A semantic Lab Notebook – Report on a Use Case Modelling an Experiment of a Microwave-based Quarantine Method

Nico Adams, Armin Haller, Alexander Krumpholz, Kerry Taylor

CSIRO, `firstname.lastname@csiro.au`

Abstract. A recent trend in a number of academic disciplines is the publication of results of experiments together with the scientific article for a better reproducibility of the published experiments and algorithms. Semantic Web technologies have the potential to aid scientists in the publishing, sharing and interlinking of this data and also in helping other scientists in the understanding of the data and the interpretation of the results of an experiment. In this paper we report on a use case on how to publish the data captured in a scientific experiment that has been conducted in the CSIRO Animal, Food and Health Sciences division as a set of ontologies and how to access this data through a set of RESTful semantic Web services. These services showcase how computational tasks that cannot be represented in the ontology can be implemented as lightweight semantic Web services to document and verify the results of an experiment. Together, the ontologies, the experimental data and the computational services constitute the elements needed for a semantically enabled lab notebook, facilitating research studies over multiple experiments, while reducing complexity and error rates.

1 Introduction

Research into different types of thermal treatments as quarantine methods against codling moth in a variety of fruit has gained much interest in recent years due to the uncertain future of chemical fumigation of food and the public concern over residues in treated products [?]. One such new thermal treatment method is currently under development in the CSIRO Animal, Food and Health Sciences division. The method proposes to use a combination of thermal treatment with microwave treatment of fruit to inactivate the fruit fly larvae from growth in different types of fruit. The experiment conducted by domain scientists in CSIRO is using a custom-built microwave-based heat treatment system that is tested on a number of different fruit for the inactivation of an induced codling moth infestation in these fruit.

Apart from the main goal of the research in establishing the effectiveness of the microwave-based heat treatment process for the inactivation of codling moth, a secondary goal of a transformational capability platform project was to showcase how the experimental data can be modelled in semantic Web languages and how the resulting ontological models can be used to support computational analysis of the experimental data. To achieve the latter, a group of ontologists and software engineers have accompanied the domain scientists during the experiment and defined models to capture the experimental data semantically. We present our methodology of how to publish the results of a scientific experiment as a set of ontologies. Further, we develop a set of services that showcase how computational tasks that cannot be represented in the ontologies themselves can be modelled and implemented as lightweight semantic Web services.

The remainder of this paper is structured as follows. In Sect. **??** we briefly describe the microwave-based heat treatment process that constitutes our use case. In Sect. **??** we describe the ontologies that are needed to capture this use case. In Sect. **??** we describe the information services that we have built on top of the ontological data representing the knowledge gathered in the use case experiment. We discuss some related work in Sect. **??**, before we conclude in Sect. **??**.

2 Microwave-based heat treatment

Our use case has been provided by scientists in the CSIRO Animal, Food and Health Sciences (CAFHS) who are developing a microwave-based heat treatment system for the treatment of fruit for the purposes of removing fruit fly larvae infestations. Briefly, for the experiment a purpose-built microwave tunnel system was built. The microwave unit also incorporates an auxiliary hot air system comprised of a heater and a fan which is also attached to the microwave system.

For the experiment that we modelled ontologically, newly harvested organically grown Mutsu and Granny Smith apples were used for treatment in this system. The apples were uniformly infested with fruit flies by making 50 pin holes on each apple at the stem end and then placed inside cages containing fruit flies.

The microwave treatment was applied by placing the apples on a small plastic stand with four protruding rods and sent through the microwave tunnel for approximately 54 min, which was preheated to $63 - 65°C$. A variable speed conveyor belt moves the fruit through the microwave tunnel where they undergo heating by microwaving to destroy fruit fly larvae and eggs. The temperature of the fruits at different points (top, flesh, core, bottom) were measured during the experiments with a fibre optic conditioner at 1 second intervals.

