

# The International Journal of Digital Curation

## Volume 8, Issue 2 | 2013

### Informative Provenance for Repurposed Data: A Case Study Using Clinical Research Data

Richard Bache,

Department of Informatics, Division of Health and Social Care Research,  
King's College London

Simon Miles,

Department of Informatics,  
King's College London

Bolaji Coker,

Division of Health and Social Care Research,  
King's College London

and

Biomedical Research Centre,  
Guy's and St Thomas' NHS Foundation Trust

Adel Taweel,

Department of Informatics, Division of Health and Social Care Research,  
King's College London

#### Abstract

The task repurposing of heterogeneous, distributed data for originally unintended research objectives is a non-trivial problem because the mappings required may not be precise. A particular case is clinical data collected for patient care being used for medical research. The fact that research repositories will record data differently means that assumptions must be made as how to transform of this data. Records of provenance that document how this process has taken place will enable users of the data warehouse to utilise the data appropriately and ensure that future data added from another source is transformed using comparable assumptions. For a provenance-based approach to be reusable and supportable with software tools, the provenance records must use a well-defined model of the transformation process. In this paper, we propose such a model, including a classification of the individual 'sub-functions' that make up the overall transformation. This model enables meaningful provenance data to be generated automatically. A case study is used to illustrate this approach and an initial classification of transformations that alter the information is created.





## Introduction

There are many situations where data collected and stored to perform some routine activity can be repurposed for different uses, such as research. Where this data comes from several heterogeneous sources and a researcher wishes to combine this data into a single warehouse with a consistent semantic representation, or indeed several distributed homogeneous warehouses, a transformation is required. Such a warehouse would typically be designed according to some information model appropriate for the new (research) purpose. There is no guarantee that mapping the data from the repurposed sources will be exact or accurate during the transformation process. Thus, for a researcher to make full and accurate use of this data, its provenance must be made explicit.

Such a case in point and the focus of this paper is clinical data collected from electronic healthcare record (EHR) systems. Data collected for the routine care of individual patients at various particular clinical sites (e.g. GP practices or hospitals) can be repurposed for epidemiological research or for identifying and recruiting patients for clinical trials. Yet to be usable, such data need to be transformed into a consistent, meaningful and standardised form suitable for the new purpose. Constructing a common clinical research database from these diverse sources is a challenging activity and requires making a number of non-trivial assumptions about the transformations required, owing to the inconsistent way in which data will be recorded and the fact that data will often be missing or incomplete. Not maintaining consistency in the transformation of the data for these assumptions across different data sources most certainly will lead to inaccuracies that may, in cases, invalidate the results of the research.

In this paper, we propose an approach for expressing the provenance of clinical data transformations, whilst recognising its application to other domains where data is transformed from one repository to another. This not only informs users how the data was derived but also makes the transformation process more transparent and thus both consistent and well formulated so that it can be reproduced in a comparable way for other data sources.

### The Scope of the Problem


The task of populating the research database here is often referred to as an ETL (Extract-Transform-Load) process. Indeed, ETL is a standard activity when migrating data from one system to another. There are tools such as TALEND<sup>1</sup> and SSIS<sup>2</sup> (SQL Server Integration Services) to support this activity. However, these tools do not utilise provenance to maintain consistency in the transformation assumptions made. This may lead to inconsistencies in the derivations made later from the data.

The problem of constructing a clinical research database from EHR data can be summarised as having some or all of the following characteristics:

---

<sup>1</sup>Talend: <http://www.talend.com>

<sup>2</sup>SQL Server Integration Services: <http://msdn.microsoft.com/en-us/library/ms141026.aspx>

- 
1. The clinical research database was designed for the purpose of research and may not be directly compatible with all potential sources in the database, which may not be known when the research database was designed;
  2. The source EHR systems were not designed for the purpose of the research and thus may only contain partial or incomplete data;
  3. The source EHR systems vary in both the data they contain, including its meaning, and the way they represent it;
  4. Much of the source data is irrelevant to the intended research purpose, e.g. information used for billing or contacting patients;
  5. Some data reveals patient identity and so must be either discarded or obfuscated;
  6. The source data may have already undergone some transformation for the purpose of standardising data from diverse sources, or been prepared for some other intended research purpose;
  7. Some information is not held explicitly in the source data but is instead held either in metadata associated with the source or implied by the context. For example, units of measurement may be defined only in the accompanying documentation or implied by the context.

We shall refer to the intended clinical research warehouse as the ‘target database’. Henceforth, we shall consider a particular source of data used to populate the target as a single ‘source database’, although in hospitals there may actually be more than one database accessed. Without loss of generality, we shall use the concept of a relational database and its related vocabulary throughout, but note that the data may actually be represented as XML files or flat files. The latter could be readily converted to such a single-table database if the fields were suitably delimited. For XML, we will instead refer to elements and attributes but note that the underlying principle would apply. We note also that the databases could be virtual, where instead of the data being physically stored, it is synthesised at runtime in accordance with some abstract information model.


A further dimension to the problem is that patient records often store clinical concepts as categorical data, in which a given attribute may hold one of hundreds or thousands of possible values. Examples are diagnoses, prescribed drugs and procedures. To avoid the ambiguity and synonymy of natural language, as well as the possibility of typographical errors, each of these clinical concepts is given a code according to one of the many clinical coding systems, such as SNOMED<sup>3</sup> or ICD-10<sup>4</sup>. Although mapping between coding systems is a non-trivial problem beyond the scope of this paper, difficulties arise when an exact match of concepts does not exist and it is necessary to map a specific concept to a more general one.

