Modelling causes for actions with the Decision and PROV ontologies

N.J. Car

ᵃ Geoscience Australia, Symonston ACT 2609
Email: nicholas.car@ga.gov.au

Abstract: Provenance modelling systems such as PROV and its precursors, PML and OPM do not have specific mechanics to deal with causes for actions. They can represent what information was used within processes and who did what, when, but not why or what rules they followed. However, modelling human decision making to represent *why* has long been of interest to philosophers, politicians, mathematicians and, more recently, computer scientists and multiple decision modelling systems have been implemented over time. Aristotle in ancient Greece, codified representations of decision logic. The Marquis de Condorcet in 18th century France, proposed fair voting systems for elections. John Dewey in the early 20th centaury described logical step-wise decision-making processes. Computer models for aspects of decisions have existed as long as computers and recently, several standards groups have implementing specifications about how to record decisions such that knowledge of them may be shared and understood by others.

In this paper, we first demonstrate conceptually mapping part of a decision representation system, the Decision Ontology (DO), to the ontology version of PROV, PROV-O. We indicate that a complete mapping is not possible due to the DO modelling templates for possible future decision making and PROV-O only dealing with past actions such as decisions already made.

We demonstrate our DO/PROV-O mapping by first modelling decisions using DO and incrementally re-modelling them as PROV-O. We do this twice for two very different decisions; one from DO's documentation and another we introduce. This exercise shows that some decisions can indeed be modelled in depth using DO and that some aspects of them can be well understood using general PROV-O provenance terms.

Next, we present an overview of our DO to PROV-O mapping and introduce a stand-alone ontology, DecPROV (http://promsns.org/def/decprov) which is a specialisation of PROV-O that captures the elements of DO we can map to it. We have chosen to implement a new ontology rather than creating a new version of DO due to the incompleteness of the mapping.

In addition to showing that certain decisions can be modelled in DO and DecPROV and thus PROV-O, we describe several common provenance scenarios and indicate causes for elements within them using a few DecPROV elements.

We conclude with a few thoughts about some of the aspects of decision modelling that the exercises related here have unveiled.

Keywords: *decision theory, decision ontology, provenance, provenance ontology, ontology alignment*

1. INTRODUCTION

PROV-O (Lebo, Sahoo, and McGuinness 2013) is the Web Ontology Language (OWL) (W3C OWL Working Group 2012) version of the PROV Data Model and is the only provenance representation system recognised by an international standards body – the Wold Wide Web Consortium. It is a generic process model that, at its most basic, uses three classes, Activity, Agent and Entity to represent steps in processes, the causative actors for them (human or machine) and the inputs and outputs (real or conceptual) of those processes. PROV-O does not contain specific classes to describe motivations for a particular course of action although this may be inferred by a representation of a process that includes instructions in Entities that were consumed by Actors who performed it. PROV-O can also only represent things that have been: it has no way of creating templates for prospective actions like many workflow models do, thus it cannot validate a provenance record according to a desired plan. A recent addition to PROV attempts to allow for "provenance templating" (L. Moreau et al. 2017).

The Decision Ontology (DO) (Nowara 2012), also formulated using OWL, was created "for describing [human] decisions and decision-making". It was published by a W3C Incubator Group in 2012 in order to be considered as an input to a process which may have resulted in a *Recommendations* (a standard) but this process was never completed. The DO contains classes and relationships that it claims can represent decisions made – *data-driven* use – and can outline decision-making scenarios – *normative* use. Combining the two uses, allows validating actual decisions against predicted decisions. In *data-drive* use, Question, Requirement and Option classes, among others, can indicate *why* a particular decision outcome was selected. In *normative* use, those classes can be used to make explicit *why* some decision should be made or at least what outcome from a decision should be expected. The DO presents elements for describing decision requirements – the information flows and processes relevant to a decision-making scenario and some elements for describing decision logic – the rules used to select one option over another, however it leaves logic description deliberately open-ended which means decision logic descriptions can be powerful – anything you like – but difficult to implement and compare. The DO is dependent on a Requirements ontology (RO) by the same author for some logic modelling but this and the DO are treated as one in this paper due to their interwoven use and the RO's small size.

