Theory and implementation of the Privacy Preferences Tool research demonstrator.

To accompany the report on an ICO-funded research project at Teesside University.
All views expressed are those of the authors.

Amended extracts from a paper presented to the TASE 2019 conference

Jim Longstaff Mengda He
School of Computing and Digital Technologies
Teesside University
Middlesbrough, Engalnd
j.j.longstaff@tees.ac.uk

We show how complex privacy requirements can be represented and processed by an extended model of Attribute Based Access Control (ABAC), working with a simple database applications pattern. During application model development, most likely based on UML (e.g. Use Case, Class Diagrams), the analyst and possibly the end user specifies ABAC permissions, and then verifies their effect by running queries on the target data. The ABAC model supports positive and negative permissions, "break glass" overrides of negative permissions, and message/alert generation. The permissions combining algorithms are based on relational database optimisation, and permissions processing is implemented by query modification, producing structurally-optimised queries in an SQL-like language; the queries can then be processed by many database and big data systems. The method and models have been implemented in a prototype Privacy Preferences Tool in collaboration with a large medical records development, and we discuss experiences with focus group evaluations of this tool. **(***Abstract***)**

Keywords—Attribute Based Access Control, Database, Medical Records

I. INTRODUCTION

In the development of web – database applications, the design and implementation of authorisation policies (the granting of access to permitted resources) has often been left to the later stages, sometimes being implemented by procedural coding. To-date, the most widely-used model of authorization has been Role Based Access Control, or RBAC [1], which is used in operating systems, databases, access control systems for specialized applications, and development environments. Attribute Based Access Control, or ABAC, is generally seen as the way forward for authorization model research, see e.g. [2]. ABAC has been considerably developed over recent years, with several NIST publications, e.g. [3], XACML standard [4], and product offerings, e.g. Axiomatics [5]. Recent research work includes identifying attributes from applications [6], and implementation schemes [7]. The central idea of ABAC is that access can be determined based on various attribute values presented by a subject. In this paper we describe our extended ABAC model which we call the Tees Confidentiality Model, version 2, or TCM2 [8, 9], and its application to healthcare and social services scenarios.

Health and Social Care information systems are undergoing substantial development at the present time, and bring very demanding challenges for authorisation. Five large health record exemplar projects are being part-funded by the UK government [10], with other projects proceeding with separate funding arrangements. These projects variously provide access for Health Care Professionals (HCPs) and Social Workers to records held in GP and social care systems. A further aim is to support medical research by granting established researchers access to Trusted Research Environments, containing data uploaded from operational systems. The issue of governance has received huge attention with the arrival of GDPR [11], the new Data Protection Act [12], and NHS and GMC systems and Guidelines [13]. There is a move towards recognising that data should be owned by the patient/citizen, who should be able to control who has access to it [14].

The GMC guidelines suggest a warning to providing full patient control over sharing their data for direct care. Paragraph 31 of [13] deals with a worst-case scenario where a patient totally objects to the sharing of information deemed necessary for their safe care – the advice given to the HCP is to refuse to extend their treatment or refer them, unless their data can be shared. The GMC advises that through argument and explanation both parties should attempt to arrive at a compromise to allow sharing. We suggest in this paper an approach to support arriving at compromise, through facilitating alerts and overrides for key HCPs.

We begin by commenting on authorisation models, their presentation to end users, and implementation. We then describe motivating scenarios and situations. Following this we describe a simple database pattern which can be part of the implementation of complex health and social care authorisation scenarios. We then introduce our TCM2 ABAC model by presenting a permissions representation for two complex healthcare scenarios, before summarising how they can be implemented using database and other technologies. Finally we give an overview of our TCM2 implementation which supports the capture and user-verification of the ABAC permissions, and its focus group evaluation.

II. METHODS, MODELS AND REQUIREMENTS ELICITATION

UML (including Use Case, Class Diagrams) is a widely-used notation for systems development. There are several widely-used methods which can take advantage of UML, including agile (e.g. Scrum, XP) + others [15]. Elements of UML, particularly Use Case, Classes are very suitable for presentation to analysts and end users during the construction of ABAC models. They can be presented with user-friendly descriptions appropriate to end users. User stories and Use Case models indicate typical users and functionality. Class Diagrams or Entity Relationship Diagrams indicate ABAC Protected Objects.