Clinical researchers using the target database will have a number of questions concerning the data they wish to gain answers for:

---

<sup>3</sup>SNOMED: <http://www.connectingforhealth.nhs.uk/systemsandservices/data/uktc/snomed>

<sup>4</sup>International Classification of Diseases (ICD): <http://www.who.int/classifications/icd/en/>

- 
1. Where did this data come from?
  2. How has it been changed from its original form?
  3. How accurate/reliable is it?

A simple example here is obfuscation of the date of birth (DOB) to protect confidentiality, so that all patients are deemed born on 1<sup>st</sup> January of the year in which they were actually born. This will cause a bias on the patients' age, making all patients – on average – six months older than they really are and this would inform the use of statistics on such data.

### **The Aims of this Paper**

This paper provides an approach by which provenance data may be generated automatically from the data transformation process. This serves two purposes:

1. The end user can use the data appropriately by gaining answers to the questions above;
2. When adding data from another similar source, it is possible to ensure that transformations are consistent with that data.

In order to represent the provenance as a sequence of distinct transformations, we first require modelling the transformation process and provide a classification of the separate transformations that enables us to reason about them. This model not only enables the provenance to be expressed, but also acts to make explicit the required assumptions. Since, when performing ETL on clinical data, there are often many different conventions that could reasonably be adopted, the act of documenting assumptions often leads to those assumptions being made in a considered manner and assures a degree of consistency. Thus the modelling of the transformation process required for provenance is a useful step in its own right.

In essence, the transformation process is driven by the needs of the intended research and thus consists of populating the research database from source data as far as this is practical. It is 'target pull' rather than 'source push'. There is, therefore, a connection between provenance and the design of the ETL process. The former addresses the question: "Where does this data come from?" the latter: "How can we capture this data and place it in our warehouse."

### **The Structure of this Paper**

In this paper, we first review the previous work on provenance. Next, we create a model of clinical research data transformation and two levels of classification over the various mappings used in transferring the information stored as data. We then describe a process for creating provenance data that may be automated, after which we offer a case study of a transformation of data from pre-processed EHR data to a research database. Provenance diagrams are created to illustrate the use of the model and also serve to document the ETL process. Finally, we offer examples of provenance questions that could be answered by this approach and some conclusions.

## Background on Provenance

While the need for keeping records and documenting processes has always been evident, work on automatic capture and query of provenance information within software systems is a topic that has been of particular interest over the past decade, with hundreds of papers on the topic in recent years (Moreau, [2010](#)).

In this time, ideas regarding data provenance in geoscience (Di et al., [2013](#)), library studies<sup>5</sup>, bioinformatics and e-science (Miles et al., [2007](#)), and other fields were brought together to allow the definition of generic, reusable models and representations of provenance, and approaches to its capture and access. These generic models are helpful skeletons around which application-specific models, such as presented in this paper, can be created.

The research has largely followed two paths. First, researchers with a focus on database systems defined approaches to determining and describing the provenance of database query results (Cheney, Chiticariu and Tanet, [2007](#)). The aim is to determine from where in the database the results obtained derive, why these particular values are in the results, and why other values are excluded from the results (Chapman and Jagadish, [2009](#)). In these approaches, the provenance is determined either by analysing the query along with the database after the query execution has been completed, or by propagating provenance information during execution from the source data through intermediary results to accompany the query result.

The second path examined how to automatically record steps in a process, and came primarily from those developing workflow systems (Davidson et al., [2007](#)). Here, the provenance of a process is captured as a side effect of its execution, and it is this approach we follow here. These approaches led to common models and representations of provenance, allowing disparate parts of a distributed system to each record their parts of the process and then combine the provenance afterwards. A widely used de facto standard is the Open Provenance Model (Moreau et al., [2011](#)), which had a strong influence on the forthcoming W3C (actual) standard, PROV<sup>6</sup>. Attempts have been made to begin to bridge the gap between the two paths discussed. For example, Acar et al. ([2010](#)) develop a data flow calculus that can describe workflow provenance but can also be used to answer questions posed regarding database provenance.

In PROV (and similarly in OPM), provenance is defined in terms of entities (things in the world, such as items of data), activities (processes which use and generate entities) and agents (those responsible for activities, most obviously people). A provenance record links these into a directed graph, documenting where an activity generated an entity, which was then used by another activity, for which an agent was responsible, and so on. In any given application, the vocabulary of PROV needs to be specialised to that of the application's domain. Ideally, the specialised model is still somewhat general, allowing the model to be reused in other applications in the same domain. In this paper, we take transformation of clinical data as our domain and provide a specialised, reusable model accordingly.

---

<sup>5</sup> For instance, the Dublin Core Metadata Initiative: <http://dublincore.org/documents/dcmi-terms/>

<sup>6</sup>W3C PROV Primer: <http://www.w3.org/TR/prov-primer/>

The relevance of provenance to healthcare data is recognised, for example, by the GridProvenance project<sup>7</sup> (Kifor et al., 2006). More detailed work has been done in using provenance to address the issue of confidentiality and unauthorised access to data (Mashima and Ahamed, 2012). The application of provenance has also been applied to the ETL process, although not specifically in the healthcare domain by Freitas et al. (2012), who argue for a minimal vocabulary for expressing the ETL transformations. Such a vocabulary is not sufficient for our needs. Specifically, the problem of repurposing data (including its anonymisation) so that data from diverse sources is in a consistent format requires non-exact transformations, which, while altering the data, are necessary for its new purpose. No such provenance model exists for this purpose.

## Modelling the Transformation Process

Taking account of the volume of data (often megabytes or gigabytes), the transformation process needs to be automatable so that, once defined, it should require no human intervention. The transformation of the source database to the target may be seen as a pure (i.e. stateless) function. Given that there may be any number of fields of data and considering each field as a dimension, we have a function in which the domain and codomain are infinite-dimensional spaces. To make this manageable, we can think of the function as being composed of many instances of a finite number of sub-functions that operate on specific entries in each database. Indeed, if we could not do this, this process could not be expressed as an algorithm implemented in a programming or scripting language. Thus this modelling process has two distinct steps: decomposing the transformation process into a number of smaller basic steps and then seeking to classify those steps.


