Healthcare data safe havens: towards a logical architecture and experiment automation

David Robertson[1], Fausto Giunchiglia[2], Stephen Pavis[3], Ettore Turra[4], Gabor Bella[2], Elizabeth Elliot[5], Andrew Morris[5], Malcolm Atkinson[1], Gordon McAllister[6], Areti Manataki[1], Petros Papapanagiotou[1], Mark Parsons[7]

[1]School of Informatics, University of Edinburgh, Edinburgh, UK
[2]Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, Trento, Italy
[3]NHS National Services Scotland, Edinburgh, UK
[4]Azienda Provinciale per i Servizi Sanitari, Trentino, Italy
[5]Usher Institute of Population Health Sciences and Informatics, University of Edinburgh, Edinburgh, UK
[6]School of Medicine, University of Dundee, Dundee, UK
[7]EPCC, University of Edinburgh, Edinburgh, UK
E-mail: dr@inf.ed.ac.uk

Published in *The Journal of Engineering*; Received on 3rd June 2016; Accepted on 10th October 2016

Abstract: In computing science, much attention has been paid to generic methods for sharing data in secure infrastructures. These sorts of methods and infrastructures are, of course, necessary for sharing healthcare data. The authors are, however, a long way away from being able to realise the potential of medical and healthcare data to support the sorts of extensive, data-intensive experiments being demanded by precision and stratified medicine. A key architectural problem remaining to be solved is how to maintain control of patient data within the governance of local data jurisdictions, while also allowing these jurisdictions to engage with experiment designs that (because of the need to scale to large population sizes) may require analyses across several jurisdictions. This study provides a snapshot of architectural work underway to provide a clear, effective structure of data safe havens within jurisdictions. It then describes how formally specified experiment designs can be used to enable jurisdictions to work together on experiments that no single jurisdiction could tackle alone. The authors' current work relates to two jurisdictions (in Scotland and in Italy), but the architecture and methods are general across similar jurisdictions.

1 Introduction

A wide variety of tools, methods and architectures are capable of enriching and analysing medical data. There is also great activity in European regions to develop carefully managed, large repositories of medical data derived from regional healthcare authorities and governed according to the practices of local jurisdictions. These safe havens are primarily maintained through the public sector and come into limited contact with researchers outside of jurisdictions. The solution to this problem is not simply to move healthcare data to a generic 'data warehouse' or 'trusted cloud'; there needs to be a framework for engagement between havens and researchers, combined with automated methods for the necessary transfer of data and provision of services. That framework must respect the governance rules of local data jurisdictions.

The scale of data resources currently held in local jurisdictions is very large. For example, in Scotland (population 5.5 million) data are collected at individual level each time a person has contact with the health service – from pre-birth through to death. Computerised information for acute and day case admissions goes as far back as 1968, with approximately 1.4 million new records added each year. There are around 90 million community prescriptions in Scotland every year and over 20 million clinical images are available across the population going back to 2010. These administrative data are held on a variety of IT platforms within secure NHS networks. The availability of unique patient identifiers within Scotland allows individuals' records to be linked across time and disease categories. A careful balance has been struck between ensuring that public benefit is derived from these national data resources, whilst simultaneously protecting the privacy of individuals. The local safe havens seek to ensure the continuation of public trust in the use of health care data via a set of relatively sophisticated governance processes and procedures.

Research on healthcare data safe havens has recently gained momentum both from an academic and a public sector or industrial perspective. This is particularly visible in the UK where political and social drivers and previously highly criticised initiatives, such as Care.data [1], have made finding solutions to key functional and ethical issues a priority. The Scottish Government, in particular, considers safe havens to be a crucial element of its health informatics strategy [2] and has the vision to set an international standard for the safe and secure use of electronic health records for research purposes [3]. It is worth noting that a big proportion of research papers published in this field concern work carried out in Scotland. The importance of safe havens has also been recognised outside the UK, with the American Medical Informatics Association highlighting the need to further investigate the use of safe havens as part of the effort towards creating a learning healthcare system [4]. There have been increasingly more discussions around governance principles for such safe havens, linked to initiatives worldwide [5–9]. For example, Ford *et al.* describe information governance challenges, including secure data transportation, record matching, data anonymisation and data access control [10]. Burton *et al.* give a focused description of data safe havens in terms of proposed socio-technical criteria that need to be fulfilled, including safety, security, ethics, accessibility, and reliability [11]. Similarly, Laurie *et al.* argue for adaptive governance models, safe and effective data management, and responsiveness to highly dynamic expectations [12]. Pavis and Morris describe the Scottish model, in which project-specific research datasets are made available via safe havens to approved researchers, who are held accountable for privacy protection [13]. Safe havens are also discussed as part of efforts towards linking health and social care data, which require data sharing between jurisdictions and collaboration between health and social care teams [14, 15].

Discussions such as the aforementioned ones, which focus on the key principles and the desired societal effects of data safe havens, are fairly widespread. However, there is limited correspondence to a more formal architecture that can lead to a data safe haven implementation that enforces the desired principles while remaining flexible and adaptive. There is also hardly any discussion around the automation of experiments within and across jurisdictions.

This paper aims to tackle precisely this gap through the discussion of a logical architecture and experiment automation approach, which respect the governance regulations of local data jurisdictions. A hierarchically structured data safe haven is proposed in Section 2, which enables the clear tracking of data management responsibility. The process of experimentation as it relates to locally controlled data is next discussed (Section 3), highlighting the central role of an experiment specification in the data sharing process. Cross-jurisdiction experiments are examined in Sections 4 and 5, clarifying the roles of data controllers and third-party data integrators. We continue, in Section 6, with a discussion of experiment automation, at the heart of which lies a formal experiment specification as a data sharing contract. In Section 7, we demonstrate the applicability of our approach through a case study that involves an experiment currently undertaken across Scotland and Trentino. We conclude, in Section 8, with an overview of the benefits of our approach and a discussion of directions for future work.

2 Data safe havens – purpose and architecture

In our view, the purpose of a data safe haven (within a governing jurisdiction) is to maximise the benefit that can be derived from its data to medical science (and, ultimately, to healthcare) through more effective data-intensive experimentation within a responsibly regulated environment.

Given this purpose, key principles underlying the architecture for a data safe haven are

(i) **Acceptability:** It must be such that a deployed version will meet privacy, integrity, security and ethical concerns necessary for approval, but also sufficient to satisfy broader public scrutiny.

(ii) **Usability:** It must be feasible, where necessary, for researchers to develop experiments using familiar methods and workflows, subject to governance constraints.

(iii) **Sustainability:** As far as possible the software used must have either known support, e.g. from vendors, or a known active open source community. Where it is appropriate as part of the architecture to contribute to open source software then it is necessary to budget for contributing to that community and be sure there is a sufficient (global) community behind the open source that it can be anticipated to continue. Where it is appropriate to lead the development of architecture-specific software, this should be developed with the Software Sustainability Institute's model of sustainable software.

(iv) **Flexibility:** The functionality of the application programming interfaces (APIs) offered by the core system to other software should allow a wide range of future analyses, that are as yet unpredictable, and should as far as practicable allow the architecture to accommodate much larger data volumes than initially encountered.

(v) **Diversity:** The architecture should be capable of evolving to support the many varieties of data formats currently in use and frequently changing. As the need arises, and subject to authorization and scrutiny by safe haven managers, participants in research should be able to create ways of handling these items.

(vi) **Scalability:** Although early instances of the architecture will rely primarily on human trust and associated manual operation of experiment workflow, as the volume of data and experimental demand increases it should be capable of adapting to include automation of workflow where this is consistent with other architectural and governance principles.