The TCM2 model presented in this paper was originally developed from the RBAC standard, and we refer to

permissions rather than rules. However our terminology corresponds to the XACML standard. A feature of our ABAC model, which does not appear in the academic literature as far as we know, is the support and processing of "break glass" overrides. The HCP or analyst can optionally define progressively stricter levels of access to sensitive data, with selectively defined overrides for restricted or emergency access.

For implementation, one approach is to modify user transactions with restrictions generated from the ABAC model. Queries are most suitably expressed in a high-level, usually SQL-like language supported by the underlying storage systems, which might be relational database, NoSQL, Hadoop or other big data systems. Many database and big data systems (e.g. NoSQL and Hadoop-based systems) support a variant of SQL; we use SQL to present the approach to implementation described in this paper.

III. Scenario 1

A. Summary

This scenario was suggested by a Consultant Transplant Surgeon during the design and development of an Electronic Health Records (EHR) system.
The scenario concerns a fictitious patient, Alice, and her GP, Fred. Alice is 50; the major events in Alice's medical history are:

1. She had a pregnancy termination when she was 16
2. Was diagnosed diabetic at 25
3. End Stage renal failure when she was 45
4. Renal transplant at 48
5. Acutely psychotic at 49
6. Crush fracture of T12 aged 50

Let us now suppose, not unreasonably, that Alice expresses the desire to place the following privacy directives on the availability of her EHR data about two of these conditions:

a) My GP (Fred) can see all my data
b) Nobody must know about my termination except my GP, any Gynaecological Consultant, and the Consultant Renal Transplant Surgeon (Bill) who operated on me.
c) My GP, Consultant Renal Transplant Surgeon (Bill) and Consultant Orthopaedic Surgeon (Bob) can see my psychosis data, but no-one else.

To show the power of our TCM2 model, consider the following contrived requirement (but still one which an EHR authorization system should be capable of handling):

d) I do not wish the members of the hospital team who carried out my termination operation to be ever able to see my psychosis data, except if they are viewing in a psychiatric role. (This directive to be in force throughout the careers of those professionals concerned).

We must add to these directives that they must be capable of being overridden in **carefully controlled and audited ways**. **The concept of overriding** can be explained in simple but accurate terms, e.g. "facilitating agreed access to sensitive data". An example of overriding follows.

B. Transaction

Consider the following transaction which requires access to sensitive restricted data, and illustrates the desirability for an override capability. The clinician user is a transplant surgeon, querying Alice's medical data. The termination data is unavailable to the surgeon and all users for a normal query except those listed in directives (a) and (b) above.

Alice has now been scheduled for a transplant (event 4). She has also now been persuaded by her GP to make her termination data available to a limited set of medical roles, the Transplant Surgeon role falling within them. However she generally wishes this to be via an extra step (an "override") to emphasise the sensitive nature of this data. Tests lead the surgeon to suspect a previous pregnancy (if the tissue type of the father is similar to the graft a very serious rejection may ensue). Alice at first refuses to confirm a previous pregnancy to the surgeon. A message is displayed, for any user in the Transplant Surgeon role, saying that he should use the override facility. The surgeon persuades Alice to let him override, and this displays the data that he needs. This allows for a specific form of treatment to be planned.

Note that this data will not be displayed upon override to roles not included with the original agreement with Alice. The permissions which implement these restrictions are described in section VI.A.

IV. Scenario 2

Omitted.

V. Healthcare Authorisation Pattern

We now describe part of a conceptual model covering health and social care data, and perhaps other data. Tables of this data model, and their contents, are the Protected Objects (POs) which are part of our TCM2 model. The data model tables correspond to the HealthGateway MIG Detailed Care Record [16], which is used in several major projects in England. Our model has other tables containing data about patients, demographics, HCPs, which are peripheral to our current theme, and not described here. In a full-scale system, the actual data might be stored in several systems, with their data models mapped onto our conceptual model. Note that the tables in our model could be used for deriving other representations, such as NoSQL Document databases. JSON representations, and others.