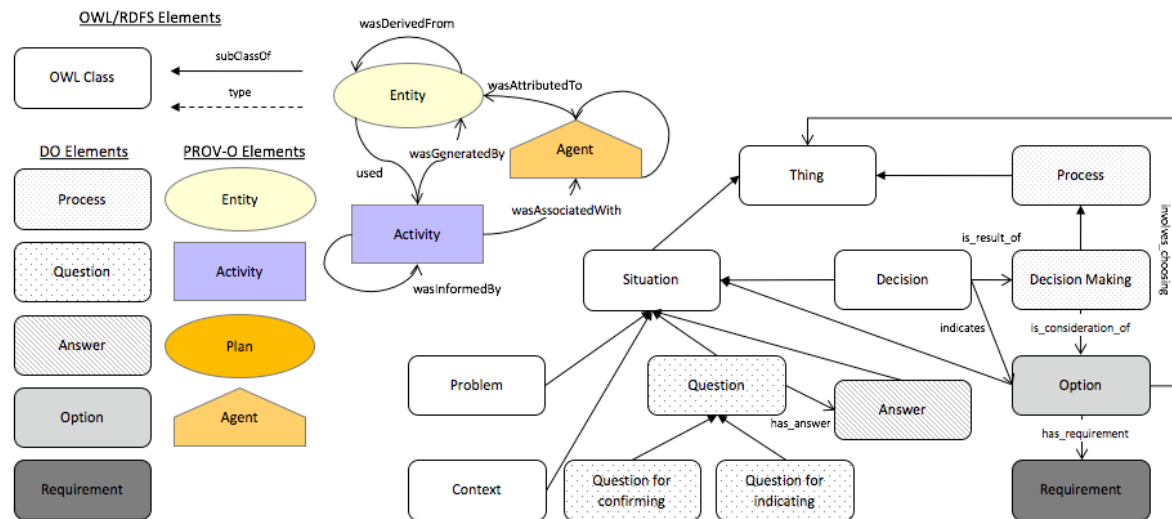Figure 1 is a key of figure elements used here; Figure 2 shows a class diagrams for PROV-O (A) and DO (B).

Figure 1: Element key for all figures   **Figure 2**: Basic classes and relationships of PROV-O (A) & DO (B)

The DO isn't the only decision modelling system; the Decision Modelling Notation (DMN) (Object Management Group 2016) and the Decision-Making Ontology (DMO) (Kornyshova and Deneckère 2010), among others, also exist. DMN is UML-based, simple, mature and very widely used within business process modelling. DMO is also UML-based and uses recent decision theory to justify its element. It is made for the information systems engineering domain but it shows no evidence of real use and lacks proper documentation.

While we recognise that many decision modelling systems could be used with PROV-O, the DO is the only OWL-based system we could find. It is thus the one most able to be directly mapped to PROV-O on a technical level. It would be possible to conceptually map DMN or others to PROV-DM and even map serialisations of DMN in XML to PROV-XML (the XML serialisation of PROV-DM) but we have chosen, in this exercise, to concentrate on Semantic Web models (Berners-Lee, Hendler, and Lassila 2001) using Linked Data (Heath and Bizer 2011) mechanics only due to our assessment of their greater future use potential.

2.   OUTLINE OF THE DECISION ONTOLOGY

The DO models decision-making as a process within which situations are encountered and data elements considered. DO Options are encountered and indicate, perhaps based on Requirements, the selection of an OWL Thing; something requiring more specification on a per-decision basis. It could be an object or the starting of a new process, including another decision-making process.

Produced before PROV-O, DO has no concept of PROV-O's general process representation, however due to the generic nature of PROV-O, some DO elements can be mapped to it, for instance a past instance of a DO Decision_Making, being a process, can be represented as a PROV-O Activity.

