Supporting Information for

Development of a repository of individual participant data from randomised controlled trials of therapists delivered interventions for low back pain

S.W Hee, M. Dritsaki, A. Willis, M. Underwood, S. Patel

Method S1 Technical details of the bespoke hybrid database

2.2.1 System architecture

Tables and columns in a relational database are represented as classes and attributes in an EAV model (Marenco et al., 2003). The term entity provides a similar role to a table row but with the significant difference of only storing a pointer to the data and not the actual data itself. We anticipated that some consistent data would be present in all randomised controlled trials (RCTs) for describing the trial and for identifying participants. Two tables, Primary Source and Subject, were created with fixed schemas to store this data (Fig. S1). The Primary Source table stores the name of the RCT (TrialName) and the repository import date (ImportDate). The Subject table stores the original identifier assigned to the participant (OriginalSubjectID), the participant trial enrolment date (EnrolDate) and a unique participant identifier generated by the system (SubjectID). A foreign key relationship is created to link each subject record to the Primary Source.

The EAV model uses a sub-schema consisting of tables for classes, attributes, objects and the EAV data. The Class table is used to hold a list of all the identified domains, e.g. Demographics. These domains generally map to a CRF but can also be used to describe a subset of repeating questions, e.g. repeated medical prescriptions. The Attribute table is used to hold a list of all identified variables that typically map to a CRF question. The Attribute table has columns for storing a short name, a verbose name, a reference to the containing class and data type details.

The Object table stores a unique identifier for each instance of a class and a reference to the class itself. A foreign key relationship is created to link each Object to a Subject. This relationship essentially makes the EAV model subject-centric, i.e. all data stored in the Object and EAV tables must be directly related to an imported subject record. Relationships between objects is possible by using an ‘ancestor column’ to store the unique identifier of a related object. This is an interpretation of the EAV with classes and relationships (EAV/CR) framework (Nadkarni et al., 1999), e.g. an object used for repeated medical prescriptions will store the unique identifier of the parent object in the ancestor column.