The master-detail pattern for health data, which we use many times, is shown in Fig. 1.

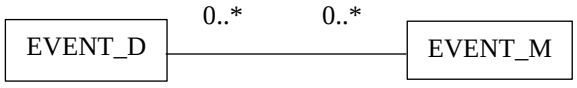

Fig. 1. Specialisations of EVENT_M (which is not shown in the diagram) are: PROBLEM, DIAGNOSIS, MEDICATION, INVESIGATION, RISKS_WARNINGS, PROCEDURES, BLOOD_PRESSURE_MEASUREMENTS.

EVENT_D contains records of every encounter involving a patient and an HCP, whether it would be notes from a telephone conversation, an admission, a referral, etc. EVENT_M and its specializations serve two purposes. Firstly to contain data, e.g. a PROBLEM record might be for Diabetes, and have attributes such as Date_of_Diagnosis, Clinician_of_Care, etc; this can be associated with all EVENT_D instances which would contain data about e.g. individual appointments, etc, for Diabetes. An EVENT_D instance can be associated with more than one EVENT_M instance. For example, an EVENT_D record of the issuing of a prescription for Diabetes would be associated with both MEDICATION and PROBLEM instances.

The full set of attributes of these tables are based on documents designed by the Professional Records Standards Body [17].

VI. Scenario Permissions Representation

A. Permissions for Scenario 1 (Health and Social Care)

As an introduction to our TCM2 model we now give examples of permissions. Abbreviations are used to facilitate concise presentation, however they correspond to the data structures and values used for implementation. We call our permissions T Permissions, or TPs, to distinguish them from RBAC permissions, and other ABAC rule formulations. TPs consist of sets of classifier values; an example of a classifier value is <UserRole, 'Psychiatrist'>. Classifier values can represent information other than attribute values, as is explained in section VII.A below. They are structured into hierarchies, which enable further TPs to be derived with more specialized classifier values, e.g. a TP with <UserRole, 'Psychiatrist'> could produce a derived TP with <UserRole, 'SeniorPsychiatrist'>.

Firstly, the EHR data for any patient is normally made available to:

a) Healthcare professionals (HCPs) such as clinicians, doctors, and administrators who have a Legitimate Relationship (LR) with the patient. This means that the patient is registered with or has been referred to them.
b) Additionally, all HCPs can exercise a Level 1 TP Override facility (see section VII.C), to exceptionally access restricted data, when they have reason to do so. Naturally, all access and overrides will be logged, and subject to audit.

The following permissions authorize this access.

TP1 Permit_TP (N): {<Database, 'EHR'>, <UserRole, 'HCP'>, <LR, 'yes'>, <Op_id, 'R_A'>}	TP2 Permit_TP(L1_Ovr): {<Database, 'EHR'>, <UserRole, 'HCP'>, <LR, 'yes'>, <Op_id, 'R_A'>}

TP1 represents the granting of read and append access to EHR database data for a clinician-user in the role of Healthcare Professional (HCP), under normal (N) processing where no override has been used. A Legitimate Relationship (LR) must exist. TP2 permits access for any HCP to any EHR data if the user has exercised a Level 1 Override.

The TPs which implement the privacy directives given in section III.A, in the order in which they are expressed, are

TP3 Deny_TP(L1): {<Database, 'EHR'>, <UserRole, 'HCP'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Termination'>}	TP4 Permit_TP (N): {<Database, 'EHR'>, <User_id, 'Fred'>, <UserRole, 'GP'>, <Op_id, 'R_A'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Termination'>}
TP5 Permit_TP (N): {<Database, 'EHR'>, <UserRole, 'GC'>, <Op_id, 'R_A'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Termination'>}	TP6 Permit_TP (N): {<Database, 'EHR'>, <User_id, <'Bill' >, <Op_id, 'R_A'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Termination'>}
TP7 Deny_TP(L1): {<Database, 'EHR'>, <UserRole, 'HCP'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Psychosis'>}	TP8 Permit_TP (N): {<Database, 'EHR'>, <User_id, 'Fred'>, <UserRole, 'GP'>, <Op_id, 'R_A'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Psychosis'>}
TP9 Permit_TP (N): {<Database, 'EHR'>, <User_id, < 'Bill', 'Bob'> <Op_id, 'R_A'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Psychosis'>}	TP10 Permit_TP(N): {<Database, 'EHR'>, <User_Coll_id, 'TermTeam'>, <UserRole, 'Psychiatrist'>, <Op_id, 'R_A'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Termination'>