(vii) **Validity:** The management and governance of the implemented architecture should ensure that potential modes of failure and misuse will be progressively identified and enumerated. As each is identified, tests for the subsequent releases of the architecture should be introduced and overseen in a clearly defined governance structure that includes independent oversight.

With these principles in mind, we are developing the safe haven architecture shown in Fig. 1. This shows the principal (conceptual) elements of data storage and transfer between NHS core systems and research safe havens. These elements are:

- **NHS core systems** from which data can be transferred to a first level safe haven only through a gateway system that permits an approved range of transfer operations (systems of approval, attribution and responsibility are discussed below).
- **Research safe havens** are environments within which data is managed and analysed by authorised researchers. The principal (first level) safe haven contains data originating directly from the NHS core system (following limited de-identification, described below) and access to it is restricted to researchers/managers approved to handle data at this (comparatively low) level of de-identification. Subsets of the data available in the first stage safe haven can be made available (via an appropriate, approved, first stage gateway) to a second level safe haven which (depending on the approval needed for the data concerned) may be accessed by a different group of researchers/managers. Fig. 1 shows two levels of safe haven but the sequence of levels can extend further (data obtained from the second level being used to generate a third level and so on).
- **Data gateways** provide data transfer from core data to the first level safe haven and between sequences of levels of safe haven. Each gateway is governed by rules of permitted data transfer (implemented either through manual control or with various degrees of automation depending on the purpose of the gateway). This allows fine-grained control of data supply plus control of access to data by researchers (depending on the safe haven to which data is extracted). For example, in Fig. 1, the core data gateway system is responsible for transfer of data from core NHS data set to the first stage data set and, in practice (because of the likely scale and diversity of first stage data), this gateway is restricted to highly trusted data managers under control of NHS. Gateway systems may differ depending on the data sets available in the safe haven to which each gateway applies and depending on the data requirements anticipated in the safe haven to which approved data will be delivered. An example of a gateway system is the Research Data Management Platform (RDMP), developed in Dundee, to provide a framework and a suite of tools for the management and curation of longitudinal research datasets [16]. It not only performs many typical data extraction, transformation and loading tasks, but also includes tools for management of the research lifecycle (including documentation of datasets, cohort linkage and reproducibility of project extracts). Of course, not all gateway systems need to be this complex, but the RDMP demonstrates the level of complexity we expect to find in a sophisticated gateway system.
- **Data paths** are routes through which data can (with appropriate use of gateways) be obtained in 'downstream' data sets from 'upstream' data sets. In Fig. 1, the second level data set (downstream) is derived on a data path leading back (via first stage gateway) to the first level data set and further back (via the core data gateway) to the core NHS data.
- **Metadata stores** are produced by data gateways to record key properties of the data transferred. Depending on the transfer operation, metadata may be added in the data haven from which a gateway has derived data and/or in the haven to which it has contributed data. For example, in Fig. 1 the core data gateway will, in practice, de-identify data elements through re-naming them,

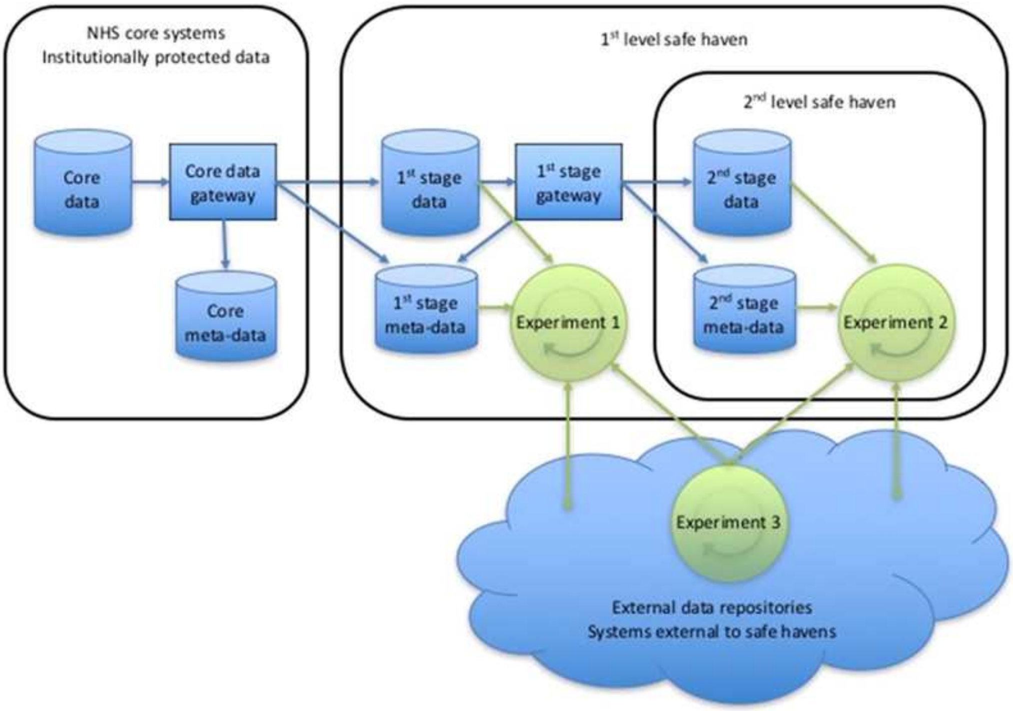


Fig. 1 *Conceptual model of data safe haven architecture*

but (for purposes of attribution) it may store core metadata (private to NHS core systems) on the relationship between NHS identifiers and synthetic identifiers used in data derived for the first stage data set.

- **Experiments** access (meta)data within the appropriate safe haven (depicted as Experiments 1 and 2 of Fig. 1). These experiments may also draw on data from external data repositories and safe havens and (subject to governance approval) may produce results data for external use (e.g. publishing). Experiments may also be run externally to safe havens for the purpose of acquiring data to enrich other experiments within safe havens (depicted as Experiment 3 of Fig. 1). We expand on formal definition of experiments, and the role of these in cross-jurisdiction data sharing, in Section 3.

The architecture above assumes that different data sets will be derived, in stages, from core NHS data. These different data sets will be accessed by different groups of researchers and data managers within a uniform, attributable, auditable system of governance. The human aspects of this system are arranged in a strictly hierarchical system as follows:

- Each safe haven has a manager with authority over and responsibility for the operation of that safe haven, the gateways and subhavens it contains and any analytics/data-management tools it contains.
- Gateways between safe havens establish paths of data management (as described above). For example, Fig. 1 contains a data management path from a first level safe haven to a second level safe haven via a first stage gateway. If there had been a second gateway, at first stage, leading to a different second stage haven then we would have two data management paths. More generally, data management paths branch as a tree from the root of the first stage.
- The responsibility for data management across a data path is then the union of data managers for the safe havens along that path. For the paths in our example of Fig. 1, the responsibilities for the path leading to the second stage data set is the union of managers for safe havens at levels 1 and 2.

The core architecture described above is concerned only with the management of data sets and with controlling responsibility/access roles related to these. The work of analysing and managing these data sets is more diverse (and perhaps specific to particular data sets). We therefore view these as extensions to the core architecture. Common forms of extension include:

- **Query and analysis systems:** provided in order to perform experimental analyses within appropriate safe havens. Their availability may differ between safe havens because it may be necessary to restrict the analyses available to some groups of researchers (e.g. on data with particular sensitivities) or it may be that some safe havens are set up for researchers who bring with them approved tools that are not available to others (e.g. because of licensing agreements).
- **Data translation systems:** provided to shift between data formats.
- **Ontology mapping systems:** provided to elate data in one naming system to data in another.
- **Data export systems:** provided for making data available externally to the safe haven. This might be through export of appropriately de-identified analytics results or it might be the generation of synthetic data, supplied as a way of engaging external computing scientists with data representative of safe haven data.