### Defining Sub-Functions

It is worth noting here that for any transformation process to be meaningful and useful, we have to understand the semantics of the data and consider the process as being one of extracting and transforming data in order to preserve its information content and meaning as far as this is possible.

Within each database there will be named tables with named fields (columns) representing the attributes of interest. It makes sense to group certain related fields together and thus treat a single molecule of data, consisting of atoms that correspond to specific values, conceptually. For example, a measurement of some physical quantity (e.g. height) may have an explicit unit of measurement (e.g. meters). When converting from one unit to another, any sub-function should map both the number and unit to another number and unit. Mapping the numbers and units in isolation loses their semantic link. Thus we define functions that map one or more fields from one row in the source to one or more fields in the target. However, more complex situations may arise:

1. There will often be a chain of transformations. For the analysis described below, we need to ensure that each basic transformation is made explicit, e.g. converting weight in pounds to kilos or rounding to nearest 0.1 kg.

<sup>7</sup> GridProvenance: <http://www.gridprovenance.org/applications/EHCRS.html>

- 
2. We will often rely on metadata to make sense of the actual data. For example, the unit of measurement may be implicit if it is always the same for some physical quantity. It might be documented elsewhere or obvious from the context.
  3. Given there is unlikely to be a direct correspondence between the database schemata, there may need to be intermediate data structures or tables to process data.

We thus propose the following algorithm for constructing a model of the transformation process. This algorithm defines the process that the member of technical staff implementing the ETL needs to perform the process to transform the data and also record the provenance. Once implemented, the ETL will run without human intervention.

For each table:

- a. Identify the fields in the target database to be populated;
- b. Group closely related fields (e.g. a measurement and its unit) together as a molecule;
- c. For each molecule:
  - c.i. Identify the fields in the source database that provide relevant data;
  - c.ii. Determine the sequence of sub-functions in the source database or metadata needed to make the transformation with appropriate intermediate data representations;
  - c.iii. Determine also any external sources of data – i.e. where the target is populated with data not from the source, e.g. a unit of measurement is implied in the source data but made explicit in the accompanying documentation;
  - c.iv. Determine the sub-function used and the data that acts as an input recursively back to the fields of the source database. This allows us to trace the sub-functions back from the one whose output is in the target back to the ones whose input is the source;
  - c.v. Assign a classification to each sub-function according (as described below).

### **Primary Classification of Sub-functions**

As we stated above, the schemata and semantic interpretation of the source and target databases should be taken as given, so there will inevitably be differences in the way that information is represented in each. The primary classification determines whether information is lost and/or added during transformation. It provides four categories that are mutually exclusive and collectively exhaustive.

There will, of course, be much information from source that is not used at all because it is not relevant to the research in question or else it compromises the privacy of the patients. However, for information that is actually used there may be situations where information is deliberately lost. For example, if the coding system used in the target database does not record data to the same level of granularity as the source, the sub-function may lose information by mapping to a coarser level of granularity e.g. ‘type 2 diabetes’ to ‘diabetes’. This is an example of information subtraction and we can classify any function as being either lossless or lossy. This is analogous to Woodruff and Stonebrake’s (1997) concept of invertible and non-invertible functions. A lossless function is reversible in that it will be possible to retrieve the original information, i.e. the function  $f$  has an inverse function  $f^{-1}$  where  $f^{-1}(f(a))=a$ . If the original information cannot be recaptured it is lossy.

The most severe case of a lossy function is ‘discard’, where the information is thrown away entirely. Suppose that a new drug has been designated a code in the source coding system but not yet in the target coding system. In this case one option would be to discard the record of that drug being prescribed, along (possibly) with associated data, such as the date prescribed and dosage.


There may be situations where information is actually added. In terms of information addition, we can classify each function as either conservative or presumptive. A function is conservative if no information is added. A presumptive function adds, and in many cases fabricates, information. Informally, we define a conservative mapping as one where nothing can be inferred from  $f(a)$  that cannot be inferred from  $a$ . More formally, we can see a conservative function as a mapping  $f: X \rightarrow Z$  whereas a presumptive mapping adds information from  $Y$  so that  $f: X \times Y \rightarrow Z$  where  $f(x, y_1) \neq f(x, y_2)$  for some distinct  $y_1, y_2 \in Y$ .

An example of a presumptive function is conversion of a date in the source to a datetime (a type that stores both date and time of day) in the target. If the source records the date of discharge (from hospital) as 23/3/2012, then, to store this as a datetime, a time of day must be added. By convention, this is midnight. However, this leads us to infer that the patient was discharged at midnight, which is almost certainly not true.

Functions may both subtract and add information at the same time (presumptive lossy). An example is the obfuscation of the DOB, used to protect confidentiality. Often each DOB is set to the first day of the year. Such a function gives the patients, with  $>0.997$  ( $=364/365$ ) probability, the wrong birthday and cannot be reversed to give the correct one. Note that if the DOB were mapped just to the year of birth, this would be conservative lossy. To say that someone, born on 22/8/1965, is born in 1965 is true. To say that he was born on 1/1/1965 would be untrue. However, for a database schema that requires a full date and not just a year as integer, a fabrication is required. For a recipient of the data to use the data correctly, he would need to be aware of this convention and ignore the date and month.

Table 1 shows the four possible types of function, with examples:





Information addition	Information subtraction	
	Lossless (invertible)	Lossy (non-invertible)
Conservative	Identity mapping Transformation of measurement scale e.g. kg > g	Loss of granularity e.g. Lung cancer > Cancer Discard
Presumptive	Converting age (in years) to DOB Date to datetime by assuming time of day	Setting all DOB to first day of the year

Table 1. Classification of functions by information loss/gain.