Deny TPs are negative permissions which prevent access. These can be very detailed, for specific users and data, TP3 denies (at Level 1) any kind of access to Alice's termination data to HCPs. However if authorized by another TP, e.g. TP12 below, a transplant surgeon could use TP Override at Level 1 to cancel the effect of the deny permission TP3.

The permissions which generate the message advising override to the transplant surgeon, and which provide the L1 Override for the transaction scenario in section III.B, are:

TP11 Deny_TP (L1):	TP12
{<Database, 'EHR'>, <UserRole, 'TransplantSurgeon'>, <LR, 'yes'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Termination'>}	Permit_TP(L1_Ovr): {<Database, 'EHR'>, <UserRole, 'TransplantSurgeon'>, <LR, 'yes'>, <Op_id, 'R_A'>, <PO_Subj_id, 'Alice' >, <PO_Problem, 'Termination'>}

The message associated with the TP11 permission could only be sent to a Transplant Surgeon who has an established LR with the patient. Also the transplant surgeon could only access the data upon Level 1 Override if he possesses an LR.

B. *Permissions for Scenario 2 (Location data)*

Omitted.

VII. TCM2 OVEERVIEW

In this section we summarize further aspects of the TCM2 authorization model. More detailed expositions can be found in [9, 10].

A. *Classifiers*

The TCM2 model is based on the RBAC concepts of users, operations and protected objects [1]; however these concepts are now represented by classifiers, as illustrated in the previous section. The simplest form of classifier corresponds to an attribute, as used in ABAC [3]. User classifiers can take the role of parameters in parameterized RBAC; extended classifiers are defined for combinations of user, operation and protected object, and collection classifiers can be created to facilitate authorizations for collections of objects. Classifier values are structured into hierarchies, which can be represented as inverted trees, with less-specialized values placed nearer the root. Classifier values can be provided by several mechanisms (stored database values, generator programs, and external applications). A classifier ordering is determined by the security architect or analyst, to indicate importance for matching (ie deciding the authorization outcome). For example if the classifier User_id was deemed to be more important than UserRole when deciding authorization, then a permission with a User_id value match would be preferred to another permission (not containing the User_id value) which was matched by a UserRole value.

The model also includes an override operation which allows a user to acquire a more specialized classifier value (if he was specifically authorized to use this type of override); e.g. a JuniorPsychiatrist might acquire the role (ie classifier value) of ConsultantPsychiatrist in an emergency situation, if he was authorized to use an override permission.

B. *T Permissions*

A TP consists of a set of classifier values, and other properties, as illustrated in section VI. Other, derived, TPs may be produced using the classifier value hierarchies for each classifier present in a TP. For example. A TP which includes the classifier value <UserRole, 'Psychiatrist'> could produce a second, derived TP which includes the classifier value <UserRole, 'SeniorPsychiatrist'>. Ranges of classifier values can be specified in a TP.

A TP will match (ie qualify to authorize a transaction) if all of its classifier values (describing the user, operation and object) are present in the transaction (described in terms of classifiers or attributes). Additionally, a TP will match if one of its derived TPs matches. This is described in section VIII below.

A TCM2 implementation builds the permission, checks that it does not repeat or conflict with existing permissions, and then generates an explanation of the permission for validation.

C. *Deny Levels*

Deny TPs are specified at increasing levels of power, called Deny Levels. A deny level contains deny permissions specified at lower deny levels. Therefore data could be denied to users who might be able to access it by Level 1 Override (if so authorized), whereas more sensitive data might be only available to more senior users who were authorized to override at Level 2 or higher. A single Level 1 Override is defined for the scenarios presented in section VI.A; the assumption is that a Level 2 Override will be specified for a healthcare application 'super user', which would provide access to all data.