The goal of the experiment was to determine the optimal configuration of the tunnel temperature, the microwaving intensity and the time of treatment in each of the stages of the treatment process to obtain 100% mortality of the fruit fly larvae and eggs that is comparable to traditional thermal treatment methods. This can be done by calculating the cumulative thermal effect for a given treatment, based on kinetic data for the thermal mortality of target insects and for product quality losses. Given a time-temperature history of $T(t)$, the cumulative thermal mortality of the microwave-based heat treatment can be calculated to an equivalent length of time in minutes, M_{52}, at a reference temperature T^{ref} of $52°C$ by using the following relationship:

$$M_{52} = \int_0^t 10^{\frac{T(t)-52°C}{z}} dt$$

where M_{52} is the equivalent time at a target temperature of $52°C$, $T(t)$ is the transient temperature profile measured by the fibre optic system, t is the time and z is the temperature change (in $°C$) required to change the value of insect mortality (lethality) by a factor of 10.

3 A semantic lab notebook

To unambiguously record the data (e.g. temperature measurements, applied power, belt speed etc.) captured in the experiment and to allow a computational analysis of the experimental data, we first need a conceptual stratification of the experiment and a common understanding of the objects and processes that are used in the experiment. In the context of this project, a fruit treatment process use-case acts as an exemplar to build up a demonstrator of a semantics driven lab notebook.

When describing experiments in electronic lab notebooks the terms used are often ambiguous. In our use case experiment, for example, the term “fruit” if used in some electronic record, may be ambiguous depending on the experiment run, as there were multiple experiments conducted, not only with Mutsu and Granny Smith apples, but also with mangoes and avocados. Even more, many of these terms exhibit polysemy: avocado, in common usage - and therefore also when used as a metadata term - may, without further specification, refer to either the “avocado fruit” or the “avocado tree”. Such distinctions are important in that they (a) determine the scope of what we can talk about in our information systems and (b) also specify - at least to a degree what sort of data is to be collected. Completely defining the meaning of something allows the specification of the relationships between entities: an “avocado fruit”, for example, is part of an “avocado tree” (at least until it has been harvested). Such a disambiguation will then, for example, allow us to talk about properties of a specific fruit (e.g. volume, firmness) and its history and provenance (this “avocado fruit” was part of an “avocado tree” which was located in a “field” which is described by “geo-coordinates” X and Y etc.). Without disambiguating the polysemous term “avocado” it would have been impossible to represent information about an avocado in such terms.

The precise definition of objects and their relationships can also help to overcome the stratification in conceptual models of the treatment system, i.e. its factory: an apple has a digital representation denoting the apple in an information system, i.e. in the electronic lab notebook – for the purposes of this description we will call it a “digital apple”. The “digital apple” is described by some “apple description”, which, in turn is a kind of “information content entity”. An “information content entity”, in turn, may be an input into a “model”, for example, a model describing the relationship between the apple volume and the required heating intensity to reach a certain core temperature in the apple.

In summary, a semantic lab notebook is an exercise in object management. For the purposes of the rest of this discussion, the term “object” denotes any entity that can be named or addressed. Objects may therefore be physical objects as well as data objects, computational service objects etc..

We have chosen to model these objects with the languages developed in the technology stack of the semantic Web [**?**]. With RDF(s) and OWL, the central components of the semantic Web stack, it is possible to attach a formal, i.e. “computable” representation of a conceptualisation of the nature of the object to the object itself. Such statements can then be evaluated by reasoners which can draw inferences over the knowledge provided. We have developed a set of ontologies for representing objects in the context of this experiment and a more general manufacturing processing model which we detail in the following sections.

3.1 Ontology stack

To develop the ontology for the use case outlined above, we used a modified and shortened version of the method described by Uschold and King [?]. For the purposes of ontology development we used documentation containing domain-specific terminology and data as elicited from our colleagues at CAFHS. Fig. **??** shows the stack of ontologies we reused and developed within this project. The figure also includes example classes that are defined within each of these ontologies, whereas the dashes denote the subsumption relations between the classes. In the following sections we describe the classes and relations in these ontologies in more detail.