The diversity of extensions allowed in any given safe haven will be determined by the appropriate manager, based on: the roles of individuals permitted in that haven; the nature of data it contains; and other governance constraints as appropriate. Within these constraints, the aim is to encourage responsible use of a diverse range of tools as extensions to the architecture.

3 Data experiments

In our view of healthcare data safe havens, data sharing is always understood in the context of experiment specifications that describe the specific analyses that researchers wish to perform on regional data. Within the experiment, specification is a formal contract for data sharing for the experiment. This is the key to effective governance because it describes precisely, at the level of experiment design, how all parties expect data to be used across an experiment. Fig. 2 shows the central role of an experiment specification in the data sharing process. Experiment requirements are described by researchers and reviewed with the relevant governance specialists, resulting in a formal experiment specification that describes the necessary interactions between data controllers and researchers in order to complete the experiment. This specification contains no patient-specific data so can be reviewed/adapted outside data safe havens. It is used within safe havens by data controllers who (subject to each controller's governance and ethics constraints) use their segment of the experiment specification as a definition of the actions expected of them by the experiment (typically this will involve querying locally controlled data sets, de-identifying the data and abstracting away detail that might identify individual people). Under control of data controllers, appropriate results can then be derived as per the experiment specification; then brought outside the local jurisdiction for experiment analysis and validation. Throughout this process, the experiment specification acts as an overall plan for the experiment, giving a form of contract between the parties involved and providing transparency of process (external to the data controllers) for review by governance bodies.

As well as providing rigour and transparency, our long term goal is to be able to maximise automation of the process of data acquisition and analysis across healthcare jurisdictions, while also being able to verify at every stage that the operations performed on data are within the safe envelope specified by the system of governance applicable to all the data concerned. This requires standardisation of data schemas, ontologies and metadata (around the group of key data assets in the safe haven); formal definition of the processes/ and analytics methods used in data management, along with the security policies (and accompanying permissions and obligations) used to ensure governance compliance. These should be defined independently of the infrastructure used to host the data, so allowing automation via compatible local (bespoke) servers, trusted cloud architectures or personal devices.

4 Cross-jurisdiction experiment architecture

Fig. 3 shows the essential components of the architecture for conducting experiments across jurisdictions. Data controllers operate through data safe havens within their jurisdictions (in the diagram we have two data controllers, each operating within a different jurisdiction). Each controller has its own control and responsibility for accessing and integrating data within its jurisdiction. A safe haven integrator is an entity that is trusted by data controllers to maintain contracts for data sharing between jurisdictions. To facilitate accurate data sharing, each controller and the safe haven must be able to align those parts of its data ontology that are relevant to appropriate data sharing contracts. As we show later, the safe haven integrator can (with appropriate design of data sharing contracts) be kept strictly separate from data operations within jurisdictions, which allows a clear separation of responsibilities in the contract and close control of data by controllers within an agreed contractual framework.

From an organisational point of view, data controllers and integrators have the following important properties:

Data controller

- Typically, there is one data controller per jurisdiction. For example, the two jurisdictions relevant to our research are NHS National Services Scotland (NSS) and the Azienda Provinciale per i Servizi Sanitari in Trentino (APSS).
- Each controller has a defined legal responsibility for data within its jurisdiction.
- Each controller holds defined data sets, and associated metadata with a data safe haven (which typically is a collection of federated databases, as in Scotland, rather than being a single physical storage site).
- Each controller controls both data sets and ontologies for that data.
- Data controllers are the only entities able to undertake data extraction from sources in their jurisdiction. They are responsible (subject to governance approval) for enacting each data access request to interrogate specified data with a specified query – thus producing experimental results.
- In the current operating model, each data query must seek to query data sets held solely by a single data controller. In Scotland the model would then be that NSS (the data controller) seeks the minimum data required to answer the specified question (for example a combination of a locally-held dataset and a national dataset). All required information is placed into a safe haven within which the experimenter (appropriately authorised) is then able to undertake analysis. This data integration process requires the transfer of both data and subsequent experimental results.
- Data controllers can support activities related to the design and interpretation of experiments. They also, however, support activities related to the operation of experiments (data extraction, integration and analysis) and these are core elements of our data sharing contracts (see Section 6.1).
- Data controller functionality via NSS in Scotland is currently supported by chargeable access to expert facilitation staff. This is not chargeable access to data or sale of data. The model operates on a cost recovery basis with opportunities for price variation dependent on customer type.

Data integrator

- A data integrator is an entity trusted by data controllers to maintain contracts for data sharing between jurisdictions.
- A key function therefore is to preserve and monitor these data sharing contracts, which are formal specifications of the process of data sharing between controllers.
- Unlike data controllers (which must be on systems controlled by local data jurisdictions), a data integrator can potentially be hosted on any system, so it could be operated via one of the data controllers or by a third party.
- A data integrator does not undertake data extraction (unless it is also a data controller).
- It enables data integration by coordinating the activities of the appropriate data controllers through reference to the data sharing contract (example of a specification language for such contracts is given in Section 6).
- A further important function of a data integrator is ontology mapping. When a data integrator completes the creation of a data sharing contract it is critical to ensure that each query is correctly interpreted by the data controller with the responsibility to execute it, so that terminology is used consistently overall. In the simplest case, we could ensure this sort of consistency by involving the relevant data controllers in the design of the data contract, ensuring that the ontologies of controllers precisely match that of the contract. In practice, however, data controllers will have differing ontologies for the data sets in their jurisdiction so it will be necessary to define mapping rules that relate terminology in contracts to ontologies in jurisdictions. We return to this issue in the context of our data sharing contracts in Section 6.2.
- A final important data integrator function relates to verification and attribution in data sharing contracts. A key benefit of the formal data sharing contracts described in Section 6 is that they can be used to retain a precise, high-level record of the interaction between (and to an appropriate level within) data controllers. In Section 6.4, we discuss this in more detail.

5 The role of third parties, external to data controllers

The data transaction examples defined in this paper could be completed, in their simplest form, between a data controller and an experimenter. As experimental or query complexity increases,

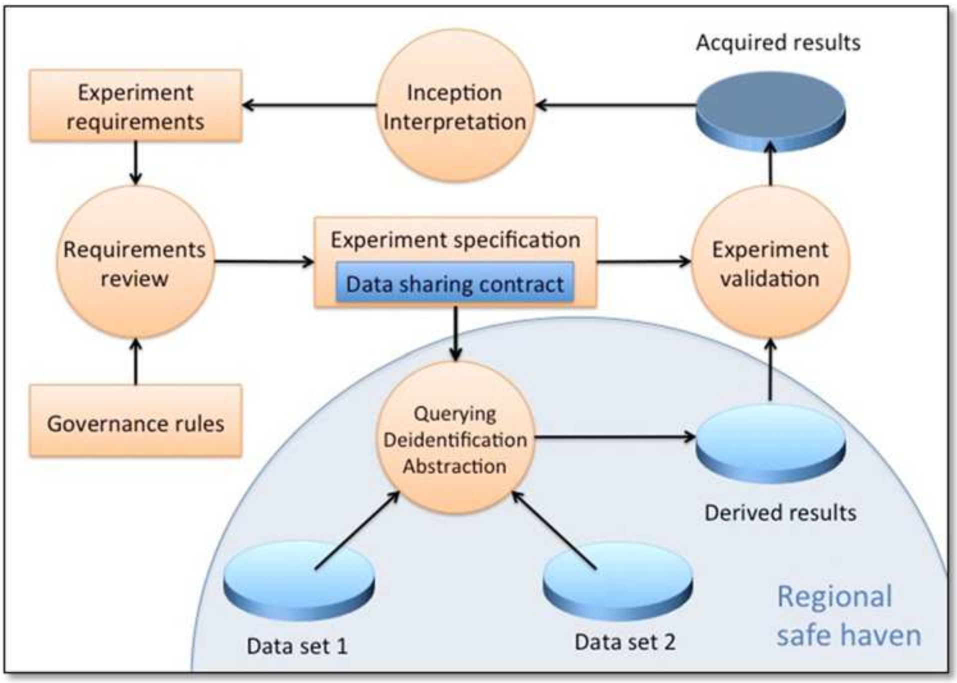


Fig. 2 *Overview of the process of experimentation as it relates to locally controlled data*

additional roles/role-holders may need to be added, for example additional data controllers, the use of a third-party data integrator and so on. In such cases, two broad categories of activity can be assigned to an external third-party:

- Experimenter support: for example data query design or results analysis and interpretation (annotation, statistics, graphical presentation etc.).
- Activities within a safe haven: through provision of tools that support, enable or accelerate processes owned by the safe haven and associated data controller.