3.   SCENARIO 1 – DO DOCUMENTATION EXAMPLE MAPPED TO PROV-O
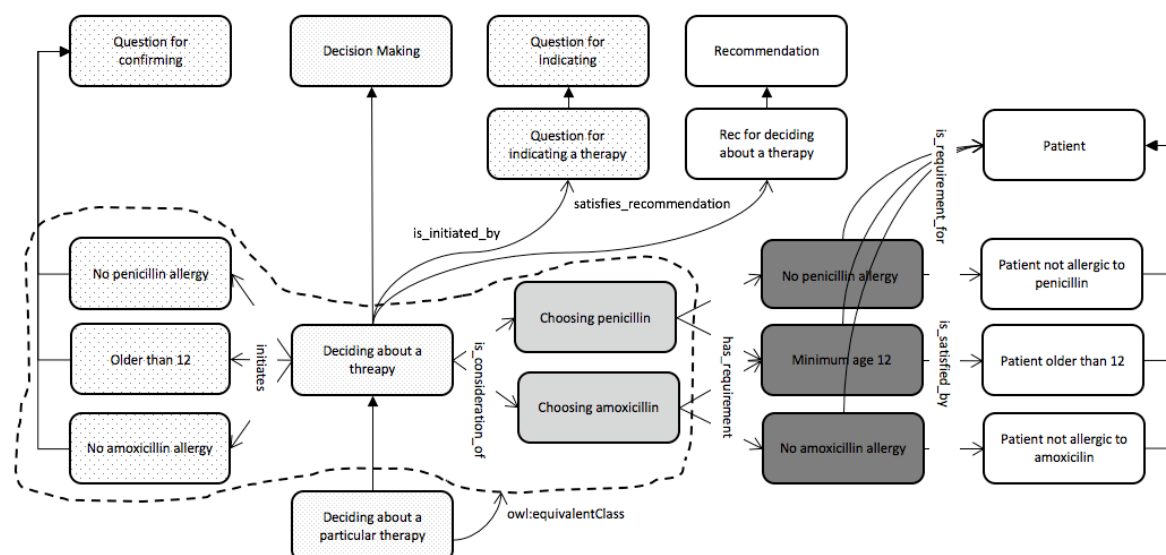

Figure 3: Diagram of the DO example "Bacterial throat infection" from (Nowara 2012)

The DO documentation supplies a single example of use and it demonstrates *normative* use. Here we convert that example to a *data-driven* use scenario by adding decision outcome and phrasing it as if it had taken place. Figure 3 shows a diagram of that DO example, a reading of which is that a decision maker (unrepresented) enters a Process to decide about a therapy which is_initiated_by a Question for indicating. The outcome of this Process is a specific Recommendation to do something. The Options considered by the Decision Maker (only two of the original example's three are shown) are Choosing penicillin and Choosing amoxicillin. These Options have various Requirements, some of which multiple Options share. Requirements are satisfied by the presence of objects with certain properties. In this example, these objects required are classes of types of Patient.

Figure 4 shows the core of Figure 3 with additions, using DO constructs, to arrive at a *data-driven* use scenario. Various elements in Figure 3 are not presented in Figure 4 such as descriptive superclasses of core classes and the Questions for confirming that the Deciding about a therapy process initiates. The Question classes are removed because the DO example does not link them to Option's actions, despite obvious associations seen on inspection due to labels. The second scenario tested here (see Section 4) will explore the use of Questions for confirming.

