParent(?x, ?y), isSibling(?y, ?z), hasChild(?z, ?w) -> isCousin(?x, ?w)$$

In the case of all the rules above, it is assumed that the isFirstDegreeRelative(), isSibling(), isCousin() are defined as symmetric properties. With this additional vocabulary defined it is then possible to express some of our sharing policies and conditions using our policy language.

5. A Policy Language for Multi-Privilege Consent

A resource, d_n, is a physical or digital object. For instance, a Database may be a collection of resources, or a resource itself. A resource has a set of properties, which are key-value pairs such as: id, name, anonymized_version, etc. A resource has a set of owners, O where o_i relates to a single individual who has ownership of the resource. Each owner can express the sharing allowances, or consent, for the resource, c_i.

Figure 3 shows a high-level grammar for a single policy defined in our policy language which operates over the resources and owners described above. Parts below that are written in bold are the language keyword. When defining the policy we need to state who is the owner of this policy , which is achieved by stating the policy owner or author in the place of authorId. A policy is not limited to a single resource, but can be attached to a set of resources. The resources to associate with a given policy should go within [URIs], where resources would be separated by delimiter ",".

The body of the policy includes two blocks: Share and Restrict. Those two blocks include rules that are applied to the resource of the policy. Relations and conditions are included in both blocks which define the rules. The first part of a rule is the relation, which can encompass generic relationships and specific identities, such as: parents(), sisters(), firstDegree(), children(Sam), Uncle(Tim). The second part of a rule is the *conditions*, using individual properties of the individuals contained by the relationships such as: age $\leq$ 15, gender==M, and age $\leq$ 112 and gender==F.

Semantics of the policy language is simple and intuitive. The id next to author declares the policy author and there can only be a single author for policy, although multiple policies can exist for a given resource, one for each associated owner. Next, the list of resources that are associated with policy are stated within URIs. The body of the policy includes share and restrict blocks: rules in *share* state the set

of people who the author wants to share the resources with. Rules in *restrict* state the set of people that the author does not want to have the associated resource. Each of the rules in share and restrict include *relations* and *conditions*. Relations evaluate to the list of individuals that the policy author has the relation with. For instance, brothers() would return the policy author's brothers. The conditions filter the people retrieved based on the relation. Filtering can be done by comparing individuals' properties such as age and gender. Boolean operators *and* and *or* can be used in order to combine several conditions. Conditions can be empty which indicates that there is no condition. Rules within share need to have a relation and conditions. An example of share rule is children():age > 15 and gender==F which means share with all policy author's daughters that are older that 15. Meanwhile, maternalUncles(): indicates the author wishes to share with the complex class of all maternal uncles, defined as a rule in the previous section.

Restrict rules return list of individuals which the policy' author does not want to gain access. There are three parts of a restrict rule. The relationship and conditions parts behave in the same manner as described above. However, the third part is called *instead*. The instead clause allows the policy author to state a list of individuals to gain access instead of the individuals retrieved by the semantics of relation and condition for the rule. The instead part itself includes list of rules which are identical to the share rules. Instead part is complete optional and can be left empty without including the '[]'.

An example of a restrict rule is nephews():age<15[parents():,Grandparents():] This is restrict rule states not to share with all nephews that are younger than 15 years old and instead share with their parents and grandparents. Meanwhile, a rule such as uncles(Mark):: in the restrict rules states not to share with specifically uncle Mark. Please note that the condition and instead were left blank but the relation can't be left blank.

The policy language allows users to state their preference on who should the resources get shared with. As identified through our requirements gathering with myKinMatters in Section 2, there is a difference between the family-tree relationships that patients use, and the pedigree relationships that clinicians use, even when they have the same names. For instance, "brother" to a clinician is a descendent from the same parent(s), while a "brother" to a patient could include step-brothers, adopted brothers, etc. Using an ontology as a part of the policy language will help clearly identify the individuals used from the relationships stated.

5.1. Expression of Use Cases

Using the example sharing policies for multiple-owner genetic data described in Section 2, we express each individual's sharing policy in the language described in Section 5. Table 2 contains the policy statements for each individual.

UC1 demonstrates very basic policy statements made by three parties: Alice, Alice's twin and the clinician. As stated in Table 1, Alice wishes to share with her siblings and children, while Alice's twin does not want to share with their brother Ali. The clinician indicates that all of Alice's first degree relatives, including siblings, parents and children, should be shared with. Please note that the clinician in this case uses a different ontology for identifying familial relationships than Alice does. At this point one could think of additional orthogonal dependencies on sharing that have to do, for example, with location (e.g., "share only with family in the UK"). Although we do not currently support spatial sharing dependencies this is something we actively investigate, in combination with other requirements that come with spatial separation (for example, a privacy law, such as the EU's GDPR. which might apply to a certain place but not to another).