D. *TP Sets*

TPs can be defined as having membership in separate, independent T Permission Sets, or TP Sets. TP Sets can be used separately to determine authorization, or combined.

Representation of different levels of processing can be accomplished with TP Sets, e.g. government regulations (TPS1), consumer-specified directives (TPS2), and directives specified by proxies for consumers (TPS3). Therefore TPS1 authorizations can be preferred to TPS2 authorizations, if this is what the security architect requires.

In the examples in section VI.A, access to health records is provided by one TP Set.

VIII. PERMISSIONS PROCESSING: MATCHED SEQUENCES

A. *Overview*

A full formal specification of TCM2 [18] has been developed using the B Method, and extracts from this specification are included in this section. Permissions processing depends on two main principles: TP Match, and Nearest Match, which we describe below.

B. *TP Match*

Firstly, a T Permission will match (ie qualify to authorize a transaction) if all its classifier values are contained in the transaction. Additionally, a TP will match if one of its derived TPs matches.

This can be expressed formally using B by the following definition:

$$\text{TPPermitAccess}(tp, acvals) \triangleq \text{bool}(\text{dom}(acvals \cap ad[tp]) = \text{dom}(tp))$$

where acvals is the transaction active classifier values (classifier values specifying the transaction), and tp is a T Permission which permits access. The set ad[tp] contains the

original ancestor classifier values as well as the set of all descendant classifier values. That is, access is granted if for every classifier in the domain of tp there exists at least one classifier value in common between the active classifier values acvals and the classifier values of tp and all their descendants. Similarly for TPDenyAccess.

C. Nearest Match TP

The second principle concerns determining which of two TPs (taken from a set of Matched TPs) is the stronger or nearer match to a transaction. This Nearest Match TP would then have a higher priority in determining the authorization outcome.

A TP is a set of classifier values. There is an ordering cfiersq on the classifiers that is set by the security architect and is a mapping of the set of integers 1,2,3,4....to the set of classifiers:

cfiersq ∈iseq (cfiers)

Given the ordering on the classifiers then for any set of classifier values cvs there exists a classifier for that set which is the most important classifier i.e. the lowest in the ordering:

CFIERL (cvs) = cfiersq (min (cfiersq- 1[dom (cvs)])

There also exists an associated ordering number for that classifier and an associated value:

NCFIERL (cvs) = min (cfiersq- 1[dom (cvs)])
VCFIERL (cvs) = cvs [CFIERL (cvs)]

Therefore, given a set of matched TPs the (set of) nearest match(es) is given by:

NearestMatch(tps) ≙{nmtp | nmtp ∈ tps ⋀
∄tp. (tp ∈ tps ⋀
(
NCFIERL(nmtp - tp ⋂nmtp) >
NCFIERL(tp - tp ⋂ nmtp)
⋁
VCFIERL(nmtp - tp ⋂ nmtp) ↦
VCFIERL(tp - tp ⋂ nmtp) ∈ ad)
)
}

where ad is the ancestor/descendant relationship.

The classifier ordering sequence cfiersq can be varied for different applications and data models, but given the data model and application, it is:

EventM_id, EventD_id, Userid, Roleid,

Regarding implementation, the sequence of Nearest Match TP can be produced by sorting tps firstly by their cfiersq classifier value, and within this by position in the classifier value hierarchy. The result of the transaction can be obtained by successively retrieving and refining the sets of objects produced by permissions in the Nearest Match TP sequence. The amount of work required for this can be reduced by optimising the sequence. Starting at the final TP, preceding TPs are analysed to determine if the final TP can be derived, ignoring the permit/deny value. If a TP later in the sequence can be derived from an earlier TP, the earlier TP can be deleted.

D. Normal TP Processing Example

For the transaction given in section III.B, the Initially-Matched set of TPs, and the Nearest Match TP sequence (following removal of all Override TPs) are:

Initially-Matched TPs

TP1	Permit_TP (N)	
TP2	Permit_TP (L1_Ovr)	
TP3	Deny_TP (L1)	
TP7	Deny_TP (L1)	
TP11	Deny_TP (L1)	
TP12	Permit_TP (L1_ovr)	

Nearest-Matched TPs (no overrides)

TP1	Permit_TP (N)	1
TP3	Deny_TP (L1)	2
TP7	Deny_TP (L1)	3
TP11	Deny_TP (L1)	4

The match strength is indicated in ascending order, starting with the weakest (i.e. 1).