Added in Figure 4 in blue is Answer Y indicating an instance of the Therapy class, Amoxicillin. The Question for indicating a therapy required that an instance of Therapy ultimately be indicated by an Answer instance, so this outcome is legal. Another Requirement for the Option, Choosing amoxicillin, is also added (not present in the original example) to demonstrate Requirements requiring satisfaction by objects in disjoint classes, in this case Patients who are either allergic or not allergic to penicillin. The instance Patient X is added to the diagram which satisfies the Requirements for Choosing amoxicillin but not those for Choosing penicillin and the wording of the main Process changed to Deciding about a particular therapy for Patient X to indicate a specific situation is being considered, not a generic one as per *normative* use. To indicate which Option has resulted in an outcome, the involves_choosing of the Choosing penicillin Option (given in the ontology example but not shown in Figure 3) is removed but the Option choosing Amoxicillin is added. Finally, in red, a Decision Maker Z has been added to indicate agency with relationships to the Decision Process and the Answer Y.
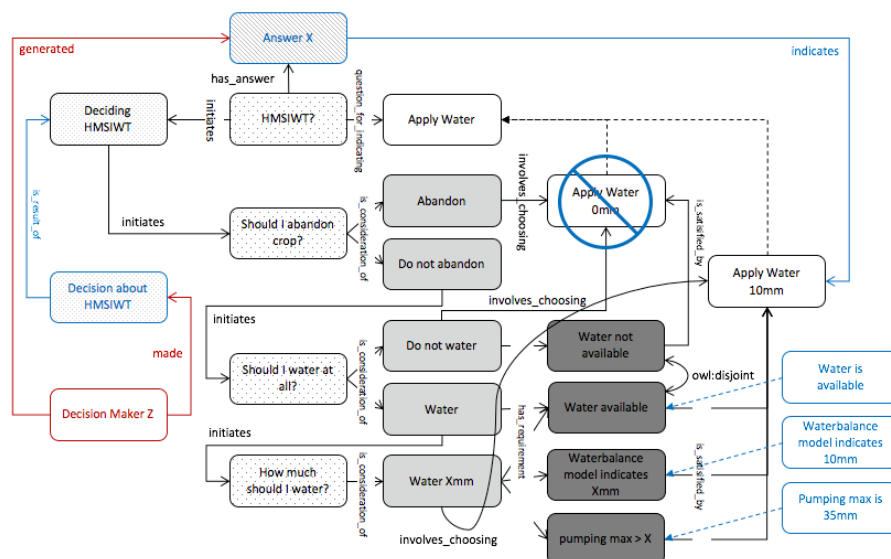
The general outcome from Figures 4 & 5 is that *data-driven* use instances of the DO can be generated from *normative* use instances of DO if Named Individuals are added but that potential outcomes must also be removed to indicate a final outcome.

Figure 4: Core elements of Figure 3 with Question/Answer linking (not in original example), an addition of Patient X & rewording of the Deciding shown in blue and a new Requirement and requirement satisfaction object addition shown in orange. The non-DO Decision Maker object and relationships added are indicated in red

Figure 5 maps the *data-driven* aspects of Figure 4 to PROV-O basic classes (Activity, Entity Agent and properties) (Lebo, Sahoo, and McGuinness 2013), laid out generally as per Figure 4 to indicate alignment. A complete mapping of DO classes to PROV-O is given in Table 1. Property mapping can be partially understood by comparing Figures 4 & 5 but for full property mappings, more detailed class mapping and the full DecPROV class model, see its ontology documentation (http://promsns.org/def/decprov).

Figure 5: A PROV-O class and property-only representation of Figure 4

One outcome from Figures 4 & 5 is that *data-driven* use instances of DO can be mapped to PROV-O. Another, not shown but derivable from Figure 5, is that instances of DO's *data-driven* use can be unambiguously represented using PROV-O properties with no need for DO properties if DO classes are used. A third outcome is that the PROV-O process model cannot represent outcomes that were not selected (the objects from Options indicated via involves_choosing) even though DO's *data-driven* use can. Indicating such un-selected outcomes using PROV-O is shown in Section 6.

Figure 6: A model of the question "How much should I water today?" (HMSIWT). *Data-driven* elements in blue, non-DO elements in red.