UC2, UC3, and UC4 exercise variations in relationships and conditions in both the share and restrict clauses. In UC4, the policy expressed by the patient indicates sharing of resource D.d1, which represents an anonymized resource, a version of resource D, which is the full version. UC5 highlights the use of the instead clause; Erica does not wish to share with her nieces and nephews, instead sharing with their parents. UC6 represents the use case that the language does not yet adequately support. As identified in later conversations with myKinMatters users, there is occasionally a temporal element to

Table 2. Sharing policy language applied to patient sharing requirements.

id	Name	Patient	Other Owner	Clinician

UC1	Basic	author Alice
		resource [A] {
		share
		siblings()::
		children()::
		}
			author Alice's twin
			resource [A] {
			restrict
			brother("Ali")::
			}
				author clinician
				resource [A] {
				share
				firstDegree()::
				}

UC2	Family vs. Pedigree	author Bob
		resource [B] {
		share
		siblings()::
		}
			–	author clinician
				resource [B] {
				share
				mother()::
				motherChildren()::
				}

UC3	Age Concerns	author Cathy
		resource [C] {
		share
		siblings()::
		children()::
		}
			author Cathy's partner
			resource [C] {
			share
			children():age>15
			}
				author clinician
				resource [C] {
				share
				firstDegree()::
				}

UC4	Anonymous	author Doug
		resource [D.d1] {
		share
		children()::
		}
			–	author clinician
				resource [D] {
				share
				mother()::
				motherChildren()::
				}

UC5	Conditions	author Erica
		resource [E] {
		share
		children()::
		restrict
		nieces()->[-parents():]
		nephew()->[-parents():]
		}
			–	author clinician
				resource [E] {
				share
				children()::
				}

UC6	Order Matters	Currently under development in our language.	–	Currently under development in our language.

UC7	Groups of Documents	author Greg
		resource [G] {
		share
		children()::
		}
		resource[G.g1 and G.g2] {
		restrict
		}
			–	–

sharing, e.g., "do not share with my sister until my mom knows." This is not a feature yet of the first version of our prototype but currently being developed in our policy language. Finally, UC7 shows how the language can reason over the sharing of multiple resources. In this case, Greg does not want both G1 and G2 to be shared, only one or the other.

As we can see, it is possible to state all policies and relationships described in the requirements in Section 2 except for the temporal dependencies, an aspect under investigation; the first deployment of our user studies currently analyses use cases to see how complex patients may make these rules, before an implementation is decided upon.

5.2. Conflict Detection

The consent statement c_i contains a group of individuals, P, identified directly or through relationships that the information should be shared with. It also contains a set of individuals N,

identified directly or through relationships, that the information should never be shared with. If an individual is in both P and N, the individual is not shared with.

Definition 1. *For all c_i, c_j sharing policy constraints of resource d_n, the policy is conflict free if $P_{c_i} \cap N_{c_j} = \emptyset$.*

Algorithm 1, Find Conflicts, is the entry point that is passed a resource ID and returns the results of who should get access or any detected conflicts. Algorithm 2 fetches relevant policies, evaluates them, interprets the semantics of the policies and puts them into a structure that detect conflicts can understand. Meanwhile, in Algorithm 3, given the semantics of policies relevant to a resource, determines the individuals who get access or for which there are conflicts.

Algorithm 1: Find Conflicts

Input: Resource *r*
Output: Print out people that passed resource should get shared with and if any conflicts exist

```
policies ← findReleventPolicies(r)
results ← detectConflitcts(policies)
print result
```

Algorithm 2: Evaluating policies

Input: Set of policies *policies*
Output: Construct a structure for every policy [Share, Never]

```
results ← ∅
for all p in policies do
    shareRules ← getRules(p,"share")
    neverRules ← getRules(p,"never")
    share ← evaluateShareRules(p.author,shareRules)
    applyNeverRules ← evaluateNeverRules(p.author,neverRules)
    never ← applyNeverRules[0]
    instead ← applyNeverRules[1]
    share ← share ∪ instead
    res ← new RuleResults(policy = p, share = share, never = never)
    results ← results ∪ {res}
return results
```

Algorithm 3: Detect conflicts

Input: List of Rules Results *results*,
Output: Return a final results , the people who should get access + conflicts

```
result ← ∅
for all i in results do
    share ← i.share
    shareWith ← i.share
    for all j in results do
        intersection ← i.share ∩ j.never
        if not(intersection== ∅) then
            result.addConflict(i.policy,j.policy,intersection)
            allNeverUpdate ← allNeverUpdate ∪ intersection
        shareWith ← shareWith \ intersection
    result.addIndividuals(shareWith)
return result
```

6. Implementation and Evaluation

6.1. Comparison to Other Privacy Management Approaches

There are several privacy and privacy policy languages that can be used to capture either user/patient privacy preferences or organisation policies. These were developed with different objectives, ranging from helping users make decisions to enforcing security guarantees. In this section, we discuss several languages and show that to the best of our knowledge there is no language that address the health data sharing requirements identified in this paper. We present an evaluation of these languages against our criteria in Table 3. Some of the languages we have analysed in this section are open-world while others are closed-word: in a closed system we do not have to guess/predefine a list of purposes, a technical language can be used to implement any purpose as long as one knows the "schema" of our data. On the other hand, the privacy languages that were developed in an open world can combine potential uses (however often written in natural language) with predefined vocabularies that try to capture possible future uses of the data. By employing an ontology-based approach in this paper we hope to achieve the best of both worlds.