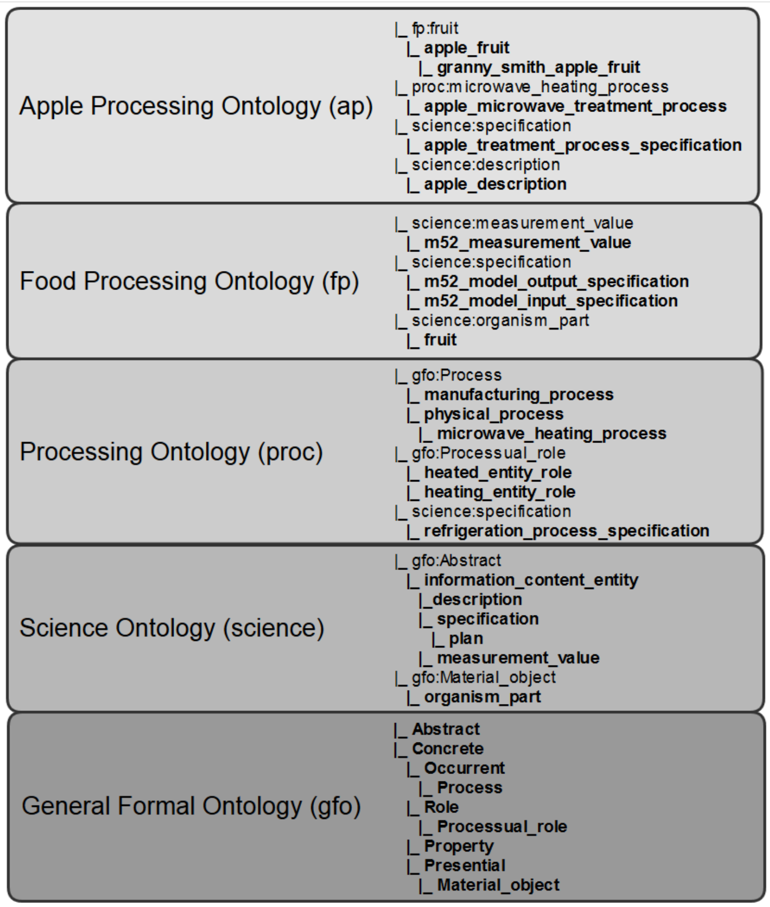

Fig. 1. Ontology Stack

Upper Level Ontology – GFO Upper– or "top level" or "foundational" –ontologies are ontologies of the most common entities in the world which are the same across all knowledge domains. For example, an upper ontology will provide a notion of what a material entity is, how material entities participate in processes and persist in time. The main purpose of an upper ontology is to facilitate semantic interoperability. A number of upper level ontologies are in use across the semantic Web community, though many Web ontologies are developed without referencing top level ontologies, often reducing the level of interoperability. Some of the most common ontologies currently in use across the semantic Web are DOLCE [?], the General Formal Ontology (GFO) [?], the Basic Formal Ontology (BFO) [?] and Cyc [?]. For the purposes of the work in this project, the General Formal Ontology was chosen, mainly for two reasons, (1) its well developed integration of objects and processes and (2) its well developed notions of time. Specifically, the GFO makes an explicit distinction between endurants (objects) and perdurants (processes) and provides convenient mechanisms for modelling how objects participate in processes. Time is taken to be primitive and time points (known as "time boundaries") can be derived. These time points can coincide which is useful for the modelling of continuous processes and change.

Science Ontology The Science Ontology [?] is a small ontology of terms which are common across all of physical science and engineering and resides underneath the General Formal Ontology. Typical terms contained in the ontology are "Information Content Entity", "Description", "Specification" including appropriate subterms. These are important for the disambiguation of the actual processes from process specifications, such as processing conditions etc..

Information Content Entities The most relevant concept that we reuse from the Science Ontology is that of an "Information Content Entity" (ICE) that is required to capture data, specifications and descriptions. ICEs are best described as entities that do not have independent existence, but rather are dependent on other entities and are in an "about-ness" relationship with those entities [?]. An ontological analysis of ICEs would conclude, that within the framework of the General Formal Ontology, these are subclasses of the $gfo{:}Abstract$ class. Abstract entities are entities which are independent from time and space, but may be dependent on other entities for their existence. Subclasses of information content entities that we reuse are, for example, "description", "measurement value" and "specification".