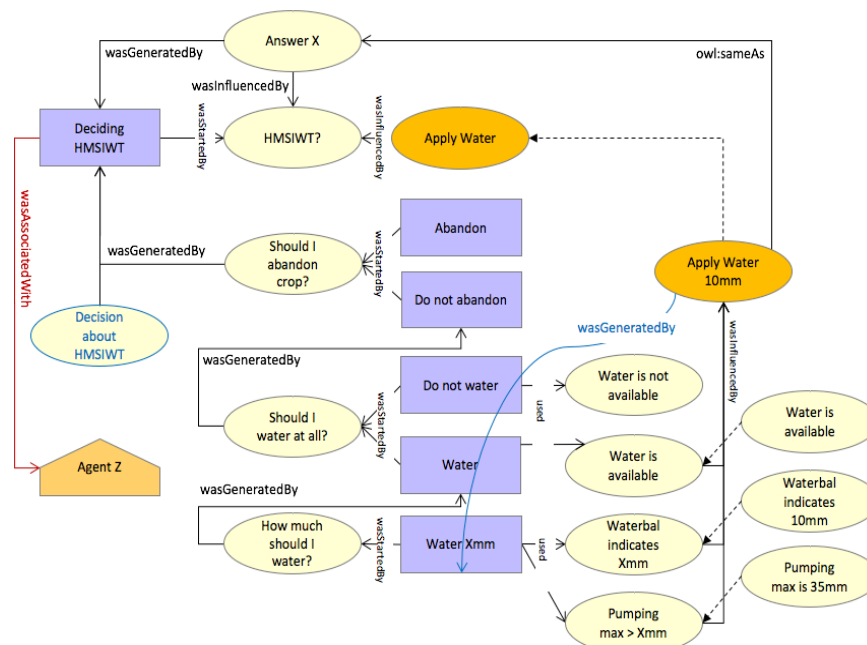**Figure 7**: A PROV-O class and property-only representation of Figure 7

4.   SCENARIO 2 – A NEW DO EXAMPLE MAPPED TO PROV-O

Using DO elements we now partially model "How much should I water today?" a question irrigation farmers face. This is in order to demonstrate both a second instance of DO/PROV-O mappings generally and also to show some other elements of DO use. Figure 6 shows the modelling with the question object HMSIWT?, initiating a Process, Deciding HMSIWT, which then initiates serial sub-questions. What is not shown here are Requirements for all the Options or the presentation of facts that satisfy all the Requirements. These are shown in the DecPROV documentation online. Figure 7 shows the mapping to PROV-O.

This is a *normative* use scenario but one that is inherently process flow-like. Indicated in blue are DO elements that can be used to convert this to a *data-driven* use scenario: instances of Answer and Decision classes and the indication of only one Option having selected an instance of an Apply Water class instance. As per Figure 6, a non-DO Decision Maker object and relationships, in red, have been added to introduce agency.

This example shows a Question for indicating (HMSIWT?) triggering sub-questions (Questions for conforming) and thus decision scenario decomposition. DO/PROV-O modelling shown in Figures 4 & 5 is confirmed here. Note here we see the expected outcome class Apply Water is a PROV-O Plan, to be interpreted as a call to an action which is not shown (the Agent actually applying water). The specific outcome is to water 10mm with 10mm shown to have been determined by a waterbalance model whose processes lie outside this modelling.

5.   SPECIALISING PROV-O FOR DECPROV

In both Figures 6 and 8, the representations of the decisions modelled probably contain few surprises for those familiar with PROV-O. The use of basic PROV-O classes with Plan being the only one outside the 3 simplest and the use of unqualified PROV-O class relations. For those familiar with DO, probably the only surprise is the interpretation of the Option class as a PROV-O Activity, implying a process with temporality.

DecPROV adopts the classes of DO listed in Table 1 and their associations with PROV-O with the single change of renaming Option to Option Selection to reinforce its temporal, process, nature. Such an interpretation allows Options to be thought of as Option Selection *processes* which consume Requirements and may or may not produce results, the process which is termed by DO as involves_choosing. DO Questions (PROV-O Entities) can then trigger multiple Option Selections only one of which, when in *data-driven* use, will be able to produce a result which either triggers further Questions (as per Figure 7) or concludes the Decision Making process with an object of the class indicated by the Answer to the Question for indicating that initiated the Decision Making. This requirement that only one of multiple Option Selection process triggered by a Question produce a result and the requirement for two or more Option Selection processes for any given triggering Question are axioms of DecPROV. These axioms could be used by a future version of the DO for strong *normative* use validation of *data-driven* use instances if circumstances describing how instances of the classes of object indicated by the Requirements of Option Selections are realised or not realised (in DO terms, satisfied).

DecPROV does not use any DO properties for such like has_requirement associating an Option with a Requirement or is_consideration_of associating a Question with Options when PROV-O is used with DecPROV classes. This is because PROV-O basic properties, used with the PROV-O superclasses of the DecPROV classes, deterministically model relationships between DecPROV classes thus it is redundant to specialise PROV-O properties further for DecPROV-specific properties. Specialisation by necessity also diminishes general understanding so it is to be avoided where redundant.