XACL [31] is XML-based and supports security policies and access policies for XML documents. **XACML** is also XML-based and is designed for access control. It supports two-way communication by implementing a request/response language [32]. There have been many extensions to XACML, such as developing profiles for usage control [33], for privacy/purpose policies [34], or some of the languages we study here. With minor effort XACML and its major extensions could capture sharing dependencies required in health data sharing but they do not capture complex classes of individuals, or the interplay of sharing and anonymising data. Moreover access control languages are closest to supporting what we here define that multi-ownership privileges.

The Enterprise Privacy Authorization Language **EPAL** [35] provides machinery to write enterprise-wide privacy policies on data and used within an enterprise. It allows positive and negative authorization privileges. To perform authorisation EPAL considers categories of purposes and of privacy actions, conditions or obligations. As such one would have to use the pre-conceived vocabulary of categories to perform some of our health data sharing requirements.

Rei [36] is a semantic web language focusing on annotating web entities with policies and on the use of distributed policy management. The policy language allows policies that contain obligations and requirements and the dynamic modification of existing policies. So Rei can express data sharing requirements such as dependencies and instigations, and as a semantic web language it can also capture complex classes of individuals. However the interplay of data sharing and anonymization and the multiple privileges on data is not captured. Another semantic web language featuring similar features and limitations is **AIR** [37] (Accountability In RDF). This policy language performs policy compliance via ontology reasoning, and provides explanations for its compliance results.

Sticky policies [38] are policies that accompany data and describe purposes, retention periods, and obligations to notify user when data is shared further. The language is able to express complex sharing dependencies, and possibly classes of individuals. It was mostly developed to allow comparing of user's privacy preferences to an organisation privacy policies, similar to P3P[39] and APPEL [40].

The now deprecated W3C standard of the **P3P** language [39] would be used by web servers to specify their data handling practices and web clients to use a language such as APPEL-P3P[40], or the XPath-based XPref[41], to state their own privacy preferences, based on an automatic match/mismatch between the two parts a webpage would be visited or not. There are several criticism points for P3P and similar "two-party" languages, such as EPAL[35], and Prime-DHP [42]. First, similar to "Terms and Conditions", often the policy is an "all-or-nothing agreement" and does not offer partial access. Second, even when users get options these are coarse-grained and limited to a set of predefined purposes and operations. Third, much criticism has focused on the security expectations that users had from these languages since most of them mix traditional adversary-oriented privacy requirements with the ability to express a bilateral agreement. Fourth, these are custom languages often with no formal, universal

semantics and with ad-hoc (if any) implementations. Fifth, many of these languages have tried to model, future and open-world potential uses of the data by a-priori exhaustively listing purposes for the user to choose from. Lastly, these ad-hoc languages many times are overly complicated, counter-intuitive and hard to use.

Another platform for enterprise wide data sharing is the Platform for Enterprise Privacy Practices **E-P3P** [43], which builds upon the P3P language discussed below. This language develops a fine-grained model for expressing privacy policies. E-P3P is similar to APPEL-P3P and both languages can exress rules to control privacy. While APPEL-P3P is used by end users, E-P3P is used by enterprises to describe their own policies. The language can describe the precedence of the rule against others, as well as the purpose for which data will be used. These features can capture our requirement for data sharing dependencies. The language also supports data categories, e.g., "contact data" or "test results" and this can be used to represent a resource in our framework. We could also be tempted to create a category of "multi-ownership" data and treat it differently, but this does not make the multi-privilege semantics that we need inherent to the system as we would like. The language also supports actions and obligations but not instigations which are not mandatory. Moreover the sole purpose of the language statements is to evaluate to a "ruling" for accessing a data resource, e.g., 'allow', 'deny', 'none', or 'error'. This setting seems complex but also binary that can evaluate only to accepting/rejecting a condition.

There is another language using the name **APPEL** [44] (The Adaptable and Programmable Policy Environment and Language) which is a very useful and simple but expressive syntax available to lay users in order for them to describe their policy containing triggers, conditions and actions. Such a language can be used to express sharing dependencies and complex classes of individuals but it would have a difficulty with the rest of our requirements.