A function that is both conservative and lossless is faithful. If all functions are faithful then defining the transformation is largely straightforward. But this will seldom be the case. So, although the transformation process is automatic, defining this process requires subjective judgment and there is a trade-off between either losing good information or adding some false information. Considering the example above, we can avoid the (probable) falsehood that the patient was discharged at midnight by discard, but we then lose also the date they were discharged. An alternative to a presumptive (and so fabricated) birthday is a conservative transformation that records no DOB at all, in which case we can infer nothing about age of the patient. In these cases a small amount of fabrication seems justified. However, if no DOB were provided in the source at all, assuming an entirely arbitrary one is perhaps a fabrication too far.

### Composition of Sub-Functions

If two or more functions are all lossless, then the composite will also be lossless. However, if any one is lossy, the composite will be lossy. Similarly, if two or more functions are conservative, the composite is conservative. However, if any one of them is presumptive then the composite is also presumptive.

Because presumptive and lossy functions affect any composite function of which they are part, they are of most interest both in making assumptions when designing the ETL process and for provenance.

### Addressing the Problems of Discard

It is normal for a database to have tables in which certain fields are mandatory (non-nullable). There is therefore a knock-on effect of discard of a mandatory field. If one mandatory entry is not given a value, then the entire row must be discarded. Furthermore, if a discarded row has a primary key that is used as a mandatory foreign key in some other table, it would necessarily result in rows of other tables also being discarded. Discard, although conservative, can be quite destructive. A degree of presumption can keep data whilst inevitably adding false information to preserve the true information. The provenance data should prevent the data recipient from drawing false inferences from the fabricated data.

## Secondary Classification of Sub-Functions

We now identify a more detailed classification of the sub-functions. Such a classification was grounded in specific examples used in the case study below and is not intended to be exhaustive. New categories will inevitably be added as other transformation processes are analysed. It is, however, intended to describe the nature of the sub-functions at a more general level. Unlike the primary classification, there is no obvious way of combining these generic transformations where two or more are applied. Table 2 identifies ten generic transformations with examples from the case study below given. We note that there are no presumptive lossy sub-functions, since these usually occur with the composition of a conservative lossy sub-function with a presumptive lossless one.

Information change	Generic transformation	Examples	Namespace
Faithful	Copy	Copying a patient ID	cwpCopy
	Faithful mapping	1. Local gender code to SNOMED mapping 2. Converting between two conventions for expressing ICD-10 codes	cwpFaithMap
	Database restructuring	Converting a single row with many diagnoses to many rows with one diagnosis each	cwpDb
	Concatenate data	Combining many data files to create one large file	cwpConcat
	Arithmetic	Calculate DOB interval from age in years and date	cwpArith
Lossy conservative	Loss of granularity mapping	OCPS to SMOMED mapping	cwpLossyMap
	Choosing one of a set of possible values	Choosing one episode for age or gender	cwpChoose
	Discard	Where there is no admission date, age is not calculated and patient is discarded	cwpDiscard
Lossless presumptive	Increased precision	All dates – assume midnight to give a datetime	cwpIncPres
	Converting a (time) interval to a single point	Putting a point date on a diagnosis and encounter	cwpInt2Point
	Adding a hard-coded value based on implied meta-data	Inserting a notional hospital name	wpInsert

Table 2. Classification of generic transformations.



## Provenance Data from the Transformation Process

For provenance data to be created automatically from the transformation process, the latter needs to be represented in a machine-readable form. This requires both a representation of the flow of information and a labelling of the sub-functions (processes) and molecules (entities). However, the machine-readable form, typically in XML, is not human readable and would be difficult for the purpose of attempting to create a set of transformations consistent with earlier ones. Provenance diagrams provide a visual representation of the particular set of transformations used to generate the actual data in the target database. This serves as a documentation of the ETL process and can inform the ETL process required for fresh data.

### Provenance Diagrams

Provenance diagrams are used to graphically represent the lineage of data. They are always backward looking and so arrows flow from the data item to the process that created it and thence back to the data used for this process. Where there is a defined ETL process, a provenance diagram can be produced statically (i.e. without actually running the ETL process) to give templates of how data would be derived in advance of running the transformation process. The sub-functions are annotated and also we colour to determine whether information addition and/or subtraction take place. The use of backward-looking provenance diagrams is apposite for the ETL process described here, since the focus is on populating the target database from whatever can be used in the source.

### Annotating the Processes and Entities


Any process or entity in a provenance diagram must be given an identifier and a namespace. The name spaces were created for the source, intermediate representations and target for the entities. For the processes, each generic transformation was given a separate namespace so that for specific sub-functions it is possible to determine which class it belongs to.

From these machine-readable names and the provenance graphs it becomes possible to create the provenance data automatically during the transformation process.

## Case Study

A research database schema was designed to hold anonymised patient details so that, when queried, it could estimate the number of patients satisfying a number of eligibility criteria for a clinical trial and thus estimate the number of patients that could potentially be recruited. This was part of the project Electronic Healthcare Records for Clinical Research (EHR4CR). In order to test the system, the database was populated with publicly available data of anonymised hospital admissions – known as hospital episode statistics (HES) – for England<sup>8</sup>. The eventual aim is to extract data from specific hospital EHR systems. This initial population of the database with the HES data provided a useful case study for the approach described

<sup>8</sup> Hospital Episode Statistics: <http://www.hscic.gov.uk/hes>



above. The ETL process was implemented using the Oracle scripting language and the provenance data was added to the database using this mechanism too. This is further explained later in this paper.