Table 1 Classes of DO and their PROV-O classes to which they are associate by being subclasses of them

DO Class	PROV-O Superclass	Note
Answer	Entity	Answer is more naturally a thing (imaginary thing), thus an Entity, rather than a subclass of Situation, as originally portrayed in DO.
Context	-	Not used: all scenarios here can be represented without Context
Decision	-	Not used: functions of Decision are subsumed by question, Answer and Decision_Making
Decision_ Making	Activity	Decision_Making is a time-bounded process, naturally an Activity. Renamed DecisionMaking
Option	Activity	Option is more naturally a process than a subclass of Situation – something to be statically encountered – as originally portrayed in DO. Renamed Option Selection
Normative Value	Entity	NVs, are documented in the DO (actually its foundational Requirements Ontology) are meant to indicate norms but the DO only demonstrates the use of Requirements. Requirements are used by Option processes to determine decision outcomes thus are either an Entity or perhaps a Plan
Process	Activity	A generic Process is a generic Activity. DecPROV uses only Activity.
Question	Entity	Like Answer, more naturally an imaginary thing
Requirement	Plan	An instruction to an Option process thus a PROV-O Plan
Situation	-	Not used: PROV-O does not represent situations as it has no state machine-like functionality: everything represented in PROV-O instances occurred in the past.

6. PROV-O SCENARIOS AND CUASATION ANSWERED WITH DO

Figure 8 shows three pure PROV-O scenarios, A. B. & C where causes for actions are posed. In scenario A, *why* (as opposed to *how*) particular Output Data is generated by a Processed Data Activity which consumed Input Data and Software Code is posed. This is answered by adding a DecPROV Question instance, Question X, which influenced the existence of Output Data and was the trigger for starting the Activity that generated it. Detailed modelling of how Question X was resolved to result in Output Data could then be modelled as per Figures 5 & 7. In scenario B, establishing *why* Input Data, as used in scenario A, was selected for use as opposed to Potential Input Data Unused requires a set of DecPROV Option Selection Activities to have taken place, one of which must have resulted in the generation of Input Data and none of which generated Potential Input Data Unused. The logic used to determine Input Data's generation and no other object's is contained within the Requirement instances used by the Option Selections. While the set of objects unused by Process Data is infinite and thus a list of instances of them cannot be related in a provenance scenario, a particular object, such as Potential Input Data Unused, may be easily assessed to have been a possible outcome of a parallel Option Selection to the one that ultimately generated the Input Data. The corollary of this is that the total set of Requirement instance must have excluded it or preferred another to its use else it would have been used.

Scenario C of Figure 8 shows that an instance of an Agent, Nick, undertook an Activity, Took out the rubbish, and *why* did he did so is asked (perhaps in relation to him possibly doing something else. Here a DecPROV Question is naïvely linked to the Activity and indicated as having started it, as this may seem appropriate: a question triggers an action. However, on reflection, this doesn't make sense: Nick must have chosen to undertake the Activity Took out the rubbish in a decision-making process before undertaking it. Thus, the answer is modelled as a set of Option Selection processes, one of which resulted in the PROV-O Plan to undertake Took out the rubbish.
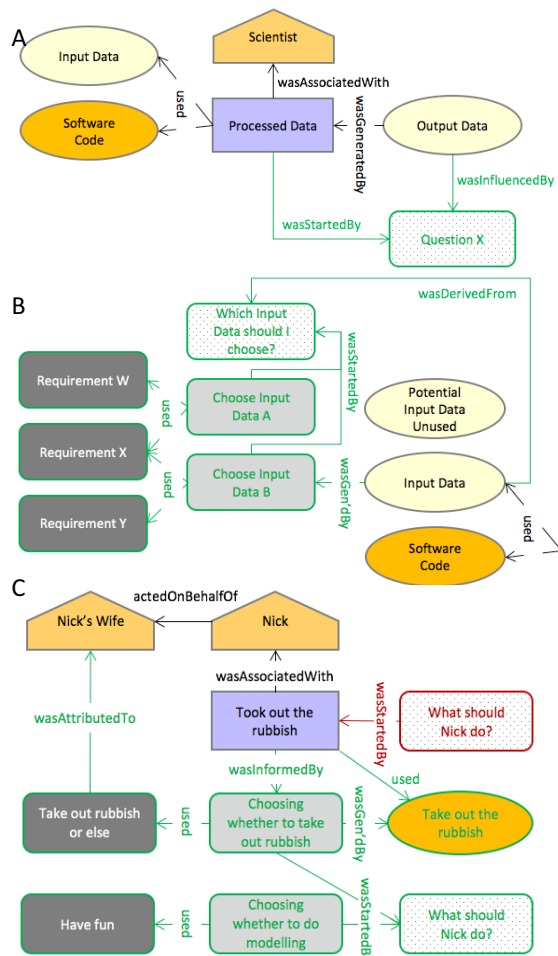
Figure 8: *Why* questions for PROV-O scenarios answered using DecPROV (green outline).