The Purpose-to-use policy language **P2U** [45] is a policy language that was designed as an improvement of P3P, to allow the sharing of information across different applications paying particular attention to not expecting an a-priori list of purposes and uses of data; instead organisation state their purpose at the data collection time - and then they can only use the data for this purpose. This extension could support of our data sharing dependencies but these languages still miss most of the the rest of our requirements.

The PrimeLife Policy Language **PPL** [46] is an extension to XACML that uses sticky policies as well in order to provide security guarantees such as the sharing of data without revealing identities. It supports authorisations and obligations and tries to do access control and data usage at the same time. Focus is given to the obligation aspect and the language gives flexibility on the way users express their privacy policies. Still, the language supports working on predefined list of authorisation purposes which seem to be ad hoc pre-agreed scenarios between parties, wrapped in an XML format; the semantics of each different rule condition in that format is determined and implemented by the parties at the time they agree on a vocabulary, which makes the whole approach very custom and relying on ad-hoc agreements. An extension to PPL is The Accountability Policy Language, **A-PPL** [47] which focuses on accountability and collection of evidence of actions. Accountability languages can capture sharing instigations that is a requirement of our framework but they still miss our other requirements.

SecPAL [48] is an authorisation policy languages where policies are expressed in logic; policy compliance is performed via translating access requests to queries against a policy database. **SecPAL4P** [49] is an extension for handling personal identifiable information. It is possible to capture our anonymization versus sharing requirements through such a language but most of or other requirements are not captured.

All these languages were developed with a variety of reasons in mind. It is evident from our discussion that no prior language captures our identified requirements for health data sharing.

	Sharing dependencies	Instigations	Complex classes	Multi-ownership	Sharing vs. anonymization
XAC(M)L ([31–34])	✓	✗	✓	✓	✗
EPAL ([35])	✓	✗	✗	✗	✓
Rei ([36])	✓	✓	✓	✗	✗
AIR ([37])	✗	✗	✓	✗	✗
Sticky Policies ([38])	✓	✓	✓	✗	✓
P3P, APPEL ([39,40])	✗	✗	✗	✗	✗
XPref ([41])	✗	✗	✗	✗	✓
E-P3P ([43])	✓	✗	✓	✗	✓
APPEL ([44])	✓	✗	✓	✗	✗
P2U ([45])	✓	✗	✗	✗	✓
PPL ([46])	✓	✗	✗	✓	✓
A-PPL ([47])	✓	✓	✗	✓	✓
SecPAL, SecPAL4P ([48,49])	✗	✗	✓	✗	✓

Table 3. Privacy languages against health data sharing requirements

6.2. Mitigating conflicts example

It is clear from the sharing statements shown in 1, there will always be inherent conflicts in how individuals desire information to be shared. In some cases, e.g., Alice clearly wants to share her information with all of her siblings, while her identical twin wishes to exclude their brother Ali in Example 1. Other, more subtle cases should be considered though, such as Cathy and her partner in Example 3. In this case, both Cathy and her partner agree that the children should be shared with; however, Cathy's partner requests that they are shared with after they reach 15 years of age.

In the case of conflict, we do not advocate a particular conflict resolution practice. The actual resolution chosen must reflect the norms and desires of the stakeholders and the culture in which they exist. For example, in the UK, the patient currently "wins" any conflict regarding sharing. However, because the clinician has a duty of care to individuals who are affected, we have prototyped a conflict resolution response within myKinMatters, as shown in Figure 4. Figure 4 shows how the clinician sets the sharing policy statements within myKinMatters based on an individual and their risk. Please note that the clinician is not writing Sharing policy statements, or using the language described in Section 4. Instead, she uses her medical knowledge of the genetic inheritance at play in a given disease, and identifies who must be informed. If the clinician's policy conflicts with a patient's stated sharing policy, an extra automated process is invoked that looks up the GP of all individuals $\in \{consultant_{share} \cap patient_{notshare}\}$ and letters are sent directly to the GP stating that the National Health Service believes this individual needs to be tested. This process is shown in Figure 5.

There are two possible extensions that can assist with conflict resolution: priorities and explanations. Allowing patients and clinicians to express priorities of sharing or not-sharing concerns may allow either fewer conflicts to develop, or to provide a way for individuals to negotiate. In addition to this, providing the ability for free-text explanations, that allow users to understand both why the information should be shared with certain groups, or the reasoning behind not sharing, may also help negotiation. In order to be tractable, approaches similar to purpose-based access control [50] could be useful. In purpose-based access control, an ontology expresses all of the purposes that data may be used for. The access request is accepted or rejected based in part on whether the purpose satisfies the release-intent. In a similar manner, an ontology may help organize reasons for people releasing or not data. No matter what, expression of these sharing concerns, priorities and explanations must not be burdensome to the users, and further work in this area must be explored.