While perdurants such as processes may map onto a time vector, the entity that we observe when we measure time, for example, is not the time vector itself, but a representation, or in better terms, a descriptions of the time vector. In our ontology, processes therefore have "process specifications" which are in an aboutness relationship to the process itself – process specifications are types (subclasses) of descriptions, which, in turn, are information content entities. We may write:

1. $science{:}process\_specification\ subclassOf\ specification$
2. $science{:}process\_specification\ equivalentClassOf$
   $(science{:}specification\ and\ (about\ only\ gfo{:}Process))$

Processing Ontology The microwave-based heat treatment process conducted in our use case is some kind of a food treatment process. Consequently, we need an ontology that defines concepts and relations of food treatment processes. However, to the best of our knowledge no such food treatment ontology exists. Defining such an ontology requires us to properly layer it on top of an Upper Level Ontology such as GFO. To do that we need to identify what constitutes a food treatment process and what are the more general concepts and relations that are needed to describe such a process. Looking at the concept of a food treatment process, it obviously involves some kind of "treatment process" that is performed on "food". A "treatment process" is performed either by a "human" or a "machine". The GFO makes a fundamental distinction between "Processes" and "Actions": ontologically, both are viewed as being of type "Occurrent" but are distinguished from each other through the involvement of an "Agent", i.e. an entity, playing an agent role. Agent roles can be played by both humans and machines. The entity that is treated during the treatment process is at its most general a gfo:*Material_object*. "Food" is – ontologically speaking – a role: a material object "becomes food" when it realises the food role (an apple sitting on a shelf, for example, is not "food" as it does not realise the food role).

From this very brief ontological analysis it becomes clear that we first need an ontology that describes objects in a factory, the roles these objects play as well as the processes in a factory and the mode of participation in those processes. A number of such manufacturing/processing ontologies exist already, however, they are either not layered on top of an Upper Level Ontology [**?**,**?**] or do not provide the detail that we require from the processing ontology to model our use case [**?**]. We therefore developed a general purpose processing ontology, drawing inspiration from the referenced manufacturing ontologies as well as the United States patent and trademark offices taxonomy on manufacturing. As depicted in Fig. 1 the processing ontology, denoted by the namespace prefix "proc" in Fig. 1, defines, for example, different types of manufacturing processes, physical, chemical and biological processes.

Processes in the processing ontology are layered on the notion of processes in GFO which are characterised by the manner in which entities participate in them, i.e.:

1. gfo:*Process* $subclassOf$ gfo:*Occurrent*
2. gfo:*Process* $subclassOf$ (gfo:*has_role some* gfo:*Processual_role*)
3. gfo:*Processual_role* $subclassOf$ (gfo:*role_of some* gfo:*Process*)

We defined a set of such roles that are common in manufacturing processes and that are played by material objects, such as a "heated_entity_role" and a "heating_entity_role". We may write:

1. *proc*:*heating_process* $subclassOf$ gfo:*Process*
2. *proc*:*heating_process* $subclassOf$ (gfo:*has_role some proc*:*heated_entity_role*)
3. *proc*:*heating_process* $subclassOf$ (gfo:*has_role some proc*:*heating_entity_role*)
4. *proc*:*heated_entity_role* $subclassOf$ (gfo:*role_of some* gfo:*Process*)
5. *proc*:*heating_entity_role* $subclassOf$ (gfo:*role_of some* gfo:*Process*)

Further, we defined "chemical_material_objects", "physical_material_objects" and "biological_material_objects" that manufacturing processes take as input or produce as output such as a "machine", an "assembly_entity" and different types of "substances".

Processes can have other processes as part which allows the modeling of complex manufacturing processes and their breakdown into small process parts. Any other process can be modeled by analogy. Processes in our ontology have time boundaries with discrete start and end timepoints. The time boundaries for processes are mapped to the notion of “Chronoids” in GFO. Time is understood in GFO to be Brentano time [**?**]. The GFO defines “Chronoids” not as sets of points, but as entities in their own right, which have two outer and an infinite number of inner “time boundaries” [**?**]. Time boundaries can overlap which allows the modeling of continuous change. Processes project to a chronoid via the *gfo:projects_to* vector (relation):

1. *gfo:Process subclassOf* (*gfo:projects_to some gfo:Chronoid*)
2. *gfo:Chronoid subclassOf* (*gfo:has_time_boundary some gfo:Time_Boundary*)

This provides all the mechanisms needed to define process durations as well as start and end times and dates.