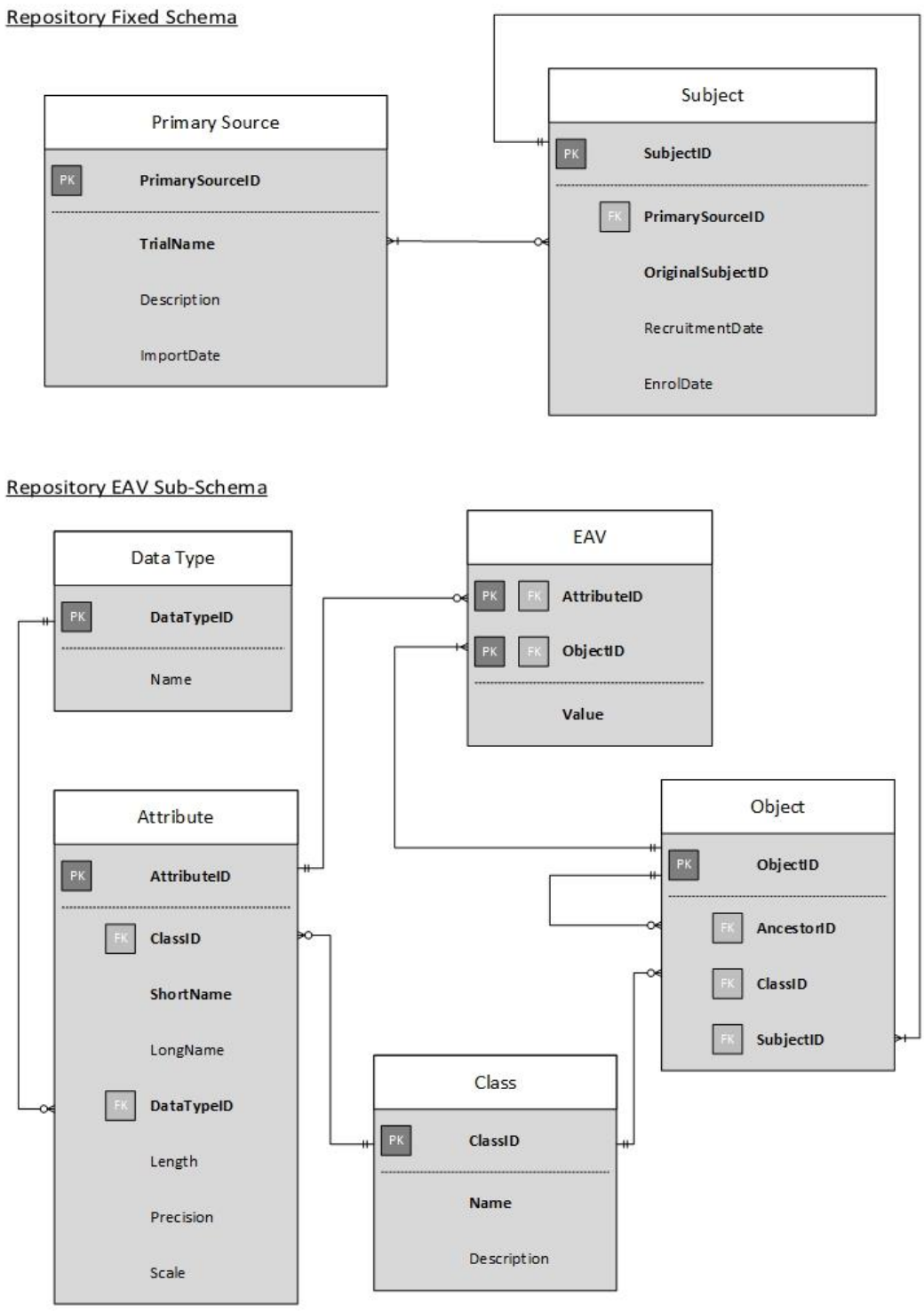

Figure S1 Entity-relationship diagram, depicting the fixed schema with the sub-scheme entity-attribute-value (EAV) tables.

The EAV data table has three columns and is used to store data from all RCTs in the repository. Two columns hold references to the related objects and attributes with the other column used for storing the actual value of each object/attribute combination. The references to the objects and attributes

take the form of foreign keys to the object and attribute tables. The format of the value is coerced into a string regardless of the intended data type e.g. integer, decimal, date or string. Values that cannot be represented as a string are not imported e.g. images and other binary formats.

Clinical trial database management systems typically incorporate study calendars to track form completion response rates based on established epochs. Patient calendars are then used to monitor actual responses for each participant. We consciously omitted these domains from the model in favour of simply recording the time point for when each form was collected within the actual object using the FU (follow-up) attribute. This was decided because it was not possible to predetermine all time points used in each RCT and the large variation made the use of a standardised study calendar superfluous. Also, there was no requirement in the repository to track form completion response rates as all data was collected retrospectively.

2.2.2 Mapping and transforming clinical data

The FU attribute stores the time point (in weeks) when clinical and/or healthcare resource-use data was collected. The time points for each form in each RCT were identified from the variable labels, trial protocols, published results or correspondence with the trial's statistician and was then hard-coded into the FU attribute. The FU attribute allows forms that share the same time point across all RCTs to be pooled for analysis.

The end users, statistician and health economist, first prepared the RCT datasets for upload by mapping the original variables to the corresponding repository attribute. Where applicable, transformation rules were added to describe how the original value will be changed to comply with the repository standardised codes. The extensible mark-up language (XML), a free and open-source standard governed by the World Wide Web Consortium (WC3), was used for describing both the mapping and transformation rules (XML Technology, 2010).

An XML schema (XSD) was created to ensure all mapping and transformation rules were specified in the correct format and order. The XSD defines the permitted XML structure and is used to validate the content of the XML document. We used Syncro Soft's oXygen software to create and edit the XML and XSD documents. This editor provides user friendly features such as drop-down lists for enumerated types, schema-aware autocomplete content, annotations and syntax error highlighting. These features allow non-programmers to create and edit the mapping and transformation rules,

forgoing the requirement to pass instructions to a programmer, saving resources and decreasing misinterpretation errors.