Both can be effectively underpinned with the 'software as a service' business model the basis for which is a periodic or ad hoc subscription model. Chargeable access might be based upon;

- Infrastructure costs for the computational footprint required (compute and/or storage).
- Application licensing for the analytics/software tools provided.
- System integration (depending on data flows to be supported and resources to be made available).
- A managed service wrapper (service desk function, first/second/third line support teams, potentially consultancy/development for new integrations).
- Bespoke elements included (and potentially charged as consultancy work) by the responsible party.

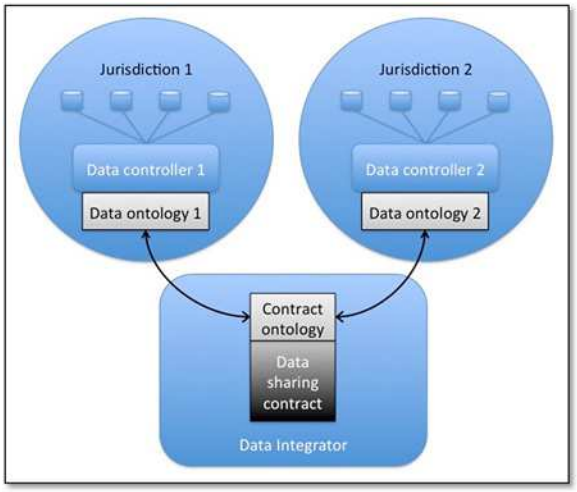


Fig. 3 *Sharing data across jurisdictions*

Multiparty solutions are also possible, with several external third-parties collaborating to deliver complex, integrated service provision – for example with subscription pricing based on cost recovery for the elements above by each partner. One vehicle for this might be contracting or service level agreements via a single frontend service provider, with 'back-to-back' contracts/service level agreements (SLAs) to the partners involved in service delivery.

In our view, the purpose of a data safe haven (within a governing jurisdiction) is to maximise the benefit that can be derived from its data to medical science (and, ultimately, to healthcare) through more effective data-intensive experimentation within a responsibly regulated environment.

6 Formal experiment specifications as data sharing contracts

A data sharing contract is a formal specification of the process of data sharing between controllers. It is specified in sufficient detail that it, potentially, can be used as a script for automating the process of data sharing (so it is an executable specification). Its principal function, however, is to describe precisely how data is shared for a given purpose. An example of a data sharing contract described in the Lightweight Coordination Calculus (LCC) [17, 18] process language is given in Fig. 4. In the example, we define what each participant in the data sharing activity is allowed to do – we refer to this as the role of the participant. The definition of each role is in terms of the communications sent to other participants (in their roles), with M⇒a(R, X) being a message sent to participant X in role R and M⇐a(R, X) being a message coming in from participant X in role R. When a communication depends on the participant performing some operation on data we attach a description of this operation using the '←'

symbol, so for instance M⇒a(R, X)←Op means that message M is sent to participant X in role R if data operation Op is performed.

In generic process languages, like LCC, we can describe many different data sharing contracts, but we supply one example in this document for illustrative purposes (see Fig. 4). Our example has three roles: an experimenter (coordinating the experiment) and two data controller roles (controllers 1 and 2). Note that the controller roles must be undertaken within the local jurisdictions, but the coordinating experimenter role could be undertaken on any system prepared to be the integrator:

- The experimenter coordinates the experiment by asking controller 1 to extract the results for query Q1; then asking controller 2 to extract the results for query Q2; then receiving from each of these the locations (L1 and L2) of the data files where the results of each query can be found; then asking controller 1 to combine data from locations L1 and L2 (this is the point at which results are shared between controllers, since L2 is not in controller 1’s jurisdiction); then finally receiving a report of the results, R, from analysis of the combined data.
- Controller 1 receives a request to extract the results for query Q1; then (if it is willing and able to enact Q1 on the data in its jurisdiction) sends the location, L1, of the results file; then it receives a request to combine data from locations L1 and L2; then sends report, R, of the analysis (if it is willing and able to perform the analysis).
- Controller 2 receives a request to extract the results for query Q2; then (if it is willing and able to enact Q2 on the data in its jurisdiction) sends the location, L2, of the results file (Fig. 4).

The specification provided in Fig. 4 is executable and enacting it on an appropriate infrastructure generates a sequence of communications between the experimenter and controllers 1 and 2. Each of the role definitions defines what is expected of the corresponding participant, so each participant’s contract for data sharing is separable and clearly defined. In the example, the data operations take place only in the two data controller roles (controllers 1 and 2), with the experimenter only sending and receiving communications to/from the controllers. This has two important consequences: that the experimenter never accesses data in controller’s jurisdictions (it only receives a report of the results of analyses) and that it makes no technical difference where the experimenter role is physically coordinated (it could be coordinated on either of the controllers or independently).

6.1 Operations available within jurisdictions

The operations available to data controllers within jurisdictions are of four broad types:

- **Data extraction operations**: reach into data sets and pull out (smaller) data sets according to query definitions.
- **Data integration operations**: combine data sets to produce (larger) data sets.
- **Data analysis operations**: apply data analytics algorithms to data sets to produce (new) results inferred from the original data.
- **Data abstraction operations**: transform data sets according to anonymisation requirements.

In our previous example, enact_query(Q, L) is a data extraction operation, while analyse_data(L1, L2) is a data analysis operation.

In the most basic form of safe haven integration we need not have a standard language for data operations (assuming that standardisation of operations can be done through agreement between the data controllers who wish to share data) but the ideal, to which we should strive, is for there to be a standard repertoire of data operations that is maintained through agreement between controllers, thus making it straightforward to execute a data sharing contract once it is approved.

The three-role data analysis scenario (involving an experimenter and two controllers) described above can then be constructed from the four basic operations as follows:

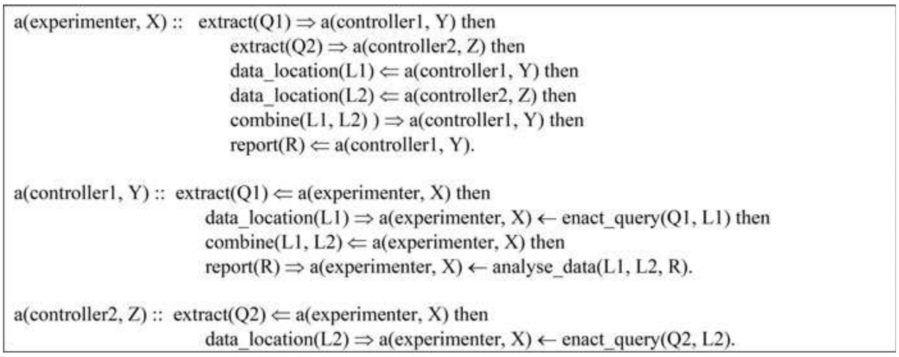
(i) Controller 1 receives a query from the experimenter:
(a) controller 1 executes the corresponding data extraction operation,
(b) controller 1 runs a data abstraction operation over the extracted data (for the purposes of anonymisation with respect to the laws of its jurisdiction),
(c) controller 1 indicates the location of the resulting dataset 1 to the experimenter.