Food Processing Ontology On top of the manufacturing processing ontology we have developed a generic food processing ontology, denoted by the namespace prefix “fp” in Fig. 1 and a more use case-specific apple processing ontology, denoted by the namespace prefix “ap”. To the best of our knowledge, there exist no such ontologies, but for an improved interoperability we have included equivalence relations to the NCI Thesaurus[1] for all the biological concepts that are defined in the food processing ontology and apple processing ontology.

Material Objects Much of the use case experiment is concerned with apple processing and hence, apples can serve as an illustration of how we handle material objects in the ontology. As discussed above, the term “apple” is potentially polysemous and hence, we need to distinguish between an “apple tree” and an “apple fruit”. Furthermore, “apple fruit” must be subdivided into several types of apples such as “apple fruit on tree”, “harvested apple fruit” or “refrigerated apple fruit” if we wish to talk about fruit still ripening on trees as opposed to harvested ones and ones which have undergone some treatment. Ontologically speaking, all of these entities are subclasses of the GFO’s “material object” class. We may therefore write in First Order Logic:

1. *ap:fruit subclassOf ap:organism_part*
2. *ap:organism_part subclassOf gfo:material_object*
3. *ap:apple_tree subclassOf ap:maleae*
4. *ap:malae subclassOf gfo:material_object*
5. *ap:apple_fruit subclassOf ap:fruit*
6. *ap:apple_fruit_on_tree equivalentClassOf*
   (*ap:apple_fruit and* (*part_of some ap:apple_tree*))

Processes The ontological treatment of processes in the food processing ontology is analogous to the description outlined above: the apple microwave treatment process has at least three discernible participants and distinct roles: (a) an apple playing the role of the heated entity, (b) the microwave oven playing the role of the treating

---

[1] `http://ncit.nci.nih.gov/`

entity and (c) an apple playing the role of the treated entity. The apple in (a) and (c) are ontologically distinct: with the beginning of treatment process, the apple has ceased to be an untreated apple and become a treated apple. In first order description logic, we may define the treatment process as follows:

1. *ap:apple_treatment_process equivalentClassOf*
   (*fp:microwave_heating_process and* (*proc:has_participant some*
   (*ap:apple_fruit and* (*gfo:plays_role some proc:heated_entity_role*))) *and*
   (*proc:has_participant some*
   (*proc:microwave_oven and* (*gfo:plays_role some proc:heating_entity_role*))) *and*
   (*has_participant some*
   (*apple_fruit and* (*gfo:plays_role some proc:treated_entity_role*))))
2. *ap:apple_microwave_treatment_process subClassOf*
   fp:microwave_heating_process
3. *fp:microwave_heating_process subClassOf gfo:Process*

4 Information services

For our proof-of-concept service implementation to analyse and verify the experimental data we have chosen to use the SADI–semantic Automated Discovery and Integration–framework [**?**]. SADI is a semantic Web service framework that is predominantly used in the bioinformatics domain. SADI comprises a set of semantic Web compliant conventions and suggested best-practices for data representation and exchange between Web services. In contrast to other semantic Web service frameworks, SADI takes some assumptions that make the protocol and the implementation much easier than, for example, OWL-S [**?**] and WSMO [**?**]/WSMX [**?**]. SADI Web services are stateless, transformative, atomic and idempotent. The distinguishing simplification in SADI is that the input and output of a Web service must share a common “base” identifier, thus assuming that all services are “annotator services”, where the Web services consume some specific input data type, and return a related output data type generated by whatever operation the service executes. Although our use case and the manufacturing domain typically require a process model to execute non-atomic manufacturing processes, we have chosen to use SADI for our first implementation for its ease-of-use. Further, SADI services are encapsulated functionalities that can be accessed over the HTTP protocol and thus, can, in the long run, be incorporated into a framework that allows for the execution of composite processes. Currently, we implement the process logic of composite processes in Java.

4.1 Architecture

We deployed our SADI information services onto an Apache Tomcat Server and use the Jena library to query the RDF Triple Store running on the same Tomcat instance (see Fig. 2). The RDF database is loaded with the ontologies as described in Sect. **??** constituting the *TBox*, while the actual data (the *ABox*), such as the temperature measurements for apples undergoing the heat treatment from different runs of the experiment, are loaded into the Triple store via scripts that transform the raw sensor data into ontology instances. The services use SPARQL queries to retrieve the required information from the Triple store, process them and return the annotated ontology instance back to the client.