The lack of outcome from the Choosing whether to do modelling Option Selection indicates that Activity was a possibility for Nick but that the total set of Requirements for the set of Option Selection processes eliminated it.

7. CONCLUSION

All elements of DO's *data-driven* use can be modelled in PROV-O and some DO classes have been mapped to PROV-O and presented in the DecPROV ontology but no properties have been as they are semantically redundant. DecPROV can both model past decisions in detail and add information about causes for some decisions modelled in PROV to provenance information.

In detailed decision modelling, describing what decision was made is easy enough, but describing what options were presented to a decision maker and then not chose is hard. The best approach with PROV-O and DecPROV seems to be to model the sets of possibilities a decision maker had before them – the Requirements classes – and then to model what they were ultimately faced – the instances of objects within the Requirements classes.

Where initial questions lead to sub-questions, as per Figure 7, the construct of DecisionMaking could be applied to each sub-question. This would allow surfacing sub-questions as whole decisions in themselves which might be useful in collections of decisions for the same sub-questions might be present in multiple, different, questions and knowing this could be useful to establish decision similarity.

In the scenarios here, the decisions (Questions for confirming and accompanying Option Selections) always result in one outcome only. Using DMN terms, this would be called a *one-hit* scenario where decision logic. DMN and other decision modelling tools also allow for multi-hit scenarios and additions to DecPROV could allow that within it too.

One aspect of DecPROV is its ability to model decisions simply (Question, DecisionMaking & Answer) or more complexly (the classes listed before plus Option Selections, Requirements and sub-questions). This allows for stated but unexplained causes or finely detailed causes for actions to be modelled giving users great choice.

REFERENCES

Berners-Lee, T, J Hendler, and O Lassila. 2001. "The Semantic Web." *Scientific American* May: 34–43.

Heath, Tom, and Christian Bizer. 2011. *Linked Data: Evolving the Web into a Global Data Space*. Edited by James Hendler. Morgan & Claypool. http://linkeddatabook.com.

Kornyshova, Elena, and Rébecca Deneckère. 2010. "Decision-Making Ontology for Information System Engineering." In *Conceptual Modeling – ER 2010*, 104–17. Lecture Notes in Computer Science. Vancouver, BC, Canada: Springer. doi:10.1007/978-3-642-16373-9_8.

Lebo, Timothy, Satya Sahoo, and Deborah McGuinness. 2013. "PROV-O: The PROV Ontology." W3C Recommendation. April 30. https://www.w3.org/TR/prov-o/.

Moreau, L., B. Batlajery, T. D. Huynh, D. Michaelides, and H. Packer. 2017. "A Templating System to Generate Provenance." *IEEE Transactions on Software Engineering* PP (99): 1–1. doi:10.1109/TSE.2017.2659745.

Nowara, Piotr. 2012. *Decision Ontology* (version 1.0). OWL ontology. Katowice, Poland. http://promsns.org/def/do.

Object Management Group. 2016. "Decision Model and Notation (DMN)." formal/2016-06-01. Object Management Group. http://www.omg.org/spec/DMN/1.1.

W3C OWL Working Group. 2012. "OWL 2 Web Ontology Language Document Overview (Second Edition)." http://www.w3.org/TR/owl2-overview/.