Figure S2 shows an example of the XML mark-up to map the sample data seen in Fig. 1 (main paper) to the equivalent repository attributes. The repository standard attributes 'SEX' from the class 'DEMOGRAPHICS' was mapped to the original variable 'Sex'. The XML mapping element 'attributeName' accepts values for the original variable name ('original Name') and the follow-up time point ('FU') as XML attributes. The value of the 'attributeName' XML element is set to the name of the repository attribute. In our example, for class 'EQ5D', the original variables 'EQ1' and 'EQ1_1', were mapped to the repository attribute 'EQ5D1' but had been assigned different time points. Unlike in the original tabular data, the repository does not store different attribute names for each time point. Instead, each time point will trigger a new object to be created. The XML 'FU' attribute is used to track which time point and original variable belongs to. The XML 'FU' attribute has been omitted for the 'DEMOGRAPHICS' class because this data is only collected once for each participant.

The original values for male and female from trial A correspond to the repository standard 1 and 2, respectively, and so no data transformation was done except that if the value was not one of these then it was considered as missing ('Null' value). On the other hand, two 'match' rules were used on the 'SEX' attribute to find the original values 'M' and 'F' from trial B. When the value 'M' was matched, the rule had been configured to update the attribute's value to '1'. Likewise, when 'F' was matched, the attribute's value was updated to '2'. The transformation for the EQ5D1, EQ5D2, EQ5D3, EQ5D4 and EQ5D5 of trial A used a 'range' rule to only allow values between 1 and 3 to be imported. If any of their values fall outside this range the system will transform the value to 'Null'.

2.2.3 Mapping and transforming healthcare resource-use data

Multiple healthcare resource-use items can be recorded at any follow-up time point. To permit this relationship the XML schema was modified to allow related classes to be defined, which in turn gets interpreted by the system to create the relationships in the Object table (see Fig. S1). The HE class is only used to define the time points for collecting the healthcare resource-use data. The actual resource-use data is defined in the HE-DATA class.

Figure S3 shows the 'HE-DATA' class being used as a child class, i.e. it has the 'HE' class as its parent. Creating child classes signifies to the system that a relationship exists between the two classes. The 'linkedValue' XML attribute on the 'childClass' element is used to specify a shared value between the

A

```
<!-- Demographics -->
<class name="DEMOGRAPHICS">
  <mapping>
    <attributeName originalName="Sex">SEX</attributeName>
  </mapping>

  <transform>
    <range min="1" max="2" operator="not in range">
    <newValue attributeName="SEX">Null</newValue>
  </transform>
</class>

<!- EQ5D-->
<class name="EQ5D">
  <mapping>
    <attributeName originalName="EQ1" fu="0">EQ5D1</attributeName>
    <attributeName originalName="EQ2" fu="0">EQ5D2</attributeName>
    <attributeName originalName="EQ3" fu="0">EQ5D3</attributeName>
    <attributeName originalName="EQ4" fu="0">EQ5D4</attributeName>
    <attributeName originalName="EQ5" fu="0">EQ5D5</attributeName>
    <attributeName originalName="EQ6" fu="0">EQ5D6</attributeName>

    <attributeName originalName="EQ1_1" fu="4">EQ5D1</attributeName>
    <attributeName originalName="EQ1_2" fu="4">EQ5D2</attributeName>
    <attributeName originalName="EQ1_3" fu="4">EQ5D3</attributeName>
    <attributeName originalName="EQ1_4" fu="4">EQ5D4</attributeName>
    <attributeName originalName="EQ1_5" fu="4">EQ5D5</attributeName>
    <attributeName originalName="EQ1_6" fu="4">EQ5D6</attributeName>
  </mapping>

  <transform>
    <range min="1" max="3" operator="not in range">
    <newValue attributeName="EQ5D1" fu="0">Null</newValue>
    <newValue attributeName="EQ5D2" fu="0">Null</newValue>
    <newValue attributeName="EQ5D3" fu="0">Null</newValue>
    <newValue attributeName="EQ5D4" fu="0">Null</newValue>
    <newValue attributeName="EQ5D5" fu="0">Null</newValue>
    <newValue attributeName="EQ5D1" fu="4">Null</newValue>
    <newValue attributeName="EQ5D2" fu="4">Null</newValue>
    <newValue attributeName="EQ5D3" fu="4">Null</newValue>
    <newValue attributeName="EQ5D4" fu="4">Null</newValue>
    <newValue attributeName="EQ5D5" fu="4">Null</newValue>
  </transform>

  <transform>
    <range min="0" max="100" operator="not in range">
    <newValue attributeName="EQ5D6" fu="0">Null</newValue>
    <newValue attributeName="EQ5D6" fu="4">Null</newValue>
  </transform>
</class>
```