7. Discussion, Conclusions and Future work

Data sharing in a clinical setting can be complex, involve multiple stakeholders, and require the need to assimilate and reconcile a range of desires and intentions alongside regulatory and disciplinary boundaries. The work presented here has explored these issues through a worked through case study of the expression and sharing of genetic test information. The requirements are derived from

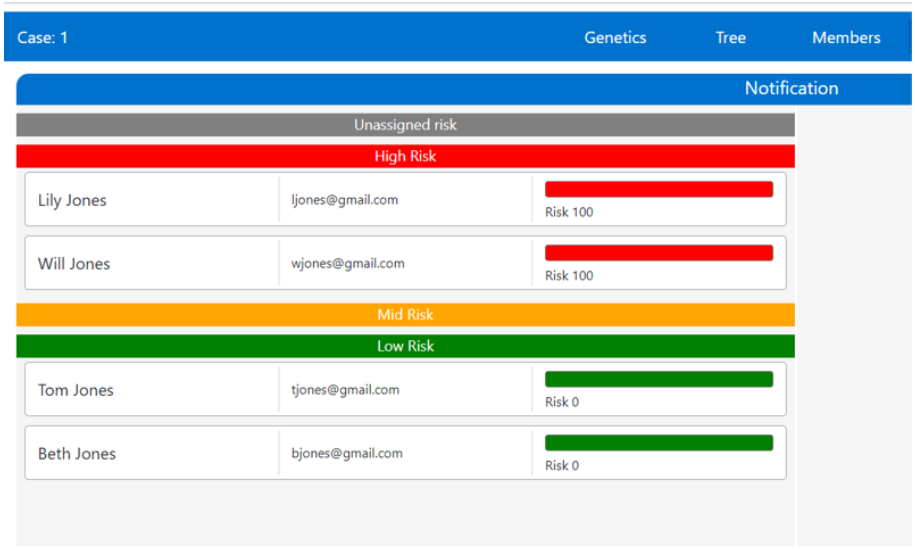

Figure 4. Conflicts can arise when a test result that is applicable to other family members occurs, and the patient is unwilling to discuss those results with the related members. Consultants can identify when familial-ownership of the data, because of disease inheritance patterns exist.

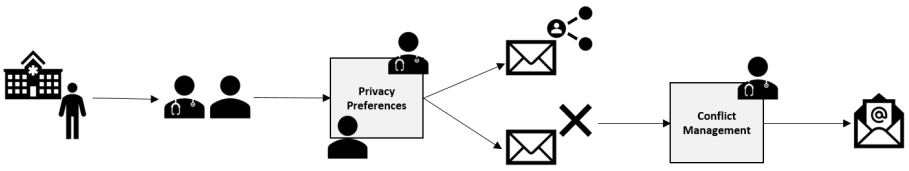

Figure 5. If the consultant and patient cannot resolve conflicts in access, an anonymized process that sends a letter to affected parties family doctor requesting additional testing can be generated and sent.

a co-design exercise that has sought to model the processes and data that exist where patients have genetic tests carried out and are then subsequently advised by clinicians that they should share this data with family members.

We advocated the need for new languages that offer machine-processable data sharing and privacy agreements. We constructed and deployed a prototype application that has allowed us to examine the processes of sharing familial data and to study user and clinician policies, preferences and consent. Using this system, and literature on patient-family data sharing, we have gathered a set of real health-based use cases. In Section 2, we presented seven illustrative use cases that highlight the challenges and complexities that can exist in the sharing of genetic test results. From these use cases we were able to distill requirements that underpin our ontological modelling of the system, our construction of the policy language, and our implementation of the conflict resolution system.

An underpinning requirement is for the policy system to understand the language in which clinicians describe their sharing desires and policies. In Section 4 we presented a conceptual ontology to describe the actors, data types and processes that require capturing as part of a permissions-based genetic data sharing system. Patients will wish to express their family tree information using more traditional genealogical family tree structures however clinicians will typically be working with pedigree family trees, that are more applicable to understanding the transference of genetic mutations through lineage. As such, our conceptual ontology and associated rules need to provide mappings between both these forms of family tree constructs, but also understand the particular language of discourse used in discussing sharing desires, in terms such as first degree relatives or cousins.

The information being shared is modelled within our ontology and our policy language caters for different versions of test results as well as notions of anonymity. As our use cases point to the need to share results anonymously, this places requirements on any eventual system to both have sharing

mechanisms that are anonymous, but also for the system to understand where the contents of test results might be appropriately anonymized or not.