Processing the Nearest-Match TP sequence authorizes the retrieval of all data except the Termination and Psychosis data. The strongest match, TP11, will exactly match the transaction, and will deny access to the Termination data for Transplant Surgeons; it will generate a message, though, just for TransplantSurgeons. TP7 and TP3 deny access to the Psychiatric and Termination data for all HCPs. TP1 permits access to all data, except the Psychiatric and Termination data for all HCPs (which is denied by the later permissions in the sequence.

E. Override TP Processing Example

Consider the transaction from section 3.B. The same initially-matched TPs are returned. However on applying Level 1 Override (L1_ovr) the sequence of Nearest-Matched TPs shown below is produced. These TPs authorize access to the termination and unrestricted data, while still denying access to the psychosis data.

Nearest Match TPs (L1_ovr)

TP1	Permit_TP (N)	1
TP2	Permit_TP (L1_Ovr)	2
TP3	Deny_TP (L1)	3
TP7	Deny_TP (L1)	4
TP12	Permit_TP (L1_Ovr)	5

We further note that, ignoring permit/deny values, TP12 can be derived from TP3, and therefore TP3 can be removed from the sequence.

IX. Permissions Processing: TCM2 Implementation

A. Overview

Consider a large implementation of the records model outlined in section V. Also that each permission corresponds to a query, specifying qualifying or denied records. The Nearest Match sequences can be expressed in SQL-like subqueries. In such a system, a query to find records permitted by the Nearest Match Sequence could first retrieve the relatively small set of data for the patient, maybe by a map-reduce approach. Then this data could be filtered by query

corresponding to an optimized SQL-like subquery to retrieve permitted records.

We firstly illustrate with the simple Protected Object example outlined in sections III.B, and then proceed to consider the wider data model scenario. The user's transaction is expressed in SQL, which is then augmented by adding WHERE clause elements which implement the Nearest Match sequence. The query is then structurally optimized using relational database techniques to produce the query to be executed by the underlying storage system.

B. Problem Transaction Query

The transaction consists of John the Transplant Surgeon, querying Alice's EHR to discover data about previous pregnancies. This data is denied to him by permission TP7 from section VI.A, under normal processing.

```
SELECT * FROM  PROBLEM
WHERE
Patient_id = 2220 /*Alice*/
```

The previous query has now been augmented with code derived from the Nearest Match TP sequence in section VIII.D. When regarded as a database query, this sequence describes all the user – operation – protected objects permitted by the transaction. When the augmented SQL is combined with the transaction SQL, there is very often huge scope for structural query optimization, as is illustrated in the example below:

```
SELECT * FROM  PROBLEM
WHERE
Patient_id = 2220
AND -- TP1 PO part
PO_id in
     (SELECT PO_id FROM EVENT_M)
AND -- TP3 PO part
     NOT PO_id IN
          (SELECT PO_id FROM PROBLEM WHERE
           PO_TYPE = 'Termination')
 AND  -- TP7  PO part
     NOT PO_id IN
          (SELECT PO_id FROM PROBLEM WHERE
           PO_TYPE = 'Psychosis')
AND -- TP11 PO part
      NOT PO_id IN
           (SELECT PO_id FROM PROBLEM WHERE
           PO_TYPE = 'Termination')
```

Expressed in SQL, rather than the internal data structures of the TCM2 system, this query is optimized to:

```
SELECT * FROM PROBLEM
WHERE
Patient_id = 2220
AND PO_id NOT IN
          (SELECT PO_id FROM PROBLEM WHERE
             PO_TYPE = 'Termination')
AND PO_id NOT IN
          (SELECT PO_id FROM PROBLEM WHERE
             PO_TYPE = 'Psychosis')
```

Similarly, the Level 1 Override permissions example would be optimized to:

```
SELECT * FROM PROBLEM
WHERE
Patient_id = 2220
AND PO_id NOT IN
          (SELECT PO_id FROM PROBLEM WHERE
             PO_TYPE = 'Psychosis')
```

C. Event Query

Now we consider the query which returns all detailed events for a patient:

```
SELECT * FROM EVENT_D
WHERE
Patient_id = 2220 /*Alice*/
```

Generally, authorization for EVENT_D records are determined by

- any EVENT_Ms they are associated with (similar to PROBLEM in the previous section), and
- other TPs they are directly matched to.