(ii) The same operations for controller 2 returning dataset 2.
(iii) Controller 1 receives from the experimenter a request for analysis of a data set combined from dataset 1 and dataset 2:

(a) controller 1 executes the corresponding data integration operation,
(b) controller 1 runs a data abstraction operation over the combined data set in order to make sure no reconstitution of personal data is possible from the combined data set,
(c) controller 1 executes the data analysis operation,
(d) controller 1 sends back the result of analysis to the experimenter.

6.2 Ontologies and ontology mapping

The final element of standardisation in our architecture (beyond the contract language and data operations) is the ontology used to express data queries and related domain-specific terminology. In our example, this would be necessary, in practice, to ensure that query Q1 could be interpreted correctly on the data sets accessible to controller 1 and (similarly) that query Q2 could be interpreted by controller 2. In the simplest case, we could ensure this sort of



```
a(experimenter, X) ::  extract(Q1) ⇒ a(controller1, Y) then
                          extract(Q2) ⇒ a(controller2, Z) then
                          data_location(L1) ⇐ a(controller1, Y) then
                          data_location(L2) ⇐ a(controller2, Z) then
                          combine(L1, L2) ) ⇒ a(controller1, Y) then
                          report(R) ⇐ a(controller1, Y).

a(controller1, Y) ::  extract(Q1) ⇐ a(experimenter, X) then
                       data_location(L1) ⇒ a(experimenter, X) ← enact_query(Q1, L1) then
                       combine(L1, L2) ⇐ a(experimenter, X) then
                       report(R) ⇒ a(experimenter, X) ← analyse_data(L1, L2, R).

a(controller2, Z) ::  extract(Q2) ⇐ a(experimenter, X) then
                       data_location(L2) ⇒ a(experimenter, X) ← enact_query(Q2, L2).
```

Fig. 4 *Example of a data sharing contract described in the LCC language*

consistency in terminology by involving the relevant data controllers in the design of the data contract, ensuring that the ontologies of controllers precisely match that of the contract. In practice, however, data controllers will have differing ontologies for the data sets in their jurisdiction so it will be necessary to define mapping rules that relate terminology in contracts to ontologies in jurisdictions. For example, query Q1 in our contract might be ‘sex = male & age > 16’ but because it is in Italy controller 1 might need to translate this into the query ‘sesso = maschile & anni > 16’. This can be done via the mappings (sex = sesso), (male = maschile) and (age = anni).

Data controllers will use a multi-layered architecture for their ontologies where language-specific terminology, language-independent (conceptual) domain knowledge, and data structures are represented as separate layers. Such an architecture decouples the problem of multilingualism (as shown in the example above), from that of the alignment of domain knowledge and of data structures, allowing for easier portability of knowledge to new languages.

To improve interoperability across data controllers, besides ontology mapping a partial pre-alignment of ontologies will be made possible. This pre-alignment is consensus-based, i.e. it is only attempted to the extent where an agreement among peers can be reached with reasonable effort. The resulting shared ontology will be hosted by the Safe Haven and made accessible to all parties. Ontology mapping will only need to be applied to knowledge not shared in this manner.

6.3 *Data integration*

Search, combination, or analysis of data sets requires some level of understanding of their underlying meaning. Such meaning is formalised with respect to reference knowledge held by data controllers. *Data integration* refers both to the process of discovery of meaning within a data set and to its alignment to reference knowledge.

We perform data integration in an *entity-centric* manner where data records are ultimately aggregated around entity instances and are stored in the data controller’s entity base. Integrated entities can subsequently be queried, searched, combined, and be subjected to data analysis operations in a consistent manner.

Generally, entity-centric data integration is a semi-automated process consisting of a series of *alignment steps*:

(i) **terminology alignment** between the terms used in the data set and the vocabulary layer of the reference knowledge;
(ii) **schema alignment** between the structure of the data set and the corresponding reference data structure (i.e. entity type);
(iii) **data alignment** to clean data values and to bring them to a form coherent with the aligned schema;
(iv) **entity alignment** to convert data records into entities and establish relations among entities;
(v) **metadata alignment** to define provenance, terms of use, privacy policies for the entities created.

6.4 *Verification and attribution in data contracts*

Since our data contracts are designed to contain the key details of the interaction needed between jurisdictions in order to share data, it is straightforward to record instances of completed contracts. These instances can then be used as evidence of appropriate data sharing, for verification and auditing purposes, and for attributing data to jurisdictions.

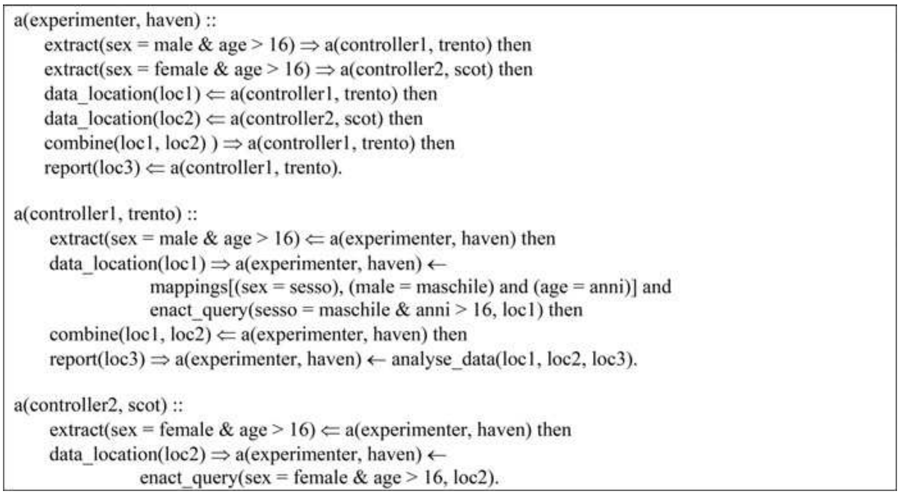
For example, the experiment contract used as our running example might have been used with two data controllers (‘trento’ and ‘scot’) and a safe haven called ‘haven’ to extract males over 16 from trento and females over 16 from scot; then combine these data for analysis at trento. The resulting contract instance, documenting the data transactions, is given in Fig. 5 (where loc1, loc2 and loc3 are locations for data).

6.5 *Security policies in contracts*

Extraction operations performed on data within jurisdictions are always performed by the appropriate data controllers, giving them familiar control over this fundamental aspect of data sharing. In our running example, this means (for instance) that the decision on whether or not enact_query(sex = female & age > 16, L2) is actually done is under the control of controller 2, which is managing those data, and controller 2 could choose not to enact the query if it felt that security/privacy rights might be compromised. It would also be possible, in data sharing circumstances where more subtle forms of assurance are needed, to build these into the formal data sharing contract. For example, we might want to have a variant of our contract for controller 1 (in the example in Fig. 4) where it does not simply analyse the data taken from its own jurisdiction (L1) in combination with that from controller 2’s jurisdiction (L2) and send the results (R) immediately to the experimenter but, instead, gives controller 2 the opportunity to check R and confirm that it is happy with these data before they are communicated to the experimenter. The contract for this would be as shown in Fig. 6.