### Target Database

The target database schema was based on a subset of the HL7 RIM (Benson, [2009](#)) data model. Specifically, it contained the following tables:

- **Subject:** Details of DOB, date of death (DOD) and gender for each patient;
- **Observation:** Diagnoses and findings (e.g. lab tests, vital signs) associated with a date, location and patient;
- **Procedures:** Procedures associated with a date, location and patient;
- **Administration:** Each administration or prescription of a drug or other substance associated with a date, location and patient;
- **Encounter:** A set of observations, procedures and administrations associated with one visit or stay;
- **Organisation:** The place where each observation, procedure or administration takes place.

Diagnoses are represented using the coding system ICD-10; all other codes used are represented as SNOMED-CT.

### Source Database

The HES data is stored in a flat files, each one representing three months of data. Standard tools were used to firstly concatenate the files and then load them into a single-table database. Both of these transformations were faithful.

Each row in the sole table represents a single hospital episode in which there may be any number of diagnoses and procedures represented in a predetermined number of columns. No finding or prescription/administration data were present, so the observation table contained only diagnoses and the administration table remained empty. Demographic data for the subject table was stored but in quite different formats. Diagnoses are represented in ICD-10, albeit using a slightly different convention. Procedures are represented using OPCS<sup>9</sup>. Conversion tables from OPCS to SNOMED were publicly available<sup>10</sup>.

### High-Level ETL Process

An initial ETL process was implemented and provenance diagrams were subsequently constructed. This process revealed the assumptions underlying the transformations and

---

<sup>9</sup> OPCS-4 Classification Health:

<http://www.connectingforhealth.nhs.uk/systemsandservices/data/clinicalcoding/codingstandards/opcs4>

<sup>10</sup> See the Technology Reference data Update Distribution (TRUD) website:

<http://www.uktcregistration.nss.cfh.nhs.uk/trud3/user/guest/group/0/home>

led to a number of changes. For example, a presumptive transformation was required to map both date of birth and date of a diagnosis from an interval to a point data. For consistency, both use the midpoint. Originally, diagnosis used the earliest possible date and birth the midpoint and so this inconsistency was resolved. Thus the set of provenance diagrams shown below were prescriptive rather than descriptive and drove subsequent modifications to the ETL implementation. Figure 1 shows the high-level ETL process at the level of each table. This high level diagram is informal since the entity ovals may represent several entities and the process boxes may represent several processes with the colour representing aggregation of the component functions. Later figures give more formal and detailed descriptions. Figure 2 gives the colour scheme used to annotate the various transformations. Note that in this and other provenance diagrams arrows from ovals to rectangles should be labelled ‘wasGeneratedBy’ and from rectangles to ovals ‘used’. This was omitted to avoid cluttering the diagram. The large number of coloured boxes and lack of white ones is due to these representing function composites – if one component process is coloured (not faithful) the whole process is coloured. The more detailed diagrams showing the basic process are mostly white (faithful). The junction table and clinical statement table were intermediate representations used to reconcile the different structures of the two databases and unflatten the data. The organisation table actual contained a single entry stating that all patients were treated at a notional KCL (King’s College London) hospital.

## Subject Table

Figures 3 and 4 show the provenance where the subject is included or discarded respectively. Since DOB is a mandatory field in the target database, if the information necessary to estimate this is absent, the whole patient is discarded. Of interest here are the determination of gender, DOB and date of death (DOD). It should be noted that patients will have one or more episodes associated with them. Since administrative gender (denoted ‘sex’ in the source) is recorded at each episode, only the first is used and others are discarded, hence it is lossy.

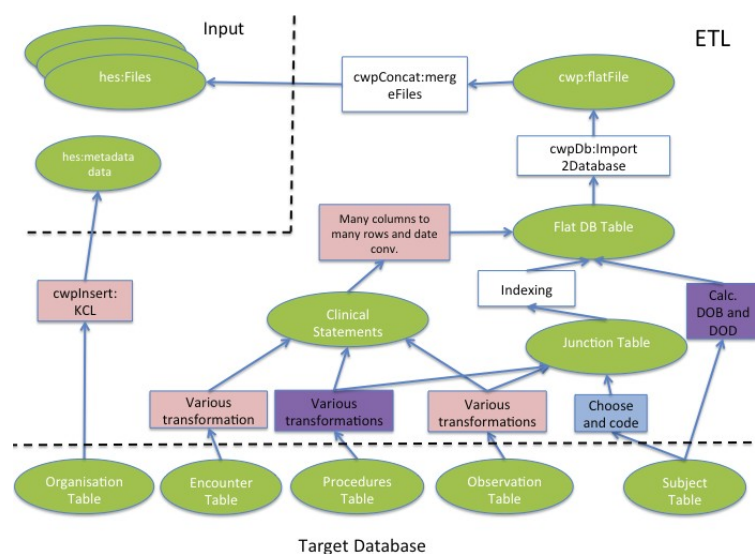


Figure 1. ETL process for entire database.

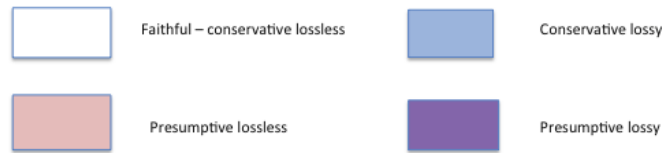


Figure 2. Colour scheme used in provenance diagrams.