B

```
<!-- Demographics -->
<class name="DEMOGRAPHICS">
  <mapping>
    <attributeName originalName="Age">AGE</attributeName>
    <attributeName originalName="Gender">SEX</attributeName>
  </mapping>

  <transform>
    <match operator="equal" value="M">
      <newValue attributeName="SEX">1</newValue>
    </match>
    <match operator="equal" value="F">
      <newValue attributeName="SEX">2</newValue>
    </match>
  </transform>

  <transform>
    <match operator="less than" value="0">
    <newValue attributeName="AGE">Null</newValue>
  </transform>
</class>

<!-- RMDQ -->
<class name="RMDQ">
  <mapping>
    <attributeName originalName="RDQ_0" fu="0">RMDQ</attributeName>
    <attributeName originalName="RDQ_3mo" fu="13">RMDQ</attributeName>
  </mapping>

  <transform>
    <range min="0" max="24" operator="not in range">
    <newValue attributeName="RMDQ" fu="0">Null</newValue>
    <newValue attributeName="RMDQ" fu="13">Null</newValue>
  </transform>
</class>
```

Figure S2 XML instructions to map the original clinical data to the equivalent repository attributes and transform the original values to the repository standard. (A) Mapping and transformation for sample clinical data from trial A (Fig. 1(a)). (B) Mapping and transformation for sample clinical data from trial B (Fig. 1(b)).

parent and child classes. In a relational database, this shared value would be created as a foreign key constraint. In Fig. S3, a 'HE' class has been defined for the 3-month (equivalent to 13 weeks) follow-up time point using the attribute 'fu': <attributeName fu="13"></attributeName>. A child 'HE-DATA' class has been defined and linked to the parent 'HE' class by specifying the value "13" for the 'linkedValue': <childClass name="HE-DATA" linkedValue="13">. This corresponds with the 3-month follow-up time point specified in the HE class.

The original tabular data required 13 columns across three rows to store all the data for the three participants (Fig. 2 (b)). Instead of creating a new column for every resource, the repository creates a new object. Thus, seven objects are created for GP visit ('Pri1'), NHS physiotherapist visit ('Pri2'), private physiotherapist visit ('nPriv1'), two instances of prescribed medicine ('pmed1', 'pmed2') and two instances of aids or medications bought over the counter ('bmed1', 'bmed2').
Child classes in the XML use 'groupName' elements to signify the number of objects that need to be created. In a relational database, this would result in adding a new 'groupName' element for every table row to be inserted. The value of the 'groupName' element has no significance except that it must be unique. In Fig. S3, although there were seven groups identified, only six needed to be created because there was no data for 'pmed2'. Thus, six groups have been created for the 3-month resource-use data: '3moResource1', '3moResource2', '3moResource3', '3moResource4', '3moResource5', and '3moResource6'. Referring to question 1 of the CRF (Fig. 2 (a)), the group '3moResource1' has been created for GP visits at the 3-month follow-up. The original variable 'Pri1' stores the number of GP visits, therefore 'Pri1' is mapped to the repository attribute 'Quantity' for the group '3moResource1'. The <staticValue/> element is used to specify fixed values that have no dependency on the original value. The recall period (RP), type of resource (Type), reason for use (Reason), location (Location), units consumed (Unit) and the payer (Payer) have been hard-coded to '13', 'GP', 'Any', 'PRI', 'Visit' and 'PHS', respectively, for the group '3moResource1'. To map data from other time points additional 'HE' objects can be created by simply adding more entries to the mapping element as shown in Figure S4 where health resource-use data can be captured for any number of follow-ups.