With the underlying data structures expressed, we then presented our ontology-based policy language in Section 5. This language allows for multiple owners, fine-grained identification of resources and sharing requirements, as well as a range of complex classes of individuals and data sharing policies derived from our use cases and co-design process. The goal here is not to prohibit the expression of policies that conflict, but rather to capture when such conflicts occur. Having articulated the challenges and requirements of data sharing agreements this then informs the design of our framework and building blocks. The analysis of this policy language against other languages with respect to meeting use case needs, detecting conflicts, dealing with conflicts in a real world setting, and instigation. These are described in Section 6. This section presented an in depth review of related work on data sharing policies and machine-enforceable contracts. As described in Section 3, a large number of policy languages already exist, and run the gamut of implementation languages, how they are attached to the data and important privacy scenarios. We map related approaches and identify their disadvantages to the use cases we have identified as important in this domain in Section 6. While there is no clear previous policy language that is an immediate fit for the concerns expressed in this work, we would like to consider the use of "obligations" as used by many privacy languages, particularly for the use of the hospital data controller who is not represented within the stakeholders of this work. In this case, obligations would need to be considered when evaluating conflicts; some conflicts would disappear as sharing would be deemed unacceptable. In the future we will extend our current work to build a fully fledged language to capture agreements, preferences, policies, and consent in health data sharing and deploy this in our developing ontology-based framework.

Author Contributions: Conceptualization, G.K., A.C., M.J.W., L.M.B. and A.M.L.; Methodology, G.K., A.C., M.J.W. and L.M.B.; Software, A.A. and L.M.B.; Validation, A.C., M.J.W., A.A. and L.M.B.; Formal analysis, G.K., A.C., M.J.W.; Investigation, G.K., A.C., M.J.W., L.M.B. and A.M.L.; Resources, G.K., A.C., M.J.W., L.M.B. and A.M.L.; Data curation, A.C., M.J.W. and A.A.; Writing—original draft preparation, G.K., A.C., M.J.W., A.A. and L.M.B.; Writing—review and editing, G.K., A.C., M.J.W., L.M.B. and A.L; Visualization, A.C., M.J.W., A.A. and L.M.B.; Supervision, G.K., A.C., M.J.W., L.M.B. and A.M.L.; Project administration, G.K., A.C., M.J.W., L.M.B. and A.M.L.; Funding acquisition, G.K., M.J.W., A.C., L.M.B. and A.M.L. All authors have read and agreed to the published version of the manuscript.

Funding: G.K. was supported by an Alan Turing Institute Fellowship. M.J.W. was supported by ESRC grant number ES/R009058/1. A.A. was supported by an ECS Centre for Health Technologies undergraduate internship. L.M.B. was funded by a Research Fellowship from Health Education England Genomics Education Programme and the Wessex Innovation Fund. A.M.L. is co-lead for the Data Science cross cutting theme of the NIHR Southampton Biomedical Research Centre and holds a Wellcome Trust Collaborative award in ethics and society.

Acknowledgments: The authors thank Don Cruickshank, Tasneem Bawendi, Liberty Gamble, Andreea Ghita, James Lister, Daniel Lockyer, Oana Paul, Luke Pullman and Maksim Romanovich for their development and maintenance efforts on myKinMatters.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lucassen, A.; Gilbar, R. Alerting relatives about heritable risks: the limits of confidentiality. *BMJ* **2018**, *361*, k1409.
2. Konstantinidis, G. The Need for Data Sharing Agreements in Data Management. *In Proceedings of the Second International Workshop on Semantic Web Technologies for Health Data Management* **2019**.
3. Batte, B.; Sheldon, J.P.; Arscott, P.; Huismann, D.J.; Salberg, L.; Day, S.M.; Yashar, B.M. Family Communication in a Population at Risk for Hypertrophic Cardiomyopathy. *J. Genet Couns.* **2015**, *24*, 336–348.
4. Blandy, C.; Chabal, F.; Stoppa-Lyonnet, D.; Julian-Reynier, C. Testing Participation in BRCA1/2-Positive Families: Initiator Role of Index Cases. *Genet. Test.* **2003**, *7*, 225–33.
5. Finlay, E.; Stopfer, J.E.; Burlingame, E.; Evans, K.G.; Nathanson, K.L.; Weber, B.L.; Armstrong, K.; Rebbeck, T.R.; Domchek, S.M. Factors Determining Dissemination of Results and Uptake of Genetic Testing in Families with Known BRCA1/2 Mutations. *Genet. Test.* **2008**, *12*, 81–91.