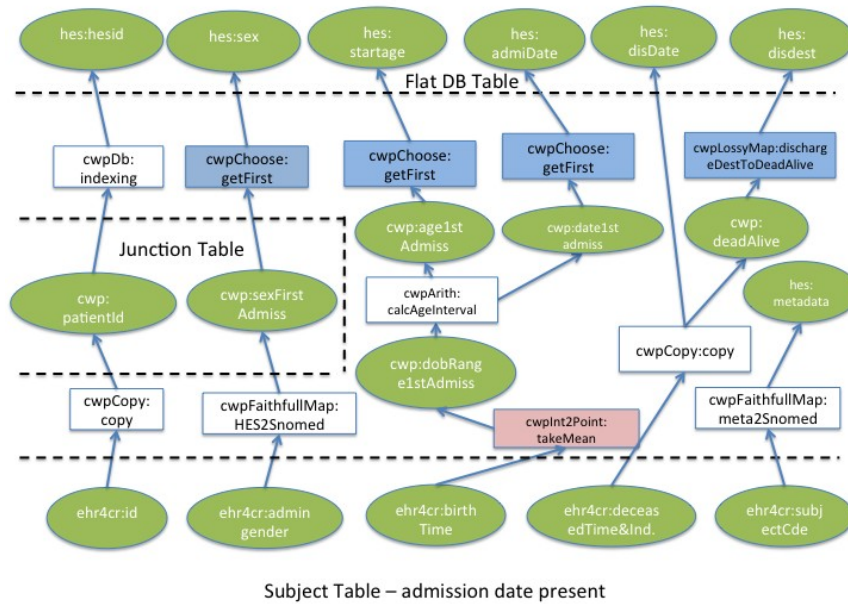


Figure 3. Provenance diagram for subject table where patient is included.

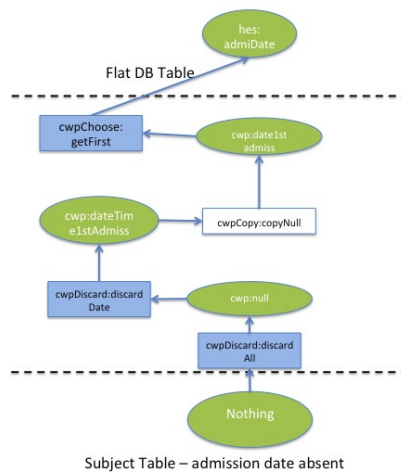


Figure 4. Provenance diagram for subject table where patient is discarded.

DOB is estimated from the age at admission (in whole years for those patients over 1 year old, slightly more complex for < 1 year) and the date of admission. Only the first episode is used, hence this is lossy. If the admission date is absent, the patient is discarded (lossy). However, an exact date cannot be obtained from this information

alone. Thus the presumption is made that it is exactly six months from the patient's birthday, taking the mean of the two limiting cases.

DOD is not recorded explicitly. However, when the patient is discharged there is a code to denote the status at discharge. One such status code denotes death, thus it is possible to determine whether the patient is dead or alive at discharge. This is lossy since we map all 'living' statuses onto a single value. The DOD may be derived from the date of discharge, provided that the patient is dead – also lossy. The discharge status is actually stored in the encounter table, so that this function is only locally lossy.

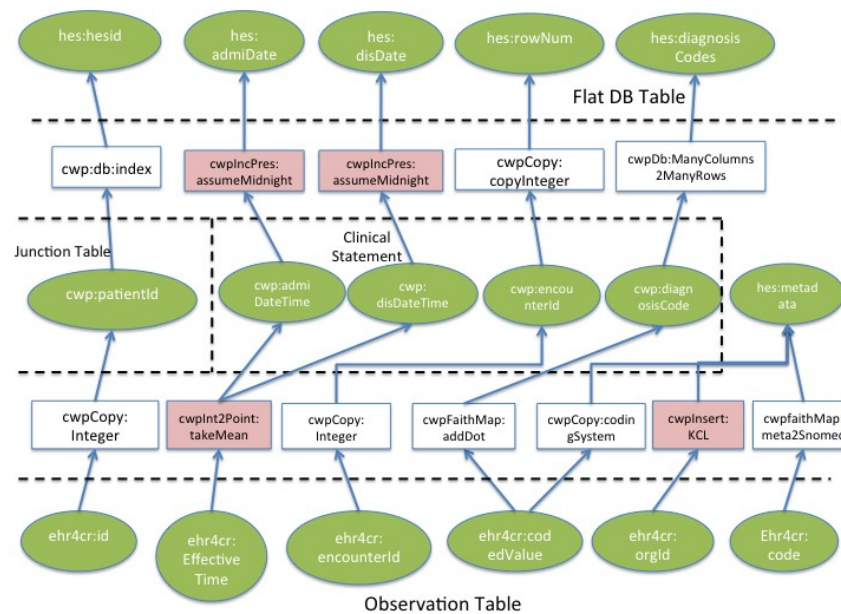


Figure 5. Provenance diagram for observation table.

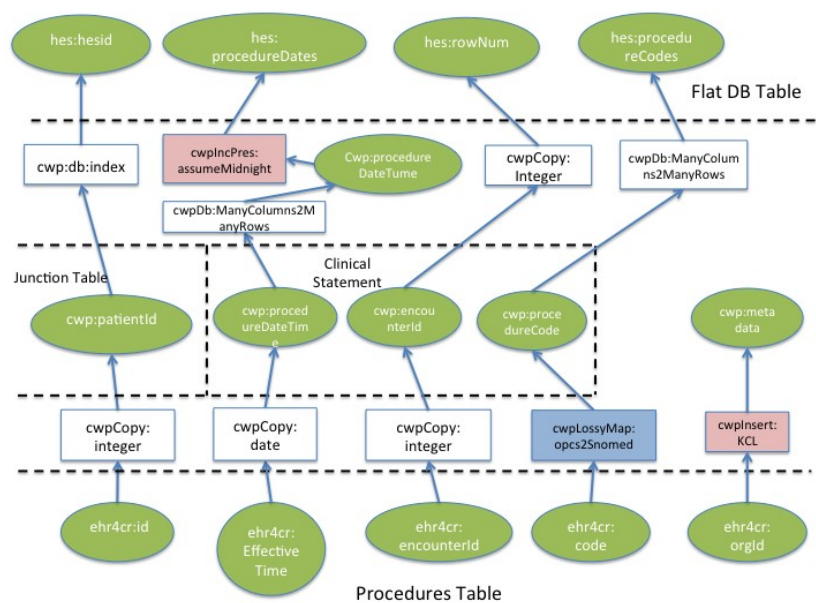


Figure 6. Provenance diagram for procedures table.