6.6 *Security in infrastructure*

The data contracts and ontology mappings described above can be specified independently of the infrastructure used to share data.

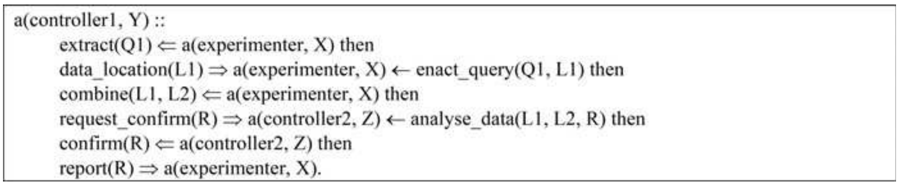


```
a(experimenter, haven) ::
    extract(sex = male & age > 16) ⇒ a(controller1, trento) then
    extract(sex = female & age > 16) ⇒ a(controller2, scot) then
    data_location(loc1) ⇐ a(controller1, trento) then
    data_location(loc2) ⇐ a(controller2, scot) then
    combine(loc1, loc2) ) ⇒ a(controller1, trento) then
    report(loc3) ⇐ a(controller1, trento).

a(controller1, trento) ::
    extract(sex = male & age > 16) ⇐ a(experimenter, haven) then
    data_location(loc1) ⇒ a(experimenter, haven) ←
                mappings[(sex = sesso), (male = maschile) and (age = anni)] and
                enact_query(sesso = maschile & anni > 16, loc1) then
    combine(loc1, loc2) ⇐ a(experimenter, haven) then
    report(loc3) ⇒ a(experimenter, haven) ← analyse_data(loc1, loc2, loc3).

a(controller2, scot) ::
    extract(sex = female & age > 16) ⇐ a(experimenter, haven) then
    data_location(loc2) ⇒ a(experimenter, haven) ←
               enact_query(sex = female & age > 16, loc2).
```

Fig. 5 *Example of a data sharing contract instance between two data controllers and a safe haven*



```
a(controller1, Y) ::
        extract(Q1) ⇐ a(experimenter, X) then
        data_location(L1) ⇒ a(experimenter, X) ← enact_query(Q1, L1) then
        combine(L1, L2) ⇐ a(experimenter, X) then
        request_confirm(R) ⇒ a(controller2, Z) ← analyse_data(L1, L2, R) then
        confirm(R) ⇐ a(controller2, Z) then
        report(R) ⇒ a(experimenter, X).
```

Fig. 6 *Example of a security policy in a contract*

Indeed, in the simplest case (where all steps in the contract are performed manually) there need not be a means of enacting contracts on computing infrastructure (and the data contract is just a more precise form of a conventional experiment description). However, a key feature of our executable specifications is that they can be used to automate sharing, given appropriate infrastructure and appropriate interpretation machinery for the formal specifications. This raises requirements for the infrastructure used in automation:

- When data located at one jurisdiction will be used in analysis at another jurisdiction (as in analyse_data(L1, L2, R) in the example above) we need to be sure that only Data controllers authorized to see the relevant data sets can actually access those data.
- The contracts themselves (if used automatically to control sharing) must be held in such a way that they cannot be maliciously adapted (for instance, to change the contract so as to send data to a third party).
- The design of contracts must be such that data with strong security restrictions is not 'leaked' into areas where weaker security is assumed. This safeguard is maintained in conventional data sharing by manual inspection of experiment design but our formal specifications could support (limited forms of) automatic verification of these sorts of properties.

7 Experiment case study

In Section 2, we described the concept of a hierarchically structured data safe haven. Then, in Sections 3 and 4, we explained how formally defined experiment designs could form contracts that enable experiments across safe havens to be automated. To give the reader an indication of the complexity of experiment designs in this context, we include in this section a practical experiment that we are undertaking between jurisdictions in Scotland and Trentino.

The purpose of the study is two-fold:

(i) To determine the absolute risks of mortality and serious vascular events among those treated with non-vitamin K oral anticoagulants following a spontaneous intracranial haemorrhage.

(ii) To compare the mortality and serious vascular event risks in those treated with non-vitamin K oral anticoagulants versus warfarin, following an intracranial haemorrhage.

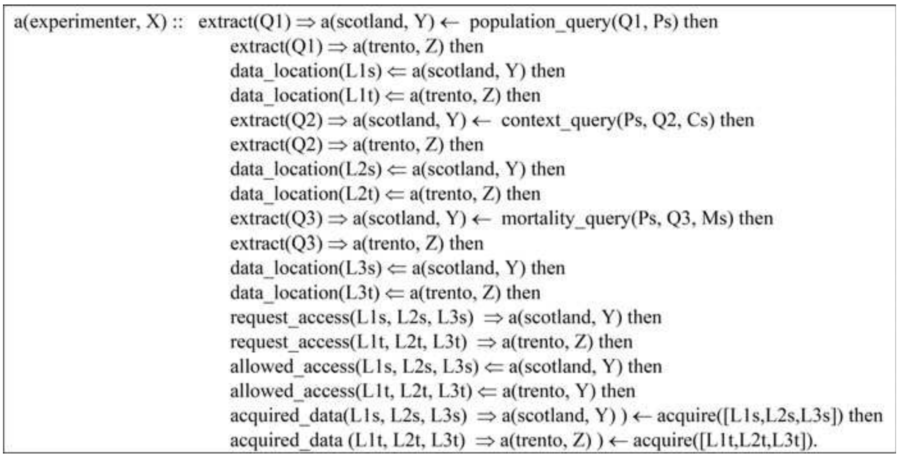
Patients who have experienced an intracranial haemorrhage will be identified from hospital admissions. Linkage to community prescription data will identify those patients who were administered non-vitamin K oral anticoagulants (Rivaroxaban, Dabigatran, Apixaban) or Warfarin during the 90 days after hospital discharge. Subsequent mortality and vascular events (ischemic stroke, systemic embolism, intracranial haemorrhage, extracranial haemorrhage) will be identified through linkage to national mortality records (which provide date and underlying cause of death) and hospital admission data.

Other covariates to be adjusted for in the analyses include: a measure of deprivation (income-based socioeconomic status assessment), age, sex, comorbidity (determined based on the co-occurring conditions which impacted treatment as recorded in the hospital discharge record and hospital admissions in the 5 years prior to the intracranial haemorrhage).

The feasibility of this cross-jurisdiction study therefore depends on the availability and capacity to link together, at individual patient level three key national health administrative datasets in each of Italy and Scotland:

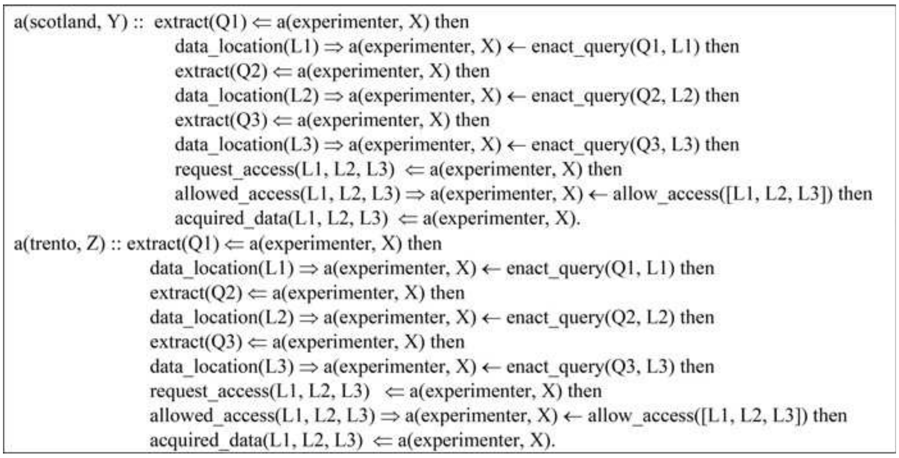
- Hospital admission data – including the ICD10 diagnostic codes and OPCS4 procedural codes.
- Mortality records – including ICD-10 coded cause of death.
- Community prescription/dispensing data – including the dosage and frequency of medications prescribed.