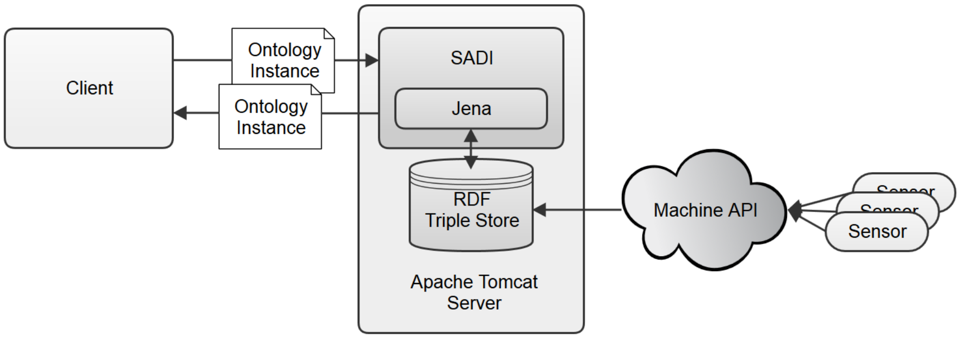

Fig. 2. System architecture of our semantic lab Notbook

4.2 Experimental Constraints Compliance service

The first semantic Web service we developed for analysing data in the semantic lab notebook allows to verify if an apple undergoing the treatment process was refrigerated properly before the experiment. The service takes as input (see Listing **??**) a specific refrigeration process specification and gives a boolean return (see Listing **??**) confirming if all the temperature observations recorded for the apples participating in the refrigeration process comply to the limits defined in the specification. The refrigeration process itself references the individual apples/batches through a *gfo:plays_role* relation. The semantic Web service uses SPARQL to query the specification of the provided process and to extract the allowed minimum and maximum temperature values for each refrigeration process. Then all temperature observations for apples playing a role in the given process are extracted and checked against the specified limits. A boolean attribute *fp:isCertifiedProcess* is then added to the instance of the process specification which is in turn returned by the service.

A false return value would indicate an interrupted cooling chain and thus nullify the results of the experiment.

Listing 1.1. Input RDF

```
<!DOCTYPE rdf:RDF
  [ <!ENTITY matinf ”http://matinf.cmse.csiro.au/”> ]>
<rdf:RDF xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
  xml:base=”@matinf;id/exp_12-09-12#”
  xmlns:fp=”@matinf;ont/owl/processing.owl#””>
  <owl:Ontology rdf:about=”@matinf;id/exp_12-09-12#”/>
  <fp:refrigeration_process_specification
    rdf:about=”@matinf;id/exp_12-09-12#refrigeration_proc_spec_1”>
  </fp:refrigeration_process_spec>
</rdf:RDF>
```

Listing 1.2. Output RDF

```
<!DOCTYPE rdf:RDF
  [ <!ENTITY matinf ”http://matinf.cmse.csiro.au/”> ]>
<rdf:RDF xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
```

```
  xml:base="@matinf;id/exp_12-09-12#"
  xmlns:fp="@matinf;ont/owl/foodprocessing.owl#">
  <fp:refrigeration_process_specification
      rdf:about="@matinf;id/exp_12-09-12#refrigeration_proc_spec_1">
    <fp:isCertifiedProcess
        rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">
      true
    </fp:isCertifiedProcess>
  </fp:refrigeration_process_specification>
</rdf:RDF>
```

This service essentially implements a complex SPARQL query using FILTERS and thus could also be expressed in a SPARQL templating language such as SPIN [?] and then executed on demand. However, the next service uses complex calculations that cannot be expressed in SPARQL. Thus, it represents one of many computations in our use case that require algebraic computations that cannot be expressed in SPARQL or RDFS/OWL directly.