Therefore if this query is executed by the Transplant Surgeon without any overrides, the same NM list would be produced. The final SQL query is:

```
SELECT EVENT_D.* FROM EVENT_D*PROBLEM
WHERE
Patient_id = 2220  AND
EVENT_D.PROBLEM_PO_ID = PROBLEM.PO_ID
AND PROBLEM_PO_id NOT IN
          (SELECT PO_id FROM PROBLEM WHERE
             PO_TYPE = 'Termination')
AND PROBLEM.PO_id NOT IN
          (SELECT PO_id FROM PROBLEM WHERE
             PO_TYPE = 'Psychosis')
```

Now suppose that the patient wished to make a particular psychiatric EVENT_D record, with PO_id = 123, available to the transplant Surgeon. This can be accomplished by creating a new permission to display this EVENT_D to a Transplant Surgeon. Because this permission is defined for an EVENT_D record, it will appear later in the Nearest Match sequence than the EVENT_M deny permission for psychiatric data, and will actually be the last in the sequence. The resulting SQL query, which retrieves it, is:

```
SELECT EVENT_D.* FROM EVENT_D*PROBLEM
WHERE Patient_id = 2220  AND
ENTRY.PROBLEM_PO_ID = PROBLEM.PO_ID
AND
(
PROBLEM.PO_id NOT IN
          (SELECT PO_id FROM PROBLEM WHERE
             PO_TYPE = 'Termination') AND
PROBLEM.PO_id NOT IN
          (SELECT PO_id FROM PROBLEM WHERE
              PO_TYPE = 'Psychosis')
)
OR
PO_id = 123
```

X. Implementation and Evaluation

We have implemented all of the authorisation functionality described above in a Privacy Preferences Tool research demonstrator. An operational implementation would take place as part of a Health Information Exchange system, which would provide the other security functionality. Our demonstrator project is being funded by the UK Information Commisioners's Office [19]; additionally we are collaborating with the Great North Care Record Project [20]. The main characteristics of the Privacy Preferences Tool are:

- Privacy Directives can be specified either during a consultation with an information specialist/HCP who receives and enters the directives, or by patients themselves if they have been provided with an online account. The same dashboard is used by a specialist and by the patient.
- The interaction is modelled on an imaginary conversation between patient and doctor. The patient is guided to identify data (the entities described in section V), and then to identify users (by team, role, or user_id) which is the subject of the permission.
- The patient identifies whether access is to be permitted or denied, and at what level (normal query or override – presented a "access to sensitive data").
- The patient inspects a simple natural language description of the privacy preference, and then tests its effect by running queries on their own data. If they, and also the medical specialist is happy, then the medical specialist turns the preference into a permission which is enforced by the medical records system. If an impasse occurs, a targeted message similar to the one described in scenario 1 section III.B can be investigated.

A significant comment from the focus group evaluations with members of the public was that developing and verifying privacy preferences of the nature described in Scenarios 1 and 2 is "straightforward". We have just completed our focus group sessions, and the demonstrator will be available via the internet once the final recommendations and improvements have been made. A description of the project is given in [21].

XI. Related Work

More details of TCM2 authorisation model and its formal specification can be found in [8,9,18]. These publications give a detailed comparison with ABAC and RBAC research literature. However this is the first paper on the use of the database pattern and model based on the MIG Detailed Care record, and our current Privacy Preferences Tool research demonstrator.

The capturing and implementation of Privacy Preferences is a substantial part of most health and social care information systems development. Recent standards in the UK are GDPR, GMC, NHS. One particularly interesting development is described in [22], where the patient controls which medical professionals can access their data. The conceptual starting point is inspection of medical records, whereas our starting point, as reflected in our implementation, is a conversation between the patient and the care professional.