6. Julian-Reynier, C.; Eisinger, F.; Chabal, F.; Lasset, C.; Nogues, C.; Stoppa-Lyonnet, D.; Vennin, P.; Sobol, H. Disclosure to the family of breast/ovarian cancer genetic test results: Patient's willingness and associated factors. *Am. J. Med. Genet.* **2000**, *94*, 13–8.
7. McGivern, B.; Everett, J.; Yager, G.; Baumiller, R.C.; Hafertepen, A.; Saal, H.M. Family communication about positive BRCA1 and BRCA2 genetic test results. *Genet. Med.* **2004**, *6*, 503–9.
8. Costalas, J.W.; Itzen, M.; Malick, J.; Babb, J.S.; Bove, B.; Godwin, A.K.; Daly, M.B. Communication of BRCA1 and BRCA2 results to at-risk relatives: A cancer risk assessment program's experience. *Am. J. Med. Genet. C.* **2003**, *119C*, 11–18. doi:10.1002/ajmg.c.10003.
9. Santos, J.M.; Santos, B.S.; Teixeira, L. Using ontologies and semantic web technology on a clinical pedigree information system. In Proceedings of the International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Heraklion, Crete, Greece, 22–27 June 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 448–459.
10. Royal College of Physicians, Royal College of Pathologists and British Society for Genetic Medicine. *Consent and Confidentiality in Genomic Medicine: Guidance on the Use of Genetic and Genomic Information in the Clinic*, 3rd ed.; Report of the Joint Committee on Genomics in Medicine. London: RCP, RCPath and BSGM, 2019.
11. Baader, F.; Calvanese, D.; McGuinness, D.; Patel-Schneider, P.; Nardi, D.; others. *The Description Logic Handbook: Theory, Implementation and Applications*; Cambridge University Press: Cambridge, UK, 2003.
12. Dwork, C. Differential Privacy: A Survey of Results. In *Theory and Applications of Models of Computation; Agrawal, Proceedings of the International Conference on Theory and Applications of Models of Computation, Xi'an, China, 25–29 April 2008; Springer: Berlin Heidelberg, 2008; pp. 1–19.*
13. *Sweeney, L. k-anonymity: A Model for Protecting Privacy.* Int. J. Uncertain. Fuzz **2002**, *10*, 557–570.
14. Machanavajjhala, A.; Venkitasubramaniam, M.; Kifer, D.; Gehrke, J. l-Diversity: Privacy Beyond k-Anonymity. ACM Trans. Knowl. Discov. Data, **2007**, *No. 1*, 3-es
15. Sandhu, R.S.; Coyne, E.J.; Feinstein, H.L.; Youman, C.E. Role-based access control models. *Computer* **1996**, *29*, 38–47.
16. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Eecurity. Alexandria, VA, USA, October, 2006; Association for Computing Machinery: New York, NY, USA; pp. 89–98.
17. Digital Imaging and Communications in Medicine, Available Online: https://www.dicomstandard.org/. (accessed on 6 April 2020)
18. W3C. ODRL Information Model 2.2, Available Online: https://www.w3.org/TR/odrl-model/. (accessed on 6 April 2020)
19. W3C. PROV-O: The PROV Ontology, Available Online: https://www.w3.org/TR/prov-o/. (accessed on 6 April 2020)
20. Bonatti, P.; Kirrane, S.; Petrova, I.; Sauro, L.; Schlehahn, E. The SPECIAL usage policy language. Technical report, V0. 1. Technical report, 2018, Available online: https://www.specialprivacy.eu/. (accessed on 6 April 2020)
21. Li, M.; Samani, R. DSAP: Data Sharing Agreement Privacy Ontology. Semantic Web Applications and Tools for Healthcare and Life Sciences, 2018. Available Online: http://hdl.handle.net/11375/23755 (accessed on 6 April 2020)
22. Konstantinidis, G.; Ambite, J.L. Scalable query rewriting: a graph-based approach. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Athens, Greece, June 2011; Association for Computing Machinery: New York, NY, USA; pp. 97–108.
23. Pérez-Urbina, H.; Rodrıguez-Dıaz, E.; Grove, M.; Konstantinidis, G.; Sirin, E. Evaluation of Query Rewriting Approaches for OWL 2. In Proceedings of the Joint Workshop on Scalable and High-Performance Semantic Web Systems (SSWS+ HPCSW 2012), Boston, MA, USA; p. 32.
24. Agrawal, R.; Bayardo, R.; Faloutsos, C.; Kiernan, J.; Rantzau, R.; Srikant, R. Auditing Compliance with a Hippocratic Database. In Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, 29 August – 3 September 2004; pp. 516–527.
25. Weitzner, D.J.; Abelson, H.; Berners-Lee, T.; Feigenbaum, J.; Hendler, J.; Sussman, G.J. Information accountability. *Comm. of the ACM* **2008**, *51*, 82.