2.2.4 Extract, transform and load (ETL)

We developed a bespoke ETL desktop application to efficiently extract, transform and load (ETL) the original trial datasets into the repository. The ETL application permits users to setup new RCTs for import, create and edit classes and attributes, and to switch between testing and live environments, giving users the flexibility and convenience of checking whether or not the instructions that they

```
<class name="HE"">
  <mapping>
    <attributeName fu="13"></attributeName>
  </mapping>

  <childClass name="HE-DATA" linkedValue="13">
    <grouping>
      <groupName>3moResource1</groupName>
      <groupName>3moResource2</groupName>
      <groupName>3moResource3</groupName>
      <groupName>3moResource4</groupName>
      <groupName>3moResource5</groupName>
      <groupName>3moResource6</groupName>
    </grouping>

    <mapping>
      <attributeName originalName="Pri1" groupName="3moResource1">Quantity</attributeName>
      <attributeName originalName="Pri2" groupName="3moResource2">Quantity</attributeName>
      <attributeName originalName="nPriv1" groupName="3moResource3">Quantity</attributeName>
      <attributeName originalName="cPriv1" groupName="3moResource3">Cost</attributeName>
      <attributeName originalName="pmed1" groupName="3moResource4">Type</attributeName>
      <attributeName originalName="pmed1" groupName="3moResource4">Text</attributeName>
      <attributeName originalName="nmed1" groupName="3moResource4">Quantity</attributeName>
      <attributeName originalName="bmed1" groupName="3moResource5">Text</attributeName>
      <attributeName originalName="cmed1" groupName="3moResource5">Cost</attributeName>
      <attributeName originalName="bmed2" groupName="3moResource6">Text</attributeName>
      <attributeName originalName="cmed2" groupName="3moResource6">Cost</attributeName>
    </mapping>

    <transform>
      <staticValue>
        <!- Public healthcare professionals: GP -->
        <newValue attributeName="RP" groupName="3moResource1">13</newValue>
        <newValue attributeName="Type" groupName="3moResource1">GP</newValue>
        <newValue attributeName="Reason" groupName="3moResource1">Any</newValue>
        <newValue attributeName="Location" groupName="3moResource1">PRI</newValue>
        <newValue attributeName="Unit" groupName="3moResource1">Visit</newValue>
        <newValue attributeName="Payer" groupName="3moResource1">PHS</newValue>
          <!- Public healthcare professionals: physiotherapist -->
        <newValue attributeName="RP" groupName="3moResource2">13</newValue>
        <newValue attributeName="Type" groupName="3moResource2">Physio</newValue>
        <newValue attributeName="Reason" groupName="3moResource2">LBP</newValue>
        <newValue attributeName="Location" groupName="3moResource2">COMM</newValue>
        <newValue attributeName="Unit" groupName="3moResource2">Visit</newValue>
        <newValue attributeName="Payer" groupName="3moResource2">PHS</newValue>
          <!- Private healthcare professionals -->
        <newValue attributeName="RP" groupName="3moResource3">13</newValue>
        <newValue attributeName="Type" groupName="3moResource3">Physio</newValue>
        <newValue attributeName="Reason" groupName="3moResource3">LBP</newValue>
        <newValue attributeName="Location" groupName="3moResource3">PTE</newValue>
        <newValue attributeName="Unit" groupName="3moResource3">Visit</newValue>
        <newValue attributeName="Payer" groupName="3moResource3">IND</newValue>
          <!- Medicine prescribed -->
        <newValue attributeName="RP" groupName="3moResource4">13</newValue>
        <newValue attributeName="Reason" groupName="3moResource4">LBP</newValue>
        <newValue attributeName="Unit" groupName="3moResource4">Px</newValue>
        <newValue attributeName="Payer" groupName="3moResource4">PHS</newValue>
          <!- Medicine bought-->
        <newValue attributeName="RP" groupName="3moResource5">13</newValue>
        <newValue attributeName="Type" groupName="3moResource5">Aid</newValue>
        <newValue attributeName="Reason" groupName="3moResource5">LBP</newValue>
        <newValue attributeName="Unit" groupName="3moResource5">Item</newValue>
        <newValue attributeName="Payer" groupName="3moResource5">IND</newValue>

        <newValue attributeName="RP" groupName="3moResource6">13</newValue>
        <newValue attributeName ="Type" groupName="3moResource6">Aid</newValue>
        <newValue attributeName="Reason" groupName="3moResource6">LBP</newValue>
        <newValue attributeName="Unit" groupName="3moResource6">Item</newValue>
        <newValue attributeName="Payer" groupName="3moResource6">IND</newValue>
      </staticValue>

      <match operator="equal" value="Ibuprofen">
        <newValue attributeName="Type" groupName="3moResource4">NSAID</newValue>
      </match>
    </transform>
  </childClass>
</class>
```