XII. Conclusions

The idea that patients should have ownership and responsibility for their health records, and that they should grant clinicians access to their data, is very different to what people thought would happen. Yet, this is now what is happening. Further predictions are that records that will be held remotely from hospitals and other providers, owned by patients, and accessed through smartphones and tablets.

We have shown how our TCM2 model can support the authorization capabilities required by this vision of the future. TCM2 can be also used for other information systems, including citizen information systems covering health and social care, financial, work-seeking and other aspects. Our TCM2 extensions to ABAC (data model, break glass overrides, alerting) can be implemented by ABAC systems based on the XACML standard.

In addition to our current Privacy Preferences Tool, several relational database demonstrators of our TCM2 model have previously been implemented by ourselves and others, and the approaches to complex authorizations and override positively evaluated within healthcare information system projects, and research and commercial ventures.

ACKNOWLEDGMENT

The authors wish to thank Tony Howitt, Professor Mike Lockyer, Professor Michael Thick and Steve Dunne for advice and contributions. This work was supported in part by grants and contracts from the England National Programme for IT (part of the England National Health Service), and is the UK Information Commissioner's Office.

REFERENCES

[1] ANSI, "Role Based Access Control, ANSI INCITS 359-2012," ANSI, 2012

[2] R. Sandhu, "The Authorization Leap from Rights to Attributes: Maturation or Chaos?," in SACMAT'12, June 20–22, 2012, Newark, New Jersey., 2012.

[3] V. Hu, D. Ferraiolo, R. Chandramouli, R, Kuhn, Attribute Based Access Control, CSRC, NIST, 2017

[4] eXtensible Access Control Markup Language (XACML) Version 3.0 Plus Errata 01. Edited by Erik Rissanen. 12 July 2017. OASIS Standard

[5] Axiomatics ABAC software, www.axiomatics.com.

[6] Charles Morisset, Tim A. C. Willemse, Nicola Zannone, "Efficient Extended ABAC Evaluation", ACM SACMAT 2018

[7] Jason Crampton, Conrad Williams, "Attribute Expressions, Policy Tables and Attribute-Based Access Control", ACM SACMAT 2017.

[8] J. Longstaff and A. Howitt, "TCM2: Supporting dynamic authorization and overrides in Attribute Based Access Control," in Case Studies in Secure Computing: Achievements and Trends, B. Issac and N. Israr, Eds., Auerbach Publications, Taylor and Francis. ISBN 9781482207064, 2014.

[9] J. Longstaff and J. Noble, "Attribute Based Access Control for Big Data applications by Query Modification," in IEEE Second International Conference on Big Data Computing Service and Applications, Oxford, 2016.

[10] LHRE Progamme. https://understandingpatientdata.org.uk/news/local-health-and-care-record-exemplars-announced

[11] GDPR. https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en

[12] Data Protection Act. https://www.gov.uk/government/collections/data-protection-act-2018

[13] GMC Confidentiality guidance. https://www.gmc-uk.org/ethical-guidance/ethical-guidance-for-doctors/confidentiality

[14] M.Thick. http://www.imsmaxims.com/blog-and-opinion/nhs-long-term-plan-cracking-da-vinci-code/

[15] K. Beck, et al, The Agile manifesto. https://agilemanifesto.org/

[16] Healthcare Gateway, MIG Detailed Care Record. https://healthcare.g.ateway.co.uk/services/detailed-care-record/

[17] Professional Records Standards Body. https://theprsb.org/standards/healthandcarerecords/

[18] A. Howitt, "Formal Specification of the Tees ConfidentialityModel," PhD thesis, Teesside University, 2008.

[19] ICO Grant Programme. https://ico.org.uk/about-the-ico/news-and-events/news-and-blogs/2017/11/ico-grants-programme-supports-independent-research-into-four-privacy-and-data-protection-challenges/

[20] Great North Care Record. https://www.greatnorthcarerecord.org.uk/

[21] ICO. https://www.youtube.com/watch?v=Iq81aIb-7xg

[22] North West London Collaboration of ClinicalCommissioning Groups. http://www.interopen.org/wp-content/uploads/2018/05/Interoperability_CaseStudy_CIE_2018.pdf