In addition, these data must be available for a sufficiently long period in order to determine outcomes occurring during at least one year follow-up following drug administration. To maximise the number of cases of intracranial haemorrhage in whom treatment with non-vitamin K oral anticoagulants or warfarin post-discharge was commenced, there is no restriction on the dates to be included. Identification of intracranial cases should commence



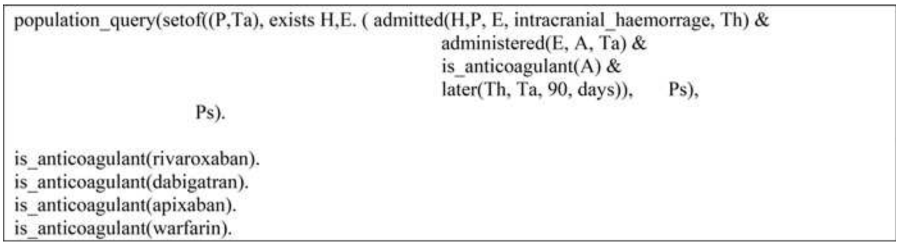
```
a(experimenter, X) ::   extract(Q1) ⇒ a(scotland, Y) ←  population_query(Q1, Ps) then
                          extract(Q1) ⇒ a(trento, Z) then
                          data_location(L1s) ⇐ a(scotland, Y) then
                          data_location(L1t) ⇐ a(trento, Z) then
                          extract(Q2) ⇒ a(scotland, Y) ←  context_query(Ps, Q2, Cs) then
                          extract(Q2) ⇒ a(trento, Z) then
                          data_location(L2s) ⇐ a(scotland, Y) then
                          data_location(L2t) ⇐ a(trento, Z) then
                          extract(Q3) ⇒ a(scotland, Y) ←  mortality_query(Ps, Q3, Ms) then
                          extract(Q3) ⇒ a(trento, Z) then
                          data_location(L3s) ⇐ a(scotland, Y) then
                          data_location(L3t) ⇐ a(trento, Z) then
                          request_access(L1s, L2s, L3s)  ⇒ a(scotland, Y) then
                          request_access(L1t, L2t, L3t)  ⇒ a(trento, Z) then
                          allowed_access(L1s, L2s, L3s) ⇐ a(scotland, Y) then
                          allowed_access(L1t, L2t, L3t) ⇐ a(trento, Y) then
                          acquired_data(L1s, L2s, L3s)  ⇒ a(scotland, Y) ) ← acquire([L1s,L2s,L3s]) then
                          acquired_data (L1t, L2t, L3t)  ⇒ a(trento, Z) ) ← acquire([L1t,L2t,L3t]).
```

Fig. 7 *Experimenter protocol in the case study*



```
a(scotland, Y) ::  extract(Q1) ⇐ a(experimenter, X) then
                   data_location(L1) ⇒ a(experimenter, X) ← enact_query(Q1, L1) then
                   extract(Q2) ⇐ a(experimenter, X) then
                   data_location(L2) ⇒ a(experimenter, X) ← enact_query(Q2, L2) then
                   extract(Q3) ⇐ a(experimenter, X) then
                   data_location(L3) ⇒ a(experimenter, X) ← enact_query(Q3, L3) then
                   request_access(L1, L2, L3)  ⇐ a(experimenter, X) then
                   allowed_access(L1, L2, L3) ⇒ a(experimenter, X) ← allow_access([L1, L2, L3]) then
                   acquired_data(L1, L2, L3)  ⇐ a(experimenter, X).
a(trento, Z) :: extract(Q1) ⇐ a(experimenter, X) then
                data_location(L1) ⇒ a(experimenter, X) ← enact_query(Q1, L1) then
                extract(Q2) ⇐ a(experimenter, X) then
                data_location(L2) ⇒ a(experimenter, X) ← enact_query(Q2, L2) then
                extract(Q3) ⇐ a(experimenter, X) then
                data_location(L3) ⇒ a(experimenter, X) ← enact_query(Q3, L3) then
                request_access(L1, L2, L3)   ⇐ a(experimenter, X) then
                allowed_access(L1, L2, L3) ⇒ a(experimenter, X) ← allow_access([L1, L2, L3]) then
                acquired_data(L1, L2, L3)  ⇐ a(experimenter, X).
```

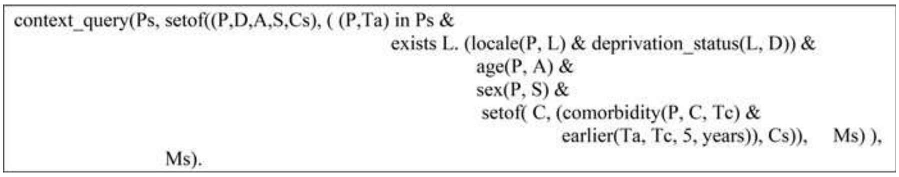
Fig. 8 *Protocol for the Scotland/Trento controllers in the case study*



```
population_query(setof((P,Ta), exists H,E. ( admitted(H,P, E, intracranial_haemorrage, Th) &
                                              administered(E, A, Ta) &
                                              is_anticoagulant(A) &
                                              later(Th, Ta, 90, days)),       Ps),
            Ps).

is_anticoagulant(rivaroxaban).
is_anticoagulant(dabigatran).
is_anticoagulant(apixaban).
is_anticoagulant(warfarin).
```

Fig. 9 *The population query in the case study*



```
context_query(Ps, setof((P,D,A,S,Cs), ( (P,Ta) in Ps &
                                        exists L. (locale(P, L) & deprivation_status(L, D)) &
                                              age(P, A) &
                                              sex(P, S) &
                                              setof( C, (comorbidity(P, C, Tc) &
                                                         earlier(Ta, Tc, 5, years)), Cs)),     Ms) ),
          Ms).
```

Fig. 10 *The context query in the case study*

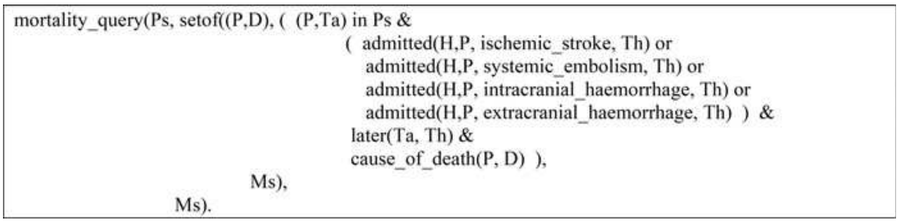
from the date at which non-vitamin K oral anticoagulants were first introduced.

Given the experimental setting above, we divide the data sharing contract for this experiment into three protocols: for the experimenter (interacting with Scotland and Trento jurisdictions), for the Scottish jurisdiction and for the Trento jurisdiction.

Experimenter protocol: This is provided in Fig. 7 and it gives the same three query requests to Scotland and to Trento (a query to determine the population for the study; a query to provide contextual information on each individual in the population; and a query to determine the subset of the population that experienced subsequent mortality). The locations of data produced by each query (not yet the data themselves) are relayed to the experimenter by Scotland/Trento. Then the experimenter requests access to the Scotland and Trento data sets and acquires the data from the appropriate locations once access is granted.