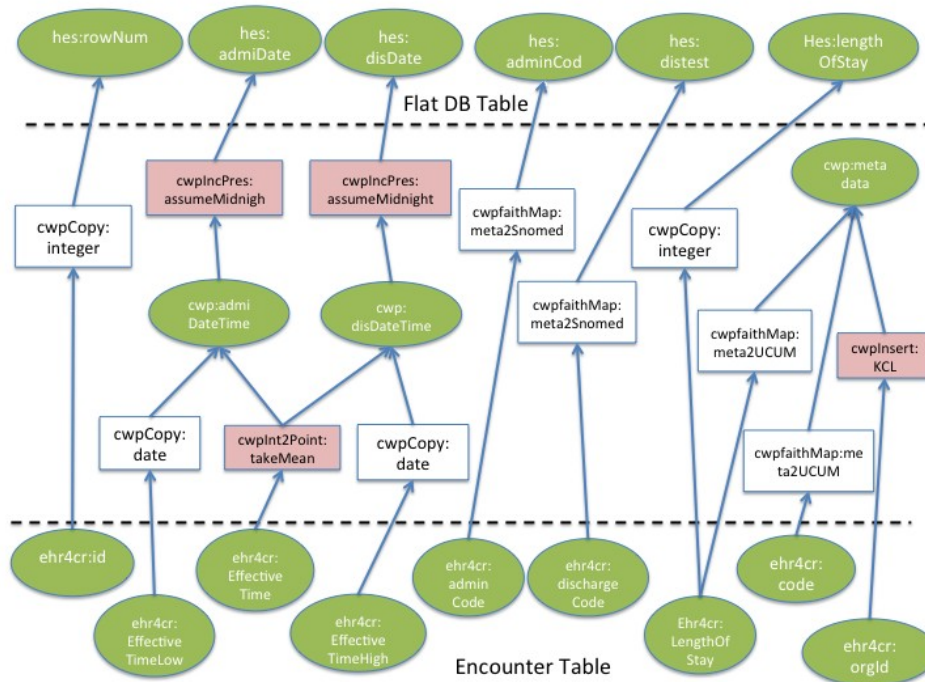


Figure 7. Provenance diagram for encounter table.

## Other Figures

Figures 5, 6 and 7 show the provenance diagrams for the observation, procedures and encounter tables. The examples of presumptive transformations are:

- Assume midnight when converting a date to a datetime;
- When no date is given for a diagnosis, use the mean of the admission and discharge date as an estimate. The same principle is used for the effective time of the encounter;
- Assume all activities take place at the notional KCL hospital and so this is a reference to the single row Organisation table.

The one example of a lossy transformation is in the mapping from OPCS to SNOMED for the procedure codes where there is a loss of granularity.

## Implementation of Provenance Data

The provenance data is stored in the target database by adding additional tables that capture the graph structure of Figures 1 and 3-7. These provenance tables store data at the level of each clinical fact. For the observation, procedures and encounter data tables there is one fact per row. For the subject table, there are three facts per row: birth, death and gender. The provenance tables are very small by comparison to the other data, since each transformation is used many times. Foreign keys in the data tables (three for subject and one for all other tables) link to one of the provenance tables so that the repeated use of the same set of transformations is normalised. Thus the provenance overhead is minimised. So for a data warehouse holding 200 patients,



the provenance increased database size (in bytes) by 56% but for one with about 24k patients, the increase was just 2.1 %.

There is currently no automatic means of producing the provenance graphs from the provenance data, although this is planned for the future. The examples produced in this paper were created manually.

## Provenance Questions

To demonstrate the use of the provenance data, we pose some typical questions that the data could answer and show how.

### 1. How accurate would an estimate of patient DOB be?

In Figure 3, following the arrows from oval ‘EHR4CR:birthtime’ shows that the estimated DOB is calculated from the earliest and latest possible dates (one year apart). Assuming that people are born uniformly throughout the year, this is an unbiased estimate with a standard deviation of three months. These dates are derived from the age at the admission date and the first admission is chosen. This choice is not a source of bias. We note from Figure 4 that if the admission date is missing, the patient is not recorded in the target database.

### 2. Can we be sure a patient is still alive or actually dead? Where does this information come from?

From Figure 3, we can see by following the arrows from the oval ‘ehr4cr:deceasedTime&Ind’ that the status ‘cwp:deadAlive’ is derived from the discharge status. Thus, if a patient is discharged dead, he is most definitely dead. However, being alive at discharge does not necessarily mean that the patient is still alive.

### 3. Is the diagnosis the same as originally recorded?

From Figure 5, starting at oval ‘ehr4cr:codedValue’, all the transformations are faithful (i.e. white). The first transformation ‘cwpDb:ManyColumns2ManyRows’ is structural and so does not change the data. The transformation ‘cwpFaithMap:addDot’ changes the format of the (IDG-10) code by placing in dots so that ‘E101’ (meaning type 1 diabetes with ketoacidosis) becomes ‘E10.1’. Up to dots, the diagnosis representation is the same. Note that ICD-10 codes may be recorded using either convention.

### 4. How accurate is the datetime of a diagnosis?

From Figure 5, starting at oval ‘ehr4cr:effectiveTime’, the transformation takes the mean of the earliest and latest possible date of diagnosis (admission and discharge). Any time of day assigned should be ignored since the transformation of a date to datetime has presumed midnight.