4.3 Continual M_{52} computational service

The second service expects an instance of an "apple_description" as input (see Listing ??) and returns a M_{52} time equivalent [?] (e.g. "m52_model_output_specification_1" in Listing ??) that has been achieved during a microwave treatment process for the given apple (described by the apple description). The returned "m52_model_output_specification_1" instance is a specification itself while the actual value that was created for the M_{52} time equivalent can be queried via the following SPARQL query:

```
SELECT ?o
WHERE { <base:m52_model_output_specification_1> <science:has_Value_Literal> ?o }
```

In our case this query returns a value of 22.62 minutes which indicates how long the embedded larvae in the specific apple would have been exposed to a reference temperature of $52°C$ in the treatment process. To calculate this value, the service retrieves all temperature observations for each of the four sensors embedded in the specific apple described by "granny_smith_apple_desc_001" and incrementally accumulates the minimum accumulated total temperature equivalent to M_{52}. This value indicates if the apple was sufficiently heat treated in the experiment to kill all embedded larvae.

Listing 1.3. Input RDF

```
<!DOCTYPE rdf:RDF
  [ <!ENTITY matinf "http://matinf.cmse.csiro.au/"> ]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xml:base="@matinf;id/exp_12-09-12#"
  xmlns:ap="@matinf;ont/owl/appleprocessing.owl#">
  <owl:Ontology rdf:about="@matinf;id/exp_2012-09-12#"/>
  <ap:apple_description
     rdf:about="@matinf;id/exp_12-09-12#granny_smith_apple_desc_001">
  </ap:apple_description>
</rdf:RDF>
```

Listing 1.4. Output RDF

```
<!DOCTYPE rdf:RDF
  [ <!ENTITY matinf ”http://matinf.cmse.csiro.au/”> ]>
<rdf:RDF xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
  xml:base=”@matinf;id/exp_12-09-12#”
  xmlns:ap=”@matinf;ont/owl/appleprocessing.owl#”
  xmlns:science=”http://purl.org/scimantica/owl/science.owl#”>
  <ap:apple_description
    rdf:about=”@matinf;id/exp_12-09-12#granny_smith_apple_desc_001”>
      <science:is_specified_by
        rdf:about=”@matinf;id/exp_12-09-12#m52_model_output_spec_1”/>
      </science:is_specified_by >
  </ap:apple_description >
</rdf:RDF>
```

5 Related Work

This is not the first time that semantically enabled electronic lab notebooks have been proposed [?,?,?]. Many of the related works describe the techniques and the methodologies on how to introduce metadata to improve the provenance of experiments. For example, in [?] it is proposed that a widely used lab notebook, ELN, be extended with semantic annotation capability to support integration with external annotation sources such as produced by problem solving environments. Other works have gone farther and actually published the data of experiments in RDF/OWL [?].

Some others have developed tools focussed on different aspects of the experimental data curation problem, for example on the scalability [?], or on capturing the relationships between results of different experiments [?].

[?] proposes an aggregation tool based on RSS feeds to ensure that the objects created during the research process are recognized, stored and indexed.

The bioinformatics community as a whole is spearheading other academic disciplines by capturing vast quantities of the knowledge published in scientific articles as ontologies in the Bioportal initiative[2].

However, we are not aware of any prior works on capturing the data produced by an experiment in ontologies combined with custom-built RESTful semantic Web services on top of the RDF data that allow to reproduce and verify the results of the experiment. The closest work to ours, but with a stronger focus on capturing the entire workflow of an experiment was proposed in [?]. The work introduces a laboratory domain specific ontology and the COW (Combining Ontologies with Workflows) software tool was developed to formalize workflows which were enhanced with ontological concepts taken from the developed domain specific ontology.

6 Conclusion

We described our prototypical implementation of a semantic lab notebook that allows data obtained in an experiment to be stored in RDF and accessed via SPARQL and custom-built semantic Web services. The system allows scientists to read the experiment related data and to combine it as part of a scientific workflow. We implemented ontologies required to model our data points via Protégé in OWL and

---

[2] `http://bioportal.bioontology.org/`

developed SADI RESTful Web services in Java that implement computational analysis functionality that cannot be expressed in the ontology directly. With an increased availability of ontologies and tools that support the capture of RDF, semantic lab notebooks can play a significant role in helping the research community to store experiment data consistently, process it faster and allow the mashup of collected datasets to facilitate research studies over multiple datasets, while reducing complexity and error rates.

Acknowledgements We are thankful to the support of our colleagues in the CSIRO Animal, Food and Health Sciences division, in particular to Dr. Mala Gamage who regularly shared the results of their heat-treatment experiments with us.