Figure S3 XML instructions to map the original healthcare resource-use data to the equivalent repository attributes and to transform the original values to the repository standard.

```
<class name="HE">
 <mapping>
  <attributeName fu="13"/>
  <attributeName fu="26"/>
  ...
  <attributeName fu="n"/>
 </mapping>

 <childClass name="HE-DATA" linkedValue="13">
  <grouping>
   <groupName>3moResource1</groupName>
   ...
   <groupName>3moResourceX</groupName>
  </grouping>
  <mapping>
   <attributeName originalName="Pri1" groupName="3moResource1">Quantity</attributeName>
   ...
   <attributeName originalName="cmedX" groupName="3moResourceX">Cost</attributeName>
  </mapping>
  <transform>
   ...
  </transform>
 </childClass>

 <childClass name="HE-DATA" linkedValue="26">
  <grouping>
   <groupName>6moResource1</groupName>
   ...
   <groupName>6moResourceX</groupName>
  </grouping>
  <mapping>
   <attributeName originalName="Pri1" groupName="6moResource1">Quantity</attributeName>
   ...
   <attributeName originalName="cmedX" groupName="6moResourceX">Cost</attributeName>
  </mapping>
  <transform>
   ...
  </transform>
 </childClass>
...

 <childClass name="HE-DATA" linkedValue="n">
  <grouping>
   <groupName>nmoResource1</groupName>
   ...
   <groupName>nmoResourceX</groupName>
  </grouping>
  <mapping>
   <attributeName originalName="Pri1" groupName="nmoResource1">Quantity</attributeName>
   ...
   <attributeName originalName="cmedX" groupName="nmoResourceX">Cost</attributeName>
  </mapping>
  <transform>
   ...
  </transform>
 </childClass>
</class>
```

Figure S4 Sample of XML instructions to map more than one time point of the original healthcare resource-use data to the equivalent repository attributes and standard.

have delineated in the XML file are correct before loading the datasets into the live database. The bespoke ETL application works by first referencing the file locations of the original dataset and the XML mapping and transformation rules. The instructions defined in the XML file are applied to the original dataset and the transformed data is loaded into the repository.

To ensure the mapping and transformation procedures worked correctly unit tests were performed at a code-level and user acceptance tests were performed on all measurements at each time point for all trials. A random sample (size of at least two) of every possible transformed value was extracted and manually cross checked against the source data. As the rules had already been tested at the code-level this check was to guard against human error and to ensure that the transformed value was submitted as intended by the statistician/health economist. Any inconsistencies were flagged and if required, the mapping and transformation instructions were amended. This process was repeated until all the data were deemed to have been transformed correctly, i.e. zero error.

2.2.5 Using EAV/CR data

Using the EAV/CR data in its raw state for any kind of analysis would be extremely difficult due to its fragmented structure. For analysis purposes the data was transformed to create a dataset that is comparable to those typically extracted from relational or flat file tabular data sources. This task is achieved by extracting the EAV's attribute/value pairs for any given class into a derived table that is then processed using a SQL pivot command to create a column for each attribute and a row for every object together with the trial name and the subject ID. The result of the query is a dataset that resembles a long format tabular structure that can easily be processed for further analysis.

References

Marenco, L., Tosches, N., Crasto, C., Shepherd, G., Miller, P. L., and Nadkarni, P. M. (2003). Achieving Evolvable Web-Database Bioscience Applications Using the EAV/CR Framework: Recent Advances. *Journal of the American Medical Informatics Association* **10**, 444-453.

Nadkarni, P. M., Marenco, L., Chen, R., Skoufos, E., Shepherd, G., and Miller, P. (1999). Organization of Heterogeneous Scientific Data Using the EAV/CR Representation. *Journal of the American Medical Informatics Association* **6**, 478-493.

XML Technology (2010). World Wide Web Consortium (W3C). www.w3.org/standards/xml/ (2010, accessed 8 January 2015).