### 5. Are the procedures codes as originally recorded?

From Figure 6, starting at oval ‘ehr4cr:code’, there is a conservative lossy mapping from OPCS to SNOMED, so detail in the original recording of procedure is lost.



## 6. How accurate is the datetime of a procedure?

From Figure 6, starting at oval ‘ehr4cr:effectiveTime’, the date is derived from a single date recording the date of procedure (hes:procedureDates). However, the time of day is presumed to be midnight. Thus the date is accurate but the time of day is not.

Each of these questions address, at least once, the general provenance questions given at the beginning of this paper.

## Conclusions

Addressing the two purposes set out at the beginning of this paper, firstly, we note that we do now have an automatable means of informing users of the provenance of each molecule of information in the warehouse. The classifications of sub-functions make explicit the modelling assumptions required to perform these transformations and without them it would be difficult to generalise to other situations or identify inconsistencies in the conventions used. We can see from the provenance questions above how we can use the provenance data to reason about the clinical data. Secondly, the description of the precise sub-functions used can be used to ensure that if data were transformed from another source, the same or similar assumptions would be made ensuring comparability of the warehouse data.

The ETL process using the HES data is simpler than a full ETL of a typical EHR. This is because the data had already been partially transformed and because other information, such as findings and medications, was absent. Nevertheless, it provides an illustration of both the problems of determining the transformation process and generating the provenance data.

More generally, the problem of repurposing data from heterogeneous sources into some research databases requires transforming the data and making assumptions in the process of defining this transformation. Although the secondary classification of the sub-functions is ETL specific, the primary classification has a wider application and provides a useful way of categorising imprecise mappings.

## Acknowledgements

This work was part funded by the IMI-funded collaborative project Electronic Healthcare Records for Clinical Research (EHR4CR), grant agreement number: 115189. The work was also part funded and supported by the National Institute for Health Research (NIHR) Biomedical Research Centre, based at Guy’s and St Thomas’ NHS Foundation Trust and King’s College London. The views expressed are those of the authors and not necessarily those of the NHS, the NIHR or the Department of Health. We would like to acknowledge the work of Mark McGilchrist in providing the schema for the research database used in the case study.

## References

- Acar, U., Buneman, P., Cheney, J., Van den Bussche, J., Kwasnikowska N. & Vansummeren, S. (2010). A graph model for data and workflow, provenance. Paper presented at the second Workshop on the Theory and Practice of Provenance. San Jose, US. Retrieved from <http://dl.acm.org/citation.cfm?id=1855803>
- Benson, T. (2009). *Principles of health interoperability HL7 and SNOMED*. Springer: Berlin.
- Chapman, A. & Jagadish, H.V. (2009). Why not? Paper presented at the SIGMOD Conference 2009. [doi:10.1145/1559845.1559901](https://doi.org/10.1145/1559845.1559901)
- Cheney, J., Chiticariu, L. & Tan, W.C. (2007). Provenance in databases: Why, how, and where. *Foundations and Trends in Databases 1*, 379-474. [doi:10.1561/19000000006](https://doi.org/10.1561/19000000006)
- Davidson, S.B., Cohen Boulakia S., Eyal A., Ludäscher B., McPhillips T.M., Bowers, S., Kumar Anand, M. & Freire J. (2007). Provenance in scientific workflow systems. *IEEE Data Engineering Bulletin 30*(4).
- Di, L., Yue, P., Ramapriyan, H.K. & King, R.L. (2013). Geoscience data provenance: An overview. *IEEE Transactions on Geoscience and Remote Sensing, 51*(3). [doi:10.1109/TGRS.2013.2242478](https://doi.org/10.1109/TGRS.2013.2242478)
- Freitas, A., Kämpgen, B., Oliveira, J.G., O’Riain, S. & Curry, E. (2012). Representing interoperable provenance descriptions for ETL workflows. Paper presented at the third International Workshop on Role of Semantic Web in Provenance Management (SWPM 2012) Extended Semantic Web Conference (ESWC). Heraklion, Crete.
- Kifor, T., Varga L.Z., Vázquez-Salceda, J., Álvarez S., Willmott, S, Miles, S., & Moreau, L. (2006). Provenance in agent-mediated healthcare systems. *IEEE Intelligent Systems, 21*(6), 38-46. [doi:10.1109/MIS.2006.119](https://doi.org/10.1109/MIS.2006.119)
- Mashima, D. & Ahamad, M. (2012). Enabling robust information accountability in e-healthcare systems. Paper presented at the third USENIX Workshop on Health Security and Privacy (HealthSec 2012). Bellevue, WA.
- Miles, S., Deelman, E., Groth, P., Vahi, K., Mehta, G. & Moreau, L. (2007). Connecting scientific data to scientific experiments with provenance. Paper presented at the IEEE International Conference on eScience and Grid Computing 2007. Bangalore, India.
- Moreau, L. (2010). The foundations for provenance on the web. *Foundations and Trends in Web Science 2*, 99-241. [doi:10.1561/18000000010](https://doi.org/10.1561/18000000010)



Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil Y., Groth, Kwasnikowska N., Miles, S., Missier P., Myers, J., Simmhan, Y., Stephan, E. & den Bussche, J.V. (2011). The open provenance model core specification (v1.1). *Future Generation Computer Systems* 27, 743-756. [doi:10.1016/j.future.2010.07.005](https://doi.org/10.1016/j.future.2010.07.005)

Woodruff, A. & Stonebraker, M. (1997). Supporting fine-grained data lineage in a database visualization environment. Paper presented at the 13th International Conference on Data Engineering. Birmingham, England. [doi:10.1109/ICDE.1997.581742](https://doi.org/10.1109/ICDE.1997.581742)