26. Denaux, R.; Dolbear, C.; Hart, G.; Dimitrova, V.; Cohn, A.G. Supporting domain experts to construct conceptual ontologies: A holistic approach. *J. Web Semant.* **2011**, *9*, 113 – 127. Provenance in the Semantic Web, doi:https://doi.org/10.1016/j.websem.2011.02.001.
27. Frezzo, T.M.; Rubinstein, W.S.; Dunham, D.; Ormond, K.E. The genetic family history as a risk assessment tool in internal medicine. *Genet. Med.* **2003**, *5*, 84–91.
28. Bennett, R.L.; French, K.S.; Resta, R.G.; Doyle, D.L. Standardized human pedigree nomenclature: update and assessment of the recommendations of the National Society of Genetic Counselors. *J. Genet. Counsel.* **2008**, *17*, 424–433.
29. Pignotti, E.; Edwards, P.; Gotts, N.; Polhill, G. Enhancing workflow with a semantic description of scientific intent. *J. Web Semant.* **2011**, *9*, 222–244.
30. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosof, B.; Dean, M.; others. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission* **2004**, *21*, 1–31.
31. Hada, S.; Kudo, M. XML Access Control Language: provisional authorization for XML documents. Available Online: http://xml.coverpages.org/xacl-spec200102.html. (accessed on 6 April 2020)
32. Parducci, B.; Lockhart, H.; Rissanen, E. Extensible access control markup language (XACML) version 3.0. *OASIS Standard* **2013**, pp. 1–154.
33. Masood, R.; Shibli, M.A.; Bilal, M.; others. Usage control model specification in XACML policy language. In Proceedings of the IFIP International Conference on Computer Information Systems and Industrial Management. Venice, Italy, 26–28 September 2012; Springer: Berlin/Heidelberg, Germany, pp. 68–79.
34. Parducci, E.; Lockhart, H.; Rissanen, E. XACML v3. 0 Privacy Policy Profile Version 1.0. *Policy* **2010**, pp. 1–11.
35. Ashley, P.; Hada, S.; Karjoth, G.; Powers, C.; Schunter, M. Enterprise privacy authorization language (EPAL). *IBM Research* **2003**, *30*, 31.
36. Kagal, L.; Finin, T.; Joshi, A. A policy based approach to security for the semantic web. In Proceedings of the International Semantic Web Conference, Sanibel, FL, USA, 20–23 October 2003; Springer: Berlin/Heidelberg, Germany; pp. 402–418.
37. Kagal, L.; Hanson, C.; Weitzner, D. Using dependency tracking to provide explanations for policy management. In Proceedings of the 2008 IEEE Workshop on Policies for Distributed Systems and Networks. 2–4 June 2008, Palisades, NY, USA; pp. 54–61.
38. Bezzi, M.; Trabelsi, S. Data usage control in the future internet cloud. In Proceedings of the Future Internet Assembly. 17–19 May 2011, Budapest, Hungary; Springer: Berlin/Heidelberg, Germany; pp. 223–231.
39. W3C. The Platform for Privacy Preferences 1.0. Available Online: https://www.w3.org/TR/P3P/. (accessed on 6 April 2020)
40. W3C. A P3P Preference Exchange Language 1.0 Available Online: https://www.w3.org/TR/P3P-preferences/. (accessed on 6 April 2020)
41. Agrawal, R.; Kiernan, J.; Srikant, R.; Xu, Y. XPref: a preference language for P3P. *Comput. Netw.* **2005**, *48*, 809–827.
42. Ardagna, C.A.; Cremonini, M.; De Capitani di Vimercati, S.; Samarati, P. A privacy-aware access control system. *J. Comput. Secur.* **2008**, *16*, 369–397.
43. Ashley, P.; Hada, S.; Karjoth, G.; Schunter, M. E-P3P privacy policies and privacy authorization. In Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society, November 2002, Washington, DC, USA; Association for Computing Machinery: New York, NY, USA; pp. 103–109.
44. Turner, K.J.; Reiff-Marganiec, S.; Blair, L.; Cambpell, G.A.; Wang, F. APPEL: Adaptable and programmable policy environment and language **2014**.
45. Iyilade, J.; Vassileva, J. P2u: A privacy policy specification language for secondary data sharing and usage. In Proceedings of the 2014 IEEE Security and Privacy Workshops. 17–18 May 2014, San Jose, CA, USA; pp. 18–22.
46. Ardagna, C.A.; Bussard, L.; De Capitani di Vimercati, S.; Neven, G.; Pedrini, E.; Paraboschi, S.; Preiss, F.; Samarati, P.; Trabelsi, S.; Verdicchio, M. Primelife policy language. In Proceedings of the W3C Workshop on Access Control Application Scenarios. 17–18 November 2009, Abbaye, Luxembourg.
47. Azraoui, M.; Elkhiyaoui, K.; Önen, M.; Bernsmed, K.; De Oliveira, A.S.; Sendor, J. A-PPL: an accountability policy language. In *Data privacy management, autonomous spontaneous security, and security assurance*; Springer, 2014; pp. 319–326.

48. Becker, M.Y.; Fournet, C.; Gordon, A.D. SecPAL: Design and semantics of a decentralized authorization language. *J. Comput. Secur.* **2010**, *18*, 619–665.
49. Becker, M.Y.; Malkis, A.; Bussard, L. A framework for privacy preferences and data-handling policies. *Microsoft Research Cambridge Technical Report, MSR-TR-2009-128* **2009**.
50. Byun, J.W.; Bertino, E.; Li, N. Purpose Based Access Control of Complex Data for Privacy Protection. In Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies, June 2005, Stockholm, Sweden; pp. 102–110.