Scotland/Trento protocols: The same protocol is followed by each of these two controllers, as shown in Fig. 8. Each of the three queries supplied by the experimenter (Q1, Q2 and Q3) is enacted and the data stored at appropriate locations (L1, L2 and L3). Then (once requested by the experimenter) access is allowed to the data at L1, L2 and L3. A final message from the experimenter flags that the data has indeed been acquired to the experimenter's satisfaction.

Population query: This query constructs (in the variable Ps) the set of data fields (P, Ta) such that P has been admitted to some hospital, H, on some event, E, involving an intracranial haemorrhage at a



```
mortality_query(Ps, setof((P,D), (  (P,Ta) in Ps &
                                    (  admitted(H,P, ischemic_stroke, Th) or
                                       admitted(H,P, systemic_embolism, Th) or
                                       admitted(H,P, intracranial_haemorrhage, Th) or
                                       admitted(H,P, extracranial_haemorrhage, Th)  )  &
                                    later(Ta, Th) &
                                    cause_of_death(P, D)  ),
                       Ms),
         Ms).
```

Fig. 11 *The mortality query in the case study*

time Th, with an anticoagulant, A, being administered at time Ta which is no more than 90 days after Th (Fig. 9).

Context query: This query constructs the set of data fields (P, D, A, S, Cs) such that the field (P, Ta) appears in the data set, Ps, extracted by our earlier population query and we can find: the deprivation status, D, of the locale, L, in which P resides; the age, A, of P; the sex, S, of P and the set, Cs, of comorbidities, C, occurring for P at a time, Tc, within 5 years previous to Ta (Fig. 10).

Mortality query: This query constructs the set of data fields (P, D) such that the field (P, Ta) appears in the data set, Ps, extracted by our earlier population query and we can find the cause of death, D, for P if P was admitted to some hospital, H, with ischemic stroke, systemic embolism, intracranial haemorrhage or extracranial haemorrhage at some time, Th, later than the patient's time of admission, Ta, in our earlier population query (Fig. 11).

8 Conclusions

In this paper, we presented a logical architecture and a formal approach to specifying experiments within and across healthcare data safe havens, which respect the governance rules of local data jurisdictions. In particular, the safe haven architecture proposed allows different data sets to be derived, in stages, from core NHS data, while establishing clear paths of data management and associated responsibility. As far as experiment specification is concerned, we argue for a formal contract for data sharing, which acts as an overall plan for the experiment, clarifying the roles and tasks of different parties. Such a formal experiment specification provides transparency for review by governance bodies and it allows the automation of the data sharing process, which is currently fully manual. It also enables cross-jurisdiction experiments, in which data controllers and third-party data integrators work together, while maintaining clearly separated responsibilities that are consistent with governance principles.

The specification methods described in this paper are not new – they are familiar from earlier work by ourselves and others on the use of formal protocol languages for knowledge sharing. There remains, however, much more development work in order to derive full benefit from the style of formal representation we have described. We have shown that we can describe the data sharing component of individual experiments and that this can be used to provide fine tuning of data sharing (crucially, in the context of the formally expressed experiment design) between appropriately structured jurisdictions. Only a portion of the experiment is currently automated – currently, we only use full automation for the data queries; not for the execution of the protocol. Technically, we could fully automate all our experiment protocols (LCC is an executable specification language) but to do so requires further work on the infrastructures necessary to build appropriate auditing, security and trust into the system. Working with the jurisdictions concerned, we are developing these infrastructures in a manner that takes into consideration both existing, local cultures of data curation and the emerging international culture of responsible data sharing in healthcare.

9 Acknowledgements

This work was partially supported by the EU FP7 SmartSociety project under grant 600854 and the EPSRC SociaM project under grant EP/J017728/1.

10 References

[1] Taylor M.: 'Information governance as a force for good? Lessons to be learnt from care.data', *ScriptEd*, 2014, **11**, (1), pp. 1–8

[2] The Scottish Government: 'A Charter for Safe Havens in Scotland: Handling Unconsented Data From National Health Service Patient Records to Support Research and Statistics', 2015, available at: http://www.gov.scot/Resource/0048/00489000.pdf

[3] Scottish Government, Health Informatics Research Advisory Group: 'A Health and Biomedical Informatics Research Strategy for Scotland: Enhancing Research Capability in Health Informatics for Patient and Public Benefit 2015–2020', 2015, available at: http://www.gov.scot/Resource/0047/00475145.pdf

[4] Cusack C.M., Hripcsak G., Bloomrosen M., ET AL.: 'The future state of clinical data capture and documentation: a report from AMIA's 2011 Policy Meeting', *J. Am. Med. Inf. Assoc.*, 2013, **20**, (1), pp. 134–140

[5] The Academy of Medical Sciences: 'Data in safe havens'. Workshop report 2014, available at: http://www.acmedsci.ac.uk/policy/policy-projects/data-in-safe-havens/

[6] Jones K.H., Ford D.V., Jones C., ET AL.: 'A case study of the secure anonymous information linkage (SAIL) gateway: a privacy-protecting remote access system for health-related research and evaluation', *J. Biomed. Inf.*, 2014, **50**, pp. 196–204

[7] Knoppers B.M.: 'Framework for responsible sharing of genomic and health-related data', *HUGO J.*, 2014, **8**, (3), pp. 1–6

[8] Safran C., Bloomrosen M., Hammond W.E., ET AL.: 'Toward a national framework for the secondary use of health data: an American Medical Informatics Association White Paper', *J. Am. Med. Inf. Assoc.*, 2007, **14**, (1), pp. 1–9

[9] Lea N.C., Nicholls J., Dobbs C., ET AL.: 'Data safe havens and trust: toward a common understanding of trusted research platforms for governing secure and ethical health research', *JMIR Med. Inf.*, 2016, **4**, (2), pp. e22

[10] Ford D.V., Jones K.H., Verplancke J.P., ET AL.: 'The SAIL Databank: building a national architecture for e-health research and evaluation', *BMC Health Serv. Res.*, 2009, **9**, (1), pp. 1–12

[11] Burton P.R., Murtagh M.J., Boyd A., ET AL.: 'Data safe havens in health research and healthcare', *Bioinformatics*, 2015, **31**, (20), pp. 3241–3248

[12] Laurie G., Ainsworth J., Cunningham J., ET AL.: 'On moving targets and magic bullets: Can the UK lead the way with responsible data linkage for health research?', *Int. J. Med. Inf.*, 2015, **84**, (11), pp. 933–940

[13] Pavis S., Morris A.D.: 'Unleashing the power of administrative health data: the Scottish model', *Public Health Res. Pract.*, 2015, **25**, (4), pp. 1–6

[14] Atherton I.M., Lynch E., Williams A.J., ET AL.: 'Barriers and solutions to linking and using health and social care data in Scotland', *Br. J. Soc. Work*, 2015, **45**, (5), pp. 1–9

[15] Witham M.D., Frost H., McMurdo M., ET AL.: 'Construction of a linked health and social care database resource – lessons on process, content and culture', *Inf. Health Soc. Care*, 2015, **40**, (3), pp. 229–239

[16] 'Research Data Management Platform', http://medicine.dundee.ac.uk/research-data-management-platform-rdmp, accessed 27 May 2016

[17] Robertson D.: 'A lightweight coordination calculus for agent systems', in Leite J., Omicini A., Torroni P., ET AL. (Eds.): 'Declarative agent languages and technologies II' (Springer, Heidelberg, 2005), (*LNCS (LNAI)*, **3476**), pp. 183–197

[18] Robertson D.: 'Lightweight coordination calculus for agent systems: retrospective and prospective', in Sakama C., Sardina S., Vasconcelos W., ET AL. (Eds.): 'Declarative agent languages and technologies IX' (Springer, Berlin, 2012), (*LNCS (LNAI)*, **7169**), pp. 84–89